

Kohei Arai *Editor*

Intelligent Systems and Applications

Proceedings of the 2023 Intelligent
Systems Conference (IntelliSys) Volume 1

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, Türkiye*

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 worldwide 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.

For proposals from Asia please contact Aninda Bose (aninda.bose@springer.com).

Kohei Arai

Editor

Intelligent Systems and Applications

Proceedings of the 2023 Intelligent Systems
Conference (IntelliSys) Volume 1



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 978-3-031-47720-1 ISBN 978-3-031-47721-8 (eBook)
<https://doi.org/10.1007/978-3-031-47721-8>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

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

Paper in this product is recyclable.

Editor's Preface

It gives me immense pleasure and privilege to present the proceedings of Intelligent Systems Conference (IntelliSys) 2023 which was held in a hybrid mode on 7 and 8 September 2023. IntelliSys was designed and organized in Amsterdam, the Netherlands, that aimed to advance and apply artificial intelligence to real world.

IntelliSys is an annual conference which provides the platform to researchers, academics and industry practitioners across the globe to share their valuable findings and insights. The conference witnessed huge international participation with delegates from all parts of the world.

A technological revolution has hit the world where Artificial Intelligence, Robotics, Machine Vision and Ambient Intelligence have gained preeminence over all the other fields. The researches in these fields have managed to give workable solutions to many intriguing problems. They also let us see through what the future would look like if artificial intelligence was entwined in our life. One of the meaningful and valuable dimensions of this conference is the way it lets researchers report and discuss these breakthroughs.

The aim was to further increase the body of knowledge in this specific area by providing a forum to exchange ideas and to build international links. Authors from 50+ countries submitted a total of 605 papers to be considered for publication. Each paper was reviewed on the basis of originality, novelty and rigorousness. After the reviews, 239 were accepted for presentation, out of which 227 papers are finally being published in the proceedings. We would like to extend our gratitude to all the learned guests who participated on site as well as online to make this conference extremely fruitful and successful and also special note of thanks to the technical committee members and reviewers for their efforts in the reviewing process. Special acknowledgment to all the distinguished keynote speakers.

We are extremely glad to bring forth the precious researches from our learned scholars and hope to whet the appetite of our readers. Your continued support and enthusiasm would motivate us to grow and evolve exponentially.

Saga, Japan

Kohei Arai

Contents

Active Restoration of Lost Audio Signals Using Machine Learning and Latent Information	1
<i>Zohra Adila Cheddad and Abbas Cheddad</i>	
Designing Cancellation Intervention System with Sliding Lead Times	17
<i>Yong Seog Kim</i>	
Automatic Generation of a Portuguese Land Cover Map with Machine Learning	36
<i>Antonio Esteves and Nuno Valente</i>	
The Use of Fuzzy Controllers in Automatic Control Systems for Quadcopters	59
<i>Ramin Rzayev, Tunjay Habibbayli, and Murad Aliyev</i>	
POWOP: Weather-Based Power Outage Prediction	75
<i>Natalie Gdanitz, Lotfy H. Abdel Khalil, Agbodzea Pascal Ahiagble, Sabine Janzen, and Wolfgang Maass</i>	
Toward a Human-in-the-Loop Approach to Create Training Datasets for RDF Lexicalisation	84
<i>Jessica Amianto Barbato, Marco Cremaschi, Anisa Rula, and Andrea Maurino</i>	
Fuzzy Time Series Forecasting on the Example of the Dow Jones Index Dynamics	102
<i>Ramin Rzayev, Parvin Alizada, and Tahir Mehdiyev</i>	
Traffic State Prediction of Perturbed and Non-perturbed Traffic Scenarios	129
<i>Teck-Hou Teng, George Rosario Jagadeesh, Thakkar Kunal, and Chong Chee Chung</i>	
“Seeing Sound”: Audio Classification Using the Wigner-Ville Distribution and Convolutional Neural Networks	145
<i>Christonasis Antonios Marios, Stef van Eijndhoven, and Peter Duin</i>	
Application and Performance Improvement of Transfer Learning on ICBHI Lung Sound Dataset	156
<i>Mohan Xu and Lena Wiese</i>	

A Computational Situationally Self-controlled Brain and Mind Interface Under Uncertainty	174
<i>Ben Khayut, Lina Fabri, and Maya Avikhana</i>	
Ethical Concerns About Personhood, Responsibility, and Privacy in Active and Passive Brain-Computer Interfaces	186
<i>Ronja Rönnback, Fenna Blom, and Maryam Alimardani</i>	
The VesselAI Methodology for AI-Powered Decision Support Systems for the Maritime Industry	201
<i>Christos Kontzinos, Spiros Mouzakitis, Carlos Agostinho, Paulo Figueiras, and Dimitris Askounis</i>	
An Energy-Efficient Reconfigurable Autoencoder Implementation on FPGA	212
<i>Murat Isik, Matthew Oldland, and Lifeng Zhou</i>	
Convergence of the Mini-Batch SIHT Algorithm	223
<i>Saeed Damadi and Jinglai Shen</i>	
Graph Autoencoder-Based Detection of Unseen False Data Injection Attacks in Smart Grids	234
<i>Abdulrahman Takiddin, Muhammad Ismail, Rachad Atat, Katherine R. Davis, and Erchin Serpedin</i>	
Causal Analysis of Artificial Intelligence Adoption in Project Management	245
<i>Egor Sarafanov, Omid Fatahi Valilai, and Hendro Wicaksono</i>	
Comparative Lightweight Scheme for Individual Identification Through Hand-Vein Patterns	265
<i>Mateo Mejia-Herrera, Juan S. Botero-Valencia, and Ruber Hernández-García</i>	
URL Classification with Intrusion Detection System	284
<i>Veeresh Uppara, Akif Iqbal, Vishal P, Vinay M V, and Sarasvathi V</i>	
NP4G: Network Programming for Generalization	301
<i>Shoichiro Hara and Yuji Watanabe</i>	
Blockchain and AI for Optimizing Bank Data Security	316
<i>Ibrahima Souare and Khadidiatou Wane Keita</i>	
Streamlining Conceptual Modeling	326
<i>Hermann Bense</i>	

E-Step Control: Solution for Processing and Analysis of IS Users Activities in the Context of Insider Threat Identification Based on Markov Chain	345
<i>Oksana Nikiforova, Vitaly Zabiniako, and Juris Kornienko</i>	
Machine Learning Based Intelligent Irrigation System Using WSN	360
<i>Benhamada Abdelhak and Kherarba Mohammed</i>	
Comparison of Artificial Neural Networks Algorithms on Datasets with Different Characteristics	371
<i>Bruno Pilosta, Dijana Oreski, and Nikola Kadoic</i>	
Boosting Federated Multitask Learning: Transfer Effects in Cross-Domain Drug-Target Interaction Prediction	386
<i>Dániel Sándor and Péter Antal</i>	
Surveying Impacts of AI in Education and Creative Practices	400
<i>Andy Deck</i>	
Active Risk Mitigation for Unmanned Aerial Systems	413
<i>Andrew Kendall and John-Paul Clarke</i>	
Towards Explainable AI: Relationship Between Twitter Sentiment, User Behaviour, and Bitcoin Price Prediction	433
<i>Qinan Zhu and Rotimi Ogunsakin</i>	
Production Portfolio Theory II—First Steps Towards a General Portfolio Theory and Numerical Exemplifications	448
<i>Bernhard Heiden and Bianca Tonino-Heiden</i>	
AI as a Threat to Education: Contrasting GPT-3 and Google in Answering Questions Along Bloom’s Taxonomy of Educational Objectives	469
<i>Nina Li</i>	
Citation Recommendation Employing Proximity-Based Heterogeneous Network Embeddings	477
<i>Zafar Ali, Irfan Ullah, Pavlos Kefalas, Nimbeshaho Thierry, Kalim Ul Haq, and Anupam Sarkar</i>	
Performance of Machine Learning Classifiers for Malware Detection Over Imbalanced Data	496
<i>Paulina Morillo, Diego Bahamonde, and Wilian Tapia</i>	
Pursuing the Optimal CP Model: A Batch Scheduling Case Study	508
<i>Giacomo Da Col and Erich Teppan</i>	

Data-Driven Decision-Making Framework for Cost-Efficient Energy Retrofit of Italian Residential Building Stock	521
<i>Ania Khodabakhshian and Fulvio Re Cecconi</i>	
Text Summarization for Call Center Transcripts	542
<i>Ishrat Ahmed, Yu Zhou, Nikhita Sharma, and Jordan Hosier</i>	
Targeted Image Reconstruction by Sampling Pre-trained Diffusion Model	552
<i>Jiageng Zheng</i>	
Hulk: Graph Neural Networks for Optimizing Regionally Distributed Computing Systems	561
<i>Zhengqing Yuan, Huiwen Xue, Chao Zhang, and Yongming Liu</i>	
The Power of Words: Predicting Stock Market Returns with Fine-Grained Sentiment Analysis and XGBoost	577
<i>Farshid Balaneji, Dietmar Maringer, and Irena Spasić</i>	
Improving Neural Network Using Jaya Algorithm with Opposite Learning for Air Quality Prediction	597
<i>Iyad Abu Doush, Khalid Sultan, Ahmad Alsaber, Dhari Alkandari, and Afsah Abdullah</i>	
Enhancing Early-Stage XAI Projects Through Designer-Led Visual Ideation of AI Concepts	607
<i>Helen Sheridan, Dympna O'Sullivan, and Emma Murphy</i>	
Using AutoML to Analyze the Effect of Attendance and Seat Location on University Student Grades	617
<i>Ac Hybl and Germán H. Alférez</i>	
The Role of the User in Meaningful Production with AI	633
<i>Iro Laskari</i>	
Personalizing Text-to-Image Diffusion Models by Fine-Tuning Classification for AI Applications	642
<i>Rafael Hidalgo, Nesreen Salah, Rajiv Chandra Jetty, Anupama Jetty, and Aparna S. Varde</i>	
ShapTime: A General XAI Approach for Explainable Time Series Forecasting	659
<i>Yuyi Zhang, Qiushi Sun, Dongfang Qi, Jing Liu, Ruimin Ma, and Ovanes Petrosian</i>	

Command Line Interface Risk Modeling	674
<i>Anthony Faulds</i>	
An Intrinsic Framework of Information Retrieval Evaluation Measures	692
<i>Fernando Giner</i>	
Substructure Discovery in Commonsense Relations Using Graph Representation Learning	714
<i>Ke Shen and Mayank Kejriwal</i>	
Artificial Intelligence Algorithm for Optimizing PID Parameters to Control Weakly Damped Systems	735
<i>Roland Büchi</i>	
Measuring Implicit Bias Using SHAP Feature Importance and Fuzzy Cognitive Maps	745
<i>Isel Grau, Gonzalo Nápoles, Fabian Hoitsma, Lisa Koutsoviti Koumeri, and Koen Vanhoof</i>	
Predicting and Explaining Variations in Software Effort Estimation Using Adaptive Fuzzy-Neural Networks with Clustering	765
<i>Riyadh A. K. Mehdi</i>	
Automated Spatiotemporal Modeling for Real-Time Data-Driven Actionable Insights	780
<i>Hugo Latapie, Mina Gabriel, Sidarth Srinivasan, Ramana Kompella, Kristinn R. Thórisson, and Pei Wang</i>	
Autoadaptive Networks of Coherent Domains for “Intelligent” Quantum Computation and Quantum Information	799
<i>Luigi Maximilian Caligiuri</i>	
HyMO-RF: Automatic Hyperparameter Tuning for Energy Theft Detection Based on Random Forest Classification	820
<i>Francisco J. S. Coelho, André L. M. Alcântara, Allan R. S. Feitosa, Jessica T. Takeuchi, Ronaldo F. Lima, and Abel G. Silva-Filho</i>	
E-ELPV: Extended ELPV Dataset for Accurate Solar Cells Defect Classification	837
<i>Marco Grisanti, Maria Ausilia Napoli Spatafora, Alessandro Ortis, and Sebastiano Battiato</i>	

Application of Deep Q Learning with Simulation Results for Elevator Optimization	849
<i>Zheng Cao, Raymond Guo, Caesar M. Tugunay, Mark Pock, Jiayi Gao, and Ziyu Wang</i>	
Pix2Pix Hyperparameter Optimisation Towards Ideal Universal Image Quality Index Score	862
<i>Dirk Hölscher, Christoph Reich, Martin Knahl, Frank Gut, and Nathan Clarke</i>	
Author Index	883



Active Restoration of Lost Audio Signals Using Machine Learning and Latent Information

Zohra Adila Cheddad¹ and Abbas Cheddad²

¹ Département of Mathematics, Université Frères Mentouri I, 250 17 Constantine, Algeria

zcheddad929@gmail.com

² Department of Computer Science, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

abbas.cheddad@bth.se

Abstract. Digital audio signal reconstruction of a lost or corrupt segment using deep learning algorithms has been explored intensively in recent years. Nevertheless, prior traditional methods with linear interpolation, phase coding and tone insertion techniques are still in vogue. However, we found no research work on reconstructing audio signals with the fusion of dithering, steganography, and machine learning regressors. Therefore, this paper proposes the combination of steganography, halftoning (dithering), and state-of-the-art shallow and deep learning methods. The results (including comparing the SPAIN, Autoregressive, deep learning-based, graph-based, and other methods) are evaluated with three different metrics. The observations from the results show that the proposed solution is effective and can enhance the reconstruction of audio signals performed by the side information (e.g., Latent representation) steganography provides. Moreover, this paper proposes a novel framework for reconstruction from heavily compressed embedded audio data using halftoning (i.e., dithering) and machine learning, which we termed the HCR (halftone-based compression and reconstruction). This work may trigger interest in optimising this approach and/or transferring it to different domains (i.e., image reconstruction). Compared to existing methods, we show improvement in the inpainting performance in terms of signal-to-noise ratio (SNR), the objective difference grade (ODG) and Hansen's audio quality metric. In particular, our proposed framework outperformed the learning-based methods (D2WGAN and SG) and the traditional statistical algorithms (e.g., SPAIN, TDC, WCP).

Keywords: Audio reconstruction · Halftoning · Steganography · Machine learning

1 Introduction

Corrupt audio files and lost audio transmission and signals are severe issues in several audio processing activities such as audio enhancement and restoration.

For example, in different applications and music enhancement and restoration situations, gaps could occur for several seconds [1]. Audio signal reconstruction remains a fundamental challenge in machine learning and deep learning despite the remarkable recent development in neural networks [1]. Audio inpainting, audio interpolation/extrapolation, or waveform replacement address the restoration of lost information in audio. The reconstruction aims to provide consistent and relevant information while eliminating audible artefacts to keep the listener unaware of any occurring issues [2]. The active reconstruction can be considered a preemptive security measure to allow for self-healing when part of the audio becomes corrupted. Active reconstruction means reconstructing lost signals by incorporating side information retrieved from pre-embedded data. Thus, the technique is not intended for any degraded audio but only for audio protected by the steganography information. To this end, and to the best of our knowledge, we found no prior research work on the active reconstruction of audio signals with the fusion of steganography (an information hiding technique), halftoning and machine learning (ML) models. The initial idea (without ML) was proposed in a PhD thesis as an application of steganography. The hiding strategy of steganography can be tailored to act as an intelligent streaming audio/video system that uses techniques to conceal transmission faults from the listener that are due to lost or delayed packets on wireless networks with bursty arrivals; thus, providing a disruption tolerant broadcasting channel [3].

While it is possible to apply passive recovery procedures to estimate missing audio segments, these procedures, whether they are based on statistical inferences or deep learning, would be futile when the missing gap exceeds half a second, thus, prohibiting their use in a real-world scenario. Hence, the corollary to that is our recommendation in this work to harness the power of machine learning methods by embedding compressed latent information. In addition to its ultimate purpose of audio recovery, our proposed approach also benefits security systems in protecting audio files from unauthorised manipulation. It may also be extended to image inpainting.

The contribution of this work is fourfold:

- A halftone-based compression and reconstruction (HCR).
- Orchestration of the three scientific disciplines: steganography, compression, and audio processing, for audio reconstruction or inpainting.
- A new framework that enables sequence-to-sequence deep learning (Seq2Seq) and Random Forest (RF), to locally train on other intact segments and latent representation.
- Unlike existing methods (traditional or deep-learning based), our approach can handle extensive gap reconstruction (e.g., 4000, 8000 ms), more information in Sect. 5.

The remainder of the paper is divided as follows: Sect. 2 discusses related work. Section 3 furnishes the necessary details regarding our proposed method, which we termed ARLAS, including our approach of imposing heavy compression on audio signals. The experimental set-up is highlighted in Sect. 4. Section 5

presents the results and compares the performance against existing competitive methods. This paper concludes with some notes in Sect. 6.

2 Related Work

In the work of Khan et al. [4], the authors proposed a modern neuro-evolution algorithm, Enhanced Cartesian Genetic Programming Evolved Artificial Neural Network (ECGPANN), as a predictor of the lost signal samples in real-time. The authors have trained and tested the algorithms on audio speech signal data and evaluated them on the music signal. A deep neural network (DNN)-based regression method was proposed in [5] for a packet loss concealment (PLC) algorithm to predict a missing frame's characteristics. Two other DNNs were developed for the training by integrating the log-power spectra and phases based on the unsupervised pre-training and supervised fine-tuning. The algorithm then provides the previous frame features to the trained DNN to reconstruct the missing frames. In [1], researchers have analyzed audio gaps (500–550 ms) and used Wasserstein Generative Adversarial Network (WGAN) and Dual Discriminator WGAN (D2WGAN) models to reconstruct the lost audio content. In Khan et al. [6], the authors proposed an audio signal reconstruction model called Cartesian Genetic Programming evolved Artificial Neural Network (CGPANN), which was more efficient than the interpolation-extrapolation techniques. The developed model was robust in recovering signals contaminated with up to 50% noise. In [2], the authors proposed a DNN structure to restore the missing audio content based on the audio gaps. The signals provided in the audio gaps in the DNN structure were time-frequency coefficients (either complex values or magnitude). In the work of Sperschneider et al. [7], the authors presented a delay-less packet-loss concealment (PLC) method for stationary tonal signals, which addresses audio codecs that utilizes a modified discrete cosine transformation (MDCT). In the case of a frame loss, tonal components are identified using the last two obtained spectra and their pitch information. Furthermore, the MDCT coefficients of the tonal components were estimated using phase prediction based on the detection of tonal components. Mokrý et al. [8] presented an inpainting algorithm called SPAIN (SParse Audio INpainter) developed by an adaptation of the successful declipping method, SPADE [9], to the context of audio inpainting. The authors show that the analysis of their algorithm, SPAIN, performs the best in terms of SNR (signal-to-noise ratio) among sparsity-based methods and reaches results on a par with the state-of-the-art Janssen algorithm [10]¹ for audio inpainting. A composite model for audio enhancement that combines the Long Short-Term Memory (LSTM) and the Convolutional Neural Network (CNN) models was proposed in [11]. Perraudeau et al. [12], proposed a reconstruction method for the loss of long signals in audio (i.e., Music signals). The concealment of such signal defects is based on a graph that encodes signal structure in terms of time-persistent spectral similarity and an intuitive optimization embedding scheme.

¹ Iteratively fits autoregressive models using a gap's all previous points for forward estimation and all its future points for backward estimation.

Mokrý and Rajmic [13] proposed a heuristic method for compensating for energy loss after running the \mathcal{L}_1 minimization. Their idea is to take the solution and modify it by entrywise multiplication of the recovered gap in the time domain by a compensation curve in order to increase its amplitude; they termed this approach the Time Domain Compensation (TDC) algorithm.

3 ARLAS: Active Restoration of Lost Audio Signals

Our method aims to reconstruct a realistic segment from audio containing corrupted or dropped regions using active embedding and machine learning. In this section, we describe the proposed method and discuss the individual stages in more detail.

3.1 Halftone-Based Compression and Reconstruction (HCR)

In the field of bit-rate reduction, or data compression, the Lempel-Ziv (LZ) or its variant, the Lempel-Ziv-Welch (LZW), are popular methods. However, they are slow and prone to failure if data corruption occurs, as in our case, there were more than 700 samples deleted from the audio signal. Therefore, finding a compression method that is more immune to data corruption and can provide heavy compression and good approximate reconstruction is desired. Hence, the proposed HCR is meant to exploit the halftoning for this purpose. The algorithm this work adopts is that of *Floyd Steinberg*, which applies error forward diffusion [14]. The rationale behind conceiving the notion of HCR is that the embedding of data in the least significant bits (LSB) of a bit-stream, necessitates dealing with binary data. Let the original audio sampled data be denoted by the vector \vec{v} ($\forall v_i \in \mathbb{R}$), which is then transformed into a matrix \mathbf{D} where $v_{xy} \in \mathbf{D}_{n,m}$ (the x -th row and y -th column in \mathbf{D}). The matrix \mathbf{D} is filled column-wise with suitable dimensions whose automatic estimation is outside of the scope of this work, see Eq. 1.

$$\vec{v} = ([v_1, v_2, v_3, \dots, v_{j-1}, v_j])^T \mapsto \mathbf{D} = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1m} \\ v_{21} & v_{22} & \dots & v_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \dots & v_{nm} \end{bmatrix} \quad (1)$$

The matrix \mathbf{D} (remapped to 8bit unsigned integers) is then passed to the dithering phase using Floyd Steinberg algorithm, which results in a binary matrix (as seen in Fig. 1b) that could be partially inverted. This contributes to the heavy compression that we obtain. For instance, the original audio sampled data and the resulting compressed vector pertaining to Fig. 1b (vectorized), show the following properties: Original Audio (1316019—**10.04 MB**—Byte) and its corresponding compressed vector (1316019—**1.26 MB**—Binary), for the length, size and type, respectively. The error diffusion algorithm exploits the

optical system of the human eye which acts as a low-pass filter removing all high frequencies resulting in the illusion of perceiving a dithered image (only binary) as a continuous tone image. Hence, it follows from such notation that in order to partially inverse the dithering operation we need to apply a low-pass filter to attenuate high frequencies (in our case, we choose a 2D Gaussian filter as in Eq. 2 with a kernel size = $2 \times [2 \times \sigma] + 1$).

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

where $\sigma = 1.50$ (determined empirically) and x, y are the coordinates of the matrix \mathbf{D} where $x \in (1, \dots, n)$ and $y \in (1, \dots, m)$. Recently, the development of deep machine learning rekindled interest in addressing the inverse halftoning problem by optimization-based filtering [15, 16]. Nevertheless, we opt to use the simple and computationally-inexpensive method insinuated above.

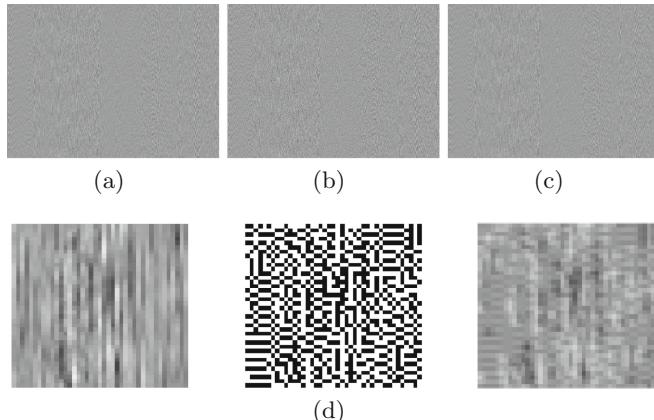


Fig. 1. HCR Visual Inspection: **a** Original Audio Data Reshaped using Eq. 1 and Visualised **b** Halftone of **(a)** (Binary Image), **c** Reconstructed **(a)** from **(b)** and **d** Small Patches Cropped from each Image Left to Right, respectively

In order to scrutinize the efficiency of the reconstruction, we calculated the correlation between the original sampled data against the estimated values from the above HCR process. The reconstruction still demonstrates a good correlation $R = 0.62$, despite the dithered version constitutes only binary values (either 0 or 1); see Fig. 2. The fitted linear regression model is depicted in Table 1. In Fig. 1c, we observe that the process captures a noisy structure and orientation of the original data shown in Fig. 1a; therefore, when this side-information is wedged to machine learning (particularly the state-of-the-art models), it leverages the quality of audio the algorithm reconstructs. This was the impetus for the initiation of this study. To gauge this performance improvement, we tested some ML models whose results appear in Sect. 3.2.

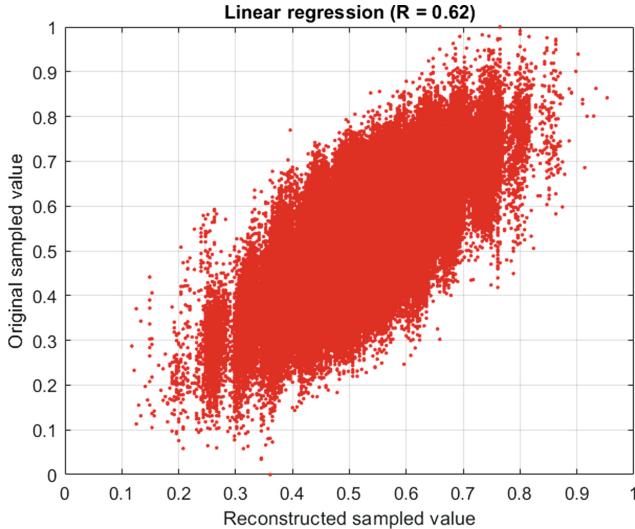


Fig. 2. A Dot Plot Depicting the Correlation between the Original Data and its Estimated Version. Note that the Reconstruction is made from Merely Binary Values (without ML and Signal Drop). This Figure Shows how much of the Information is Lost in the Compression Process; however, we Rely on our Deep Learning Model to Learn Reconstructing the Original Signal from this Approximation

Table 1. Effect-estimates and *P-values* (Wald Tests) from fitting a linear regression model for the data plotted in Fig. 1. Number of observations: 1316019, Error degrees of freedom: 1316017 root mean squared error: 0.0527 R-squared: 0.384, Adjusted R-squared: 0.384

	Estimate	SE	tStat	P-value
(Intercept)	0.23791	0.00032051	742.29	0
x1	0.54719	0.00060376	906.3	0

3.2 Deployed Deep/Machine Learning Architectures

In this section, we list down the different machine learning models we deploy in this study.

Shallow Machine Learning *Random Forest (RF)*—RF regressor is a supervised decision learning technique for regression that employs the ensemble learning method. The most important parameter is the number of trees which is set to 100 as per the recommendation in the literature [17]. It has been demonstrated that RF is more robust than other similar approaches for handling small samples, high dimensional and nonlinear fitting [18, 19].

Deep Machine Learning *Long short-term memory (LSTM)*—LSTM is a type of recurrent neural network (RNN) models; it is best suited to make predictions

based on time series data by learning long-term dependencies [20]. We have implemented a multi-layer LSTM recurrent neural network to predict the missing signal values with the Keras TensorFlow library (more details in Sect. 4.4). The LSTM is inherently able to learn some long-term dependencies and find patterns over time, which ultimately makes its next prediction more accurate. The latter distinguishes it from traditional shallow ML models. The LSTM also overcomes the vanishing gradient problem that other RNN-based architectures are prone to [21]. This is the impetus for using LSTM in audio processing, whose signals can be viewed as time-series inputs.

Training and Testing Segments This section briefly discusses how the two deep/machine learning models are trained and tested. In Fig. 3 (top row), the original audio sampled signal is displayed for mere comparison. Figure 3 (second row) shows the stego-audio signal with the embedded copy and a simulation of signal loss (i.e., empty segment). Note that the stego-audio² and the original audio look identical since all what was flipped is the last LSB value. Subsequently, we extract the hidden data from the LSB plane, and the values in the extracted binary vector are rearranged by using the same secret key (e.g., for simplicity, we choose the length of the audio segment as the key). The vector is then transformed to a matrix using Eq. (1) (which should correspond to the dithered version), and then a 2D Gaussian filter is applied using Eq. (2). The result is then vectorised to yield the plot shown in Fig. 3 (third row). Part of this vector will be used for training and validation (i.e., the red segments), and the tuned model is finally applied to the test data (i.e., the green segment) to predict (reconstruct) the lost segment.

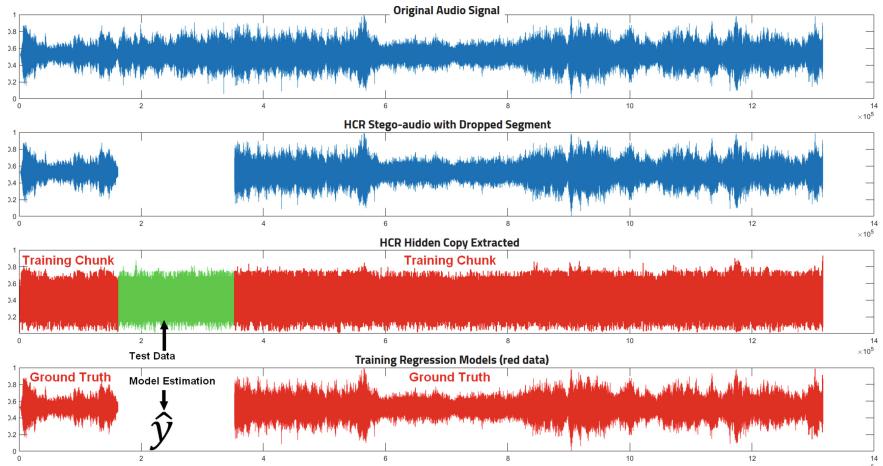


Fig. 3. Audio track segments to train (Red) and test (Green) machine learning models

² In the domain of Steganography, a stego-carrier is a signal similar to the original one except that it carries in it concealed data.

Performing global tuning based on locally adaptive learnt statistics was discussed in the literature though in a different context [22]. Figure 4, along with the Algorithms 1, 2, and 3, provide an in-depth detailed description of our approach. In essence, Algorithm 1 describes our embedding approach, while Algorithm 2 provides feature extraction based primarily on the extracted embedded data and on its derived augmented feature space using scalogram (namely, the continuous 1-D wavelet transform) [23], which shall form the backbone for the training of the machine learning models.

s

4 Experimental Set-Up

The experimental samples in this paper are commonly used audio file samples exhibiting different music instruments. As this research focuses on active reconstruction, the audio input is in *.wav* file format (PCM 16 bit). The main reason for selecting this format is the sound quality, as it preserves the originality of analogue audio without any compression.

4.1 State of the Art Methods

The state-of-the-art sparsity-based inpainting techniques to which we compared our approach are the non-learning methods ‘ASPAIN’ [8], Chambolle-Pock (CP) [24], Douglas-Rachford based (DR) [25], Janssen Autoregressive [10], Orthogonal Matching Pursuit (OMP) [26], ‘SSPAIN-H’ [8], Revisited and re-weighted methods (‘reCP’, ‘reDR’, ‘wCP’, ‘wDR’) [13], and the Time Domain Compensation (TDC) [13]. Moreover, we also included a deep-learning based method (D2WGAN) [1] inspired from [27], and a graph-based (Similarity Graphs-SG-) [12]. Our approaches are labeled ‘RF’, and ‘LSTM’. The RF and the LSTM

Algorithm 1: HCR-Embedding Algorithm

Data: *Audio file S*
Result: *Self-embedded file SS*

```

 $S' \leftarrow$  Scale the vector S to the interval  $[[0,1]*255]$ ;  

 $S'' \leftarrow$  Reshape  $S'$  into a matrix ; /* dimensions are calculated  

adaptively */  

 $D \leftarrow$  Perform image dithering on  $S''$ ; /* heavy lossy compression */  

;  

 $D' \leftarrow$  Flatten D into a vector;  

SET SEED  $\alpha = \text{length}(D')$ ;  

 $D'' \leftarrow$  Permute bit stream positions of  $D'$  with  $\alpha$ ;  

; /* to avoid localised distortion */  

 $SS \leftarrow$  embed  $D''$  into LSB (least significant bits) of  $S'$ ;  

 $SS \leftarrow$  rescale(SS);  

Return SS;
```

Algorithm 2: HCR-Feature Extraction Algorithm

Data: Self – embedded file SS with gap G
Result: Feature descriptors file F

$S' \leftarrow$ Scale the vector SS to the interval $[[0,1]^*255]$;
 $S'' \leftarrow$ extract the LSB bit stream of S' ;
SET SEED $\alpha = \text{length}(S'')$;
 $E \leftarrow$ Invert permutation of the bit stream positions of S'' with α ;
 $E' \leftarrow$ Reshape E into a matrix ;
 $E'' \leftarrow$ Apply 2D Gaussian filter as in Eq. 2 and flatten it;
 $W \leftarrow$ Compute scalogram from E'' using continuous 1-D wavelet transform
and L_1 optimisation;
 $F \leftarrow$ concat vertically $\Re(W) \& E''$;
; /* to form the augmented feature space */
Return F ;

Algorithm 3: HCR-RF/LSTM Algorithm

Data: Self – embedded file SS with gap G
Data: Feature descriptors file F
Result: Reconstructed audio file \hat{S}

$M \leftarrow \text{length}(SS)$;
 $N \leftarrow 1$;
 $TrainX \leftarrow []$;
 $TrainY \leftarrow []$;
 $TestX \leftarrow []$;
 $TestY \leftarrow []$;
while $N \leq M$ **do**
 if $N \in G$ **then**
 $| TestX \leftarrow F[N]$; /* Extract features pertaining to the gap G */
 else
 if $N \notin G$ **then**
 $| TrainX \leftarrow F[N]$;
 $| TrainY \leftarrow SS[N]$; /* Response variables are those regions
 of the uncorrupt signal SS */
 end
 end
end
 $HCRModel \leftarrow$ Train RF/LSTM using {TrainX, TrainY};
 $\hat{G} \leftarrow HCRModel \{TestY\}$; /* Estimate the gap using the trained model */
 $\hat{S} \leftarrow SS + \hat{G}$;
Return \hat{S}

are trained on the extracted pre-embedded data to perform the reconstruction of the missing gap.

4.2 Data Set (Audio Samples)

We used the same audio benchmark as in [8], which comprises 10 files: *Violin*, *Clarinet*, *Bassoon*, *Harp*, *Glockenspie*, *Celesta*, *Accordion*, *Guitar sarasate*, *Piano schubert*, and *Wind ensemble stravinsky*. The gap length of 100 ms and 300 ms were chosen randomly from the set so that the gaps do not overlap. Therefore, we ran a total of 280 tests ($14(\text{methods}) \times 10(\text{segments}) \times 2(\text{gaps})$). These tests (Phase I) were run on all methods reported in the previous section except the D2WGAN and SG. In the second stage (Phase II), we deploy the deep learning and the SG methods. We first select the top-performing methods from Phase I. Then, we perform additional tests using the *Piano* and the *Acoustic Guitar and Orchestra* (Mixed instruments) audio files, the gaps' length of 520, 820, 865, and 883 ms were chosen following the paper [1].

4.3 Evaluation Metrics

We utilise three commonly used reference-based metrics to evaluate the performance of the different audio restoration algorithms. The first is the objective difference grade (ODG) which is described in the Perceptual Evaluation of Audio Quality (PEAQ) standard algorithm used for objective measurements of perceived audio quality. It is believed to be based on generally accepted psycho-acoustic principles that approximate the subjective difference grade used in human-based audio tests [28–30]. The second is related to ODG but uses Hansen’s method and a different model for speech quality estimation [31]; this metric is termed herein “QC”. The third performance measure is the scale-invariant source-to-noise ratio (SI-SNR), which has recently been advocated for as the preferred evaluation metric in lieu of the standard source-to-distortion ratio (SDR), SI-SNR (hereforth abbreviated as SNR) is defined as follows [32]:

$$\begin{cases} \mathbf{s}_{\text{target}} := \frac{\langle \hat{\mathbf{s}}, \mathbf{s} \rangle \mathbf{s}}{\|\mathbf{s}\|^2} \\ \mathbf{e}_{\text{noise}} := \hat{\mathbf{s}} - \mathbf{s}_{\text{target}} \\ \text{SI-SNR} := 10 \log_{10} \frac{\|\mathbf{s}_{\text{target}}\|^2}{\|\mathbf{e}_{\text{noise}}\|^2} \end{cases} \quad (3)$$

where $\hat{\mathbf{s}} \in \mathbb{R}^{1 \times T}$ and $\mathbf{s} \in \mathbb{R}^{1 \times T}$ are the reconstructed and original clean sources, respectively, and $\|\mathbf{s}\|^2 = \langle \mathbf{s}, \mathbf{s} \rangle$ denotes the signal power. Scale invariance is ensured by normalizing to zero-mean prior to the calculation. All of the metrics capture objective measures of signal quality. The ODG metric’s values range from 0.0 (imperceptible audio distortion) to -4.0 (very annoying distortion). The SNR is measured in dB (decibel); the higher it is, the better the reconstructed signal is. The objective speech quality measure (QC) would ideally reach to 1.000 with a perfect reconstruction.

4.4 Software Environment

The RF and other algorithms are executed using MATLAB (R2022a); for the LSTM implementation, we used Python (Keras, Pandas, Scipy, Numpy and TensorFlow libraries). The LSTM properties are as follows: model (Sequential()),

Dropout (0.2), Optimizer (Adam), learning rate (0.0001), and the loss function for our model was measured using MSE (mean squared error). Finally, we fit the models with a batch size of 32 and 40 epochs (to avoid overfitting, as the training process had a stable loss within the first five iterations).

5 Results and Analysis

The comparison is performed between the original and the reconstructed audio signals in this study. After extracting the required training data from the sequence (see Fig. 4), the data is then augmented using a scalogram. The aggregated data (features) are then passed to RF and LSTM models for training; see Algorithm 2. The test set is the hidden data embedded in the stego-audio whose dynamic range is extended using 2D Gaussian filter and whose feature space is extended using the scalogram; see Algorithms 2 and 3. The evaluation of the reconstructed signals to the original signal (which acts as the reference for validation) is observed by calculating the statistical metrics reported in Sect. 4.3. Many state-of-the-art audio inpainting algorithms (in Phase I) had their performance deteriorating when exceeding the range of very short gaps (≈ 45 ms); see Fig. 5 where we show a visual summary of the performance with gaps of 100 and 300 ms (20 tests for each method). In Phase II, we observed good performance of the proposed approach for more extended gap reconstruction (e.g., 800 ms). Fig. 6 depicts a ranking summary, and Fig. 7 shows a real example of an audio gap reconstruction whose blow-up is presented in Fig. 8, allowing closer scrutiny of the reconstruction quality. Additional examples are furnished online.³ Moreover, the ability to handle lengthy missing gaps (e.g., 4000ms, 8000ms) teases our approach apart from other methods. The already existing methods, which we examined, are not designed nor able to handle these large gaps' reconstruction. The audio files of our experiments (i.e., LSTM, RF) are available online.⁴ The experiments reinforce that deep learning reconstruction approaches can benefit from embedded side-information if designed carefully, compare D2WGan to our approaches in Fig. 6 and Table 2.

6 Conclusions

The aim of this paper is to put forward a new framework which proposes the fusion of audio dither-based steganography with machine/deep learning for the active reconstruction of lost signals. The results show that the proposed solution is feasible and can enhance the reconstruction of lost audio signals (readers may wish to listen to the audio online, see URL in Sect. 5). We conducted experiments on several types of signal drops of (100, 300 ms), (500–800 ms) and (4000, 8000 ms (shown online)). As a proof-of-concept, we can assert that, in general, the LSTM and the RF models are good models to utilise. Our approach

³ Audio Clips (medium gaps): <https://ardisdataset.github.io/ARLAS2/>.

⁴ Audio Clips (lengthy gaps): <https://ardisdataset.github.io/ARLAS/>.

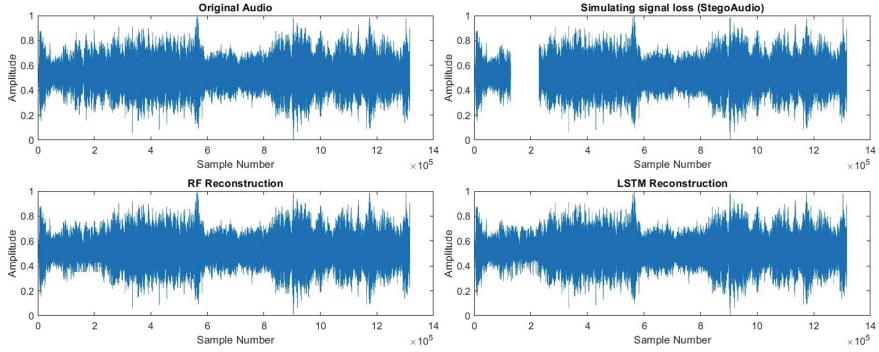


Fig. 4. Reconstruction of a short audio signal using RF and LSTM

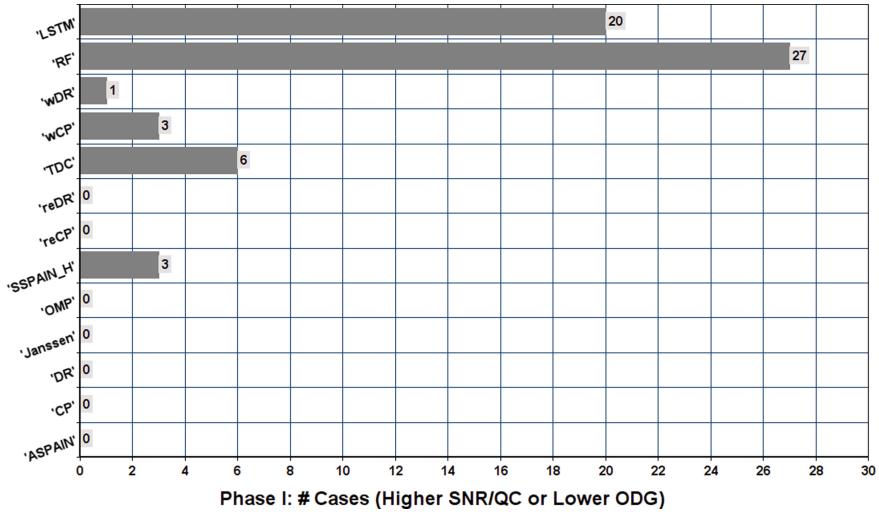


Fig. 5. Phase I: determining the best performing non-learning methods. from the results of 20 tests on each of these methods, we have two competitive methods (CP and TDC) that will be tested in phase II. Although ASPAIN has had no winning cases, since it is a recent algorithm, we opt to upvote it for relevance to phase II (see Sect. 4.2). The X-axis The X-axis denotes the number of cases a given algorithm outperforms other algorithms in terms of SNR or ODG (20 Tests were Measured). Detailed numerical results, on which this figure is based, are furnished in the supplementary files (https://github.com/ARDISDataset/ARLAS/tree/main/Excel_Sheet)

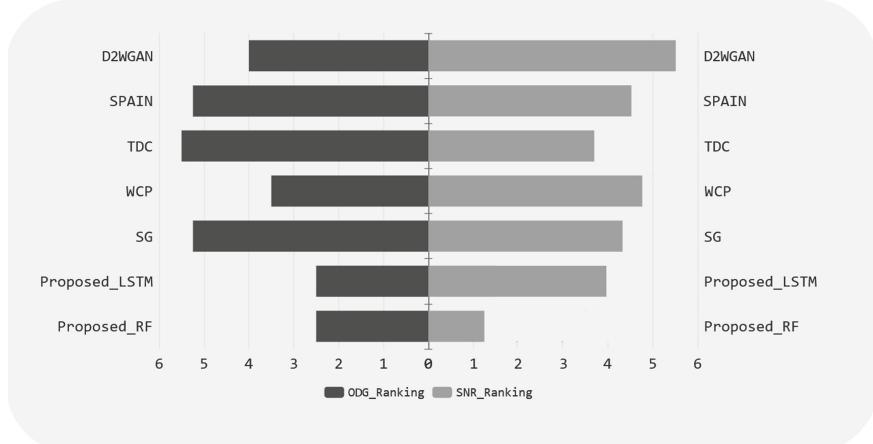


Fig. 6. Phase II: determining the best-performing methods, including deep learning-based (D2WGAN) and Graph-Based (SG). From the results of 8 tests on each of these methods, we can observe that RF and LSTM (Sequence-to-Sequence Modelling) exhibit promising results on both metrics (SNR and ODG), which ranked them at the top of the list. Detailed numerical results, on which this figure is based, are furnished in the supplementary files (https://github.com/ARDISDataset/ARLAS/tree/main/Excel_Sheet)

Table 2. Average ranking of the different methods in phase II (8 Tests) measured using the three audio quality metrics. ODG and SNR are depicted graphically in Fig. 6

Method	Rank-ODG	Rank-QC	Rank-SNR
Proposed-LSTM	2.50	2.50	4.00
Proposed-RF	2.50	2.00	1.25
wCP [24]	3.50	5.25	4.75
D2WGAN [1]	4.00	2.75	5.50
SG [12]	5.25	5.75	4.25
SPAIN [8]	5.25	4.50	4.50
TDC [13]	5.50	5.50	3.75

is not meant to replace current inpainting audio methods but rather to assist them by providing latent side information. It can also benefit security systems in protecting audio files from unauthorised manipulation. The paper supplies extensive experiments, which we believe are compelling evidence of the efficacy of our proposed approach, a corollary when combining halftoning, steganography and machine learning. To our knowledge, we found no similar implementation in the literature for audio missing-segment reconstruction. Thus, we conclude this paper by stating that the fusion of steganography and state-of-the-art machine

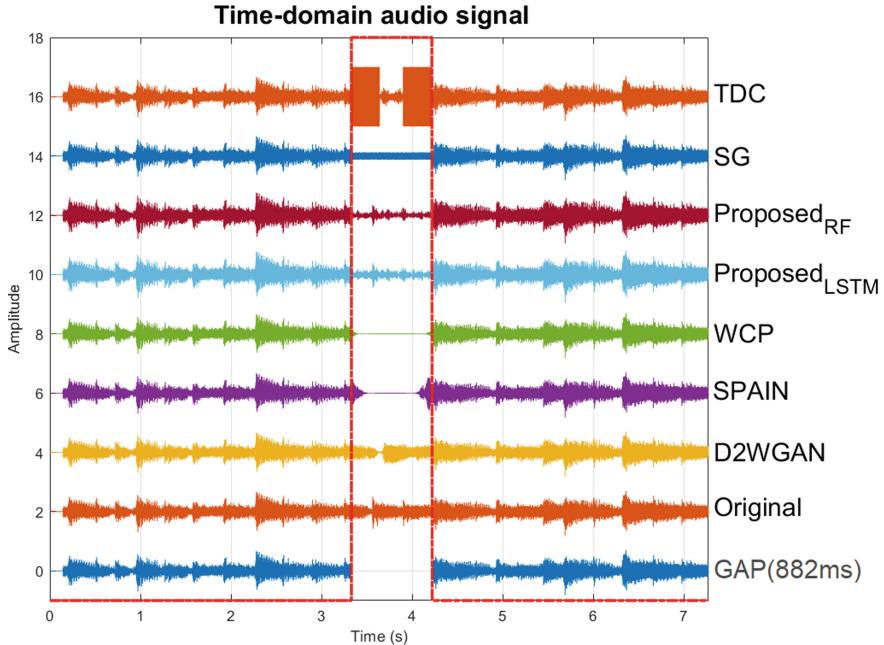


Fig. 7. Phase II—audio reconstruction test:- performance of the different audio inpainting algorithms on an example audio sample (*extend-solo-1-real-36*) [1]. The GAP signal is merely the original audio sample but with the gap (Highlighted in Red Dashed Line) simulating dropped signal

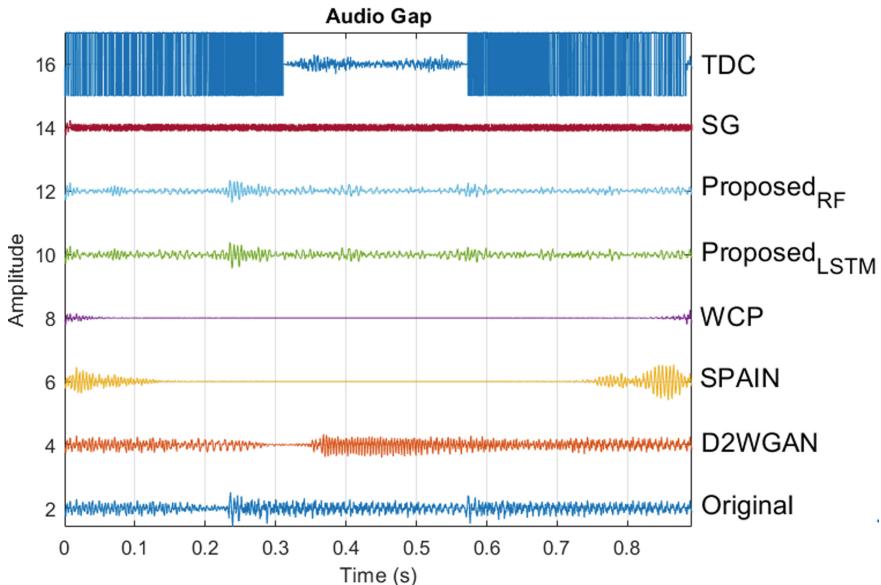


Fig. 8. A zoom-in into the reconstructed gap of Fig. 7. Notice how RF and LSTM predict the level of the amplitude despite the length of the gap (i.e., 882 ms)

learning algorithms can be considered for the active reconstruction of audio signals. However, as we pointed out in the discussion section, there is room for enhancement, for example, enhancing the algorithm for inverse-halftoning, which is an ill-posed problem.

References

1. Ebner, P.P., Eltelt, A.: Audio inpainting with generative adversarial network (2020). [arXiv:2003.07704](https://arxiv.org/abs/2003.07704)
2. Marafioti, A., Perraудин, N., Holighaus, N., Majdak, P.: A context encoder for audio inpainting. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* **27**(12), 2362–2372 (2019)
3. Cheddad, A.: Steganoflage: A New Image Steganography Algorithm, Ph.D. thesis School of Computing and Intelligent Systems, Faculty of Computing and Engineering, University of Ulster, United Kingdom (2009)
4. Khan, G.M., Khan, N.M.: Real-time lossy audio signal reconstruction using novel sliding based multi-instance linear regression/random forest and enhanced CGPANN. *Neural Process. Lett.* 1–29 (2020)
5. Lee, B.-K., Chang, J.-H.: Packet loss concealment based on deep neural networks for digital speech transmission. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* **24**(2), 378–387 (2015)
6. Khan, N.M., Khan, G.M.: Audio signal reconstruction using cartesian genetic programming evolved artificial neural network (CGPANN). In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 568–573. IEEE (2017)
7. Sperschneider, R., Sukowski, J., Marković, G.: Delay-less frequency domain packet-loss concealment for tonal audio signals. In: 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 766–770. IEEE (2015)
8. Mokrý, O., Záviška, P., Rajmic, P., Veselý, V.: Introducing SPAIN (SParse Audio INpainter). In: 2019 27th European Signal Processing Conference (EUSIPCO), pp. 1–5 (2019). <https://doi.org/10.23919/EUSIPCO.2019.8902560>
9. Kitic, S., Bertin, N., Gribonval, R.: Sparsity and cosparsity for audio declipping: a flexible non-convex approach. In: Proceedings: 12th International Conference on Latent Variable Analysis and Signal Separation. Liberec, Czech Republic (2015)
10. Janssen, A.J.E.M., Veldhuis, R.N.J., Vries, L.B.: Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes. *IEEE Trans. Acoust., Speech Signal Process.* **34**(2), 317–330 (1986)
11. Hasannezhad, M., Zhu, W.-P., Champagne, B.: A novel low-complexity attention-driven composite model for speech enhancement. In: IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5. IEEE (2021)
12. Perraудин, N., Holighaus, N., Majdak, P., Balazs, P.: Inpainting of long audio segments with similarity graphs. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* **26**(6), 1083–1094 (2018). <https://doi.org/10.1109/TASLP.2018.2809864>. June
13. Mokrý, O., Rajmic, P.: Audio inpainting: revisited and reweighted. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* **28**, 2906–2918 (2020). <https://doi.org/10.1109/TASLP.2020.3030486>
14. Floyd, R.W., Steinberg, L.: An adaptive algorithm for spatial greyscale. *Proc. Soc. Inf. Disp.* **17**(2), 75–77 (1976)

15. Kim, T.H., Park, S.I.: Deep context-aware descreening and rescreening of halftone images. *ACM Trans. Graph.* **37**(4), 1–12 (2018)
16. Li, Y., Huang, J.B., Ahuja, N., Yang, M.H.: Deep joint image filtering. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *Computer Vision—ECCV’16*. 2016. Lecture Notes in Computer Science, vol. 9908. Springer, Cham (2016)
17. Dasari, S.K., Cheddad, A., Andersson, P.: Random forest surrogate models to support design space exploration in aerospace use-case. In: MacIntyre, J., Maglogiannis, I., Iliadis, L., Pimenidis, E. (eds.) *Artificial Intelligence Applications and Innovations. AIAI 2019*, vol. 559, pp. 532–544. Springer, Berlin (2019)
18. Dasari, S.K., Cheddad, A., Andersson, P.: Predictive modelling to support sensitivity analysis for robust design in aerospace engineering. *Struct. Multidiscip. Optim.* **61**, 2177–2192 (2020)
19. Espinosa, R., Palma, J., Jiménez, F., Kamińska, J., Sciavicco, G., Lucena-Sánchez, E.: A time series forecasting based multi-criteria methodology for air quality prediction. *Appl. Soft Comput.* **113**, 107850 (2021)
20. Sun, L., Du, J., Dai, L., Lee, C.: Multiple-target deep learning for LSTM-RNN based speech enhancement. In *Proceedings: Hands-Free Speech Communications and Microphone Arrays (HSCMA’17)*, pp. 136–140. IEEE, San Francisco (2017)
21. Graves, A.: Long short-term memory. In: *Supervised Sequence Labelling with Recurrent Neural Networks. Studies in Computational Intelligence*, vol. 385. Springer, Berlin (2012)
22. Yogarajah, P., Condell, J., Curran, K., McKevitt, P., Cheddad, A.: A dynamic threshold approach for skin tone detection in colour images. *Int. J. Biom.* **4**(1), 38 (2012)
23. Lilly, J.M.: Element analysis: a wavelet-based method for analysing time-localized events in noisy time series. *Proc. R. Soc. A: Math., Phys. Eng. Sci.* **473**(2200), 20160776 (2017). <https://doi.org/10.1098/rspa.2016.0776>
24. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imag. Vis.* **40**, 120–145 (2011)
25. Combettes, P., Pesquet, J.: Proximal splitting methods in signal processing. *Fixed-Point Algorithms Inverse Probl. Sci. Eng.* **49**, 185–212 (2011)
26. Adler, A., Emiya, V., Jafari, M.G., Elad, M., Gribonval, R., Plumley, M.D.: Audio inpainting. *IEEE Trans. Audio, Speech, Lang. Process.* **20**(3), 922–932 (2012)
27. Donahue, C., McAuley, J., Puckette, M.: Adversarial audio synthesis. In: *Proceedings of International Conference on Learning Representations (ICLR)* (2019)
28. Thiede, T., Treurniet, W., Bitto, R., Schmidmer, C.: PEAQ—The ITU standard for objective measurement of perceived audio quality. *J. Audio Eng. Soc.* **48**(1/2) (2000)
29. Kabal, P.: An Examination and Interpretation of ITU-R BS.1387: Perceptual Evaluation of Audio Quality. Dept. Electrical and Computer Engineering, McGill University, TSP Lab Technical Report (2002)
30. Huber, R., Kollmeier, B.: PEMO-Q-a new method for objective audio quality assessment using a model of auditory perception. *IEEE Trans. Audio Speech Lang. Process.* **14**(6), 1902–1911 (2006)
31. Hansen, M., Kollmeier, B.: Objective modelling of speech quality with a psychoacoustically validated auditory model. *J. Audio Eng. Soc.* **48**(5), 395–409 (2000)
32. Luo, Y., Mesgarani, N.: Conv-TasNet: surpassing ideal time-frequency magnitude masking for speech separation. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* **27**(8), 1256–1266 (2019). Aug.



Designing Cancellation Intervention System with Sliding Lead Times

Yong Seog Kim^(✉)

Utah State University, Logan, UT 84322, USA
yong.kim@usu.edu

Abstract. In this paper, we propose a new cancellation intervention system to minimize possible revenue loss of business entity in tourism sector from last-minute booking cancellation. The proposed system automatically sends e-reminders to travelers who are most likely to cancel their bookings using calibrated prediction models on the subsets of bookings with different lead times. In particular, cost-sensitive learning methods with varying class weights in machine learning community are adopted to overcome hurdles from imbalanced class distributions. Finally, this study introduces cumulative gain charts to provide general guidelines on how to maximize the expected benefits from the proposed system.

Keywords: Booking cancellation · Cancellation intervention system · Machine learning · Cost-sensitive learning · Cumulative gain chart

1 Introduction

It is inevitable that tourism sector suffers from various factors that influence travelers to cancel their bookings due to the abrupt changes of business plans, weather conditions, political turmoil in destination places, Covid-19 like pandemic regulations, or personal health—or family-related reasons [1–3]. The abrupt cancellation of bookings or no-shows negatively affect the revenue of business entities in tourism sector [4, 5]. Therefore, numerous studies have developed various models to forecast the volume of tourism demand or estimate the probability of which bookings are most likely to be cancelled in advance.

The first notable such attempt is observed in several studies that investigate the key determinants and estimate the probabilities of cancellations based on identified determinants. In particular, several studies report few traveler-related variables that significantly impact on booking cancellations. For example, few studies [6, 7] report that, because of cultural and personal differences in attitudes towards uncertainty, Asian travelers are more likely to change their travel plans and cancel their bookings than those from other countries. In another study [8], the authors recommend that business entities in tourism sector need to identify and communicate with highly attractive segments of tourists who are resistant to external or internal crisis events.

Booking-related variables such as length of stay, planning horizon, travel distance, booking and arrival day of the week, arrival month and arrival year and room category

are also known to affect the likelihood of booking cancellations [9]. According to a study [10], bookings with longer stay are likely to have lower cancellation rates because travelers are more careful for their travel plan. It is also expected that bookings with shorter travelling distances are more likely to be cancelled because they are typically domestic trips with lower cost and priority than international travel plans with more exotic and top priority [11].

Strong relationship between booking cancellations and reservation channels is also reported [12]. According to this study, cancellation rates are found to be highest in online bookings followed by offline bookings and travel agency bookings due to the fact that online travel agent sites often provide the free cancellation option. This finding is consistent with the fact that more business entities in tourism sector promote the best deals to deal-seeking travelers via online travel agents, which encourages travelers to place several bookings first and then cancel the rest except the most convenient deal [4]. This is particularly true as web portals supported by information technologies make travelers easily access cancellation policy or pricing strategy of one business entity and compare them with those from other entities.

It is also shown that bookings involving family or children are less likely to be cancelled because tourism is considered as a social activity with the purpose of strengthening kinship [13, 14]. Similarly, groups with less homogeneous attitudes or with different preferences among members are likely to be canceled [9, 15]. It is also reported that bookings with longer lead times are more likely to be canceled due to their higher exposure to unexpected internal and external risks that are uniformly distributed over lead times [10].

The ultimate goal of afore-mentioned studies is to provide revenue managers in tourism sector with descriptive and predictive models to profile which bookings are most likely to be cancelled so that they operate cancellation policies to minimize revenue loss from last-minute cancellations or no-shows. While there is no booking cancellation policy universal for all business entities in tourism sector, a prevailing principle of such policies is to impose higher cancellation fees if bookings are canceled at times closer to the date of arrival. Several studies [4, 16] report that cancellation policies often increase the revenues of hotels by 8% or decrease no-show rate to a lower rate of 5%.

We believe that key determinants of cancellation and prediction models based on them are very useful to describe the common characteristics of canceled bookings. We, however, believe that these determinants and current prediction models based on them are not very useful as a means of deterring booking cancellation or minimizing revenue loss from unexpected cancellation. For example, consider the fact that cancellation rates are found to be higher in online bookings compared with offline bookings and travel agency bookings. However, it is almost inconceivable to suggest a cancellation policy that ask revenue managers to encourage travelers to make bookings only through offline or travel agency instead of online agency.

A similar argument can be made on lead times. While several studies report that bookings with longer lead times are more likely to be canceled, it is very difficult for business entities to use lead times to curb high cancellation rates in bookings with longer lead times. That is, business entities in tourism sector cannot deny bookings with long lead times because they are most likely to be canceled. Nor can they accept only

bookings with short lead times. Of course, it is possible for them to impose a heavier cancellation fees on canceled bookings with short lead times to discourage the last-minute cancellation. However, even in such circumstances, many business entities do not intend to impose any cancellation fees as long as bookings are canceled at least two business days before the arrival date, invalidating the usefulness of lead times to control cancellation rates.

Understanding these subtle limitations of lead times (or booking channels) is very critical. That is, while many prior studies identify them as one of the most descriptive variables to predict which bookings are likely to be canceled and even use them to estimate the cancellation probability of given bookings, we emphasize their limitations within the newly proposed cancellation intervention system. Therefore, instead of using lead times variable directly in prediction models, we used it to divide the entire booking datasets into few subsets with distinct lead times so that prediction models can be calibrated on each subset independently. In addition, we propose that prediction models should focus on “identifying” as many canceled bookings as possible with recall (or sensitivity) metric, not on “predicting” both canceled and non-canceled bookings correctly with accuracy metric.

This study contributes to the literature by making theoretical and practical contributions that help revenue managers minimize revenue loss from unexpected cancellation or no-shows. First, we calibrate prediction models separately for each subset of booking data with varying values of lead times so that models are able to estimate the probability of being canceled more accurately and identify as many canceled bookings as possible. In addition, as we will present in following sections, each subset of bookings displays very different (often severely imbalanced) class distributions. Therefore, we adopt an alternative way of calibrating prediction models using cost-sensitive learning [17, 18], where different class weights (or cost metrics) are used for booking instances in canceled or non-canceled classes so that misclassifying canceled bookings can be considered more critical within prediction models. Finally, this research framework for developing a separate prediction model for each subset is an extension of studies [19], proposing that cancellation policies that incorporate cancellation deadlines with the appropriate size of the deadlines increment are highly recommended.

The body of this paper is organized as follows. First, we begin with a brief literature review on several studies on overbooking, cancellation policies, and pricing wars that many business entities in tourism sector adopt to avoid revenue loss. Then multiple studies and prediction models therein for forecasting tourism demands or booking cancellation are summarized with their limitations to highlight the needs of this study. Then, we introduce our main dataset and share several steps we go through for data cleansing and engineering. We will then present calibrated models with class weights for each subset of bookings with varying lead times and provide managerial suggestions on how to incorporate such models within the cancellation intervention system. Finally, we will address limitations and possible research directions.

2 Literature Review on Cancellation Policies and Forecasting

Multiple internal and external factors make it very difficult for business entities to forecast tourism demand and schedule room occupancy, which, in return, affect negatively more fundamental tasks like budget planning and revenue management. As an effort to reduce such risks, business entities in tourism sector implement various risk management strategies.

Overbooking strategy intends to minimize risks by accepting reservations above the capacity of the business entity to prepare for cases that some bookings will be cancelled [20]. Several studies report that overbooking policy is a very common practice to reduce the revenue loss from last minute cancellations [21, 22]. Note, however, that overbooking strategy may incur extra costs such as guest compensation, relocation, or even negative reputation when the business unit cannot accommodate overbooked demands [23, 24]. The author in [5] investigates optimal room allocation policies to maximize the expected sales assuming business entity operates early discount service and for overbooking policy. Not only does they find optimal room allocations for early discount and the number of overbookings, they also derive an optimal allocation for early discount without cancellations and overbookings.

As business entities in tourism sector are more reluctant to overbooking strategy due to customer relation concerns, they operate various cancellation policies with different cancellation fees and deadlines to discourage last minute cancellations and no-shows [4]. While cancellation policies may discourage cancellations for certain bookings, charging a cancellation fee for canceled booking may present negative impression on service quality and increase travelers' risk perception, which makes the business entity less desirable from travelers' perspective [10, 25, 26]. Therefore, [27] divides U.S. hotels into two mutually exclusive segments with unique characteristics through a two-step cluster analysis and propose that differences in hotel characteristics should be considered for setting cancellation policies with cancellation fees and deadlines. Other strategies such as price wars can be considered although they may bring negative impact on the business sustainability in the long-term [28].

Regardless of what cancellation policies or pricing strategies business entities adopt, it is imperative to estimate reliably and accurately future demand or, more preferably, which bookings are most likely to be cancelled based on the estimated probability of cancellation using prediction models. Once business entities can reduce uncertainty in estimating future demand, they can operate their resources more efficiently by optimizing operations and minimizing costs of idle resources.

To this end, a stream of studies adopts various time-series and simulation models including the Autoregressive Integrated Moving Average model (ARIMA), seasonal ARIMA model (SARIMA) and space-time ARMA model (STARIMA) model [29–31] and Monte Carlo simulation [32, 33] to predict future tourist demand exploiting vast amount of historical data. Therefore, their usage for the cancellation intervention system is somewhat limited because the system requires control instruments to curb cancellation at each booking level, not at aggregated booking level.

Another stream of studies adopts various machine learning and data mining models such as artificial neural network (ANN), support vector machine (SVM) and decision tree (DT), to forecast tourism demand with great success [34–37]. Unlike these studies, other

studies focus on forecasting booking cancellation (rates) with various models [10, 38]. For example, [10] formulates booking cancellation prediction as a classification problem using booking-related variables such as number of previous bookings not cancelled, previous stays, distribution channel or days of week of booking dates, then calibrate several DT-, ANN- and SVM-variant models and report high success with accuracy results in excess of 90%.

Several more complex and advanced hybrid models return greater success than a single model. For example, [39] uses genetic algorithms (GA) as a wrapper around ANN to configure the best set of ANN structures to forecast the sales revenue of a travel agency. Two other studies [40, 41] obtain better performance by using a hybrid model of a neural network with fuzzy time series or Grey–Markov models to forecast the number of tourists. In particular, [42] reports that their hybrid model of genetic algorithms (GA) and ANN returns the best performance, significantly out-performing the 2nd best model (i.e., Random Forest (RF)) in terms of accuracy (98 vs. 80%), recall (98.7 vs. 79.9%), F1 score (97.9 vs. 80.9%). Most complex models in which deep learning models such as long short-term memory (LSTM) and its variants are combined with other models can be found in recent studies [43, 44].

While all these machines learning and data mining models can be used to estimate the probability of cancellation for each booking so that a set of bookings that are most likely to be canceled are selected and managed with care through the cancellation intervention system, almost all prior studies use these models as one of many candidates to find the best model in terms of accurately predicting both canceled and non-canceled bookings. In addition, ANN (and hybrid models with ANN and deep learning models such as LSTM) is often-considered a black-box algorithm because its complex final structure for forecasting makes it very difficult to make casual inferences [45].

Only few studies proposed the importance of dynamic nature of cancellation policies and revenue management. For example, a revenue management model based on six key drivers such as organizational culture, demand forecasting, dynamic distribution channels, dynamic and customized pricing and daily reviewing is proposed to improve efficiency during the downturn like during Covid-19 [46]. In another study [38], authors calibrate several machine learning models to explicitly consider the time-dependency of the importance of input variables. That is, according to them, the impact of input variables on the probability of being canceled is dependent on the different stages of the booking horizon. Therefore, they calibrate different models on different subsets of bookings over time t . In [19], multiple models are calibrated to forecast future reservations at 3 different horizons of 7 days, 14 days, 30 days prior to the date of stay, and report that ANN performs relatively well in terms of Mean Absolute Error (MAE) and Mean Squared Error (MSE) against traditional models including Regression and Logistic Regression even if booking window shifts.

While we share the idea of building multiple models for different subsets of bookings, this study is very different from prior studies whose primary purpose is to forecast the overall cancellation rates or maximize the value of accuracy. In contrast, this study intends to calibrate dynamic models on each subset of bookings with the imbalanced class distributions (i.e., less than 10% of canceled bookings) with cost-sensitive learning and estimate the probability of being canceled for each booking while identifying as many

canceled bookings as possible (i.e., maximizing recall value of canceled bookings). Note that models with higher classification accuracy usually are not good at providing probability estimates for each class [47].

3 Data Description and Preliminary Analysis

3.1 Data Description, Data Cleansing, and Data Engineering

The study uses datasets obtained from a study [48] that describes two datasets with hotel booking data. These datasets include booking records to arrive between the 1st of July of 2015 and the 31st of August 2017 for a resort hotel (40,060 records) and a city hotel (79,330 records) located in Lisbon, Portugal. These datasets include both bookings arrived and canceled along the common 31 variables (except an additional “hotel” variable which is included to indicate two types of hotel) shown in Table 1. Since this study provides a minimal description of datasets for self-containment purpose, readers who like to have a full description of the dataset are strongly suggested to refer to the original study [48].

The first step of data cleansing and engineering we take is to remove variables with little predictive value due to its ID-related categorical values (e.g., “company” and “agent”). Few other variables such as “country” (because of too many categorical values for prediction models), “reservation_status_date” and “reservation_status” (because of redundant information with “is_canceled”) are also removed from further analysis. Next all records with missing and duplicated values are removed. Several other minor cleaning steps are also implemented. For example, “meal” variable with “undefined” values are replaced with “SC (self catering).” In addition, few derived columns are created: “kids” variable as the sum of “children” and “babies,” “total_members” variable as the sum of “kids” and “adults”, and “total_nights” variable as “stays_in_week_nights” and “stays_in_weekend_nights.” Finally, each value of the chosen categorical variable, a dummy variable is created to check their relationships with a class variable, “is_canceled.”

As a result of data cleansing and engineering, the final dataset includes 86,534 records across 78 numerical and dummy variables. This dataset is randomly divided into two parts, a training set (70% of the original dataset, resulting in 60,573 records) and a test set (25,961 records) to calibrate the prediction model and evaluate the performance of the calibrated model, respectively.

3.2 Identifying the Determinants of Booking Cancellation

As stated in the previous introduction and literature review section, many studies have been carried out to estimate the probability of or identify the key determinants of booking cancellation. While this study does not intend to replicate the exactly same findings or calibrate better prediction models than those from prior studies, it is necessary to re-visit major findings from those studies in this section to extend them with our own unique objectives.

As the first step to validate findings from prior studies, this study estimates the importance of input variables in terms of their usefulness to predict whether or not a

Table 1. List of variables.

Variable	Type	Description
Hotel	Categorical	Hotel types: Resort or City
is_canceled	Categorical	If the booking was canceled (1) or not (0)
lead_time	Integer	Number of days between booking and arrival date
arrival_date_year	Integer	Year of arrival date
arrival_date_month	Categorical	Month of arrival date
arrival_date_week_number	Integer	Week number of the arrival date
arrival_date_day_of_month	Integer	Day of the month of the arrival date
stays_in_weekend_nights	Integer	Number of weekend nights the guest booked
stays_in_week_nights	Integer	Number of week nights the guest stayed or booked
Adults	Integer	Number of adults
Children	Integer	Number of children
Babies	Integer	Number of babies
Meal	Categorical	Type of meal booked
Country	Categorical	Country of origin
market_segment	Categorical	Market segment designation
distribution_channel	Categorical	Booking distribution channel
is_repeated_guest	Categorical	Booking was from a repeated guest (1) or not (0)
previous_cancellations	Integer	Number of previous bookings cancelled
previous_bookings_not_canceled	Integer	Number of previous bookings not cancelled
reserved_room_type	Categorical	Code of room type reserved
assigned_room_type	Categorical	Code for the type of room assigned to the booking
booking_changes	Integer	Number of changes made to the booking
deposit_type	Categorical	Indication on if the customer made a deposit
Agent	Categorical	ID of the travel agency
Company	Categorical	ID of the company/entity that made the booking
days_in_waiting_list	Integer	Number of days the booking was in the waiting list

(continued)

Table 1. (*continued*)

Variable	Type	Description
customer_type	Categorical	Type of booking
Adr	Numeric	Average Daily Rate
required_car_parking_spaces	Integer	Number of car parking spaces required by the guest
total_of_special_requests	Integer	Number of special requests made by the customer
reservation_status	Categorical	Reservation last status,
reservation_status_date	Date	Date at which the last status was set

certain booking will be canceled. To this end, this study runs a correlation analysis and summarizes a part of its outcome with 10 most important variables in Table 2.

Table 2. Key determinants of booking cancellation.

Correlation analysis		Feature importance score from decision tree	
List of top 10 variables	Score	List of top 10 variables	Score
market_segment_Online TA	0.210	lead_time	0.198
lead_time	0.183	adr	0.124
deposit_type_Non Refund	0.165	arrival_date_day_of_month	0.087
distribution_channel_TA/TO	0.150	total_of_special_requests	0.074
customer_type_Transient	0.128	arrival_date_week_number	0.064
adr	0.126	total_nights	0.043
total_of_special_requests	-0.122	market_segment_Online TA	0.042
market_segment_Offline TA/TO	-0.124	stays_in_week_nights	0.036
deposit_type_No Deposit	-0.156	deposit_type_Non Refund	0.036
required_car_parking_spaces	-0.184	required_car_parking_spaces	0.029

According to Table 2, the most positively correlated input variable with the booking cancellation indicator turns out to be “market_segment_Online TA” (i.e., bookings made via Online Travel Agent (TA) are more likely to be canceled) followed by “lead_time” (i.e., bookings with longer lead times are more likely to be canceled) and “deposit_type_Non Refund” (i.e., bookings with non-refundable deposit are more likely to be canceled). In contrast, the most negatively correlated input variable turns out to be “required_car_parking_spaces” (i.e., bookings with higher number of required parking spaces are less likely to be canceled) followed by “deposit_type_No Deposit” (i.e., bookings with no deposit are more likely to be canceled), “market_segment_Offline TA/TO” (i.e., bookings made via offline TA are less likely to be canceled). Note that all these findings are consistent with findings from prior studies as reviewed in previous sections.

This study also calibrates one of the most popular prediction models, Decision Tree (DT), to obtain the feature importance score based on information gain ratio [49] for each input variable to predict the value of “is_canceled” class. The first 10 input variables with the highest feature importance scores are shown in the second column of Table 2. We immediately note that two methods to identify key determinants of booking cancellation, correlation analysis from traditional statistical analysis and feature importance score from DT, return a very similar list of input variables. For example, several common variables such as “lead_time”, “adr”, “market_segment_Online TA” and “required_car_parking_spaces” are listed as most predictive variables by two methods. However, few variables such as “arrival_date_day_of_month,” “arrival_date_week_number,” “total_nights” and “stays_in_week_nights” are listed as unique key determinants from DT. Overall, however, these findings are very similar and consistent with findings from prior studies.

3.3 Calibrating the Prediction Models of Booking Cancellation

One of the most important components of the newly proposed cancellation intervention system is the accurately calibrated prediction models to estimate the probability of being canceled given input variables. To this end, this study calibrates multiple prediction models including DT, K-Nearest Neighbor (KNN) and ANN with default settings for easy replication of the results reported in this study. This studies also validate few additional models such as Gaussian Naïve Bayes (GNB), Support Vector Machine with linear and non-linear kernel (SVM-linear and SVM-nonliner) [50]. Note that ANN, KNN and DT are based on non-parametric approach, while GNB and SVM take a parametric approach deeply rooted in mathematical and statistical theory.

Out of these models, ANN with scaled input variables performs best with 80% of accuracy followed by SVM-linear model (78%) and DT (75%) while GNB performs worst (57%). The relatively lower performance of GNB model is expected in advance mainly because several input variables are correlated and hence the assumption that they are conditionally independent of each other given the value of the class indicator is not satisfied. Readers who are interested in these algorithms and their introductory tutorials may read several studies and other citations therein [51].

Finally, Random Forest (RF), an ensemble model based on DT classifier [52], is calibrated with 100 DTs with 5 randomly chosen variables used to determine the best split at each intermediate node of each DT. As expected in advance, RF with no additional parameter tuning returns the highest accuracy at 81% by reducing bias and variance errors. In addition, this study discovers that RF and DT models return almost identical list of the most predictive variables. Considering all of these preliminary outcomes, RF model is chosen as the prediction model to estimate the probability of booking cancellation for further analysis. The selection of RF model is also consistent with the finding in a study [53] where RF algorithms family usually return the best performance out of 179 classifiers.

3.4 Missing Links Between Prediction and Cancellation Intervention Systems

While all identified determinants of booking cancellation and prediction models to estimate the probability of booking cancellation in previous section can be useful, this study likes to point out gaps between calibrating prediction models and implementing the cancellation intervention system. Note that the most expected utility of developing prediction models is to accurately identify booking instances that are most likely to be canceled so that the proposed intervention system sends e-reminders to travelers and property units of canceled booking can be entered the reservation system to minimize the loss from last-minute cancellations or no-shows.

For example, consider lead_time variable, which is identified as one of the most predictive variables to predict booking cancellation. Also consider a very well-established cancellation policy in tourism sector that imposes a certain amount of cancellation fees on all canceled bookings unless they are cancelled at least two business days before the arrival date. While this cancellation policy may indeed discourage last-minute cancellation, it does not prevent from revenue loss due to booking cancellation. In fact, it implies that travelers can cancel their bookings anytime without fees as long as they cancel them at least two business days before the arrival date (i.e., $\text{lead_time} > 2$). To supplement this cancellation policy, this study proposes a new cancellation intervention system as shown in Fig. 1.

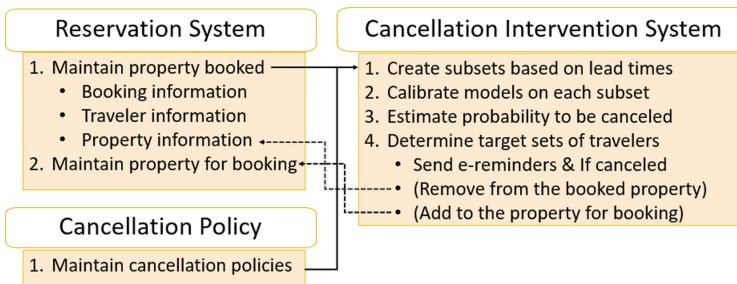


Fig. 1. Proposed cancellation intervention system.

The proposed system will receive a copy of booking data (denoted as solid arrow lines) from the reservation system and create subsets of bookings based on (cancellation fees and) lead times from the current cancellation policy. Then, the system will calibrate prediction models on each subset to estimate the probability of being canceled and determine target sets of travelers to send e-reminders. Note that optimal sets of travelers can vary depending on the characteristics of subsets with lead times. Once travelers respond to e-reminders with their confirmation to cancel their bookings, the intervention system sends requests (denoted as dotted arrow lines) to remove properties associated with canceled bookings from booking information and add them back to the list of properties available for bookings to the reservation system.

The core function of the intervention system allows operators of the system to send out e-reminders to a set of travelers who are most likely to cancel their bookings. The e-reminders ask the recipients to respond only if they like to cancel their bookings within

few days specified. To encourage the response from travelers, the contents of e-reminders may include the exemption of cancellation fees or a discount coupon for future booking if they respond back with booking cancellation t days earlier (say, $t = 4$ days before) than the arrival dates of original bookings. If travelers respond to e-reminders with booking cancellation, operators of the system may inform managers of reservation system so that they accept new booking on the cancelled property units, thus expecting additional revenues.

Fundamentally, this new intervention system utilizes extra times obtained by asking travelers to reveal their desire to cancel their booking in advance, rather than waiting until the last moment. Then this extra time will be used to place canceled property units back into a reservation system for accepting new bookings on them.

Another requirement of the new intervention system is to allow operators of the system to determine the optimal proportion of travelers to contact with e-reminders. That is, the performance of prediction models should not be based on the overall accuracy of predicting over all booking instances. Instead, it should be able to be updated incrementally over a subset of booking instances dependent on varying values of lead_time. To this end, it is necessary to divide the entire datasets into s different subsets (in this study, s is set to 3) depending on the value of lead_time. The first subset, $D_{<2}$, includes all records whose lead_time value is less than 2 from the current time t . The other two subsets, $D_{2 \leq t \leq 4}$ and $D_{>4}$, are a set of booking records whose lead_time is between 2 and 4, and lead_time is greater than 4, respectively.

Note that the assumed cancellation policy imposes fees only on canceled bookings with $\text{lead_time} \leq 2$. Therefore, for the first subset, $D_{<2}$, it may not be necessary to calibrate prediction models and implement the intervention system due to very limited time for travelers to respond and operators of the system to react. However, it is still possible to accrue additional revenue from new bookings on canceled property units as long as they are canceled before the arrival date. In terms of an optimal portion of travelers for e-reminders from this subset of booking instances, it is recommended to target larger portion of travelers (or booking instances) than in other subsets of booking instances due to the smallest number of subset size and narrowest time window to implement the intervention system.

For the subset of bookings with lead_time greater than 4 (i.e., $D_{>4}$), it is recommended to target a smaller portion of booking instances only with high probability to cancel because of large number of booking instances and response rate to e-reminders is relatively low. The impact of the intervention system can be maximized on the subset of bookings with lead_time between 2 and 4 (i.e., $D_{2 \leq t \leq 4}$) because these bookings may have at least few more days to accept new bookings once travelers notify their intention to cancel their bookings. The optimal scope of targeting on this subset of booking instances would be somewhere in the middle of those of two other subsets of booking instances.

4 Intervention Strategy on the Bookings with Short Lead Times

4.1 Calibrating Prediction Models with Class Weights

$D_{<2}$ contains a total of 8,914 records whose lead_time is less than 2 and, among these records, only 549 records are canceled bookings, resulting in a 6% of cancellation rate. To calibrate a RF model, the first 50% of $D_{<2}$ records are used as a training set (4,457 records with 266 cancelled bookings) and the second 50% are reserved for a test set (4,457 records with 283 cancelled bookings). Note that a relatively higher portion (50%) of dataset is reserved for a test set due to a very small number of records of canceled bookings.

The calibrated RF model with typical training process returns 93% of accuracy, which can be considered very accurate at first sight. However, this performance is very mis-leading because the calibrated RF model simply predicts almost all records as “non-canceled (Class 0),” which results in about 93% of accuracy corresponding to the proportion of records in Class 0. This claim can be easily verified in terms of a very low recall value (0.02, that is, only 2% out of all canceled records are identified) for “canceled” (Class 1) and a very high recall value (1, that is, 100% of all non-canceled records are identified) for “non-canceled” (Class 0) class.

These findings imply that two important modifications are needed to calibrate prediction models for the intervention system. First, accuracy metric should not be used as an appropriate evaluation metric any longer because of the severely imbalanced class distributions. In addition, since the RF model should identify as many canceled bookings as possible for the new intervention system, recall (=sensitivity = the proportion of correctly identified canceled bookings) metric should be used as a more appropriate metric.

Second, a new training method is also needed to calibrate models on datasets with severely imbalanced class distributions. One of the most popular ways of training prediction models in such a case is to assign different class weights on records in different classes [17, 18]. For example, if a dataset consists of records in two classes (e.g., 100 records in Class 0, and 1 record in Class 1), it is reasonable to set a weight of 1 on records in Class 0 and a weight of 100 on records in Class 1 so that correctly identifying records in Class 1 is perceived as 100 times more important as correctly identifying records in Class 0.

However, since any set of class weights can be used, this study adopts GridSearchCV module in Python to find the optimal set of parameters for RF model of 100 DTs while varying class weights to maximize the recall value of canceled bookings. The best performance of the RF model is calibrated on the training set when the class weights of 1:100 for non-canceled and canceled bookings is used with additional sets of parameters to control the grow of DTs. The optimally configured RF model significantly improves the recall value of canceled bookings from 2 to 92%.

4.2 Implications of Prediction Models on Cancellation Intervention Strategy

For the successful implementation of the proposed intervention system, it is imperative to visually display how the calibrated model performs in terms of recall metric as top

x % of bookings that are most likely to be canceled are chosen. For this purpose, this study utilizes the cumulative gain chart, which displays the cumulative recall values on y-axis across chosen proportion of booking records sorted in the descending order of probability to be canceled on x-axis. Ideally, the cumulative gain chart of the highly performing model should be able to identify 100% of canceled bookings using only small proportion of booking records, which makes it plotted closer to upper left corner.

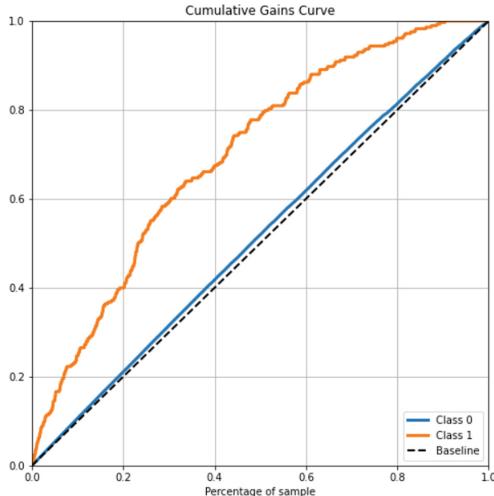


Fig. 2. Cumulative gain chart from $D_{<2}$.

The dotted line in Fig. 2 shows the cumulative recall value of canceled booking records (i.e., the cumulative % of identified canceled bookings) as x % of bookings are randomly selected, which will be always represented as a diagonal line. Therefore, it is expected that 20% of canceled booking will be identified when 20% of bookings are randomly selected. In contrast, the cumulative gain chart in orange color in Fig. 2 shows the cumulative recall value of canceled bookings as top x % of bookings that are most likely to be canceled are chosen by the calibrated RF model. According to Fig. 2, when 20% of bookings that are most likely to be canceled are selected, 40% of canceled bookings are identified. Similarly, when 30% of bookings that are most likely to be canceled are selected, 60% of canceled bookings are identified.

The usefulness of this cumulative gain chart with the new cancellation intervention system is that it helps operators of the intervention system determine how many e-reminders should be sent to travelers who would or would not notify whether they cancel their bookings. In a hypothetical situation, operators may want to send out as many e-reminders as possible to identify more (to-be) canceled bookings. However, in such a case, they may face very high tangible (e.g., any financial incentives offered to encourage travelers' reply and operational costs to take care of more volumes) and intangible costs (i.e., customer dissatisfaction due to unwanted e-reminder). In another situation, decision makers may want to send out e-reminders to only top 20% of bookings that are most

likely to be canceled, while being satisfied with the fact that they may identify only 40% of canceled bookings and they are able to accept new bookings on only those booking units.

As a general guideline, this study suggests a small top $x\%$ of bookings only (e.g., 20%) on $D_{<2}$ not only because the cancellation rate is very low (about 6%) in this subset of booking instances with a very short lead time (< 2 days) but also because there is a small window of opportunity to rent canceled booking units.

5 Intervention Strategy on the Bookings with Longer Lead Times

5.1 Calibrating Prediction Models with Class Weights

This section presents the process of calibrating RF models with class weights on two subsets of bookings with intermediate ($D_{2 \leq t \leq 4}$) and long lead times ($D_{>4}$). $D_{2 \leq t \leq 4}$ consists of 4,617 non-canceled and 499 canceled bookings, resulting in a 9.7% of cancellation rate. The first half of randomly split records of $D_{2 \leq t \leq 4}$ is used as a training set (2,558 records with 240 cancelled bookings) and the second half is reserved for a test set (2,558 records with 259 cancelled bookings).

As noted in analysis on $D_{<2}$, a RF model trained without considering imbalanced class distributions will result in about 90% of accuracy (and 4% of recall) by predicting all bookings as non-canceled bookings due to severely imbalanced class distributions. Another GridSearchCV module is implemented to search for an optimal set of class weights for the RF model of 100 DTs while maximizing the recall value of canceled bookings. The best performance of the RF model is found with the class weights of 1:100 for non-canceled and canceled bookings, returning significantly improved recall value of canceled bookings from 4 to 99%.

The identical steps are taken to calibrate prediction models on $D_{>4}$ that contain 49,661 non-canceled and 22,843 canceled bookings, resulting in a 31.5% of cancellation rate. Note that $D_{>4}$ has a much-balanced class distribution than two other subsets of bookings because it contains many bookings with long lead times and bookings with long lead times has higher chance of being canceled.

As in two other subsets of bookings, the first half of randomly split records in $D_{>4}$ is used as a training set (36,552 records with 11,384 cancelled bookings) and the other half is reserved for evaluation purpose (36,252 records with 11,459 cancelled bookings). Again, a GridSearchCV module is executed on the training set to configure the set of parameters for RF model of 100 DTs while varying class weights to optimize the recall value of canceled bookings. The best performance of the RF model is found when the class weights of 1:100 for non-canceled and canceled bookings is used with additional sets of parameters to control the grow of DTs. The calibrated RF model improves the recall value of canceled bookings from 54% to 100%.

5.2 Implications of Prediction Models on Cancellation Intervention Strategy

The cumulative gain charts from $D_{2 \leq t \leq 4}$ and $D_{>4}$ are presented in Fig. 3a and b, respectively. According to Fig. 3a, when top 20 and 40% of bookings that are most

likely to be canceled are selected, 47 and 77% of canceled bookings in the test set of $D_{2 \leq t \leq 4}$ are identified, respectively. In fact, this cumulative chart of Class 1 (canceled bookings) displays a better performance of the calibrated RF model on $D_{2 \leq t \leq 4}$ than the RF model on $D_{>4}$.

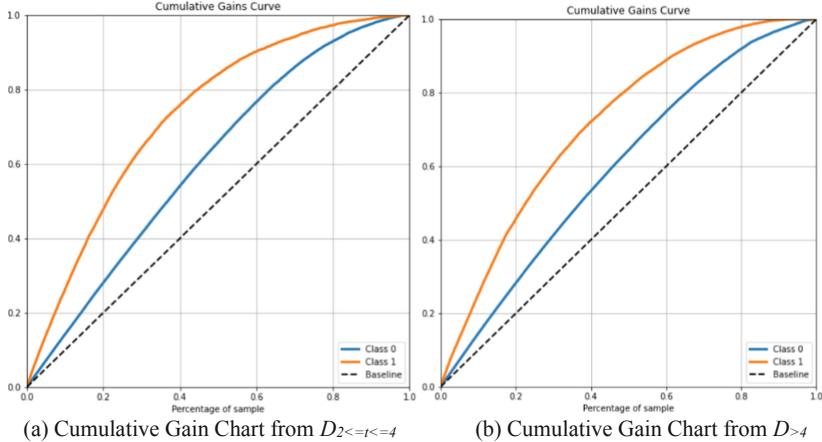


Fig. 3. Cumulative gain chart for bookings with longer lead time.

In particular, this study pinpoints that the maximum benefit from the intervention system would be realized from $D_{2 \leq t \leq 4}$. Note that the success of the intervention system depends on how fast travelers will respond to e-reminders and hence how much extra time reservation managers will have to accept new bookings on canceled property units. Since bookings in $D_{2 \leq t \leq 4}$ has lead times at least two days longer than bookings in $D_{<2}$, reservation managers are likely to have more times to place canceled units into reservation system to accept new bookings, which will result in a higher make-up for revenue loss due to unexpected cancellation.

Still, it is important to determine how many bookings should be targeted for e-reminders. As general guidelines for effective operation of the intervention system, this study suggests that a significant portion of bookings (e.g., between 40 and 60%) on $D_{2 \leq t \leq 4}$ that are most likely to be canceled should be targeted to generate e-reminders. This is mainly because bookings in $D_{2 \leq t \leq 4}$ show a higher cancellation rate and hence it is reasonable to target more bookings to identify canceled bookings. In addition, travelers are more likely to respond to e-reminders because they want to avoid cancellation fees if they do not act quickly and hence property managers may have ample opportunity to accept new bookings on canceled property.

Figure 3b shows the cumulative gain chart of the RF model on $D_{>4}$. According to Fig. 3b, when top 20 and 40% of bookings that are most likely to be canceled are targeted, 45 and 74% of canceled bookings in the test set of $D_{>4}$ are identified, respectively. This is a significant improvement over a random baseline model, resulting in an improvement of 2.25 (=45/20%) and 1.85 (=74/40%), respectively. All these findings validate that the

calibrated RF model is able to identify canceled bookings quite well and can be used with the intervention system.

Note also that there exists a wider window of opportunity to accept new bookings on canceled property units because all bookings in $D_{>4}$ have at least 4 days of lead time. Therefore, if the intervention system sends e-reminders to travelers who are most likely to cancel and they indeed respond to e-reminders promptly, the impact of the intervention system can be significant for bookings in $D_{>4}$. However, travelers of bookings in $D_{>4}$ are less likely to respond to e-reminders because no cancellation fees will be imposed on their bookings. In addition, travelers may not respond to e-reminders because they would not know for sure if they have to cancel their bookings due to great uncertainty associated with longest lead times of their bookings.

Considering all these observations, this study suggests that only a small portion of bookings (e.g., less than 20%) on $D_{>4}$ that are most likely to be canceled should be targeted to generate e-reminders to travelers in the intervention system. This suggestion may sound counter-intuitive considering the fact that bookings in $D_{>4}$ not only contains the largest number of booking records but also shows the highest rate of cancellations (31.5%). However, with all bookings with longer lead times in $D_{>4}$, it is easy to imagine that travelers are less likely to respond to e-reminders due to various internal and external uncertainties they perceive during longer lead times. Finally, since this subset of booking records contains a majority of booking records, even targeting a small portion of booking records may result in very higher tangible and intangible costs.

6 Conclusion and Future Work

This paper presents a new cancellation intervention system to supplement widely adopted cancellation strategies (e.g., cancellation fees and overbooking policies) and minimize the possible loss from last-minute cancellation or no-shows. This new cancellation intervention system automatically sends e-reminders to the travelers of bookings that are most likely to be canceled, which can be identified with the help of the calibrated prediction model. In particular, prediction RF models are calibrated independently on the different subsets of bookings with varying intervals of lead times using cost-sensitive learning scheme with different class weights to offset biased learning tendency of prediction models toward majority class within imbalanced class distributions. At the same time, this study introduces cumulative gain charts that visually present the cumulative recall values of canceled bookings class over top $x\%$ of bookings chosen. Finally, this study provides several general guidelines on how to maximize the expected benefits from the proposed system for different subsets of bookings using cumulative gain charts.

This new intervention system is based on the fact that while few key determinants of booking cancellation identified from prior studies (e.g., lead time or online travel agent) cannot be very useful for the purpose of mitigating the revenue loss from last minute cancellation. This observation is very important from practitioners' perspective because the ultimate goal of predictive/descriptive analytics with machine learning algorithms is to apply meaningful patterns from analytics to practical application for real business problems. Therefore, for example, simply identifying lead time as one of variables with the highest correlation score with the indicator of booking cancellation does not bear any practical values.

From theoretical perspective, this study re-confirms the strong need of incorporating class weights to remedy severe imbalance of class distributions to calibrate prediction models. At the same time, the most popular metric, accuracy should not be used to evaluate the performance of the prediction models in such a case. Instead, recall (the proportion of correctly predicted records of target class out of entire records of target class) or precision (the proportion of correctly identified records of target class out of entire records predicted to be target class) should be used after considering relative costs of misclassification. This study uses recall mainly because the success of the intervention system is dependent on identifying as many bookings to be canceled as possible while avoiding to misclassify canceled booking cases as non-canceled booking cases.

Several directions to extend the current research are possible. The most obvious step to take next is to extend the current model into a dynamic intervention system that can be applied continuously to new sets of booking records. That is, the current system is designed to split the entire booking records into three subsets of bookings based on three lead times (i.e., < 2 , ≤ 4 , and > 4 days) from t and apply the intervention program for each subset. However, the proposed system should be re-applicable to new bookings from near future (say, $t + 7$ days), which can be dependent on the needs of business entity. Another simple extension of the current research is to explore the impact of changes of three lead times to define subsets of booking records (e.g., from < 2 , ≤ 4 , and > 4 days to < 3 , ≤ 7 , and > 7 days).

The most significant extension of the current research is to model the probability of receiving travelers' responses to e-reminders generated from the intervention system. For example, it will be interesting to see if travelers from a specific geographical area, demographics, or booking related characteristics that may affect travelers' willingness to respond to e-reminders. At the same time, it may be also interesting to incorporate the probability of renting canceled property successfully to assess the impact of the proposed intervention system.

References

1. Deyá-Tortella, B., Leoni, V., Ramos, V.: COVID-led consumption displacement: A longitudinal analysis of hotel booking patterns. *Int. J. Hosp. Manag.*, **107**, Article 103343, (2022)
2. Samitas, A., Asteriou, D., Polyzos, S., Kenourgos, D.: Terrorist incidents and tourism demand: Evidence from Greece. *Tour. Manag. Perspect.* **25**, 23–28 (2018)
3. Song, H., Lin, S., Witt, S., Zhang, X.: Impact of financial/economic crisis on demand for hotel rooms in Hong Kong. *Tour. Manage.* **32**(1), 172–186 (2011)
4. Chen, C., Schwartz, Z., Vargas, P.: The search for the best deal: how hotel cancellation policies affect the search and booking decisions of deal seeking customers. *Int. J. Hosp. Manag.*, **30**(1), 129–135 (2011)
5. Koide, T., Ishii, H.: The hotel yield management with two types of room prices, overbooking and cancellations. *Int. J. Prod. Econ.* **93–94**(10), 417–428 (2005)
6. Karl, M.: Risk and uncertainty in travel decision-making: tourist and destination perspective. *J. Travel Res.* **57**(1), 129–146 (2018)
7. Seddighi, H.R., Nuttal, M.W., Theocharous, A.L.: Does cultural background of tourists influence the destination choice? An empirical study with special reference to political instability. *Tour. Manage.* **22**(2), 181–191 (2001)

8. Hajibaba, H., Gretzel, U., Leisch, F., Dolnicar, S.: Crisis-resistant tourists. *Ann. Tour. Res.* **53**, 46–60 (2015)
9. Park, S., Fesenmaier, D.R.: Travel decision flexibility. *Tour. Anal.* **19**(1), 35–49 (2014)
10. Antonio, N., Almeida, A., Nunes, L.: Predicting hotel booking cancellations to decrease uncertainty and increase revenue. *Tour. & Manag. Stud.* **13**(2), 25–39 (2017)
11. Chew, E., Jahari, S.: Destination image as a mediator between perceived risks and revisit intention: A case of post-disaster Japan. *Tour. Manag.*, **40**(C), 382–393 (2014)
12. Falk, M.T., Vieru, M.: Modelling the cancellation behaviour of hotel guests. *Int. J. Contemp. Hosp. Manag.* **30**(10), 3100–3116 (2018)
13. Sirakaya, E., Woodside, A.G.: Building and testing theories of decision making by travellers. *Tour. Manage.* **26**, 815–832 (2005)
14. Thornton, P., Shaw, G., and Williams, A.: Tourist group holiday decision-making and behavior: the influence of children. *Tour. Manag.*, **18**(5), 287–298 (1997)
15. Milliken, F., Martins, L.: Searching for common threads: Understanding the multiple effects of diversity in organizational groups. *Acad. Manag. Rev.* **21**, 402–433 (1996)
16. DeKay, F., Yates, B., Toh, R.: Non-performance penalties in the hotel industry. *Int. J. Hosp. Manag.* **23**(3), 273–286 (2004)
17. Elkan, C.: The Foundations of cost-sensitive learning. In: Proceedings of the Seventeenth international joint conference of artificial intelligence, pp. 973–978. Morgan Kaufmann, Seattle, Washington (2001)
18. Ting, K.M.: An instance-weighting method to induce cost-sensitive trees. *IEEE Trans. Knowl. Data Eng.* **14**(3), 659–665 (2002)
19. Webb, T., Schwartz, Z., Xiang, Z., Singal, M.: Revenue management forecasting: The resiliency of advanced booking methods given dynamic booking windows. *Int. J. Hosp. Manag.*, **89**, Article 102590, (2020)
20. Liberman, V., Yechiali, U.: On the hotel overbooking problem: an inventory system with stochastic cancellations. New York University, Graduate School of Business Administration (1978)
21. Bertsimas, D., Popescu, I.: Revenue management in a dynamic network environment. *Transportation Science, INFORMS* **37**(3), 257–277 (2003)
22. Karaesmen, I., van Ryzin, G.: Overbooking with substitutable inventory classes. *Oper. Res.* **52**, 83–104 (2004)
23. Lindenmeier, J., Tscheulin, D.: The effects of inventory control and denied boarding on customer satisfaction: The case of capacity-based airline revenue management. *Tour. Manage.* **29**(1), 32–43 (2008)
24. Wangenheim, F., Bayón, T.: The chain from customer satisfaction via word-of-mouth referrals to new customer acquisition. *J. Acad. Mark. Sci.* **35**, 233–249 (2007)
25. Fruchter, G.E., Gerstner, E.: Selling with “Satisfaction Guaranteed.” *J. Serv. Res.* **1**(4), 313–323 (1999)
26. Mann, D.P., Wissink, J.P.: Money-back contracts with double moral hazard. *Rand J. Econ.* **19**(2), 285–292 (1988)
27. Chen, C., Xie, K.: Differentiation of cancellation policies in the U.S. hotel industry. *Int. J. Hosp. Manag.*, **34**, 66–72 (2013)
28. Gehrels, S., Blanar, O.: How economic crisis affects revenue management: the case of the Prague Hilton hotels. *Res. Hosp. Manag.* **2**(1–2), 9–15 (2013)
29. Chu, F.L.: Forecasting tourism demand with ARMA-based methods. *Tour. Manage.* **30**(5), 740–751 (2009)
30. Claveria, O., Datzira, J.: Forecasting tourism demand using consumer expectations. *Tourism Review* **65**(1), 18–36 (2010)
31. Pfeifer, P.E., Bodily, S.E.: A test of space-time ARMA modelling and forecasting of hotel data. *J. Forecast.* **9**(3), 255–272 (1990)

32. Song, H., Li, G.: Tourism demand modelling and forecasting—a review of recent research. *Tour. Manage.* **29**, 203–220 (2008)
33. Zakhary, A., Atiya, A., El-Shishiny, H., Gayar, N.: Forecasting hotel arrivals and occupancy using Monte Carlo simulation. *J. Revenue Pricing Manag.* **10**, 344–366 (2011)
34. Claveria, O., Torra, S.: Forecasting tourism demand to Catalonia: Neural networks vs. time series models. *Econ. Model.*, **36**(1), 220–228 (2014)
35. Claveria, O., Monte, E., Torra, S.: Tourism demand forecasting with neural network models: Different ways of treating information. *Int. J. Tour. Res.* **17**(5), 492–500 (2015)
36. Huang, H.C.: A study on artificial intelligence forecasting of resort demand. *J. Theor. Appl. Inf. Technol.* **70**(2), 265–272 (2014)
37. Wu, D.C., Song, H., Shen, S.: New developments in tourism and hotel demand modeling and forecasting. *Int. J. Contemp. Hosp. Manag.* **29**(1), 507–529 (2017)
38. Morales, D.R., Wang, J.: Forecasting cancellation rates for services booking revenue management using data mining. *Eur. J. Oper. Res.* **202**(2), 554–562 (2010)
39. Huang, H.C., Hou, C.I.: Tourism demand forecasting model using neural network. *Int. J. Comput. Sci. & Inf. Technol.* **9**(2), 19–29 (2017)
40. Hu, Y.C., Jiang, P., Lee, P.C.: Forecasting tourism demand by incorporating neural networks into Grey–Markov models. *J. Oper. Res. Soc.* **70**(1), 12–20 (2019)
41. Moutinho, L., Huarng, K.H., Yu, T.H.K., Chen, C.Y.: Modeling and forecasting tourism demand: the case of flows from Mainland China to Taiwan. *Serv. Bus.* **2**, 219–232 (2008)
42. Sánchez-Medina, A.J., C-Sánchez, E.: Using machine learning and big data for efficient forecasting of hotel booking cancellations. *Int. J. Hosp. Manag.*, **89**, Article 102546, (2020)
43. Huang, L., Zheng, W.: Novel deep learning approach for forecasting daily hotel demand with agglomeration effect. *Int. J. Hosp. Manag.*, **98**, Article 103038, (2021)
44. Wu, D.C.W., Ji, L., He, K., Tso, K.F.G.: Forecasting tourist daily arrivals with a hybrid SARIMA–LSTM approach. *J. Hosp. & Tour. Res.* **45**(1), 52–67 (2021)
45. Hyndman, R. J., Athanasopoulos, G.: *Forecasting: Principles and practice*, 3rd ed. OTexts (2021)
46. Zaki, K.: Implementing dynamic revenue management in hotels during Covid-19: Value stream and wavelet coherence perspectives. *Int. J. Contemp. Hosp. Manag.* **34**(5), 1768–1795 (2022)
47. Provost, F., Domingos, P.: Tree induction for probability based ranking. *Mach. Learn.* **52**(3), 199–215 (2003)
48. Antonio, N., Almeida, A., Nunes, L.: Hotel booking demand datasets. *Data Brief* **22**, 41–49 (2019)
49. Quinlan, J.R.: *C4.5: Programs for machine learning*. Morgan Kaufman, San Mateo, CA (1993)
50. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**, 273–297 (1995)
51. Hand, D., Mannila, H., Smyth, P.: *Principles of data mining*. The MIT Press, Cambridge (2001)
52. Breiman, L.: Random forests. *Mach. Learn.* **45**, 5–32 (2001)
53. Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D.: Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* **15**(1), 3133–3181 (2014)



Automatic Generation of a Portuguese Land Cover Map with Machine Learning

Antonio Esteves¹(✉) and Nuno Valente²

¹ ALGORITMI Research Centre/LASI, University of Minho, Braga, Portugal
esteves@di.uminho.pt

² University of Minho, Braga, Portugal
a81986@alunos.uminho.pt

Abstract. The application of machine learning techniques to satellite imagery has been the subject of interest in recent years. The increase in quality and quantity of images, made available by Earth observation programs, such as the Copernicus program, led to the generation of large amounts of data. Among the various applications of this data is the creation of land cover maps. The present work aimed to create machine learning models capable of accurately segmenting and classifying satellite images to automatically generate a land cover map of the Portuguese territory. Several experiments were carried out with the spectral bands of the Sentinel-2 satellite, with vegetation indices, and with several sets of land cover classes. Three machine learning architectures were evaluated, which adopt two different techniques for image classification. One of the classification techniques follows an object-oriented approach, and in this case the architecture adopted in our models was a U-Net artificial neural network. The other classification technique is pixel-oriented, and the machine learning models tested were random forest and support vector machine. The overall accuracy of the results obtained ranged from 68.6% to 94.75%, depending strongly on the number of classes into which the land cover is classified. The result of 94.75% was obtained when classifying the land cover only into five classes. However, a very interesting accuracy of 92.37% was achieved by the model when trained to classify eight classes. These results are superior to those reported in the related bibliography.

Keywords: Machine learning · Deep learning · Remote sensing · Land cover map

1 Introduction

Recent scientific advances in remote sensing (RS) have resulted in easy access to satellite imagery. Among the countless applications of satellite imagery, the present work highlights the creation of land use land cover (LULC) maps. LULC refers to human constructions and natural features of the earth's surface. LULC are used in various fields of study, such as urban planning, natural resource

management, carbon circulation, epidemiology, and climate change. Using the Portuguese territory as a case study, this work intends to apply machine learning (ML) techniques to reproduce the results of the Corine Land Cover (CLC) European Union project.

One of the expected outcomes for this project was to train a model capable of successfully classifying satellite imagery into a LULC map. In a LULC classification task, the term of comparison is an overall accuracy of 85% and where none of the classes have an accuracy of less than 70% [14]. If a trained model performs better than this threshold, it will be considered successful.

RS tasks, such as LULC classification, exhibit some unique specificities. Although there are huge amounts of satellite imagery, most of this data is not classified or it is outdated, therefore not being useful for training deep learning (DL) models [18]. The seasons introduce variability, and hence complexity, especially due to changes in phenology [13]. However this variability can be captured by DL methods, provided it is reproduced in the training data [20].

Several techniques can be implemented for LULC classification, however, these can be divided into two categories: pixel-oriented and object-oriented. Pixel-oriented techniques are more traditional and consider each pixel as an independent unit, classifying each pixel according to its spectral values. Due to their technical limitations, pixel-oriented methods should not be used with high-resolution images [9], as they lower model accuracy and generate images that suffer from the salt and pepper problem, as mentioned by [24]. These methods have two additional limitations [25]:

- Can't handle mixed pixels, a phenomenon that occurs when features from multiple classes are present in a single pixel.
- Don't take advantage of the content of adjacent pixels and their contextual information.

Object-oriented methods, also called geographic object-based image analysis (GEOBIA), group pixels into segments that ideally represent real-world objects. Typically GEOBIA takes place in two phases: segmentation and classification. Note that the segmentation process, not present in pixel-oriented techniques, can also introduce errors into the model, especially in cases of sub-segmentation [12].

2 Related Work

Several papers reporting the application of machine learning to remote sensing have been published recently. This section focus on related approaches to pixel- and object-based land cover classification, the employed ML models, the datasets, the considered land cover classes, and other techniques such as the inclusion of spectral indexes.

A comparison of five ML models was documented in [20]. The chosen models were Random Forest (RF) and four Convolutional Neural Networks (CNNs). These models classified Sentinel-2 imagery into eight classes using the four bands

with a spacial resolution of 10 m (red, green, blue, NIR). The achieved results were compared to TOP10NL data from the Infrastructure for Spatial Information in the European Community (INSPIRE). The RF was the worse model with an overall accuracy of 81%, and the best model obtained 86% accuracy. The main conclusions from this were (i) CNNs and RFs are capable of classifying land cover classes, (ii) hyper-parameter optimization has reduced effect on results when adequate amount of training data is available, (iii) seasonal variety can be handled by introducing it into the training set, (iv) in three out of four CNN models, the size of the input impacts the classification results, and (v) transfer learning shows acceptable results, making the usage of several additional data valid when the application targets a European map.

Abdi [1] presents another comparison of ML models, including Support Vector Machines (SVM), extreme gradient boosting (XGBoost), RF, and an Artificial Neural Network. Their case study was the boreal climate and the considered surface area has a dimension of $10\text{ km} \times 12\text{ km}$. The models were feed with four images, one per season, which improved the classification of some classes. The model with the best result was a SVM with an accuracy of 75.8%.

The work presented in [14] has objectives and methodology similar to ours and can therefore be used as a comparison term. The paper evaluates the feasibility of applying the U-Net neural network to classify the land cover. It achieved a classification accuracy of 92% using the 5 CLC level 1 classes, which decreases to 84% when 13 CLC level two classes are considered. The model obtains the worst results when using only RGB bands, on the contrary, the best model was obtained with a combination of spectral bands and computed spectral indexes such as NDVI.

Van Tricht et al. [21] reports a successful application of a RF, over a combination of Sentinel-1 and Sentinel-2 imagery data, to crop mapping in Belgium. The Model mapped the Belgium territory in 12 classes and two steps. The first step classifies the objects in one out of 4 classes: built-up, water, forest, and crop. The second step expands the crop class into nine more specific classes. They achieved an 82% overall accuracy.

The paper from [19], published in 2019, introduced a new large-scale dataset for training ML models to classify or segment satellite imagery. The dataset is called BigEarthNet and contains 590,326 image patches. This significant amount of data alleviates the problem encountered in RS, the lack of large training sets, a bottleneck that prevents the use of the more recent and complex deep learning models.

3 Methodology

3.1 BigEarthNet Dataset

The main dataset we adopted to train ML models was the mentioned BigEarthNet [19]. This dataset consists of 590,326 Sentinel-2 image patches, composed of 12 spectral bands with 10, 20, and 60 m of spacial resolution, and each patch is

labeled with one or more CLC classes. The bands with 60 m spacial resolution were discarded.

Although this dataset is aimed for classification problems, a layer was added to each patch, obtained from a Web Map Service, containing the corresponding 2018 CLC map. This layer was necessary to train the models for pixel-wise classification, instead of patch-wise classification. From the total of 590,326 images, 16,110 were removed, either because they contain clouds or because there was no CLC map available. Although the BigEarthNet dataset contains images covering the four seasons, land cover classes are not balanced. The CLC class corresponding to glaciers and perpetual snow is totally absent from the dataset.

3.2 LandCoverPT Dataset

The BigEarthNet dataset gathers images from several European countries, some of which have biomes drastically different from Portugal. The LandCoverPT was created with the objective of having dataset more appropriate to train ML models capable of generating a Portuguese land cover map.

The creation of the dataset used 26 Sentinel-2 products, captured in June and August 2019, and the products where divided into 153,347 patches with the same size as the BigEarthNet patches (120×120).

A few aspects to take in consideration when analysing the results produced with the LandCoverPT dataset:

1. It does not include seasonal variety.
2. A thorough examination to identify the presence of clouds was not carried out, and so there may be a residual amount of clouds not detected by manual inspection.
3. Some level 3 CLC classes are missing, since they do not exist in Portuguese territory.

3.3 Models

The first attempt to classify the land cover was carried out with a Support Vector Machine (SVM) [2, 7]. The work reported in [1] compares the SVM to the random forest, the extreme gradient boosting (XGBoost), and a deep neural network. An SVM constructs a hyperplane, in a high dimensional space, to separate each pair of classes. A good separation is achieved by the hyperplane that has the largest distance to the nearest training samples, in order to minimize the generalization error of the classifier. The separating hyperplane depends on a subset of the training data, called the support vectors. A hard margin SVM tries to fit a decision boundary that maximizes the distance between the support vectors of the two classes, but this type of SVM classifier is very sensitive to outliers and it only works on data that is linearly separable. The soft margin SVM addresses these problems by allowing some samples to be located on the boundary region. Thus, a soft margin classifier deals with a trade-off between maximizing the

width of the separating margin and minimizing the misclassifications. The trade-off is controlled by the C hyperparameter of scikit-learn SVC classifier.

In machine learning, kernels can help to construct non-linear decision boundaries using linear classifiers. A kernel function only calculates the relationships between every pair of samples as if they were in a higher dimensional space. This trick, consisting in calculating the high-dimensional relationships without actually transforming the samples to the higher dimension, is called the kernel trick. The kernel trick reduces the amount of computation required by SVMs by avoiding the transformation of the data from a lower to higher dimensional space. There are several types of kernels, such as polynomial and Gaussian kernels. The (Gaussian) Radial Basis Function kernel, computed with the pair of samples x_i and x_j , is expressed by:

$$K(x_i, x_j) = e^{-\gamma ||x_i - x_j||^2} \quad (1)$$

Parameter C can be interpreted as the inverse of regularization. Parameter γ controls the influence that the classification of a given training sample has over the classification of its neighbors, where a larger γ means that only closer samples are affected. Natively, SVC only supports binary classification, but it was extended with a one-versus-one approach to allow multi-class classification. All attempts to classify the land cover with SVMs, were done with five classes and the scikit-learn SVC classifier, which is based on LibSVM [4, 10].

The second ML model evaluated was Random Forest (RF), a supervised Machine Learning algorithm based on the concept of ensemble learning [3]. An example of a successful application of a RF model to land cover classification is documented in [21]. RF improves the Decision Tree (DT) algorithm, and emerged with the objective of minimizing its main limitations: they are prone to overfitting and even a small change in the training data can result in a huge difference on the decision tree structure. The random forest overcomes these limitations by taking the prediction from each tree and based on the majority votes from the trees (Fig. 1). It uses bagging and feature randomness when building each individual tree, in order to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree. Randomness is built into RF mainly in two ways: each tree is fitted on a subset of the entire dataset, and each tree can grow differently, by virtue of the randomized order or subset of the features considered for optimum split in the Decision Tree.

The Random Forest uses an ensemble technique called Bootstrap Aggregating, or Bagging. First, each decision tree is trained independently with a different bootstrapped set, obtained from the entire dataset using sampling with replacement (bootstrap step). During inference, a prediction is made by each decision tree, and the final prediction by the random forest is returned as a majority vote (aggregation step). The cost function, or criterion, used more often during the learning process to split a node of the decision tree is called the Gini Impurity. It is basically a concept to quantify how homogeneous or “pure” a node is. A node is considered pure ($G = 0$) if all training samples in the node belong to the

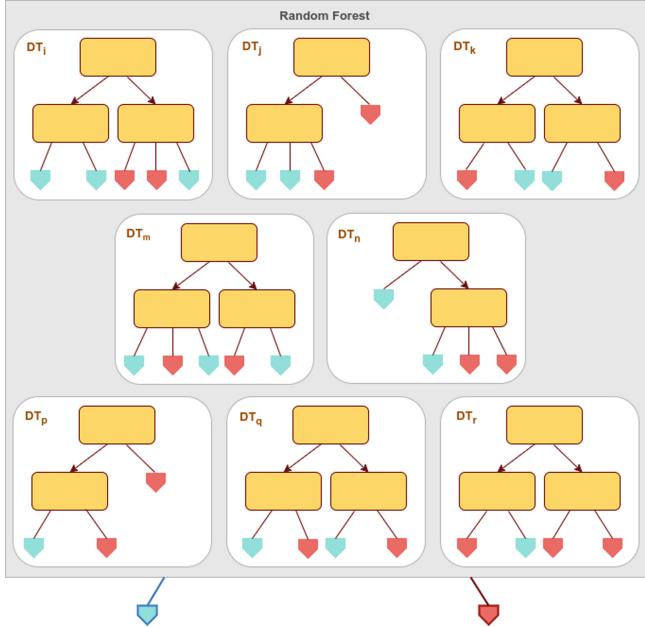


Fig. 1. A RF model is a forest of decision trees

same class, while a node with many training samples from many different classes will have a Gini Impurity close to 1. The Gini impurity at a node is computed by Eq. 2.

$$G = 1 - \sum_{c=1}^{NC} \frac{n_c}{n}^2 \quad (2)$$

Where NC is the number of classes, n_c is the number of samples belonging to class c on the node, and n is the total number of samples on the node.

In the present work RFs were implemented with the scikit-learn `RandomForestClassifier`. The most relevant hyperparameters of `RandomForestClassifier` are the number of trees the algorithm builds (`n_estimators`), the maximum number of features considered when splitting a node (`max_features`), and the minimum number of samples that must be allocated to each leaf node to be created (`min_sample_leaf`).

The last model, and the one that was most thoroughly evaluated, to classify the land cover was the neural network U-Net. U-Net is a CNN model initially developed for biomedical image segmentation and to be trained with few images [15]. However, both U-Net and other variants of it, were successfully applied to the RS domain, as reported in works [14, 17, 25, 26].

As can be seen in Fig. 2, U-Net comprises two parts, a contracting path that captures context (top part), and a symmetric expanding path that enables pre-

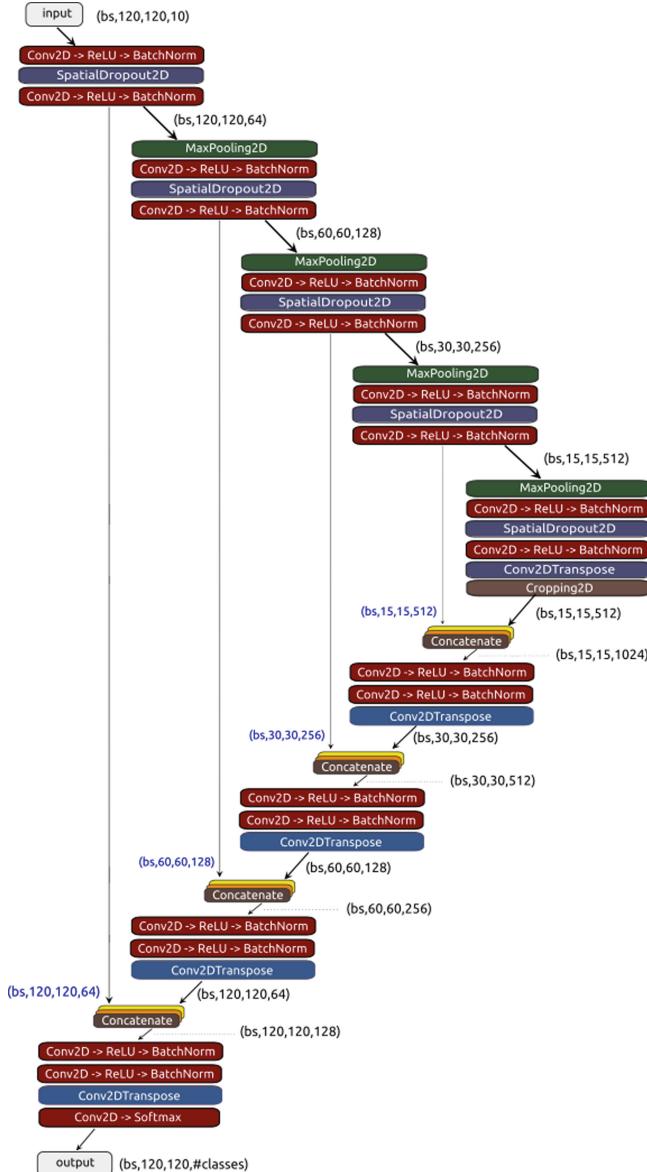


Fig. 2. U-Net model trained on patches with 120×120 pixels and 10 bands

cise localization of features (bottom part). The contracting part extracts features through convolutions with 3×3 filters and max pooling layers. The expanding part uses convolutions and transposed convolutions to reduce the number of feature maps from 512 to 64, while it increases their dimensions from 15×15 to 120×120 . Feature maps from the contracting part of the network are copied

to the expanding part to avoid losing spatial information. The copy is implemented by the 4 vertical skip connections in Fig. 2. The copied features are then concatenated with same size features from the expanding path.

In our experiments, the U-Net receives as input 120×120 patches and outputs `#classes` segmentation masks with the same size, one mask per land cover class. As documented in the next section, experiments were carried out with different numbers of land cover classes.

4 Experiments and Results

4.1 Support Vector Machine Classifier

In the first experiment with SVMs, the model was trained with unbalanced samples from 256 image patches of size 120×120 pixels and 10 bands. The model was trained with $C = 2.0$, the RBF kernel, `gamma='scale'`, unlimited number of iterations, and `decision_function_shape = 'ovr'`. The achieved validation accuracy was 79.3%. Since the dataset is quite unbalanced, a reasonable high accuracy is achieved by a model that is tuned to classify correctly the three most frequent classes (1, 2, 5) and misclassifying the least frequent ones (0, 3). The next step was to balance the dataset, considering the same number of samples for all the classes. The considered number of samples was defined as the minimum value of the occurrences among the five classes.

Another direction that was explored was applying Principal Component Analysis (PCA) to reduce the number of features per sample from 10 (bands) to three (principal components), those that explain around 99% of the variance. Figure 3 shows the result of plotting the samples, after being projected on a 2D/3D space, defined by the two/three principal components of PCA that explain most of the variance. The projection on 3D makes it easier to visualize the clustering of the samples belonging to the same class. The visual analysis of this figure reveals that the classes exhibit a significant overlapping on the 3D space, which will make separation difficult. It was applied grid search cross-validation (CV) to find best values for the hyperparameters C and $gamma$ of the SVM model. It was found that $C = 1.0$ and $gamma = 5.0$ allow the best accuracy. When using a $C \geq 10000$ it was observed that the computation time, necessary to run a “batch” with a combination of hyperparameters, became extremely high. The global test accuracy of the model was 59.1%. Finally, we dropped PCA and keep the 10 original features per pixel. Considering 2048 image patches, 10 features per pixel (corresponding to 10 Sentinel-2 bands) which allowed us to achieve the highest accuracy with SVM, 68.6%.

Evaluation metrics for the best SVM model are presented in Table 1 and the confusion matrix is in Fig. 4. Considering the F1-score, the worst result belongs to the wetlands class (class 3). Although the improvement of the SVM model after balancing the dataset, optimizing the hyperparameters and reducing the number of features with PCA is not a satisfactory result and reveals that SVM is not the best fit to classify the land cover. Moreover, even a moderated number of image patches, such as 1024, turns the training very slow.

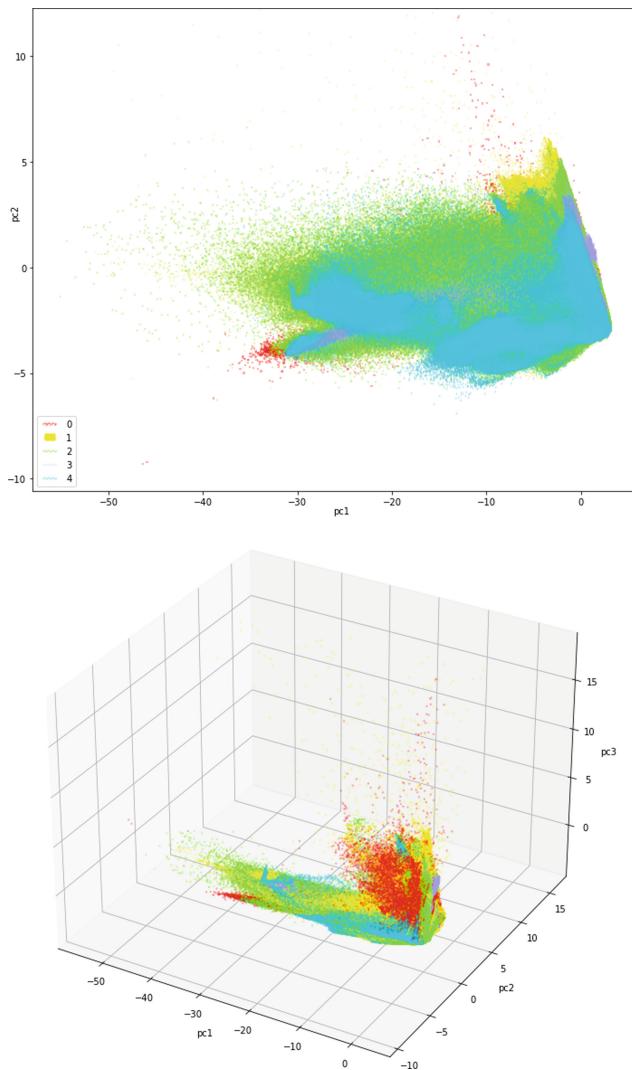


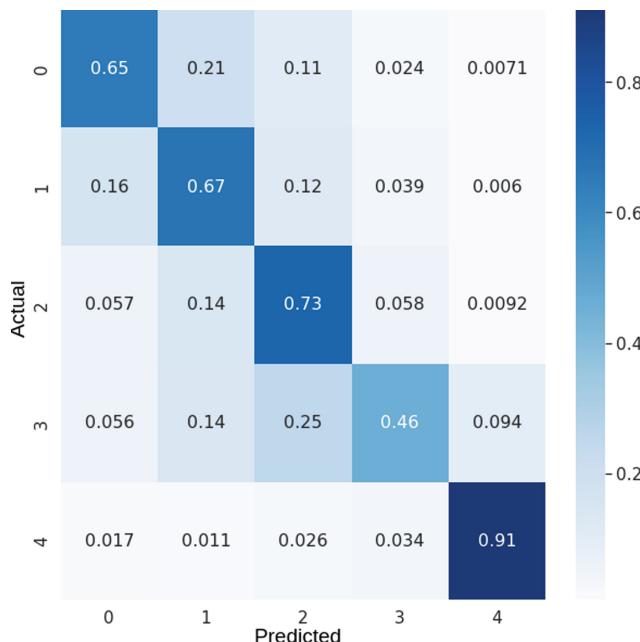
Fig. 3. Plotting the samples after being projected on a 2D/3D space defined by the two/three principal components of PCA

4.2 Random Forest Classifier

Training of the RF was done with 1024 image patches of 120×120 pixels each, the number of land cover classes was five, classes were balanced by considering a number of pixels per class equal to the least frequent class, PCA was applied to select the three features that explain most of the variance, the criterion used to evaluate the splits was `log_loss`, the RF included 100 decision trees,

Table 1. Results for SVM model, using 10 features per sample, and the 5 CLC Level 1 classes

Class	Precision	Recall	F1-score
0—Artificial surfaces	0.69	0.65	0.67
1—Agricultural areas	0.57	0.67	0.62
2—Forest and semi-natural areas	0.59	0.73	0.66
3—Wetlands	0.75	0.46	0.57
4—Water bodies	0.89	0.91	0.90

**Fig. 4.** Normalized confusion matrix for the classification in 5 classes with SVM

`bootstrap = False` meaning the whole dataset is applied to train each tree. The test accuracy score achieved by the trained model is 0.557.

Next, the number of image patches was increased to 2048, the number of features per pixel remained on three, the evaluation criterion was changed to `gini`, the number of decision trees was kept on 100, `bootstrap = True` and `max_samples = 0.8`. The test accuracy score achieved by the trained model is 0.573. It was also tried increasing the number of decision trees to 200, but there was no improvement on the model performance.

Since using only three features per pixels resulted in poor results, it was decided to remove PCA and keep the 10 original features per pixel. Considering 2048 image patches, 10 features per pixel (corresponding to 10 Sentinel-2 bands),

5 land cover classes, balancing the frequency of the classes, with the `gini` evaluation criterion, `bootstrap = True`, `max_samples = 0.8`, and `max_features = 3`, the test accuracy score achieved by the trained model raised to 0.706. The confusion matrix is presented in Fig. 5. This confusion matrix reveals that the percentage of samples correctly classified is 68.0% for class 0, 67.0% for class 1, 72.0% for class 2, 55.0% for class 3, and 93.0% for class 4. Precision, recall, and F-score metrics for the trained RF model are shown on Table 2.

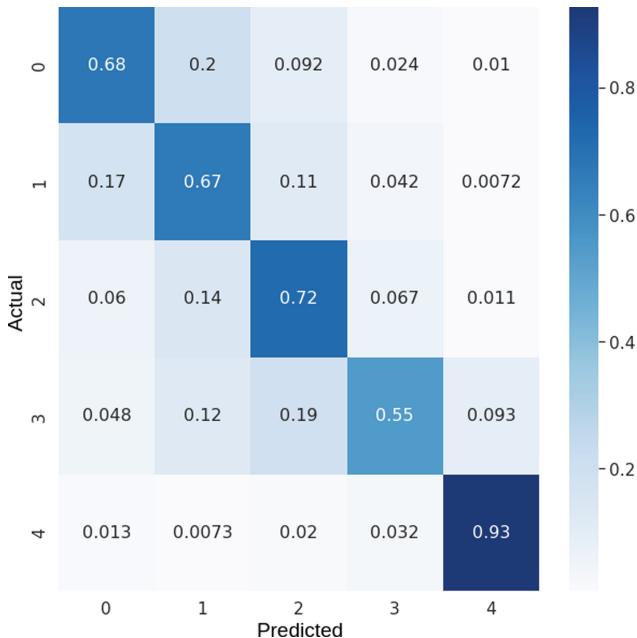


Fig. 5. Normalized confusion matrix for the classification in 5 classes with RF

4.3 U-Net

U-Net model was described with the TensorFlow library, especially the Keras API, it was trained with Adam optimizer, the categorical cross-entropy loss, the `ModelCheckpoint`, `EarlyStopping`, and `ReduceLROnPlateau` callbacks, during 200 epochs. Models were evaluated based on accuracy, precision, recall, and F1-score metrics.

Table 2. Results for RF Model, using 10 features per sample, and the 5 CLC Level 1 classes

Class	Precision	Recall	F1-score
0—Artificial surfaces	0.70	0.68	0.69
1—Agricultural areas	0.59	0.67	0.62
2—Forest and semi-natural areas	0.63	0.72	0.67
3—Wetlands	0.77	0.55	0.64
4—Water bodies	0.88	0.93	0.91

A list of all experiments carried out, as well as the results obtained, can be seen in the Table 3. The set of experiments accomplished with U-Net and the BigEarthNet dataset will be summarized now.

Table 3. Summary of the different experiments

Model	Classes	Dataset	Overall accuracy
SVM	5	BigEarthNet	68.6%
RF	5	BigEarthNet	70.6%
U-Net	43	BigEarthNet	82.32%
U-Net + NDVI	43	BigEarthNet	77.95%
U-Net	15	BigEarthNet	87.11%
U-Net	11	BigEarthNet	86.88%
U-Net	8	BigEarthNet	92.37%
U-Net	5	BigEarthNet	94.75%
U-Net	5	LandCoverPT	87.26%

The experiment with all 43 CLC level three classes will work as our baseline, i.e, with all the other experiments we will try to improve the results of the baseline. The overall accuracy achieved was 82.32% with most misclassifications being within very similar classes, such as continuous urban fabric and discontinuous urban fabric. The class with the lowest results was green urban areas, being misclassified as urban fabric or forests.

The second experiment tried to improve the results of the previous attempt through the insertion of the Normalized Difference Vegetation Index (NDVI). NDVI was chosen because of its popularity in the literature, for example in [6,8]. The final results were worse than in the previous scenario, analysing each class individually shows that some classes were being completely misclassified and this did not happen in the previous experiment. Taking into consideration these results the idea of using other spectral indexes was abandoned.

The next step taken to improve the results was to reduce the number of land cover classes. The experiment with 15 CLC level two classes improved the overall

accuracy to 87.11%. The normalized confusion matrix for the segmentation in 15 classes with U-Net is shown in Fig. 6. While the baseline presented some values for the F1-score metric of the order of 0.4, this model presents 0.65 as the lowest value.

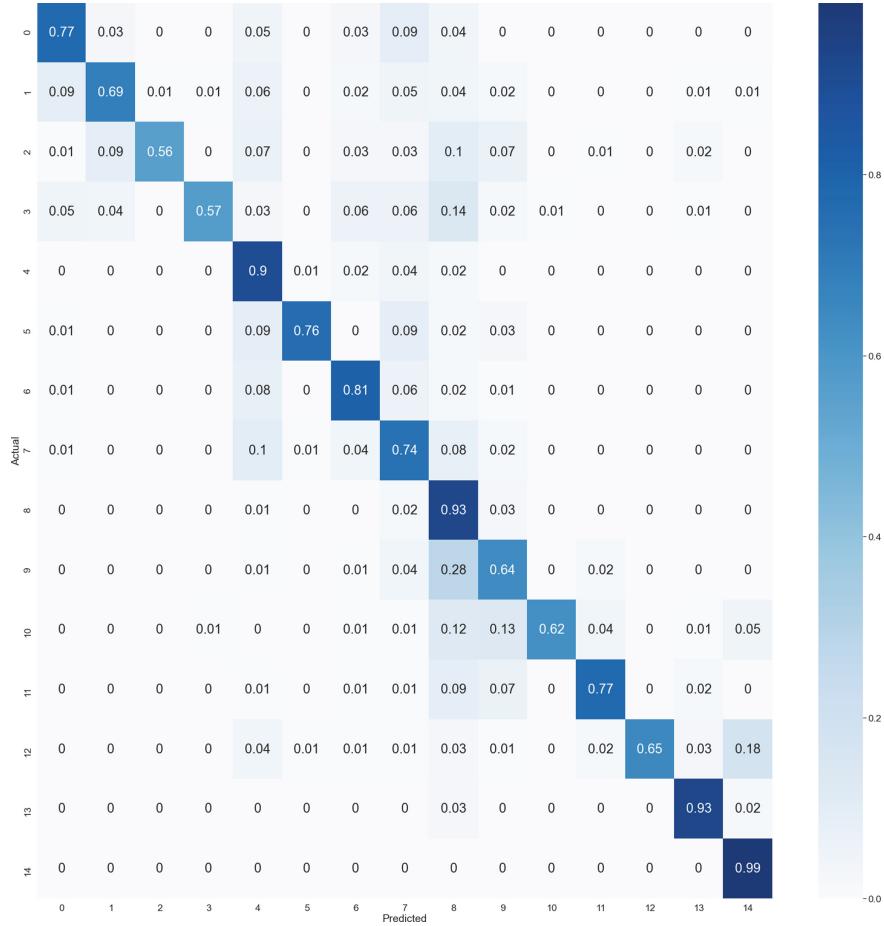


Fig. 6. Normalized confusion matrix for the segmentation in 15 classes with U-Net

The automatic classification of land cover in 15 classes is still a very ambitious objective, and therefore another model was trained to classify the land cover only in the 5 CLC level one class. The trained U-Net model achieved a 94.75% overall accuracy, the best result among all experiments. Evaluation metrics for this model are presented in Table 4 and the confusion matrix is in Fig. 7. Considering the F1-score, the worst result belongs to the wetlands class (class 3).

Table 4. Results for U-Net model, using 10 spectral bands and the 5 CLC Level 1 class

Class	Precision	Recall	F1-score
0—Artificial surfaces	0.86	0.82	0.84
1—Agricultural areas	0.94	0.94	0.94
2—Forest and semi-natural areas	0.95	0.95	0.95
3—Wetlands	0.77	0.80	0.78
4—Water bodies	0.98	0.99	0.98

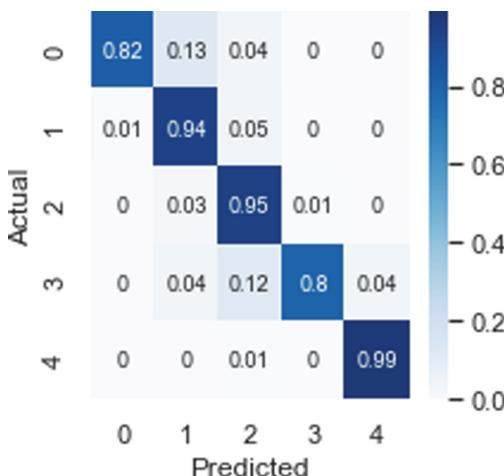


Fig. 7. Normalized confusion matrix for the segmentation in 5 classes with U-Net

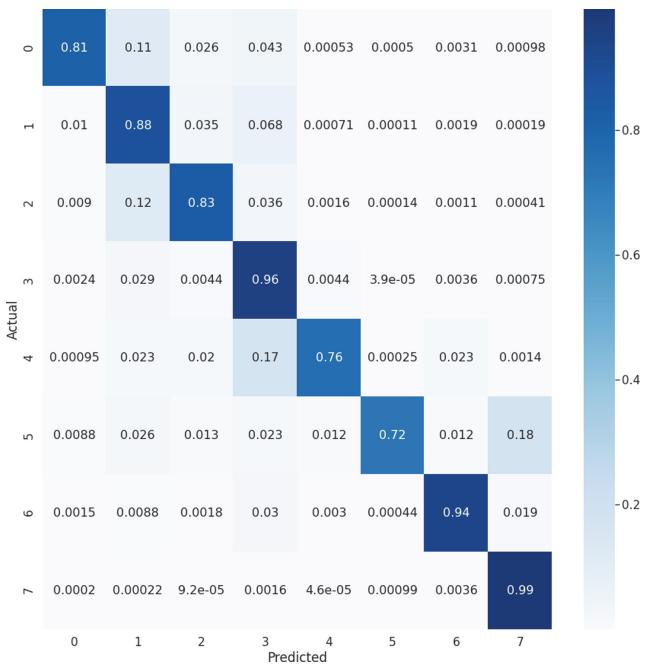
Two attempts with a combination of CLC classes from levels 1 and 2 were realized. The first one used 11 classes and obtained an overall accuracy of 86.11%, a result worse than the experiment with 15 classes.

The second attempt used eight classes and its overall accuracy was 92.37%, a result very similar to the experiment with five classes (Table 5). The chosen level two classes are those that were best classified by U-Net trained with the level two classes. The remaining level two classes were collapsed into the corresponding level 1 class. The CLC hierarchy was maintained, i.e., only level two classes that would be part of the same level 1 class were gathered. Confusion matrix analysis (Fig. 8) shows that 11% of the samples belonging to class 0 (artificial surfaces) are classified as class 1 (agricultural areas), 12% of class 2 (pastures) is classified as class 1 (agricultural areas), 17% of class 4 (inland wetlands) is classified as class 3 (forest and semi-natural areas), and 18% of class 5 (maritime wetlands) is classified as class 7 (maritime waters).

Experiments with the LandCoverPT dataset, the U-Net model, and level 1 land cover classes, were also accomplished. The results of these experiments

Table 5. Results for U-Net model, using 10 spectral bands and 8 CLC Level 1 and Level 2 classes

Class	Precision	Recall	F1-score
0—Artificial surfaces	0.84	0.81	0.83
1—Agricultural areas	0.91	0.88	0.90
2—Pastures	0.81	0.83	0.82
3—Forest and semi-natural areas	0.94	0.96	0.95
4—Inland wetlands	0.78	0.76	0.77
5—Maritime wetlands	0.79	0.72	0.76
6—Inland waters	0.94	0.94	0.94
7—Maritime waters	0.99	0.99	0.99

**Fig. 8.** Normalized confusion matrix for the segmentation in 8 classes with U-Net

were worse than those achieved with the BigEarthNet dataset, quantified as an overall accuracy of 87.26%. Classes with the worst results in this experiment were artificial surfaces and wetlands. A possible explanation for this results can be the low number of samples containing those classes in the LandCoverPT dataset.

The visual inspection to the predictions with the trained models, and to the correspondent ground-truth, revealed that the classification errors were predominantly located at the boundary of the patches (Fig. 9). The most likely

explanation for this fact is the existence of mixed pixels. Another explanation, mentioned in the literature, is the lower ability of U-Net to correctly segment pixels at the object's boundaries.

Figure 9 shows a satellite image patch, randomly chosen from the test set. The leftmost column of the figure shows the ground truth masks for the 5 level 1 class (0–4). The next column shows the model prediction for the same classes. In the upper right corner are presented 4 of the 10 bands of the input patch, in this case the ones with the best spatial resolution: red, green, blue and near infrared. Pixels that were misclassified are shown in yellow in the central right part of the figure.

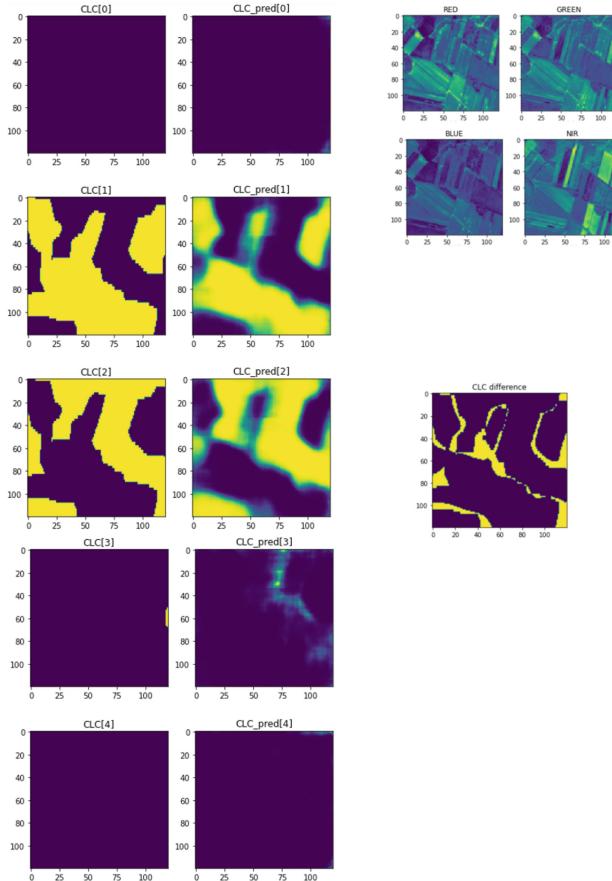


Fig. 9. Random satellite image patch from the test set: input masks (Left), predicted masks (Middle), Input visible bands (Top Right), and misclassified pixels (Center Right)

When we evaluate the trained models with a dataset distinct from the training set, the results are inferior. It was observed that several land cover parcels, classified as agricultural areas in the 2018 CLC map (yellow regions in Fig. 10), are misclassified by our models as artificial surfaces (red regions in Fig. 10) or forests and semi-natural areas (green regions in Fig. 10).

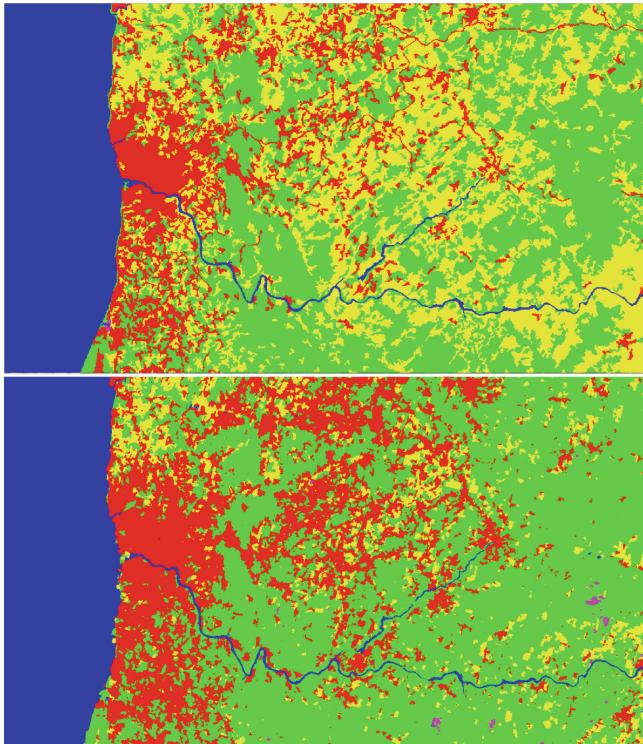


Fig. 10. Ground truth 2018 CLC map with 5 classes, for the Northwest region of Portugal (Top) and corresponding map generated by the trained U-Net model (Bottom). Color scheme: artificial surfaces (Red), Agricultural areas (Yellow), Forest and semi-natural areas (Green), Wetlands (Magenta), Water bodies (Blue)

Another problem, observed in some parts of the automatically generated map, is the discontinuity between patches. This problem occurs because the masks generated by the model are obtained patch by patch, where the patch size is 120×120 . A possible solution is to discard the pixels on the periphery of the patches and use only the inner part (Fig. 11). The innermost pixels have more contextual information and better accuracy than peripheric pixels, as it can be seen in Table 6. The drawback of this solution is the longer time it takes to generate the land cover map. For example, considering only an inner part of 20×20 pixels on each patch, the time to classify the same land area will increase $6 * 6$ times.

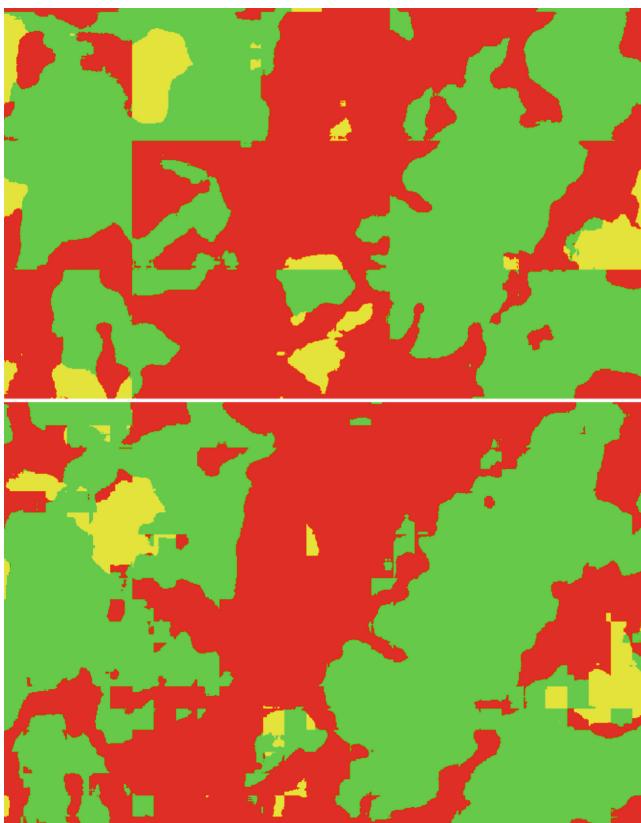


Fig. 11. Discontinuity problem in the segmentation of patches (Top) and its mitigation (Bottom)

Figure 12 contains a complete and continuous land cover map for continental Portugal. This map was generated with the U-Net model, trained on the BigEarthNet dataset and five classes. Sentinel-2 products, downloaded from the <https://scihub.copernicus.eu> website, were used to generate the full map. Images were captured by the satellite on July 7, 2021 and August 22, 2021, and have a maximum cloud percentage of 5%. Because products with a minimum cloud percentage were needed, it was impossible to use all the images from the same day. To visualize the map we used the QGIS tool, where the various parcels of the map generated by the model were merged and trimmed with the help of a shapefile that defines the boundaries of the Portuguese mainland.

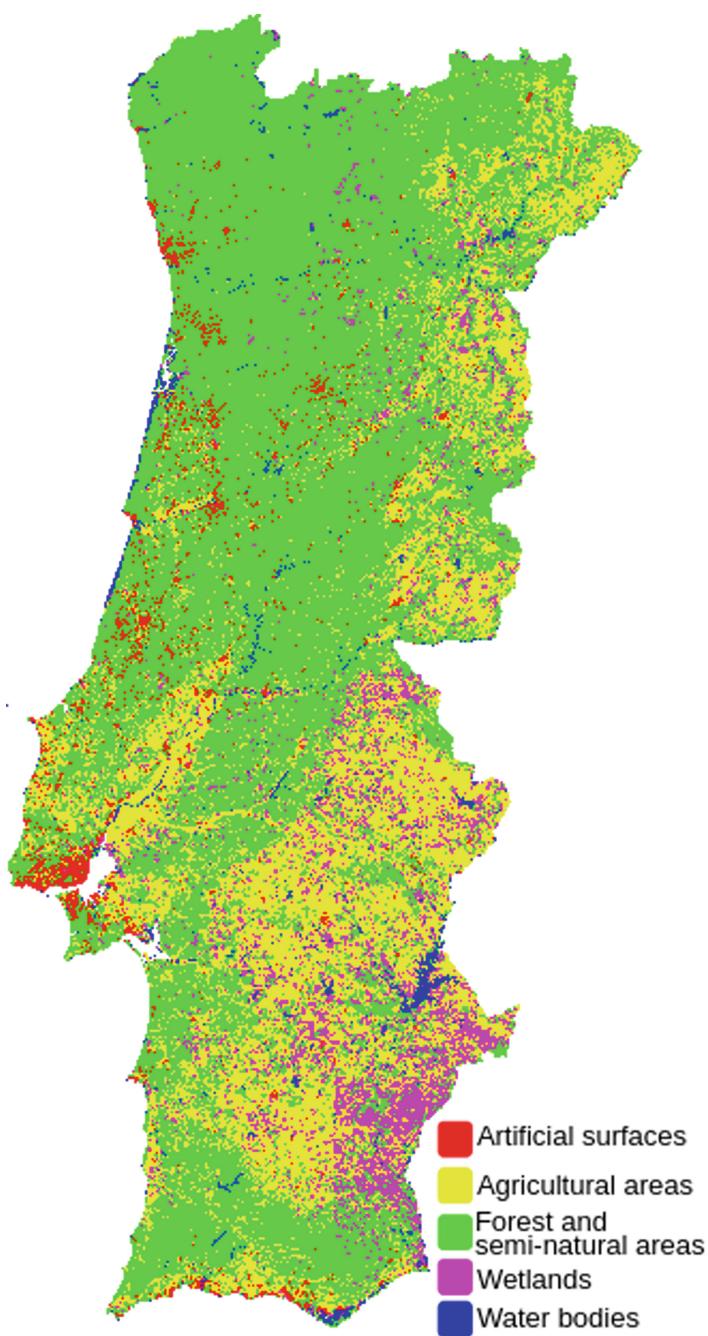


Fig. 12. Land cover map for the Portuguese mainland generated by the U-Net model

Table 6. Accuracy Achieved when using only the Inner Part of Each Patch, for Different Sizes of the Considered Area

Size of the area used on each patch	43 CLC classes	15 CLC classes	5 CLC classes
120 × 120	82.38%	87.02%	94.75%
110 × 110	82.71%	87.27%	94.87%
100 × 100	82.91%	87.42%	94.94%
90 × 90	83.07%	87.51%	95.00%
80 × 80	83.17%	87.60%	95.03%
70 × 70	83.26%	87.68%	95.08%
60 × 60	83.37%	87.74%	95.08%
50 × 50	83.43%	87.76%	95.08%
40 × 40	83.50%	87.83%	95.09%
30 × 30	83.55%	87.88%	95.11%
20 × 20	83.59%	87.89%	95.11%
10 × 10	83.62%	87.93%	95.10%

5 Conclusions and Future Work

The results achieved in the present work provide an evidence that it is possible to automatically and reliably generate an updated land cover map. Thus, the results of this study are relevant for those working in the field of remote sensing.

The biggest difficulty encountered in the course of the work was the processing of large amounts of data from a dataset such as the BigEarthNet or the Sentinel-2 satellite products. To overcome these difficulties, techniques such as feeding the training loop with data stored in TFrecords files and adopting iterative processes whenever possible.

The best trained model achieved an overall accuracy of 94.75%, which can be increased to 95.11% if only the central pixels of the patches are considered during the segmentation of each patch. Although this result is very good, it should however be taken into consideration that the visual comparison between the official 2018 CLC map and the map generated by the developed model, for the same geographical area and the same year, shows that the overall quality of the generated map is lower than 94.75%.

When classifying land cover into five classes, a consistent result across all models is a greater difficulty in identifying artificial surfaces (class 0) and wetlands (class 4). The explanation lies in the similarity between the spectral characteristics of artificial surfaces and agricultural areas (class 1), and between wetlands and semi-natural areas (class 2). In the case of Portuguese territory, the identification of class 3 constitutes an added problem because wetlands are not frequent.

The latest official CLC map is relative to 2018 and required a production time of about one and a half year while training, tuning and generating the land cover map with the proposed model requires a time of less than a month. The ML model will never have a higher accuracy than the CLC project since the model learns from the official CLC map data. However, given the time difference needed to produce the maps, the maps generated with ML models are feasible in several scenarios, because they may be more up-to-date than the official CLC map.

To conclude it is necessary to point out that the CLC maps and the Land Use and Land Cover charts have a human error, and when these maps are used to train ML models the error remains. In case of the 2018 CLC map, each participating country commissioned a team to create their map, but all countries used the same methodology and nomenclature, to ensure an accuracy higher than 85%.

Although the best model achieved good results, some alternatives remained to be explored. Here are some possibilities to improve the presented results:

- Test other segmentation models that address some of the U-Net limitations, such as models based on Feature Pyramid Networks [11, 16, 22, 23, 27] and DeepLab [5].
- Test other datasets, improve and increase the tested LandCoverPT dataset, which exhibit some limitations to obtain optimal results. Another possibility is to improve the dataset would be to optimize the size of the patches into which the Sentinel-2 products were divided.
- Implement other strategies to minimize the segmentation problem at the periphery of patches.
- Take a more consistent approach to optimizing model hyperparameters, for example by using a library such as Optuna or TPOT.
- Add other types of data to the optical images, such as radar images collected by the Sentinel-1 satellite.
- Test spectral indexes with the random forest model.

Acknowledgments. This work has been supported by FCT—Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

References

1. Abdi, A.M.: Land cover and land use classification performance of machine learning algorithms in a boreal landscape using sentinel-2 data. *GIScience Remote. Sens.* **57**(1), 1–20 (2020)
2. Boser, B., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: Proceedings of the 5th Workshop on Computational Learning Theory, pp. 144–152 (1992). <https://doi.org/10.1145/130385.130401>
3. Breiman, L.: Random forests. *Mach. Learn.* **45**, 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
4. Chang, C.-C., Lin, C.-J.: Libsvm: A Library for Support Vector Machines (2001)

5. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)
6. Clerici, N., Calderón, C.A.V., Posada, J.M.: Fusion of sentinel-1a and sentinel-2a data for land cover mapping: a case study in the lower magdalena region, colombia. *J. Maps* **13**(2), 718–726 (2017)
7. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**, 273–297 (1995). <https://doi.org/10.1007/BF00994018>
8. Gašparović, M., Zrinjski, M., Gudelj, M.: Automatic cost-effective method for land cover classification (alcc). *Comput. Environ. Urban Syst.* **76**, 1–10 (2019)
9. Georganos, S., Grippa, T., Lennert, M., Vanhuysse, S., Johnson, B., Wolff, E.: Scale matters: spatially partitioned unsupervised segmentation parameter optimization for large and heterogeneous satellite images. *Remote Sens.* **10**(9), 1440 (2018)
10. Hsu, C.-W., Chang, C.-C., Lin, C.-J.: A Practical Guide to Support Vector Classification (2016)
11. Kirillov, A., Wu, Y., He, K., Girshick, R.B.: Pointrend: Image segmentation as rendering. *CoRR* (2019). [ArXiv:abs/1912.08193](https://arxiv.org/abs/1912.08193)
12. Liu, D., Xia, F.: Assessing object-based classification: advantages and limitations. *Remote Sens. Lett.* **1**(4), 187–194 (2010)
13. Ma, L., Li, M., Ma, X., Cheng, L., Peijun, D., Liu, Y.: A review of supervised object-based land-cover image classification. *ISPRS J. Photogramm. Remote Sens.* **130**, 277–293 (2017)
14. Mäyrä, J.: Land cover classification from multispectral data using convolutional autoencoder networks. Master's thesis, University of Jyväskylä (2018)
15. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241 (2015)
16. Seferbekov, S.S., Iglovikov, V.I., Buslaev, A.V., Shvets, A.A.: Feature pyramid network for multi-class land segmentation. *CoRR* (2018) [ArXiv:abs/1806.03510](https://arxiv.org/abs/1806.03510)
17. Stoian, A., Poulain, V., Inglada, J., Poughon, V., Derksen, D.: Land cover maps production with high resolution satellite image time series and convolutional neural networks: adaptations and limits for operational systems. *Remote Sens.* **11**(17), 1986 (2019). Aug
18. Sumbul, G., Charfuelan, M., Demir, B., Markl, V.: Bigearthnet: a large-scale benchmark archive for remote sensing image understanding. In: IGARSS 2019–2019 IEEE International Geoscience and Remote Sensing Symposium, pp. 5901–5904 (2019)
19. Sumbul, G., Charfuelan, M., Demir, B., Markl, V.: Bigearthnet: a large-scale benchmark archive for remote sensing image understanding. In: IGARSS 2019–2019 IEEE International Geoscience and Remote Sensing Symposium, pp. 5901–5904 (2019)
20. Syrris, V., Hasenohr, P., Delipetrev, B., Kotsev, A., Kempeneers, P., Soille, P.: Evaluation of the potential of convolutional neural networks and random forests for multi-class segmentation of sentinel-2 imagery. *Remote Sens.* **11**(8), 907 (2019)
21. Van Tricht, K., Gobin, A., Gilliams, S., Piccard, I.: Synergistic use of radar sentinel-1 and optical sentinel-2 imagery for crop mapping: a case study for belgium. *Remote Sens.* **10**(10), 1642 (2018)
22. Yuan, Y., Fang, J., Lu, X., Feng, Y.: Spatial structure preserving feature pyramid network for semantic image segmentation. *ACM Trans. Multimedia Comput. Commun. Appl.* **15**(3) (2019)

23. Yuan, Z., Liu, Z., Zhu, C., Qi, J., Zhao, D.: Object detection in remote sensing images via multi-feature pyramid network with receptive field block. *Remote Sens.* **13**(5) (2021)
24. Zhang, C.: Deep Learning for Land Cover and Land Use Classification. Ph.D. thesis, Lancaster University (2018)
25. Zhang, X., Han, L., Han, L., Zhu, L.: How well do deep learning-based methods for land cover classification and object detection perform on high resolution remote sensing imagery? *Remote Sens.* **12**(3) (2020)
26. Zhang, Z., Liu, Q., Wang, Y.: Road extraction by deep residual u-net. CoRR (2017). [ArXiv:abs/1711.10684](https://arxiv.org/abs/1711.10684)
27. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. CoRR (2016). [ArXiv:abs/1612.01105](https://arxiv.org/abs/1612.01105)



The Use of Fuzzy Controllers in Automatic Control Systems for Quadcopters

Ramin Rzayev^{1(✉)}, Tunjay Habibbayli², and Murad Aliyev³

¹ Institute of Control Systems of ANAS, Vahabzadeh str. 9, AZ1141 Baku, Azerbaijan
raminrza@yahoo.com

² Institute of Information Technology of ANAS, Vahabzadeh str. 9, AZ1141 Baku, Azerbaijan
³ Intelpro LLC, Jabbarly str. Globe Center 609, AZ1065 Baku, Azerbaijan

Abstract. The purpose and objectives of this work are to develop methods for designing fuzzy control systems for weakly formalized technical objects relative to helicopter-type unmanned aerial vehicles, namely quadcopters. The result of the paper is to obtain analytical information on the existing structures of fuzzy regulators, understanding the features of their design, and practical use. The basic paradigm of this approach is the verbal interpretation of the distance to possible obstacles in the frontal horizontal and vertical planes in the direction of the quadcopter flight. As a tool for formalizing verbal assessments, it is proposed to use fuzzy sets that can reflect the terms (or values) of linguistic variables as inputs of the basic quadrotor control model under overland monitoring. Taking into account the presence of obstacles, the maneuvering of the quadcopter is carried out on the basis of the fuzzy signals generated by the Fuzzy Inference System, which reflect the change in the angle of rotation in the horizontal and vertical planes, as well as the velocity of the quadcopter.

Keywords: Quadcopter · Overland monitoring · Fuzzy inference · Fuzzy controller

1 Introduction

The use of unmanned aerial vehicles (UAVs) is becoming very popular, which is explained by the presence of significant advantages for this type of aviation equipment [1, 2]; the cost of conducting and preparing flights, and scheduled maintenance is reduced. The versatility allows the use of UAVs for solving a wide variety of tasks, and the absence of a person on board makes it possible to remove restrictions on the use of UAVs under dangerous to the life and health of the crew. The specifics of the use of different types of UAVs (aircraft and helicopter types) are determined by their technical capabilities. In particular, helicopter UAVs have certain limitations, however, for a number of tasks, their use is more preferable than the use of aircraft-type UAVs. A typical task solved on the basis of helicopter UAVs is the overland monitoring in mountainous and wooded areas [1].

One of the most popular representatives of rotary-wing UAVs among researchers is a quadcopter. The quadcopter is a helicopter with four rotors. Its distinguishing feature

from helicopters is that each of the propellers is involved in the formation of elevating power, and flight stability is higher. The quadcopter is a complex, multi-connected technical device with unsimulated dynamics, subject to the influence of external disturbances [1, 2]. Therefore, the solution of the quadcopter control problem is interesting from scientific and practical points of view, which causes significant interest in this topic.

2 Problem Definition

As any physical agent, a quadcopter has six degrees of freedom, which indicates its ability to perform geometric movements in three-dimensional space (in a Cartesian three-dimensional coordinate system), namely: to move forward/back, up/down, left/right, as well as perform Euler turns around each of the three mutually perpendicular axes (yaw, pitch, roll). At the same time, the movement of a quadcopter along any vector in space can be represented as the sum of three elementary movements along the base vectors along each of the axes, and each such elementary movement cannot be derived from the other two.

Assuming that the movement of the quadcopter along each of the three axes does not depend on the movement along the other two and rotation around any of the axes, it is proposed to develop a fuzzy control model based on a fuzzy controller that provides automatic maneuvering of the quadcopter in three-dimensional space to avoid obstacles under overland monitoring. The fuzzy controller itself must be formed by using the Fuzzy Inference System, in which the input linguistic variables are the viewing latitudes in the sectors of space (Fig. 1), within which the presence of obstacles and the distance to them are analyzed at the verbal level. At the same time, the terms of the three output linguistic variables should be reflected in the form of control actions to overcome obstacles by changing the angle of rotation in the horizontal plane (left/right), the flight altitude (up/down) and the quadcopter speed.

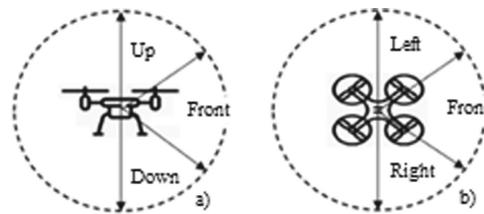


Fig. 1. Quadcopter view sectors: **a** view in the vertical plane, **b** view in the horizontal plane.

3 Quadcopter Control Based on Fuzzy Logic

Despite the development of methods of the modern theory of automatic control and the results obtained on their basis, such controllers as PI, PD, PID are mostly often used in robotics. Control systems built on the basis of PI-, PD-, PID-controllers for solving the problem of quadcopter flight stabilization do not provide the necessary quality, since the calculation of their parameters requires an accurate mathematical model of the control object and disturbances, which is extremely difficult to obtain.

At the same time, the mathematical apparatus of fuzzy logic makes it possible to use intuitive data, expert experience (heuristic knowledge) about the control object, which in a certain sense compensates the absence of the accurate mathematical model in cases where it is simply impossible to build it [3]. The disadvantage is the iterative process of synthesis of fuzzy controllers, due to the need to select a number of parameters due to the absence of sufficient information support. At present, the application of the mathematical apparatus of fuzzy logic to automation problems is possible in two ways. The first one consists in designing a classifier of situations that determines the objectives of the system functioning. The second approach is more popular and is based on the direct regulation of the parameters of the control object. Despite these disparities, these approaches are similar to each other.

Figure 2 shows the general structure of the fuzzy control system for the quadcopter flight process [4], which provides autonomous maneuvering to overcome obstacles in the viewing sectors indicated in Fig. 1. The fuzzy control system consists of four equivalent fuzzy controllers that generate general control (u) and correction factors (Δu) for each of the variables: in the horizontal plane—Left/Right, in the flight altitude—Up/Down, and the quadcopter speed according to a PID-like law. The calculation of the final control actions for each of the quadcopter engines is carried out in the “Aggregation” block by summing the values of the controller outputs.

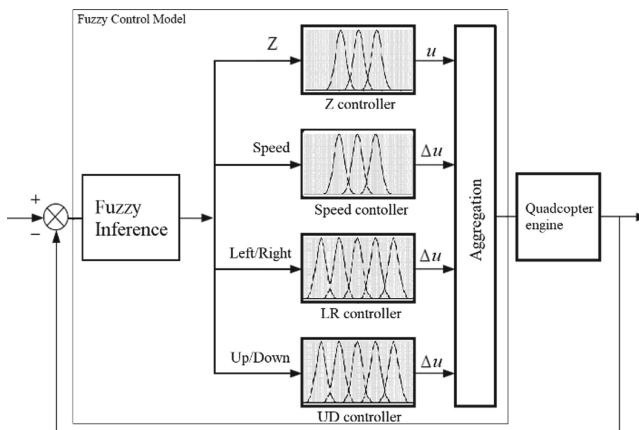


Fig. 2. Fuzzy control system for the quadcopter flight process.

Due to the fact that fuzzy controllers are synthesized without using a mathematical model and there are currently no conventional methods for the mathematical analysis of fuzzy controllers, it is impossible to assert that the solutions obtained on their basis are optimal.

4 Fuzzy Inference System for Automatic Maneuvering of the Quadcopter

The development of the algorithm for the formation of the autonomous flight trajectory of a quadcopter was carried out on the basis of the empirical analysis of its behavior in five sectors. The result of this analysis was the limited set of logically consistent verbal judgments formulated as the following information fragments:

d_1 : "If obstacle is not detected or it is too far away along the flight path of the quadcopter, then there is no necessity to change height, direction and to reduce velocity";

d_2 : "If the lidar detects any obstacle at a medium distance along the quadcopter motion path and the left sector in the horizontal plane is free, then it is essential to lose speed to the average, turn slightly to the left and do not change flight altitude";

d_3 : "If the lidar detects any obstacle at the close distance along the quadcopter motion path and the left sector in the horizontal plane is free, then it is essential to lose speed to a minimum, turn sharply to the left and do not change flight altitude";

d_4 : "If the lidar detects any obstacle at the medium distance along the quadcopter motion path, there is also any obstacle in the left sector in the horizontal plane at a not remote distance and the right sector in the horizontal plane is free, then it is essential to lose speed to the average, turn slightly to the right, and do not change flight altitude";

d_5 : "If the lidar detects any obstacle at the close distance along the quadcopter motion path, there is also any obstacle in the left sector in the horizontal plane at a not remote distance and the right sector in the horizontal plane is free, then it is essential to lose speed to the minimum, turn sharply to the right, and do not change flight altitude";

d_6 : "If the lidar detects any obstacle at the average distance along the quadcopter motion path, there are also obstacles in the left and right sectors in the horizontal plane at a not remote distance, and the upper sector in the vertical plane is free, then it is essential to lose speed to the average, slightly increase the flight altitude, and do not turn horizontally";

d_7 : "If the lidar detects any obstacle at the close distance along the quadcopter motion path, there are also obstacles in the left and right sectors in the horizontal plane at a not remote distance and the upper sector in the vertical plane is free, then it is essential to lose speed to the minimum, sharply increase the flight altitude, and do not turn horizontally";

d_8 : "If the lidar detects any obstacle at the average distance along the quadcopter motion path, there are also obstacles in the right, left and upper fields of vision at a non-remote distance, and the lower sector in the vertical plane is free, then it is essential to lose speed to the average, slightly reduce the flight altitude and do not turn horizontally";

d_9 : "If the lidar detects any obstacle at the close distance along the quadcopter motion path, there are also obstacles in the right, left and upper fields of vision at a not remote distance, and the lower sector in the vertical plane is free, then it is essential to lose speed to the minimum, sharply reduce the flight altitude, and do not turn horizontally";

d_{10} : "If the lidar detects any obstacle at the average distance along the quadcopter motion path, and the obstacle is also detected at the average distance to the left sector in the

horizontal plane, obstacles are detected at a not remote distance to the right horizontal plane, up and down, then it is essential to lose speed to the average while maintaining the direction and altitude of the flight”;

d_{11} : “If the lidar detects any obstacle at the close distance along the quadcopter motion path, and the obstacle is also detected at the average distance to the left in the horizontal plane, obstacles are detected at a not remote distance to the right in the horizontal plane, up and down in the vertical plane, then it is essential to lose speed to the minimum, turn sharply to the left, and do not turn vertically”;

d_{12} : “If the lidar detects any obstacle at the medium distance along the quadcopter motion path, and the obstacle is detected a close distance to the left in the horizontal plane, the obstacle is detected at the average distance to the right in the horizontal plane, and obstacles are detected at a not remote distance from down and up along the quadcopter motion path, then it is essential to lose speed to the average, turn sharply to the left without changing the direction and altitude of the flight”;

d_{13} : “If the lidar detects any obstacle at the close distance along the quadcopter motion path, and the obstacle is also detected at a close distance to the left in the horizontal plane, the obstacle is detected at the average distance to the right in the horizontal plane, and obstacles are detected from down and up along the direction at a not remote distance, then it is essential to lose speed to the minimum, turn sharply to the right, and do not turn vertically”;

d_{14} : “If the lidar detects any obstacle at the medium distance along the quadcopter motion path, the obstacle is detected at a close distance to the left in the horizontal plane, the obstacle is also detected at a close distance to the right in the horizontal plane, the obstacle is detected at the average distance from up, and the obstacle is detected at a not remote distance from down, then it is essential to lose speed to the average, turn sharply to the right without changing the direction and altitude of the flight”;

d_{15} : “If the lidar detects any obstacle at close distance along the quadcopter motion path, the obstacles are detected at the average distance to the left and to the right in the horizontal plane, as well as the obstacle is detected at a not remote distance from up, then it is essential to lose speed to the minimum, sharply increase flight altitude, and do not turn horizontally”;

d_{16} : “If the lidar detects any obstacle at the medium distance along the quadcopter motion path, obstacles are detected at the close distance to the right, left and higher along the direction, and the obstacle is detected at the average distance down the course, then it is essential to lose speed to the average without changing the direction and altitude of the flight”;

d_{17} : “If the lidar detects the obstacles at close distance along the quadcopter motion path, as well as to the right, left and up, however, the obstacle is detected at the average distance down the direction, then it is essential to lose speed to the minimum, sharply reduce the flight altitude, and do not turn horizontally”;

d_{18} : “If any obstacle is detected at the medium distance along the quadcopter motion path, obstacles are detected at the close distance to the right, left, up and down the direction, then it is essential to lose speed to the average without changing the direction and altitude of the flight”;

d_{19} : “If in all fields of the view the detected obstacles are at a close distance, then it is essential to lose speed to the minimum, turn sharply to the left in the horizontal plane, and do not turn vertically”.

Maneuvering to the right (or to the left) in the horizontal plane from the frontal impasse due to the presence of obstacles in all 5 fields of view (see fragment d_{19}), the quadcopter continues to traverse and, thereby, creates the new motion path for itself in accordance with principles formulated by $d_1 \div d_{19}$. Thus, test of all potential scenarios of collision with difficulties made it possible to establish the limited set of linguistic variables (see Table 1) and implications for generate the Fuzzy Inference System (FIS) able to fly a quadcopter under overland monitoring. Appropriate FIS is formulized as follows:

$$d_1: (x_1 = X_{11}) \Rightarrow (y_1 = Y_{11}) \& (y_2 = Y_{23}) \& (y_3 = Y_{33});$$

$$d_2: (x_1 = X_{12}) \& (x_2 = X_{21}) \Rightarrow (y_1 = Y_{12}) \& (y_2 = Y_{22}) \& (y_3 = Y_{33});$$

$$d_3: (x_1 = X_{13}) \& (x_2 = X_{21}) \Rightarrow (y_1 = Y_{13}) \& (y_2 = Y_{21}) \& (y_3 = Y_{33});$$

$$d_4: (x_1 = X_{12}) \& (x_2 = \neg X_{21}) \& (x_3 = X_{31}) \Rightarrow (y_1 = Y_{12}) \& (y_2 = Y_{24}) \& (y_3 = Y_{33});$$

$$d_5: (x_1 = X_{13}) \& (x_2 = \neg X_{21}) \& (x_3 = X_{31}) \Rightarrow (y_1 = Y_{13}) \& (y_2 = Y_{25}) \& (y_3 = Y_{33});$$

$$d_6: (x_1 = X_{12}) \& (x_2 = \neg X_{21}) \& (x_3 = \neg X_{31}) \& (x_4 = X_{41}) \Rightarrow (y_1 = Y_{12}) \& (y_2 = Y_{23}) \& (y_3 = Y_{34});$$

$$d_7: (x_1 = X_{13}) \& (x_2 = \neg X_{21}) \& (x_3 = \neg X_{31}) \& (x_4 = X_{41}) \Rightarrow (y_1 = Y_{13}) \& (y_2 = Y_{23}) \& (y_3 = Y_{35});$$

$$d_8: (x_1 = X_{12}) \& (x_2 = \neg X_{21}) \& (x_3 = \neg X_{31}) \& (x_4 = \neg X_{41}) \& (x_5 = X_{51}) \Rightarrow (y_1 = Y_{12}) \& (y_2 = Y_{23}) \& (y_3 = Y_{32});$$

$$d_9: (x_1 = X_{13}) \& (x_2 = \neg X_{21}) \& (x_3 = \neg X_{31}) \& (x_4 = \neg X_{41}) \& (x_5 = X_{51}) \Rightarrow (y_1 = Y_{13}) \& (y_2 = Y_{23}) \& (y_3 = Y_{31});$$

$$d_{10}: (x_1 = X_{12}) \& (x_2 = X_{22}) \& (x_3 = \neg X_{31}) \& (x_4 = \neg X_{41}) \& (x_5 = \neg X_{51}) \Rightarrow (y_1 = Y_{12}) \& (y_2 = Y_{23}) \& (y_3 = Y_{33});$$

$$d_{11}: (x_1 = X_{13}) \& (x_2 = X_{22}) \& (x_3 = \neg X_{31}) \& (x_4 = \neg X_{41}) \& (x_5 = \neg X_{51}) \Rightarrow (y_1 = Y_{13}) \& (y_2 = Y_{21}) \& (y_3 = Y_{33});$$

$$d_{12}: (x_1 = X_{12}) \& (x_2 = X_{23}) \& (x_3 = X_{32}) \& (x_4 = \neg X_{41}) \& (x_5 = \neg X_{51}) \Rightarrow (y_1 = Y_{12}) \& (y_2 = Y_{23}) \& (y_3 = Y_{33});$$

$$d_{13}: (x_1 = X_{13}) \& (x_2 = X_{23}) \& (x_3 = X_{32}) \& (x_4 = \neg X_{41}) \& (x_5 = \neg X_{51}) \Rightarrow (y_1 = Y_{13}) \& (y_2 = Y_{25}) \& (y_3 = Y_{33});$$

$$d_{14}: (x_1 = X_{12}) \& (x_2 = X_{23}) \& (x_3 = X_{33}) \& (x_4 = X_{42}) \& (x_5 = \neg X_{51}) \Rightarrow (y_1 = Y_{12}) \& (y_2 = Y_{23}) \& (y_3 = Y_{33});$$

$d_{15}: (x_1 = X_{13}) \& (x_2 = X_{23}) \& (x_3 = X_{33}) \& (x_4 = X_{42}) \& (x_5 = \neg X_{51}) \Rightarrow (y_1 = Y_{13}) \& (y_2 = Y_{23}) \& (y_3 = Y_{35});$

$d_{16}: (x_1 = X_{12}) \& (x_2 = X_{23}) \& (x_3 = X_{33}) \& (x_4 = X_{43}) \& (x_5 = X_{52}) \Rightarrow (y_1 = Y_{12}) \& (y_2 = Y_{23}) \& (y_3 = Y_{33});$

$d_{17}: (x_1 = X_{13}) \& (x_2 = X_{23}) \& (x_3 = X_{33}) \& (x_4 = X_{43}) \& (x_5 = X_{52}) \Rightarrow (y_1 = Y_{13}) \& (y_2 = Y_{23}) \& (y_3 = Y_{31});$

$d_{18}: (x_1 = X_{12}) \& (x_2 = X_{23}) \& (x_3 = X_{33}) \& (x_4 = X_{43}) \& (x_5 = X_{53}) \Rightarrow (y_1 = Y_{12}) \& (y_2 = Y_{23}) \& (y_3 = Y_{33});$

$d_{19}: (x_1 = X_{13}) \& (x_2 = X_{23}) \& (x_3 = X_{33}) \& (x_4 = X_{43}) \& (x_5 = X_{53}) \Rightarrow (y_1 = Y_{13}) \& (y_2 = Y_{21}) \& (y_3 = Y_{33}).$

Table 1. FIS parameters and their characteristics.

<i>Inputs</i>			
Symbol	Variable name	Universe	Term set
x_1	Obstacle distance along the flight path	$[0, 1]$	$\{X_{11} = \text{SIGNIFICANT}, X_{12} = \text{AVERAGE}, X_{13} = \text{INSIGNIFICANT}\}$
x_2	Obstacle distance to the left in the horizontal plane	$[0, 1]$	$\{X_{21} = \text{SIGNIFICANT}, X_{22} = \text{AVERAGE}, X_{23} = \text{INSIGNIFICANT}\}$
x_3	Obstacle distance to the right in the horizontal plane	$[0, 1]$	$\{X_{31} = \text{SIGNIFICANT}, X_{32} = \text{AVERAGE}, X_{33} = \text{INSIGNIFICANT}\}$
x_4	Obstacle distance to the higher in the vertical plane	$[0, 1]$	$\{X_{41} = \text{SIGNIFICANT}, X_{42} = \text{AVERAGE}, X_{43} = \text{INSIGNIFICANT}\}$
x_5	Obstacle distance to the lower in the vertical plane	$[0, 1]$	$\{X_{51} = \text{SIGNIFICANT}, X_{52} = \text{AVERAGE}, X_{53} = \text{INSIGNIFICANT}\}$
<i>Outputs</i>			
y_1	Velocity	$[0, 1]$	$\{Y_{11} = \text{FULL}, Y_{12} = \text{AVERAGE}, Y_{13} = \text{ZERO}\}$
y_2	Left/Right	$[-0.5, 0.5]$	$\{Y_{21} = \text{SHARPLY TO THE LEFT}, Y_{22} = \text{SLIGHTLY TO THE LEFT}, Y_{23} = \text{IS ABSENT}, Y_{24} = \text{SLIGHTLY TO THE RIGHT}, Y_{25} = \text{SHARPLY TO THE RIGHT}\}$
y_3	Up/Down	$[-0.5, 0.5]$	$\{Y_{31} = \text{SHARPLY UP}, Y_{32} = \text{SLIGHTLY UP}, Y_{33} = \text{IS ABSENT}, Y_{34} = \text{SLIGHTLY DOWN}, Y_{35} = \text{SHARPLY DOWN}\}$

The membership functions of fuzzy sets, which describe the terms of the input and output linguistic variables of the FIS, have been established empirically. In the notation of the MATLAB package, the membership functions of fuzzy inputs and outputs of the FIS are shown in Figs. 3 and 4, respectively.

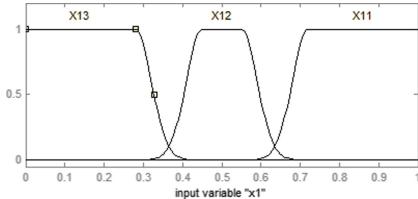


Fig. 3. Terms of input linguistic variable “Obstacle distance along the flight path”.

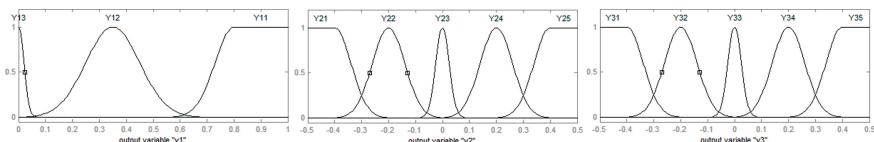


Fig. 4. Terms of output linguistic variables: y_1 —Velocity, y_2 —Left/Right, y_3 —Up/Down.

Table 2 presents the responses of the FIS to the relevant inputs provided by the implicative rules $d_1 \div d_{19}$. The corresponding interactive functional window in the notation of the MATLAB\FIS editor is shown in Fig. 5.

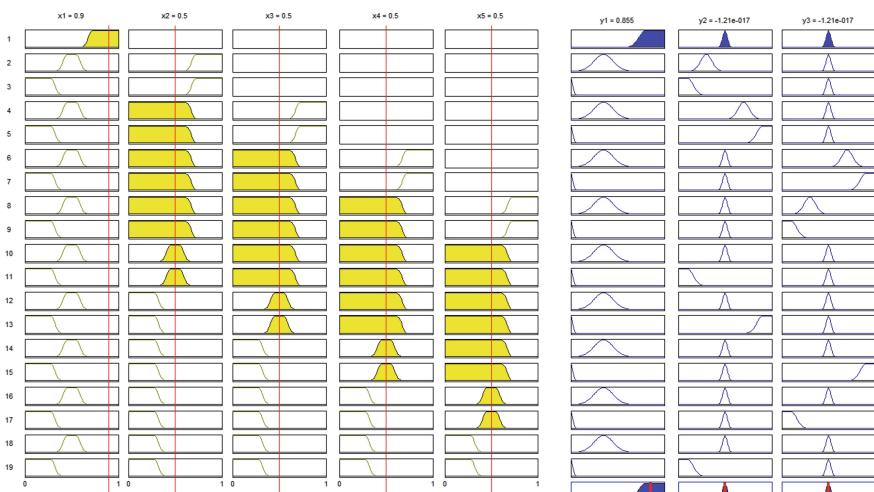


Fig. 5. Functional window in the MATLAB\FIS notation.

Table 2. Numerical implementation of the rules $d_1 \div d_{19}$ using the MATLAB\FIS editor.

№	Inputs					Outputs		
	x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3
1	0.95	0.50	0.50	0.50	0.50	0.855	0	0
2	0.50*	0.95	0.50	0.50	0.50	0.350	-0.200	0
3	0.10	0.90	0.50	0.50	0.50	0.013	-0.411	0
4	0.50	0.50	0.90	0.50	0.50	0.350	0.200	0
5	0.10	0.50	0.90	0.50	0.50	0.013	0.411	0
6	0.50	0.50	0.50	0.90	0.50	0.350	0	0.200
7	0.10	0.50	0.50	0.90	0.50	0.013	0	0.411
8	0.50	0.50	0.50	0.50	0.90	0.350	0	-0.200
9	0.10	0.50	0.50	0.50	0.90	0.013	0	-0.411
10	0.50	0.50	0.40	0.40	0.40	0.350	0	0
11	0.10	0.50	0.50	0.50	0.50	0.013	-0.411	0
12	0.50	0.10	0.50	0.50	0.50	0.350	0	0
13	0.10	0.10	0.50	0.50	0.50	0.013	0.411	0
14	0.50	0.10	0.10	0.50	0.50	0.350	0	0
15	0.10	0.10	0.10	0.50	0.50	0.013	0	0.411
16	0.50	0.10	0.10	0.10	0.50	0.350	0	0
17	0.10	0.10	0.10	0.10	0.50	0.013	0	-0.411
18	0.50	0.10	0.10	0.10	0.10	0.350	0	0
19	0.10	0.10	0.10	0.10	0.10	0.013	-0.411	0

* If the rule does not mention the presence or absence of any obstacle in each of the five review sectors, then the corresponding input action is assigned the value of 0.5, as the average value over the universe [0, 1].

5 Quadcopter Control Based on the Artificial Neural Network

Artificial neural networks (ANNs) represent an alternative tool for the development of intelligent control systems to fuzzy controllers. The use of ANNs for solving control problems is different [5–9]: they can be used as a tool for evaluation, approximation, prediction of parameters of control objects, and direct regulation. Interest in ANNs as the basis for organizing control systems is explained by their ability to work with fuzzy, inaccurate data, which is typical for many technical control objects, including quadcopters. Another positive property of ANNs is the possibility of organizing robust controllers based on them [7], which have significant flexibility to changes in the parameters of control objects and the environment, as well as sensor noise. This provides a significant increase in the accuracy of the controller and expands the range of its practical use in comparison with classical PID-, PI-, PD-controllers.

Features of the ANN make it possible to implement complex, non-linear dependencies on its basis. When designing regulators, this property facilitates the synthesis of the control system, since ANN-based implementation of interconnected control systems does not require the developer to have exact numerical knowledge about the mutual influence of internal quantities. As a disadvantage, one can point out the need for a certain number of iterations of the controller synthesis, the high dependence of the design process on the developer's intuition, the need to obtain and prepare training samples in advance, which in itself is not an easy task.

The presence of the ability of dynamic adaptation makes it possible to actively use the ANN as the basis of self-adjusting controllers [7]. The possibility of flexible settings and versatility make the use of ANNs a powerful tool for the developer. The construction of such control systems requires certain assumptions, for example, the presence of an idealized (conceptual) mathematical model that allows you to set the desired behavior of the control object.

The complexity of the practical use of ANNs is due to the large number of existing neural network architectures and the inability to determine in advance which of them is most suitable for solving a practical problem. The problem is solved in various ways, for example, by conducting experiments with subsequent choice or on the basis of expert experience (heuristic knowledge). Quite often, for use as the basis of the regulator, feedforward neural networks are used. To ensure that the dynamics of the object is taken into account, specialized delay elements are introduced into the network structure [7].

Actually, our further reasoning is based on this paradigm.

Table 3 presents various scenarios for the behavior of a quadcopter in the presence of obstacles in the five sectors, which was obtained by MATLAB\FIS editor. At the same time, the flight trajectory is formed based on the operational five-criteria assessment of the presence (or absence) of obstacles in all quadcopter viewing sectors, where the terms of input linguistic variables x_k ($k = 1 \div 5$) act as qualitative evaluation criteria. An analytical approach to such assessment makes it possible to compare alternative quadcopter flight routes using the desired vector (y_1, y_2, y_3) , reflecting the relative influence of factors x_k in the form of the mapping $F: R^5 \rightarrow R^3$. Therefore, it is advisable to represent the working model for the formation of the flight path of the quadcopter in the form of a "black box", where inputs and outputs are initially determined by expert judgments that constitute a heuristic knowledge base (an external representation of the given problem) in the form of information fragments $d_1 \div d_{19}$. Thus, to build an analytical model for the formation of the quadcopter flight trajectory, it is advisable to use a multilayer neural network, which, as known, is an effective tool for knowledge compilation relative to behavior of a quadcopter under overland monitoring.

Table 3. Quadcopter maneuvering scenarios under overland monitoring.

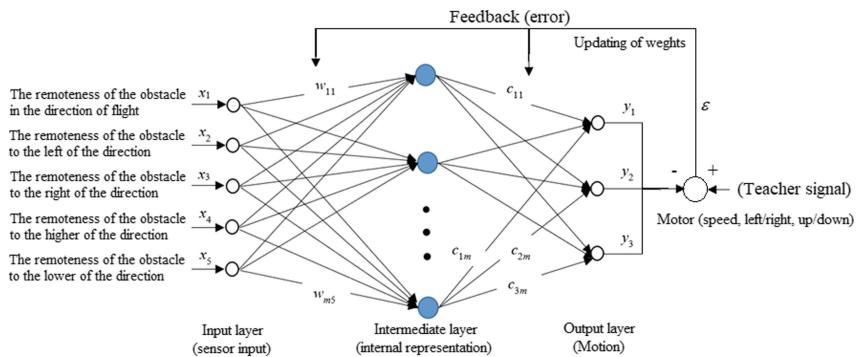
№	Inputs					Outputs		
	x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3
1	0.433	0.291	0.628	0.076	0.961	0.350	0.055	-0.190
2	0.611	0.824	0.460	0.862	0.855	0.380	-0.194	0.000
3	0.374	0.622	0.139	0.750	0.702	0.337	-0.154	0.215
4	0.712	0.738	0.568	0.438	0.469	0.854	0.000	0.000
5	0.922	0.535	0.320	0.107	0.102	0.855	0.000	0.000
6	0.396	0.283	0.318	0.925	0.001	0.349	0.000	0.204
7	0.026	0.563	0.311	0.177	0.959	0.013	-0.001	-0.410
8	0.627	0.457	0.127	0.826	0.117	0.467	0.000	0.171
9	0.570	0.359	0.197	0.404	0.663	0.351	0.000	-0.169
10	0.296	0.631	0.224	0.938	0.827	0.019	-0.097	0.384
11	0.426	0.026	0.464	0.954	0.842	0.350	0.000	0.200
12	0.602	0.563	0.781	0.415	0.181	0.363	0.197	0.000
13	0.510	0.253	0.232	0.894	0.429	0.350	0.000	0.200
14	0.333	0.292	0.856	0.275	0.762	0.131	0.386	0.000
15	0.645	0.632	0.452	0.743	0.643	0.658	-0.103	0.103
16	0.524	0.833	0.781	0.985	0.546	0.350	-0.200	0.000
17	0.152	0.557	0.453	0.289	1.000	0.013	-0.001	-0.410
18	0.709	0.310	0.242	0.138	0.410	0.853	0.000	0.000
19	0.389	0.113	0.828	0.625	0.121	0.347	0.209	0.000
20	0.133	0.292	0.972	0.499	0.693	0.013	0.411	0.000
21	0.534	0.148	0.532	0.721	0.969	0.350	0.000	0.200
22	0.492	0.921	0.376	0.131	0.585	0.350	-0.200	0.000
23	0.283	0.965	0.307	0.287	0.305	0.015	-0.410	0.000
24	0.178	0.407	0.298	0.385	0.427	0.016	-0.394	0.000
25	0.004	0.521	0.704	0.257	0.895	0.013	0.387	-0.088
26	0.557	0.562	0.291	0.562	0.910	0.350	-0.001	-0.200
27	0.296	0.592	0.922	0.910	0.104	0.019	0.400	0.000
28	0.189	0.123	0.004	0.101	0.411	0.015	-0.010	-0.399
29	0.410	0.491	0.864	0.569	0.522	0.350	0.201	0.000
30	0.605	0.961	0.766	0.756	0.306	0.367	-0.196	0.000
31	0.702	0.644	0.219	0.000	0.740	0.851	-0.001	-0.001

(continued)

Table 3. (continued)

№	Inputs					Outputs		
	x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3
32	0.433	0.629	0.741	0.405	0.838	0.350	0.144	0.000
33	0.465	0.793	0.395	0.745	0.086	0.350	-0.200	0.000
34	0.065	0.004	0.789	0.553	0.318	0.013	0.411	0.000
35	0.026	0.245	0.480	0.651	0.787	0.014	0.000	-0.195

In the problem under consideration, “external knowledge” about 35 possible scenarios for the formation of the quadcopter’s flight path is presented in the form of the following information model (see Table 3) $\{(x_{1j}, x_{2j}, x_{3j}, x_{4j}, x_{5j}) \rightarrow (y_{1j}, y_{2j}, y_{3j})\}_{j=1}^{35}$. Then the mapping F can be approximated by the three-layer feedforward neural network (FNN) (see Fig. 6), which induces signal for each of the r -th outputs ($r = 1 \div 3$) in the following form $z_{rj} = \sum_{k=1}^m c_{rk} \varphi[\sum_{i=1}^5 w_{ki} x_{ij} - \theta_k]$ ($j = 1 \div 35$), where m is the number of nonlinear neurons in the “hidden” layer, selected by the user during the simulation by rule-of-thumb method; w_{ki} и c_{rk} ($k = 1 \div m$) are the weights of input and output synaptic connection, respectively; θ_k is the bias (threshold) of the k -th nonlinear neuron from the “hidden” layer; $\varphi(\cdot)$ is the activation function of the non-linear neuron from the “hidden” layer, for example, sigmoidal type $\varphi(t) = 1/(1 + e^{-t})$.

**Fig. 6.** The three-layer neural network.

The neural network, which compiles the heuristic knowledge relative to the flight routing of the quadcopter, processes input vectors with components according to the number of criteria for assessing the “proximity” of obstacles in all sectors of the view, which are represented as numbers from the segment $[0; 1]$. As a result of adjustment of the parameters (weights of synaptic connections and thresholds), the neural network

can be approximate the imaginary mapping $F: R^5 \rightarrow R^3$, presented in tabular form (see Table 3). For each case, at its output, the neural network must generate command control of the quadcopter flight path in the form of the three-component vector $(y_1, y_2, y_3) = (\text{speed}, \text{left/right}, \text{up/down})$. In particular, the neural network must respond with the output vector $(0.350, 0.000, 0.200)$ at the input vector position $(0.510, 0.253, 0.232, 0.894, 0.429)$ (see scenario 13 from Table 3). As shown in Fig. 7, the network has one hidden layer, which consists of 5 nonlinear neurons with log-sigmoid activation functions, the range of which allows to realize: *speed* within the interval $[0, 1]$, *left/right* within the interval $[-0.5, 0.5]$, *up/down* within the interval $[-0.5, 0.5]$.

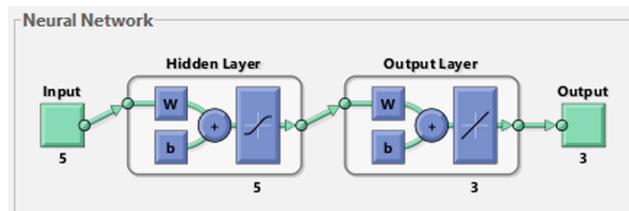


Fig. 7. Three-layer feedforward neural network in MATLAB notation.

After training, testing and validation of the neural network (see Fig. 8), the corresponding products are formed (results of pairs of the “input–output” type) and summarized in Table 4.

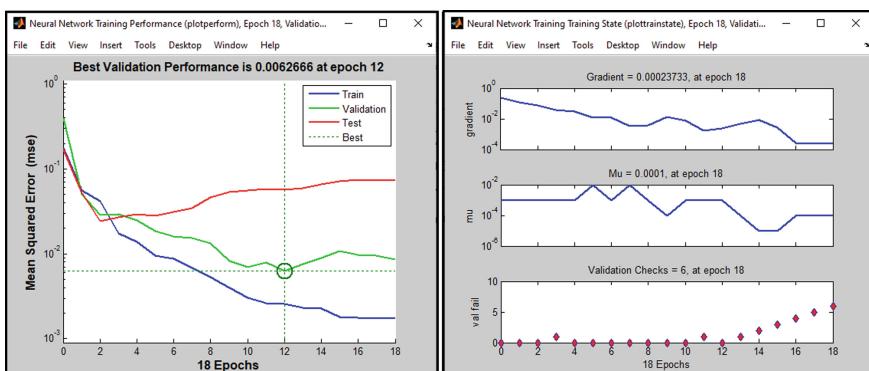


Fig. 8. Results of training, testing and validation of the neural network in MATLAB notation.

Table 4. Control signals are generated by FNN.

Scenario	Inputs				Outputs					
					Using FIS				Using FNN	
	x_1	x_2	x_3	x_4	y_1	y_2	y_3	z_1	z_2	z_3
1	0.433	0.291	0.628	0.076	0.961	0.350	0.055	-0.190	0.3805	0.0638
2	0.611	0.824	0.460	0.855	0.380	-0.194	0.000	0.3720	-0.2092	0.0435
3	0.374	0.622	0.139	0.750	0.702	0.337	-0.154	0.215	0.1603	-0.1010
4	0.712	0.738	0.568	0.438	0.469	0.854	0.000	0.000	0.8450	-0.0334
5	0.922	0.535	0.320	0.107	0.102	0.855	0.000	0.000	0.8925	0.0019
6	0.396	0.283	0.318	0.925	0.001	0.349	0.000	0.204	0.3445	-0.0126
7	0.026	0.563	0.311	0.177	0.959	0.013	-0.001	-0.410	0.0161	-0.0592
8	0.627	0.457	0.127	0.826	0.117	0.467	0.000	0.171	0.4858	0.0041
9	0.570	0.359	0.197	0.404	0.663	0.351	0.000	-0.169	0.3067	0.0290
10	0.296	0.631	0.224	0.938	0.827	0.019	-0.097	0.384	0.1858	-0.1288
11	0.426	0.026	0.464	0.954	0.842	0.350	0.000	0.200	0.3424	0.0212
12	0.602	0.563	0.781	0.415	0.181	0.363	0.197	0.000	0.8288	-0.1886
13	0.510	0.253	0.232	0.894	0.429	0.350	0.000	0.200	0.3598	-0.0156
14	0.333	0.292	0.856	0.275	0.762	0.131	0.386	0.000	0.1945	0.3646
15	0.645	0.632	0.452	0.743	0.643	0.658	-0.103	0.103	0.6319	-0.0920
16	0.524	0.833	0.781	0.985	0.546	0.350	-0.200	0.000	0.3832	-0.0907
17	0.152	0.557	0.453	0.289	1.000	0.013	-0.001	-0.410	0.0147	0.0025

(continued)

Table 4. (continued)

Scenario	Inputs	Outputs					Using FNN				
		x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	z_1	z_2
18	0.709	0.310	0.242	0.138	0.410	0.853	0.000	0.000	0.8375	0.0191	-0.0102
19	0.389	0.113	0.828	0.625	0.121	0.347	0.209	0.000	0.2847	0.2029	0.0528
20	0.133	0.292	0.972	0.499	0.693	0.013	0.411	0.000	0.0004	0.4498	-0.0840
21	0.534	0.148	0.532	0.721	0.969	0.350	0.000	0.200	0.3557	0.0093	0.1589
22	0.492	0.921	0.376	0.131	0.585	0.350	-0.200	0.000	0.4234	-0.0783	-0.1872
23	0.283	0.965	0.307	0.287	0.305	0.015	-0.410	0.000	0.0327	-0.3269	-0.1404
24	0.178	0.407	0.298	0.385	0.427	0.016	-0.394	0.000	0.0261	-0.3941	-0.0363
25	0.004	0.521	0.704	0.257	0.895	0.013	0.387	-0.088	-0.0061	0.4013	-0.0879
26	0.557	0.562	0.291	0.562	0.910	0.350	-0.001	-0.200	0.1962	-0.0842	-0.2522
27	0.296	0.592	0.922	0.910	0.104	0.019	0.400	0.000	0.0552	0.3839	-0.0457
28	0.189	0.123	0.004	0.101	0.411	0.015	-0.010	-0.399	0.0470	-0.1082	-0.3173
29	0.410	0.491	0.864	0.569	0.522	0.350	0.201	0.000	0.3144	0.2509	0.0694
30	0.605	0.961	0.766	0.756	0.306	0.367	-0.196	0.000	0.7688	-0.1998	0.3088
31	0.702	0.644	0.219	0.000	0.740	0.851	-0.001	-0.001	0.8111	-0.0211	-0.0081
32	0.433	0.629	0.741	0.405	0.838	0.350	0.144	0.000	0.3401	0.1108	-0.0560
33	0.465	0.793	0.395	0.745	0.086	0.350	-0.200	0.000	0.1829	-0.2772	0.1751
34	0.065	0.004	0.789	0.553	0.318	0.013	0.411	0.000	0.0047	0.4062	-0.0850
35	0.026	0.245	0.480	0.651	0.787	0.014	0.000	-0.195	0.0159	-0.0078	-0.0427

6 Conclusion

Technical complexity, topological multi-coupling of the structure and non-linearity of multi-copters, as well as the presence of unidentified parameters dictate the need to find alternative solutions to control problems of such dynamic objects. Expert-empirical studies provide some opportunity to get certain ideas about how to control quadcopters under overland monitoring, about the features of their practical application: advantages and disadvantages. Therefore, it seems possible to compile heuristic knowledge obtained by expert-empirical analysis using the methods of neural network and/or neuro-fuzzy (hybrid) modeling. In particular, it is promising to develop the quadcopter control by a structured hierarchical neural network, which combined two types of multilayer neural networks: so-called “reason” and “instinct” networks, which is connected to each other by short-term memory units used in process time-dependent data. But this is the subject of the next study.

References

1. Michael, D.S.: Simulation and control of a quadrotor unmanned aerial vehicle. http://www.uknowledge.uky.edu/gradschool_theses/93. Last accessed 21 Sep 2022
2. Beloglazov, D.A. et al.: Analysis of the features of the practical use of controllers for automatic control systems for quadcopters. Eng. Bull. Don **3**, 1–18 (2015), <http://www.ivdon.ru/ru/magazine/archive/n3y2015/3078>. Last accessed 21 Sep 2022 (in Russian)
3. Sharma, A., Barve, A.: Controlling of quad-rotor UAV using PID controller and fuzzy logic controller. Int. J. Electr., Electron. Comput. Eng. **2277–2626**, 38–42 (2012)
4. Lopez, V., Morata, F.: Intelligent fuzzy controller of a quadrotor. In: International conference on intelligent system and knowledge engineering, pp. 141–146 (2010)
5. Nicoli, C., Macnab, C., Ramirez-Serrano, A.: Robust neural network control of a quadrotor helicopter. http://www.academia.edu/8226428/ROBUST_NEURAL_NETWORK_CONTROLLER_OF_A_QUADROTOR_HELICOPTER. Last accessed 21 Sep 2022
6. Burka, A., Foster, S.: Neato Quadcopters. <http://www.web.cs.swarthmore.edu/~meeden/cs81s12/papers/AlexSethPaper.pdf>. Last accessed 21 Sep 2022
7. Shepherd, J., Tumer, K.: Robust neuro-control for a micro quadrotor. In: Genetic and evolutionary computation conference, p. 8, Portland, Oregon, USA, (2010)
8. Vijaya Kumar, M. et al.: A direct adaptive neural command controller design for a nun stable helicopter. Eng. Appl. Artif. Intell. **22**, 181–191 (2009)
9. Suresha, S., Sundararajan, N.: An on-line learning neural controller for helicopters performing highly nonlinear maneuvers. Appl. Soft Comput. **12**, 360–371 (2012)



POWOP: Weather-Based Power Outage Prediction

Natalie Gdanitz^(✉), Lotfy H. Abdel Khaliq, Agbodzea Pascal Ahiagble,
Sabine Janzen, and Wolfgang Maass

Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Saarbrücken,
Germany

{natalie.gdanitz, lotfy.abdel_khaliq, agbodzea_pascal.ahiagble,
sabine.janzen, wolfgang.maass}@dfki.de

Abstract. The worldwide energy-crisis poses a critical risk to the energy-intensive process industry. Rising costs for gas lead to increased usage of electrical power (e.g. for heating) that network operators are not prepared for. Weather-dependent energy-sources (e.g. windparks, solar panels) lead to additional fluctuations within the power grid. In worst case a simultaneous and prolonged loss of gas supply and electricity will lead to network bottlenecks, or complete network shutdowns—blackouts. For manufacturers, power outages thereby lead to severe consequences (i.e. waste, broken machines, additional costs), with only limited options to prevent them. Within this paper we highlight the implementation of POWOP, a weather-based service for *POWer Outage Prediction* that increases the resilience within the German process industry (e.g. paper, glass or chemical production). By using a predictive analytics forecasting model and a knowledge graph consisting of semantically enhanced Scenario Patterns, we are able to predict regional power outages for the next 7 days and to provide action recommendations for potential actors. Our publicly available web-application was evaluated for 15 locations of paper manufacturers in the German region Bavaria and will be demonstrated within a screencast.

Keywords: Energy-driven crisis · Outage prediction · Weather · Process industry · Scenario pattern · Service · Demo

1 Introduction

Hardly any topic is currently discussed as much as the worldwide energy crisis, in particular the restricted gas supply [10]. Widespread power outages as potential consequence pose a critical issue, specifically in the energy-intensive process industry, e.g. glass and paper production or chemical industry. Minimal fluctuations within the power grid, below 49.8 and over 50.2 H, can already lead to serious effects due to uncontrolled shutdowns of frequency sensitive machines [8]. Ultimately this means that within automated processes chemical reactions are spontaneously interrupted, tons of waste are produced and machine parts

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

K. Arai (Ed.): IntelliSys 2023, LNNS 822, pp. 75–83, 2024.

https://doi.org/10.1007/978-3-031-47721-8_5

can break resulting in high repair costs and personnel expenses for manufacturers [1,2,8]. In case of gas shortages and a related increase in costs, industry as well as private households will switch to alternative methods such as electrical power in order to heat, for their energy supply of machines etc. As distribution network operators and their networks are not prepared for such a setting, this will lead to network bottlenecks, voltage drops due to load shortages as well as network shutdowns [11,13,21]. What contributes to this situation and causes additional fluctuations within the grid, is the increased usage of weather-dependent energy-sources for power generation [4] in order to compensate for the occurring gas shortage. Wind turbines and solar panels can thereby increase instability within power grids [9,12] as they produce power inconsistently based on current weather conditions (e.g. storms, sun hours). One example for this can be seen within the German region Bavaria, as they increasingly depend on installed solar systems that do not produce enough power in the winter season [14]. Worst case will be a simultaneous and prolonged loss of electricity and gas supply—a blackout. Although such power outages and grid fluctuations are a well known problem and current threat, there are not many available measures to prevent them. For manufacturers these include the acquisition of expensive proprietary power plants or speculative measures like an implicit gut feeling based on expertise [3,7]. Network operators on the other hand focus on dispatch and re-dispatch measures to balance occurring fluctuations in the grid [8,14]. In order to enable early preventive actions to outages and fluctuations based on weather changes especially in the upcoming fall and winter seasons, a prediction of potential events is needed.

Authors of previous research already addressed weather-related outage prediction focusing on predictive analytic methods for outages caused by extratropical storms [18], logistic regression or decision trees [17], graph neural networks [16] and regression trees [5]. Orsato et al. developed a tool for anticipating energy disruptions based on climate changes [15]. In contrast, the authors of [19,20] focus on simulated storm data, while calling out for contextual crisis descriptions of outage events. In addition to missing context, these papers do not focus on presenting their predictions within a demonstration of an intuitive service that also recommends actions for the actual user. Based on this gap, we formerly developed AISOP [8]—a model for AI-based scenario planning in the prediction and description of crisis situations. AISOP addresses crisis events (e.g. outages) within four steps: (1) Learning, (2) anticipating, (3) monitoring and (4) responding to a crisis. Initially, semantically enhanced Scenario Patterns are filled based on historical data describing crisis scenarios [8]. The description includes contextual information of a crisis, reason, impact and location, involved actors, measures, resources involved within these measures, the data source and interlinked historical events [8]. By applying predictive analytic methods to historical datasets, a forecasting model is generated and applied to current data to make a prediction. The predicted outage events are mapped onto the Scenario Pattern structure and given as result to the user [8].

By using AISOP [8] as basis, we implemented POWOP as a service for *POWer Outage Prediction* that enables decision makers to anticipate outages early, to receive contextual information about crisis events and recommendations to react accordingly. An exemplary application and demonstration of the service will be presented within a screencast. POWOP was implemented as a web application and will be provided as publicly available open source code, in order to increase manufacturer's resilience within the upcoming seasons.

2 POWOP Service Implementation

By enhancing the presented PoC within [8], we implemented an intuitive user service for manufacturers within the energy-intensive process industry. Our service architecture consists of three levels (cf. Fig. 1): A **Data Repository** for data allocation, a logic level for generating **Scenario Patterns** and state-of-the-art **Machine Learning** application and a **User Interface** for presenting the results to the user.

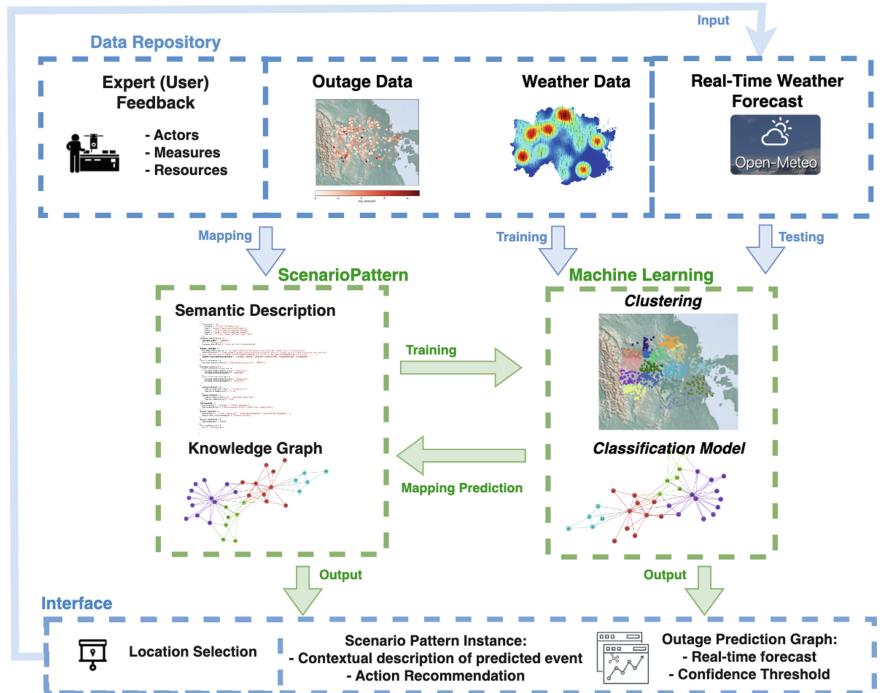


Fig. 1. Technical architecture of POWOP.

2.1 Data Repository

The **Data Repository** collects datasets from different sources to train and test our framework: The *Outage dataset* contains location-specific information on power outages (e.g. date, time, duration, city, reason, planned/unplanned occurrence) collected from the German Federal Network Agency,¹ while the *Weather dataset* includes weather characteristics (e.g. windspeed, wingust, temperature, rain, thunder) obtained from the NCEI database² [8]. Both datasets collect information for the period 2012–2020. In addition to these datasets, *User input* on actors that react onto an outage event, on measures that should be taken and on resources to be used were collected.

2.2 Scenario Patterns

Furthermore, we implemented **Scenario Patterns** within a *semantic description* using JSON-LD as format. Semantic web standards such as the vocabularies schema.org,³ DCMI,⁴ DCAT⁵ and PROV Ontology⁶ were applied (cf. Listing 1). Respectively historical information from the collected datasets were filled into the Scenario Pattern’s structure as historical crisis scenarios via a simple mapping of the included data features. By using the JSON-LD format and by registering related events within the history entity [8], the resulting descriptions form a network of Scenario Patterns within a knowledge graph, that can further be transferred into a knowledge graph database such as Neo4j⁷ using Cypher script.⁸

Listing 1. JSON-LD Instance of a filled Scenario Pattern

```

1 {  
2   "@context": [{"  
3     "schema": "http://schema.org",  
4     "dct": "http://purl.org/dc/terms/",  
5     "dcat": "http://www.w3.org/ns/dcat#",  
6     "prov": "http://www.w3.org/ns/prov#",  
7     "pairs": "https://www.pairs-projekt.de/"  
8   }],  
9   "schema:identifier":{  
10    "schema:name": "Outage",  
11    "@id": "Outage_987",  
12    "schema:DateTime": "2020-09-13T13:38:56+0000"  
13  },  
14  "pairs:Context":{
```

¹ https://www.bundesnetzagentur.de/DE/Fachthemen/ElektrizitaetundGas/Versorgungssicherheit/Versorgungsunterbrechungen/Auswertung_Strom/start.html.

² <https://www.noaa.gov/>.

³ <https://schema.org/>.

⁴ <http://purl.org/dc/terms/>.

⁵ <https://www.w3.org/ns/dcat>.

⁶ <https://www.w3.org/TR/2013/REC-prov-o-20130430/>.

⁷ <https://neo4j.com/>.

⁸ <https://neo4j.com/labs/apoc/4.3/cypher-execution/>.

```

15   "schema:description": "Outage based on shutdown windturbines; power
16   grid fluctuation",
17   "dcat:Dataset": "temp ,dewp ,slp ,stp ,visib ,wdsp ,mxpsd ,gust ,max ,min ,prcp ,
18   fog ,rain_drizzle ,snow_ice_pellets ,thunder
19   ,423.32.9,31.2,1029.7,995.5798903987854,6.9,2.7,7.0,...",
20   "pairs:InfluentialFactors": "Autumn Season , weather conditions ,
21   Thunderstorm , Windspeed"
22 },
23 "dct:Provenance": {
24   "schema:Organization": ["Bundesnetzagentur" , "NCEI"]
25 },
26 "schema:location": {
27   "pairs:ScenarioLocation": {
28     "schema:addressLocality": "Raubling" ,
29     "schema:addressRegion": "Bavaria"
30   },
31 },
32 "pairs:Reason": {
33   "pairs:Precondition": "Thunderstorm" ,
34   "pairs:Probability": "0.78"
35 },
36 "pairs:Effect": {
37   "pairs:Postcondition": "machine downtime" ,
38   "pairs:Complexity": "Low"
39 },
40 "prov:Agent": {
41   "prov:Role": ["Worker" , "Plant Manager"] ,
42   "prov:Activity": ["Maintenance work" , "Technical expertise"]
43 },
44 "pairs:Measure": {
45   "prov:Plan": ["Plan downtime" , "Plan maintenance" , "Controlled Shutdown
46   "],
47   "pairs:ActivityCategory": "precautionary"
48 },
49 "pairs:Resource": {
50   "prov:Entity": "None"
51 },
52 "dct:hasVersion": {
53   "@id": "Outage_430"
54 }

```

2.3 Machine Learning

Within the **Machine Learning** component, we first prepared our datasets for further use and merged weather and outage data respectively by their date. In doing so, we found a scarcity of occurring outages per city which led to an imbalanced situation [8]. In particular, only 2% of the data could be used effectively. This percentage was further reduced to 1.5 % as more than one outage could occur per city on each day. To cater this issue, we used a K-means *clustering approach* to group nearby cities with similar weather behaviour into one cluster and aggregated their outages, which increased the data quality to work with [8]. In particular, given the set of n cities' coordinates $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, with $K = 19$, K-means groups the cities into 19 clusters according to the distance using latitude and longitude. Once the algorithm converges, the data records for each cluster are aggregated and normalized to mitigate bias of high range

features. The final dataset contains features such as cluster-id, weather-station-id, date and weather-related features while an outage is used as a binary label [8].

In order to train our forecasting model, the pre-processed dataset is split into 70% for training, 10% for validation and 20% for testing purposes. The training set is used as input to our *classification model*, for which we use the state-of-the-art gradient boosting algorithm *XGBoost* [6]. However, XGBoost has many hyperparameters that need to be tuned which makes it infeasible to perform a Grid-Search given the large hyperparameter space [6]. Instead, OPTUNA⁹ was used as framework to perform automatic hyperparameter optimization.

Furthermore, the forecasting model has been applied on real-time weather information in order to predict outages for the next 7 days. A location selected by the user within the **User Interface** is therefore used as input to the weather API *OpenMeteo*,¹⁰ which collects real-time weather information. These features are then passed to the Machine Learning component, are normalized and used as input to XGBoost to get the prediction.

Our model was tested on 15 chosen locations of paper manufacturers within Bavaria, Germany. Experimental results showed the effectiveness of our model in capturing both outage and non-outage events, achieving a 81.2% overall accuracy and 70% sensitivity, stating the correct identification of outages specifically [8]. The implementation of the Machine Learning component was realized in Python.

2.4 Graphical User Interface

After the user has selected a *location* for generating prediction results, a regional forecast of power outages for the next 7 days is shown within a *prediction graph* (cf. Fig. 2). The position of each bubble on the graph depends on the predicted day and the probability of the actual occurrence of the predicted event (confidence level).

Potential outages are represented by red bubbles, while green bubbles illustrate that there will be no outage on that day. Figure 2 thereby shows a potential outage for the city Raubling with a probability of 74%. The orange line represents a threshold for the confidence level, meaning that events above are more likely to occur. In case of a predicted outage, a filled *Scenario Pattern Instance* is generated and integrated using Highcharts¹¹ in order to provide potential decision makers with relevant information about the predicted event (cf. Fig. 3). We used JavaScript for the implementation of the User Interface component. A demonstration of the developed service¹² is presented within a screencast.¹³

⁹ <https://optuna.org/>.

¹⁰ <https://open-meteo.com/en>.

¹¹ <https://www.highcharts.com/>.

¹² <https://tinyurl.com/OutagePredictor>.

¹³ <https://youtu.be/Y5UaESiOn9k>.

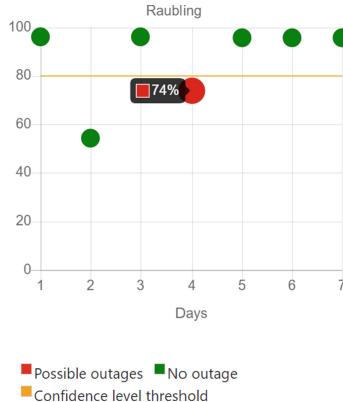


Fig. 2. Outage prediction.

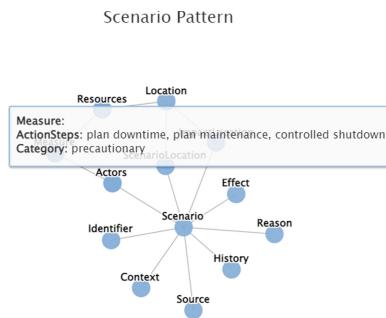


Fig. 3. Scenario pattern.

3 Conclusion

In this paper we described the development of the intuitive service POWOP for weather-based outage prediction within the energy-intensive process industry in Germany. The service allows potential decision makers to predict regional power outages for the next 7 days. As database weather information and outage data were collected in order to train a predictive analytics forecasting model. Additionally, contextual insights and action recommendations were derived from these datasets, enriched by additional user input and mapped onto semantically enhanced Scenario Patterns. The evaluation of our service resulted in values of 81.2% for accuracy and 70% sensitivity for 15 locations of paper manufacturers in Bavaria. A demonstration of the service is presented within a screencast.

References

1. Ahadu, E.: The effect of electric blackout on the operation and productivity of small manufacturing enterprises. *IJRRIS* **6**, 11–21 (2019)
2. Ahmed, I., Baddeley, M., Coffman, D., Meikle, J., Oseni, M., Sianjase, G.: The Cost of Power Outages to Zambia's Manufacturing Firms (2019)
3. Bialek, J.: What does the gb power outage on 9 august 2019 tell us about the current state of decarbonised power systems? *Energy Policy* **146**, 111821 (2020)
4. Statistisches Bundesamt. Energieerzeugung (2022). <https://www.destatis.de/DE/Themen/Branchen-Unternehmen/Energie/Erzeugung/.inhalt.html>. Accessed 28 Oct 2022
5. Cerrai, D., Wanik, D.W., Bhuiyan, M.A.E., Zhang, X., Yang, J., Frediani, M.E., Anagnostou, E.N.: Predicting storm outages through new representations of weather and vegetation. *IEEE Access* **7**, 29639–29654 (2019)
6. Chen, T., Guestrin, C.: XGBoost. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2016)
7. Cissokho, L.: The productivity cost of power outages for manufacturing small and medium enterprises in senegal. *J. Ind. Bus. Econ.* **46**(4), 499–521 (2019)
8. Janzen, S., Gdanitz, N., Abdel Khaliq, L., Munir, T., Franzius, C., Maass, W.: Anticipating energy-driven crises in process industry by ai-based scenario planning. *HICSS* (2023)
9. Kim, J., Kim, T., Lee, J., Ham, K.S.: Predicting 24-hours ahead photovoltaic power output using forecast information. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 1462–1464 (2019)
10. Lambert, L.A., Tayah, J., Lee-Schmid, C., Abdalla, M., Abdallah, I., Ali, A.H., Esmail, S., Ahmed, W.: The eu's natural gas cold war and diversification challenges. *Energ. Strat. Rev.* **43**, 100934 (2022)
11. Li, T., Eremia, M., Shahidehpour, M.: Interdependency of natural gas network and power system security. *IEEE Trans. Power Syst.* **23**(4), 1817–1824 (2008)
12. Mills, A., Ahlstrom, M., Brower, M., Ellis, A., George, R., Hoff, T., Kroposki, B., Lenox, C., Miller, N., Milligan, M., et al.: Dark shadows. *IEEE Power Energ. Mag.* **9**(3), 33–41 (2011)
13. Moghavvemi, M., Faruque, M.: Power system security and voltage collapse: a line outage based indicator for prediction. *Int. J. Electr. Power Energy Syst.* **21**(6), 455–461 (1999)
14. Netztransparenz. Sonderanalysen Winter 2022/2023 (2022). <https://www.netztransparenz.de/weitere-veroeffentlichungen/sonderanalysen-winter-2022-2023>. Accessed 28 Oct 2022
15. Orsato, R.J., Barakat, S.R., de Campos, J.G.F.: Organizational adaptation to climate change: learning to anticipate energy disruptions. *Int. J. Clim. Chang. Strat. Manag.* (2017)
16. Owerko, D., Gama, F., Ribeiro, A.: Predicting power outages using graph neural networks. In: 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 743–747. IEEE (2018)
17. Saidi, A.Y.N., Ramli, N.A., Muhammad, N., Awalin, L.J.: Power outage prediction by using logistic regression and decision tree. *J. Phys.: Conf. Ser.* **1988**, 012039 (2021). IOP Publishing
18. Tervo, R., Láng, I., Jung, A., Mäkelä, A.: Predicting power outages caused by extratropical storms. *Nat. Hazard.* **21**(2), 607–627 (2021)

19. Watson, P.L., Koukoula, M., Anagnostou, E.: Influence of the characteristics of weather information in a thunderstorm-related power outage prediction system. *Forecasting* **3**(3), 541–560 (2021)
20. Watson, P.L., Spaulding, A., Koukoula, M., Anagnostou, E.: Improved quantitative prediction of power outages caused by extreme weather events. *Weather. Clim. Extrem.* **37**, 100487 (2022)
21. Zhang, X., Shahidehpour, M., Alabdulwahab, A., Abusorrah, A.: Hourly electricity demand response in the stochastic day-ahead scheduling of coordinated electricity and natural gas networks. *IEEE Trans. Power Syst.* **31**(1), 592–601 (2015)



Toward a Human-in-the-Loop Approach to Create Training Datasets for RDF Lexicalisation

Jessica Amianto Barbato^{1(✉)}, Marco Cremaschi¹, Anisa Rula²,
and Andrea Maurino¹

¹ University of Milano-Bicocca, Milan, Italy

{JessicaAmianto.Barbato,Marco.Cremaschi,Andrea.Maurino}@unimib.it

² University of Brescia, Brescia, Italy

anisa.rula@unibs.it

Abstract. Datasets that include alignments between natural language and Knowledge Graphs are fundamental to a wide variety of Natural Language Processing and Generation tasks. Current state-of-the-art aligned datasets, though, are significantly impacted by reduced size and scarcity of covered domains, and their quality is difficult to evaluate. To compensate for these issues, we introduce **SEAllon**, a tool for extracting RDF triples from natural language textual corpora based on a human-in-the-loop approach. We present our first results of **SEAllon**'s approach, paving the way for further researches in the field of human-in-the-loop triple extraction.

Keywords: Natural language processing · Natural language generation · Human-in-the-loop · Relation extraction

1 Introduction

Handling human knowledge during the creation of high-quality resources is of crucial importance for many research tasks related to Natural Language Processing (NLP), such as Question Answering (QA), Knowledge Base Population (KBP) [10], Entity Linking (EL), Named Entity Recognition (NER) and Natural Language Generation (NLG). As a prerequisite of these tasks, the alignments between natural language and structured Knowledge Graph (KG) are fundamental for the training of Machine Learning (ML) approaches. Such alignments can be briefly described as an inter- or cross-document association between a sentence as shown in Example 1 and its representation in Resource Description Framework (RDF) triples as shown in Example 2. However, the datasets of this kind that are currently available suffer from several issues related primarily to their reduced size, the narrowness of the domain they address, and/or the quality of the results obtained [7].

Example 1. “Elliott” Smith was an American singer, songwriter, and multi-instrumentalist.

Example 2. ⟨ dbr:Elliott_Smith, dbo:occupation, dbr:Songwriter ⟩, ⟨ dbr:Elliott_Smith, dbo:occupation, dbr:Singer ⟩, ⟨ dbr:Elliott_Smith, dbo:occupation, dbr:Multi-instrumentalist ⟩.

Several works have been provided for the creation of aligned datasets, which consider two main operations: NER and Relation Extraction (RE). RE via distant and weak supervision, regardless of how inclusive is the fundamental assumption they are based on, are proven to produce rather noisy data [20, 23, 26, 27]. Even though the distant supervision assumption is largely relaxed in our approach, it would be naive to consider that our resulting data will not be noisy as well: this explains why a fruitful denoising approach is necessary in order to improve the quality of such data. Relation extraction is a challenging task for computers to complete fully automatically, mostly due to tasks like entity extraction and predicate linking. Therefore triple alignment could easily benefit from intellectual crowds [3]. Crowdsourcing has been efficiently employed to address information extraction and data labelling tasks by employing crowds' abilities in disparate domains. Moreover, the public availability of crowdworking platforms like Amazon Mechanical Turk (AMT)¹ [1, 20] and Figure Eight² (formerly known as CrowdFlower) [9, 21] has eased the process of collecting data from crowdworkers.

Following the intuitions provided in [14], there are several important issues related to crowdsourced data processing: (i) **Task design**: previous works that leverage crowdsourcing in a human-in-the-loop approach to information extraction present a worker with a single choice scenario, in which they have either to decide whether a sentence expresses a target relation [15] or pick one possible label for each portion of text in a sentence [12]; (ii) **Quality control**: spiteful workers could deliberately give the wrong answers, and improperly trained workers could not complete the annotation task successfully; (iii) **Cost control**: many examples to be annotated, even at very low retribution per worker and task, correspond to a relevant monetary effort on the part of the requester. Human-in-the-loop approaches are inherently less impacted by cost-related issues since the amount of tasks submitted to each crowdworker is a small remainder of the tasks that cannot be automatically completed by the system; (iv) **Latency control**: when the task is too difficult, not appealing, nor interesting, crowdworkers might take too much time to complete their job. It could be tempting, though, to raise the retribution in order to reduce the latency, but it might not always be the best solution [8].

In this work, we propose a human-in-the-loop approach to produce high-quality aligned RDF-text datasets. When referring to *human-in-the-loop* with respect to our approach, a process will be outlined in which manually revised input is included within an RDF annotation and triple extraction process in which both tasks are performed automatically. We present **SEAllon** (SEmantic

¹ <https://www.mturk.com/>.

² <https://appen.com/>.

ALIgnment),³ an annotation tool to assist the users. Our approach is different with respect to previous approaches that exploit human-in-the-loop and from those systems that use active learning. Indeed, in these cases, the user is mostly asked to exhibit agreement or a preference concerning an output provided by the system. Moreover, **SEAllon** is able to limit the issues due to the quality of alignments and the narrowness of the reference domain, highlighted in [7].

The main contributions of this paper are as follows: (i) A systematic analysis of the state-of-the-art of the RDF-text datasets to highlight their characteristics and creation process; (ii) The definition of a pipeline for the generation of RDF-text datasets characterised by high-quality alignments; (iii) The introduction of a crowdsourcing approach for the relation extraction process; (iv) A user-friendly Web app that allows users to validate the alignments; (v) A first empirical evaluation of the proposed workflow.

The remainder of the paper is organised as follows: Sect. 2 presents an overview of currently available aligned datasets and their significant characteristics; Sect. 3 introduces **SEAllon**'s approach by presenting the RDF extraction pipeline it is based on; Sect. 4 provides a detailed description of how the crowdsourcing module is designed; Sect. 5 discusses the assessment of **SEAllon**'s results in terms of both automatic and manual evaluation. Eventually, Sect. 6 formalises the main contributions of this work.

2 State of the Art

State-of-the-art open-domain datasets employed in data-driven NLG and other applications are T-REx [7], CC-DBP [10] and the WebNLG Dataset [4, 9, 24].

T-REx is one of the broadest and most recent among the publicly available open-domain training datasets for NLG tools; it features 6.2 million sentences from DBpedia abstracts aligned with 11 million triples from Wikidata. It has been previously employed, fully or partially, for testing Named Entity Linking (NEL) approaches, language models, unsupervised models for relation and information extraction, self-supervised frameworks for Open Relation Extraction and Knowledge Base Population systems; it has also been employed for training and testing analogy models for learning relations [13, 16, 25].

CC-DBP aligns 173 million sentences from the Common Crawl⁴ corpora with data from DBpedia, resulting in 3 million aligned triples. Common Crawl is a valuable source of open-domain textual contents, but the model that produced CC-DBP can only account for small portions of such texts. The dataset has been mostly employed to evaluate KBP systems [11] and for the training and testing steps of transfer learning systems development [2].

The **WebNLG** dataset has been employed to complete the 2017 WebNLG challenge (updated in late 2019 and re-released in 2020 for the 2020 WebNLG+ challenge), and it has been used in the development and experimental steps for

³ The version 0.1 of the tool is available at <https://sealion.ml/>. The source code can be downloaded from the Git repository https://bitbucket.org/disco_unimib/sealion/.

⁴ <https://commoncrawl.org/>.

NLG from structured data. As per the original version, the English dataset contains 13,211 entries with 35,426 lexicalisations and 372 unique RDF properties in the training set and 1,667 entries with 4,464 lexicalisations and 290 distinct RDF properties in the development set; the RDF triples in the set belong to fifteen DBpedia categories, including WrittenWorks, Artist, Athlete and Food. Note that the WebNLG dataset includes original and modified DBpedia triples, the former featuring only DBpedia predicates and the latter containing more human-understandable relations; the more recent 3.0⁵ version includes one more DBpedia category for a total of 16 topics and the syntactic shape of each sentence. It has been created by leveraging a novel method that combines a content selection module to extract varied, relevant and coherent data units from DBpedia with a crowdsourcing process for associating data units with human authored textual content to capture their meaning. The dataset is far more precise when compared to others in both entity and relation extraction, and it features both multiple lexicalisations of the same triple and multiple triples that can be aligned to one, more complex, sentence; still, it is rather restricted as for both the length and intricacy of the sentences and the domain of interest. Interesting examples of neural network models trained on the WebNLG dataset are present in literature [4], but they only apply to the restricted range of domains featured in the dataset; moreover, the structural complexity of such output texts is not comparable to human-produced natural language corpora.

In Table 1, it is possible to view three examples extracted from the datasets described above. It can be seen that there are significant differences between WebNLG, T-REx, and CC-DBP. In WebNLG, the sentences have a much simpler structure, and there is a close correspondence between the information contained in the sentence and the information expressed by the triples. In T-REx and CC-DBP, on the other hand, the triples do not express all the information contained in the sentences, which have a more complex structure.

As previously indicated, the three datasets were created to enable different tasks: T-REx for RE and KBP, CC-DBP to evaluate KBP systems and for the training and testing transfer learning systems. WebNLG for training NLG systems.

Related to NLG, many approaches have been proposed for the last twenty years and can be divided into two main categories: **pipeline** and **end-to-end** systems [5]. The former have better performances but are more complex: they generally have a complex architecture composed of several modules; this kind of architecture is prone to error propagation, however, it is less susceptible to hallucinations, which happen when the system adds information that is not present in the input (basically, when they say things that are not true) [5]. The latter, the **end-to-end** systems, have much simpler architectures but are less precise: they are more prone to hallucinations, omissions and repetitions. Nevertheless, they excel in some tasks such as discourse ordering and text structuring [5]. Recent end-to-end approaches such as T5⁶ [22] have closed the gap between the

⁵ Available at <https://gitlab.com/shimorina/webnlg-datas>.

⁶ <https://github.com/google-research/text-to-text-transfer-transformer>.

Table 1. Examples of alignments contained in the analysed datasets

Dataset	Triple	Sentences
T-REx	⟨Ljubljana, country, Slovenia⟩ ⟨Ljubljana, capital of, Slovenia⟩ ⟨Slovenia, capital, Ljubljana⟩ ⟨Ljubljana, capital of, Slovenia⟩	Jason has connections outside the classical world, being the mythical founder of the city of Ljubljana, the capital of Slovenia.
CC-DBP	dbr:The_Beatles, dbr:Tony_Sheridan ⟨dbo:associatedBand, ⟨odp:isMemberOf, ⟨dbo:associatedMusicalArtist, ⟨odp:hasMember	Ain't She Sweet was an American album featuring four tracks recorded in Hamburg in 1961 by The Beatles featuring Tony Sheridan (except for the title ...
WebNLG	⟨Aarhus, leaderName, Jacob_Bundsgaard⟩ ⟨Aarhus_Airport, city, Aarhus⟩	Aarhus airport serves the city of Aarhus whose leader is Jacob Bundsgaard.

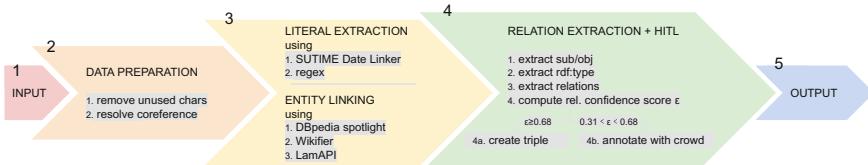
two architectures, surpassing pipelines in many tasks and showing excellent generalisation capabilities but still suffering from problems such as hallucinations, omissions and repetitions. On the other hand, pipeline systems like DualEnc [28] have demonstrated the importance of relations between the triples to create semantically correct sentences, which can be represented using graph structures such as Knowledge Graphs (KGs). This analysis suggests that, even if progress has been made in language generation approaches, these are closely related to the dataset's quality used in the training phase.

Other approaches of RE in the literature yield datasets whose quality is comparable to that of CC-DBP [10]. This entails that such datasets are unlikely to be employed in an NLG context like the one outlined here. In contrast, the goal of this work aims to achieve a quality that can be approached to that of WebNLG without relying on a major use of manual annotation.

3 Defining a Pipeline for RDF Extraction from Text

We introduce a pipeline for RDF triple extraction from text, as shown in Fig. 1, which follows in the footsteps of the process described in [7].

Note that the pipeline in [7] has been fully re-implemented to make several adjustments aimed at improving the overall functionality of the system. Key

**Fig. 1.** The SEALion pipeline described in Sect. 3

components of the pipeline are presented in the following sections. One of the most significant differences is modifying the process to consider a user intervention and adding User Interface elements to provide appropriate support to the User Experience during the annotation tasks.

3.1 Data Import and Preparation

The first step in our pipeline is **data import**; as a matter of convenience, the pipeline is fed with text (see Example 3).

Example 3. “Elliott” Smith was a singer. Elliott Smith was born in 1969. He lived in Portland, Oregon.

When a textual corpus has been uploaded, the application performs some **pre-processing** tasks including: (i) **Removal of unused characters** refers to the application of regular expressions to remove unusable portions of the text such as asides between hyphens and dashes, and phonetic transcriptions to improve the efficacy of the system; (ii) **Coreference resolution** leverages spaCy⁷ and Stanford CoreNLP⁸ through API calls to resolve implicit subjects in the text. An implicit subject is found in a sentence when the agent of the action is not directly expressed. Along with POS tagging, it is necessary to differ subject from object entities in the text.

The output of the pre-processing step is exemplified in Example 4; if compared to the same, raw input in Example 3, we notice that some portions of text (e.g., birthplace and date between parenthesis in the first sentence) have been removed and implicit subjects (“he”) have been replaced by the actual subject *Elliott Smith*, underlined in the example.

Example 4. Elliott Smith was a singer. Elliott Smith was born in 1969. Elliott Smith lived in Portland, Oregon.

3.2 Entity Resolution

The second step in SEALion’s pipeline is **entity resolution** which consists of two tasks: (i) **Entity Linking**: Wikifier’s and DBpedia Spotlight’s⁹ APIs are employed; both services are capable of providing automatic annotations of entities in the text with links to either Wikipedia concepts or DBpedia entities. Further support to the entity linking process is provided by the LamAPI¹⁰ tool, which supports fast queries for DBpedia and Wikidata using an ElasticSearch instance. A set of annotated entities for each service is produced, the union of which will constitute the pool of entities present in the text. Then, the same services are used to retrieve the classes to which the entities belong.; (ii) **Literal**

⁷ <https://spacy.io/>.

⁸ <https://stanfordnlp.github.io/CoreNLP/>.

⁹ <https://www.dbpedia-spotlight.org/api/>.

¹⁰ <https://lamapi.ml/>.

extraction: literals in text are extracted with the help of SUTime Date Linker¹¹ and regular expressions to identify geo-coordinates, URLs, e-mail addresses, IP addresses and ISBNs [6]. The result of the entity resolution is shown in Example 5.

Example 5. `dbr:Elliott_Smith` was a singer. `dbr:Elliott_Smith` was born in 1969. `dbr:Elliott_Smith` lived in `dbr:Portland,_Oregon`.

After entity resolution is completed, subject and object entities, along with literals, are identified in the text; given a document $D = (S_0, S_1, S_2, \dots, S_z)$ with $0 \leq n \leq z$ where S_n is a sentence in the document D , in input similar to the one in Example 3, we obtain a set of tuples $t_{n,m} = (sub_{n,m}, obj_{n,m})$ with $t_{n,m}$ in the sentence S_n , each representing the m -th couple of subject and object entities in the n -th sentence. For clarity, we provide an example of the output of the current module (see Example 6), which will be sent to the relation extraction step.

Example 6. Sentence: Elliott Smith was a singer. Elliott Smith was born in 1969. Elliott Smith lived in Portland, Oregon

Tuples: (`dbr:Elliott_Smith`, `dbr:singer`), (`dbr:Elliott_Smith`, `xsd:date`),
(`dbr:Elliott_Smith`, `dbr:dbr:Portland,_Oregon`)

3.3 Relation Extraction

Given a document D and a set of tuples of type $t_{n,m} = (sub_{n,m}, obj_{n,m})$, the **relation extraction** process is aimed at identifying predicates that relate those entities in the context of the sentence they belong to. Note that we assume that each sentence might be represented by more than one triple, but we also admit the case in which no triple is extracted that can express the content of the sentence. Our relation extraction process is based on three different steps: (i) **Step 1**: looks for candidate predicates employing the LamAPI tool. The result of this step is a list of the top 5 most suitable predicates; (ii) **Step 2**: identifies the correct predicate by means of a set of syntactic and semantic metrics and appropriate thresholds. (iii) **Step 3**: activates the human-in-the-loop step for ambiguous situations.

The input in **Step 1** consists of a pool of tuples of type $t_{n,m} = (sub_{n,m}, obj_{n,m})$ for each sentence $S_n \in D$. For each tuple, predicate linking is performed by leveraging LamAPI, with which it is possible through an API specification to match labels with full-text search capability and fuzzy matching. With the terms `SubjectType` and `ObjectType` we refer to the ontology class (e.g., `rdf:type`) or a datatype, so that the aforementioned triple comes to represent the presence of instances of two given classes that are linked to each other via properties `predicate`. With LamAPI, we perform a first query to retrieve the predicates $pred_{n,m}$ between $sub_{n,m}$ and $obj_{n,m}$. A second query permits to obtain an ordered set (based on frequency) of at most five predicates $pred_{n,m,j}$

¹¹ <https://stanfordnlp.github.io/CoreNLP/sutime.html>.

between $typeSub_{n,m}$ and $typeObj_{n,m}$ where n is the sentence in the document D , m is the tuple $t \in S_n$ and $0 \leq j \leq 4$ is the predicate retrieved from LamAPI.

In **Step 2** each predicate is compared to the predicate $pred_{S_n,m}$ in the sentence. To assess the similarity of each relation $pred_{n,m,j}$ to the predicate $pred_{S_n,m}$ the following similarity metrics are applied: (i) **String matching**: we look for a string matching between the RDF predicate and a portion of text. Note that we are not limiting the search to predicates since many RDF relations are made up of one or more nouns and adjectives; (ii) **Synonyms string matching**: a set of synonyms for the RDF predicate is retrieved via RelatedWords,¹² a service for retrieving word or n-grams synonyms; (iii) **Word embeddings similarity**: we use Word2Vec [18,19] to generate word embeddings, which are vectorial representations, and compute similarities. Unlike the string matching comparison, we need to limit our search to predicates in the sentence because the process would be otherwise too expensive in terms of computation, and we might identify similar words that are not representative of any relation between a couple of entities. To do so, we employ part-of-speech tags to extract a predicate from the text and compute its cosine similarity to the RDF predicate.

As for the word embedding similarities, a point has to be made; imagine that the verb in a sentence $S_n \in D$ is phrasal or that the predicate we have collected from LamAPI is multi-word. In either situation, we would not be able to compute a single vector that expresses the meaning of that relation since Word2Vec can only calculate single-word vectors. We are now presented with two different scenarios: (i) The predicate $pred_{n,m,j}$ in the triple has multiple words (e.g., dbo:associatedMusicalArtist): in this case, we create a single vector for each word in $pred_{n,m,j}$ and then we compute the average of those embeddings. We are here assuming that the meaning of the words in the predicate are similar, otherwise, the average vector would be meaningless; (ii) The predicate $pred_{S_n,m}$ in the sentence is phrasal or multi-word (e.g., “bring out”): phrasal verbs cannot be vectorialised by computing an average of its components since lexical vector representations of particles like “out” or “up” do not significantly participate in distinguishing different phrasal verbs derived from the same verb (e.g., “bring out” and “bring up”). We decided to replace each phrasal verb with a single-word verb collected by leveraging WordNet¹³ synsets and then compute its embedding; by doing so, we can calculate the cosine similarity between $pred_{S_n,m}$ and $pred_{n,m,j}$.

A single value associated to each predicate $pred_{n,m,j}$ is returned so that the likelihood $L_{pred_{n,m,j}}$ of it representing the relation expressed in the sentence can be assessed; the highest scoring predicate $\max_{pred_{n,m,j}}$ is selected for triple creation. We compute two thresholds $\varepsilon_{min}, \varepsilon_{max} \in (0, 1)$ with $\varepsilon_{min} < \varepsilon_{max}$ so that:

- if $\max_{pred_{n,m,j}} \leq \varepsilon_{min}$ the triple $T_{n,m,j}$ is discarded and no triple is returned for sentence S_n ;

¹² <https://relatedwords.org/>.

¹³ <https://wordnet.princeton.edu/>.

- if $\varepsilon_{min} < \max_{L_{pred_{n,m,j}}} < \varepsilon_{max}$ the triple $T_{n,m,j}$ is sent to selected crowd-workers for manual annotation;
- if $\max_{L_{pred_{n,m,j}}} \geq \varepsilon_{max}$ the triple $T_{n,m,j}$ is created and stored.

Because the selection of thresholds is highly dependent on the corpus chosen for Word2Vec training, we specify that in the validation process, the Brown corpus, available among the Python NLTK library corpora, was chosen to initialize the model. The thresholds have been thus empirically set to $\varepsilon_{min} = 0.425$, $\varepsilon_{max} = 0.645$. Consider the sentence and the tuples: dbr:Elliott_Smith, dbr:Bass_Guitar and dbr:Elliott_Smith, dbr:Guitar in Example 6. The sentence can be split into two verb phrases:

Example 7. Sentence: Elliott Smith’s primary instrument was the guitar

Tuple: (dbr:Elliott_Smith, dbr:Guitar) converted to (dbo:MusicalArtist, dbo:MusicalArtist)

Example 8. Sentence: Elliott Smith also played piano, clarinet, bass guitar, drums, and harmonica

Tuple: (dbr:Elliott_Smith, dbr:Bass_Guitar) converted to (dbo:MusicalArtist, dbo:MusicalArtist)

Examples 7 and 8 show the alignments between the two sentences and the tuples that represent their subject and object. We can now query LamAPI with the converted tuples; note that the tuples containing each entity’s types are equal, so the output of the query will be the same, as described in Example 9.

Example 9. Predicates: dbo:associatedMusicalArtist, dbo:associatedBand, dbo:recordLabel, dbo:instrument, dbo:influencedBy

A value $L_{pred_{n,m,j}}$ for each $pred_{n,m,j}$ returned by LamAPI is computed following the aforementioned methodology, so that we are capable of extracting the predicate dbo:instrument, with $L_{dbo:instrument} > \varepsilon_{max}$, for the sentence in Example 7. For the sentence in Example 8 we observe that $L_{dbo:instrument} \simeq L_{dbo:associatedMusicalArtist}$ with $L_{max} = \max(L_{dbo:instrument}, L_{dbo:associatedMusicalArtist})$ and $L_{max} \in (\varepsilon_{min}, \varepsilon_{max})$.

In **Step 3** the sentence in Example 8 is sent to the crowdsourcing module for human annotation. It is remarkable, though, that the ⟨ dbr:Elliott_Smith, dbo:instrument, dbr:Guitar ⟩ triple does not exist in DBpedia.

4 Crowdsourcing-Based Relation Extraction

We have already briefly depicted the introduction of crowdsourced annotations in the RDF extraction pipeline in a human-in-the-loop approach in Sect. 1. The following sections will extensively describe the process of selecting the crowd-workers and assigning them appropriate tasks; we will also focus on the answer aggregation step, which will be detailed in Sect. 4.2.

4.1 User Training

The training step of the annotation tool is necessary to direct the user to the type of annotation we require. It should be recalled that the goal of this work is to obtain a dataset that contains alignments such that triples describe only the content of the corresponding sentence and describe it in its entirety. To do this, we have the annotator perform some preliminary training, which is described as follows. Training is performed on sentences belonging to the WebNLG dataset that have been manually reviewed for correct alignments. For each of these sentences, entity and predicate annotations are collected using LamAPI tool. Short descriptions are also collected that can help the annotator choose the appropriate annotation, as in Example 10.

Example 10. Synthpop

Music genre in which the synthesizer is a key instrument

In each sentence, an entity (indifferently subject or object of the sentence) or a predicate is highlighted. The user is asked to choose the appropriate label for the highlighted portion of text from six alternatives, i.e., five possible annotations and the *None of the above* option. Each annotator must work on 15 sentences to complete the training and provide the correct answer for each before going on to the next step in SEALion. If an answer is selected that is not the correct one, the user will receive one of the following types of feedback: (i) The information conveyed by the sentence does not match with that of the answer you selected; (ii) The concept expressed by the entity in the sentence is more general than the one in the option you selected; (iii) The concept expressed by the option you selected is more general than the one in the entity in the sentence; (iv) There is at least one correct answer provided among the remaining options—when the *None of the above* option is improperly selected.

Such feedback is intended to draw the annotator to generate triples that meet the requirements of completeness and correctness of the concept expressed by the triple w.r.t. the sentence.

4.2 Task Design and Quality Control

In our pipeline, we model micro-tasks, assigned to crowdworkers as single choice questions with at most six options to choose among, that is, the predicates returned by LamAPI described in Sect. 3.2 and a *None* option if no predicate represents the relation.

Unlike the work described in Li et al. [15], which presents the user with a two-option question about whether a sentence expresses a target relation or not, we opted for a wider range of choices; note that when the worker is given multiple options to consider, as in Grosman et al. [12], relation constraints are generally imposed to avoid impossible or meaningless triples. Such constraints become less necessary in our case since LamAPI generally returns predicates capable of expressing a relation between two given entity classes and the crowdsourcing component wanes the chances of very unlikely relations. In a human-in-the-loop

framework like ours, some major questions have to be answered before proceeding to the actual annotation process thus we need to define: (i) **Who are the annotators**: similar studies that involve a human-in-the-loop approach do not detail the workers selection process; usually a small sample of skilled annotators that are already familiar with the domain they are asked to deal with [12,15]. Picking the right intellectual crowd is fundamental to avoiding poor performances and low-quality data at the end of the process, but how can we assess a worker’s quality? Modelling a worker requires choosing an accurate strategy to evaluate a worker’s performance without relying on self-reports (e.g., questionnaires, interviews), which cannot reflect the actual expertise of the respondent. Instead, a worker probability model, detailed in Sect. 4.2, appears to be much more satisfactory for our purpose; (ii) **Whom to assign a task**: when a task and a group of workers whose qualities are known are defined, we have to identify a subset of our crowd to whom we want to assign that task. Such a problem is commonly known as the Jury Selection Problem [3] and is detailed in Sect. 4.2; (iii) **How to handle contrasting annotations**: answer aggregation is a key step in collecting responses from crowdworkers. Also known as voting strategy, it briefly consists of assigning the same task to multiple workers and aggregate their results; a Weighted Majority Voting strategy is considered in this work, considering each worker’s quality and their answer to the task. Our inter-annotator metric will be discussed in Sect. 4.2.

Moreover, to improve the quality of the annotation tasks, we have designed a User Interface that serves to disguise some of the complexity that derives from the tools and processing steps we have employed.

Who are the Annotators When relying on the wisdom of crowds to derive an answer to our tasks, we assume that, even though some diverging participant responses may exist, the overall judgement is still well-grounded. Different background knowledge, expertise or understanding of the task typically result in varying inter-annotator answers that need to be aggregated to resolve the uncertainty. Following the works in [3,29] we model an Individual Error Rate $q_i \in [0, 1]$, corresponding to the probability that the i th participant will give a contrasting answer with regards to the correct task response.

Definition 1. The Individual Error Rate q_i of the i th participant corresponds to the probability that a worker provides a wrong answer

$$q_i = Pr(\text{the worker's vote} — \text{the task's correct answer } A) \quad (1)$$

A ground truth A , which is a binary value 0 (false) or 1 (true), is therefore required to assess the participants’ quality; in detail, we employ a subset of the WebNLG dataset described in Sect. 2, for which RDF triples-text alignments are already known, as a Gold Standard and randomly inject it into real tasks. We will only pick simple sentences in order to minimise the chance of misinterpretation. Based on the participant’s answers to such golden tasks, we can compute their Individual Error Rate; participants whose $IER \geq 0.9$ are directly discarded.

Unlike training tests [12], submitted before the actual task, the worker is not aware that an evaluation is being performed, thus reducing the chances of biased responses.

Whom to Assign a Task Once the Individual Error Rate for each participant is computed, we need to identify a subgroup of workers capable of showing the best performance throughout the tasks they are presented with. Such a problem is known as the Jury Selection Problem, presented in [3] as:

Definition 2. Given a candidate worker set U with size K , a budget $B \geq 0$, a crowdsourcing model M , the Jury Selection Problem (JSP) consists in selecting a subset $J_y \subseteq U$ with size $1 \leq y \leq K$, so that J_y is allowed according to M and the Jury Error Rate $JER(J_y)$ is minimised.

In SEAllon's pipeline, we assume that there exist a set of workers who, for any altruistic reason (e.g., students and researchers), are capable of performing high-quality annotations without further financial incentives, so that any group of candidates is allowed in J_y , but other models can be considered [3]. Following Definition 2, we compute the Jury Error Rate $JER(J_y)$ of a subset of workers J_y by computing the probability, given J_y , that the number of mistaken workers C , where $0 \leq C \leq y$, is a minority in J_y

$$JER(J_y) = Pr(C \geq \frac{y+1}{2} | J_y) \quad (2)$$

How to Handle Contrasting Annotations When the selected subset of our crowdworkers has completed their tasks, their answers have to be aggregated. The clearest way of drawing a single decision from disparate opinions is Majority Voting [3, 15], which outputs the answer supported by at least half of the participants in our selected subset plus one. Given a group of workers J_y , to compute Majority Voting, we need y , which is the size of our subset, to be odd.

Definition 3. Given a voting V_y , a set of binary values representing the response of a crowd J_y with size y , the Majority Voting strategy is defined as follows:

$$MV(J_y) = \begin{cases} 0 & \text{if } \sum j_i \geq \frac{y+1}{2} \\ 1 & \text{if } \sum j_i \leq \frac{y-1}{2} \end{cases} \quad (3)$$

We adopt a more sophisticated metric, the Weighted Majority Voting [17], which also takes into account the Individual Error Rate q_i of each participant in the crowd J_y , computed as in Definition 1, so that more importance is given to votes expressed by highly-reliable workers (i.e. those with low Individual Error Rate).

5 Validation

SEALLon's results have been assessed following similar applications validation processes [4] by leveraging both automatic and manual evaluation. A brief description of common metrics has been provided in the definitions in Sect. 5.1. When addressing the issue of false negatives, we must consider that, as represented in Table 4, our Gold Standard aligns sentences to triples that are not adequately representative of their content and mainly express information inferred from the sentences themselves. The **SEALLon** process purposely considers these triples to be erroneous because they are not directly related to the meaning of the sentence.

We validate our pipeline on a subset of sentences—disjoint from the Gold Standard described in Sect. 4.2—from the WebNLG+ test set and the T-REx dataset. The sentences have been randomly selected to avoid biased annotations towards a specific domain. The test set counts 120 sentences, equally parted between the two datasets, aligned with 280 triples (180 from the WebNLG dataset and 100 from T-REx) to be used as a reference. The quantity of the sentences was defined in accordance with what has been done in other works in the state of the art [7]. After the automatic annotation, performed as described in Sect. 3.3, we collected 130 triples and sent another 220 predicted triples belonging to 100 sentences to crowdworkers for manual annotation. Each sentence was reviewed at least 45 times by a pool of graduate and undergraduate students in Computer Science related studies and researchers who were given two days to complete the task, including 15 real sentences and 5 Gold Standard sentences to assess Participant Quality. Table 2 summarises statistics about **SEALLon** performance regarding the WebNLG and T-REx datasets samples; note that **SEALLon** is capable of extracting a total of 350 triples aligned with 120 sentences, whereas the reference sets only included 280 triples.

5.1 Automatic Evaluation

We calculate Precision P , defined as the number of correct relation annotations over the total number of positive relation annotations, by comparing the results of our pipeline with reference datasets; note that, since our approach has resulted in a larger number of alignments than those in our samples, we need to assume that: (i) Triples that include the same subject and object entities of the reference triple, related by equal or equivalent (e.g., *location* and *cityLocation*) predicates, and are classified as correct by **SEALLon** are to be considered true positive candidates; (ii) Triples that include the same subject and object entities of the reference triple, related by different predicates (e.g., *knownFor* in **SEALLon** alignments and *notableWorks* in reference alignments), and are classified as correct by **SEALLon** are to be considered false-positive candidates; (iii) Triples present in **SEALLon** alignments but absent from reference datasets are to be omitted at this stage.

The results of the automatic evaluation are presented in Table 3; precision is calculated here before the triple is submitted to the crowdsourcing annota-

tion module and after the sentence has been re-annotated in order to assess the improvement made by using a human-in-the-loop approach. **SEALLon** performs best on WebNLG sentences even without human annotation, while it does not accomplish high precision on the T-REx dataset. Another precision value has been computed with regards to the system’s prediction before human annotations are performed; the automatic relation extraction module highest-scoring predicate (see Sect. 3.3 for a detailed description of the process) has been considered as a prediction of the correct predicate between a pair of entities. Such prediction has been then compared with the crowdsourcing module’s outcome, which could confirm or modify the system’s prediction. After manual review, 80% of triples are confirmed, reaching an average precision of 0.88. Although an increase in precision resulting from manual annotation is to be expected, it should be emphasised that a minor human effort is sufficient to achieve a significant improvement in the quality of the final output.

The results presented in Table 3 will be better discussed in Sect. 5.2.

Table 2. Statistics of **SEALLon**’s results for each dataset’s test sample

	WebNLG	T-REx	Total
Total extracted triples	140	210	350
Triples to be reviewed by crowdworkers	50	170	220
Triples automatically accepted	90	40	130
Average triples per sentence	2	4	3

Table 3. Precision Calculated with Regards to Reference Triples in WebNLG and T-REx

	WebNLG sample		T-REx sample	
	No crowd annotations	With crowd annotations	No crowd annotations	With crowd annotations
Precision P	0.84	0.92	0.40	0.80

5.2 Manual Evaluation

A sample of **SEALLon**’s output (50 sentences, 142 triples) has been collected and manually checked for correct relations by a small group of domain experts, each of which was asked to accept an alignment as correct if and only if the triple was explicitly mentioned in the sentence. Note that, as stated in the previous section, **SEALLon** was capable of extracting and aligning more triples than those present in reference datasets, especially when compared to T-REx. An example of alignments proposed to manual reviewers is presented in Table 4. Such triple-sentence couples’ quality could not be assessed with automatic evaluation since

Table 4. Triple-text alignments in **SEALLon** compared to the same alignments in reference datasets' samples

The population of the metropolitan area of Ciudad Ayala, a part of Morelos, Mexico, located at 1147 above sea level, is 1 777 539, and the UTC offset for this Pacific Daylight Time zone area is -6.

SEALLon	WebNLG
Ciudad Ayala — part — Morelos	C.A. — populationMetro — 1777539
C.A. — populationMetro — 1777539	Ciudad Ayala — utcOffset — -6
Ciudad Ayala — utcOffset — -6	Ciudad Ayala — isPartOf — Morelos
C.A. — minimumElevation — 1147	Ciudad Ayala — country — Mexico
	C.A. — elevationAboveTheSeaLevel — 1147
	C.A. — timeZone — PacificDaylightTime

Saul Swimmer was an American documentary film director and producer best known for the movie The Concert for Bangladesh (1972).

SEALLon	T-REx
The Concert for Bangladesh (film) — director — Saul Swimmer	Saul S. — countryOfCitizenship — American
Saul S. — genre — documentary film	Bangladesh — diplomaticRelation — American

no reference triples could be found in WebNLG and T-REx. Consider the first sentence in Table 4: **SEALLon** was capable of correctly extracting four triples out of six references, two of which, $\langle \text{CiudadAlaya}, \text{timeZone}, \text{PacificDaylightTime} \rangle$ and $\langle \text{CiudadAlaya}, \text{country}, \text{Mexico} \rangle$ are not directly mentioned in the text and are rather inferred from its content. A similar case is represented by the second sentence, where the $\langle \text{Bangladesh}, \text{diplomaticRelation}, \text{American} \rangle$ triple is not only presumably inferred from the text, but it is also incapable of accounting for the informational content of the sentence. According to manual reviewers, **SEALLon** results in an accuracy of 93% in aligning triples and sentences so that the former are relevant to the informational content of the latter.

6 Conclusions and Future Research

In this work, a pipeline has been defined, including an automatic pre-processing and entity recognition module and a crowdsourcing component aimed at collecting a small amount of manually annotated data to be reinserted in the process to reduce the error rate in the extraction of RDF triples. We have then focused on the most critical issues that arise when a crowd of human respondents is considered, from selecting a proper set of annotators to the definition of a metric for computing inter-annotator agreement. **SEALLon** proves to be a useful tool to perform RDF extraction from natural language and unstructured text. However, more testing needs to be performed to assess its accomplishments outside of controlled textual sources. As depicted, the main goal of the **SEALLon** solution

is the definition of a methodology for RDF-text alignment. Therefore it is possible to use this methodology to create new datasets that meet the user's needs (e.g., specific domain dataset). We purportedly decided to focus on evaluating a restricted number of alignments, concentrating on testing the improvements provided by the human-in-the-loop approach compared to fully automatic triple extraction pipelines. Still, an analysis of how many epochs are necessary for a single textual source before manual annotation is no longer required and what quality is achieved than has to be carried out. Moreover, further research should be conducted on the interaction between the user and the system in both face-to-face and remote annotation settings to supply sufficient support for high-quality relation extraction in either environment. Of course, different paths could be explored for the techniques and formulas we have presented in this work. It would be interesting to compare state-of-the-art methodologies in a human-in-the-loop approach like ours. Also, long-time testing to verify how much previous manual annotations can impact future automatically extracted triples is foreseeable.

Acronyms

AMT	Amazon Mechanical Turk
EL	Entity Linking
KB	Knowledge Base
KBP	Knowledge Base Population
KG	Knowledge Graph
KGs	Knowledge Graphs
ML	Machine Learning
NEL	Named Entity Linking
NER	Named Entity Recognition
NLG	Natural Language Generation
NLP	Natural Language Processing
QA	Question Answering
RDF	Resource Description Framework
RE	Relation Extraction

References

1. Angeli, G., Tibshirani, J., Wu, J., Manning, C.D.: Combining distant and partial supervision for relation extraction. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1556–1567. Association for Computational Linguistics, Doha, Qatar (2014)

2. Bhattacharjee, B., Kender, J.R., Hill, M., Dube, P., Huo, S., Glass, M.R., Belgodere, B., Pankanti, S., Codella, N., Watson, P.: P2l: Predicting transfer learning for images and semantic relations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 760–761 (2020)
3. Cao, C.C., She, J., Tong, Y., Chen, L.: Whom to ask? jury selection for decision making tasks on micro-blog services. *Proc. VLDB Endow.* **5**(11), 1495–1506 (2012)
4. Ferreira, T.C., Gardent, C., Ilinykh, N., van der Lee, C., Mille, S., Moussallem, D., Shimorina, A.: The 2020 bilingual, bi-directional WebNLG+ shared task: overview and evaluation results (WebNLG+ 2020). In: Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+), pp. 55–76. Association for Computational Linguistics, Dublin, Ireland (Virtual) (2020)
5. Ferreira, T.C., van der Lee, C., van Miltenburg, E., Krahmer, E.: Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. In: Proceedings of the EMNLP-IJCNLP, pp. 552–562. Association for Computational Linguistics, Hong Kong, China (2019)
6. Cremaschi, M., De Paoli, F., Rula, A., Spahiu, B.: A fully automated approach to a complete semantic table interpretation. *Futur. Gener. Comput. Syst.* **112**, 478–500 (2020)
7. Elsaifar, H., Vougiouklis, P., Remaci, A., Gravier, C., Hare, J., Laforest, F., Simperl, E.: T-REx: A large scale alignment of natural language with knowledge base triples. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), pp. 3448–3452. European Language Resources Association (ELRA), Miyazaki, Japan (2018)
8. Faridani, S., Hartmann, B., Ipeirotis, P.G.: What's the right price? pricing tasks for finishing on time. In: Proceedings of the 11th AAAI Conference on Human Computation, AAAIWS'11-11, pp. 26–31. AAAI Press (2011)
9. Gardent, C., Shimorina, A., Narayan, S., Perez-Beltrachini, L.: Creating training corpora for NLG micro-planners. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 179–188. Association for Computational Linguistics, Vancouver, Canada (2017)
10. Glass, M., Gliozzo, A.: A dataset for web-scale knowledge base population. In: The Semantic Web, pp. 256–271. Springer International Publishing, Cham (2018)
11. Glass, M., Gliozzo, A., Hassanzadeh, O., Mihindukulasooriya, N., Rossiello, G.: Inducing implicit relations from text using distantly supervised deep nets. In: The Semantic Web—ISWC 2018, pp. 38–55. Springer International Publishing, Cham (2018)
12. Grosman, J.S., Furtado, P.H.T., Rodrigues, A.M.B., Schardong, G.G., Barbosa, S.D.J., Lopes, H.C.V.: Eras: improving the quality control in the annotation process for natural language processing tasks. *Inf. Syst.* **93**, 101553 (2020)
13. Hu, X., Wen, L., Xu, Y., Zhang, C., Yu, P.: SelfORE: Self-supervised relational feature learning for open relation extraction. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 673–3682. Association for Computational Linguistics (2020)
14. Li, G., Wang, J., Zheng, Y., Franklin, M.J.: Crowdsourced data management: A survey. *IEEE Trans. Knowl. Data Eng.* **28**(9), 2296–2319 (2016)
15. Li, M., Jin, J., Wu, W., Yang, Y., He, L., Yang, J.: A crowdsourcing based human-in-the-loop framework for denoising uus in relation extraction tasks. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2019)

16. Lin, X., Li, H., Xin, H., Li, Z., Chen, L.: Kbpearl: a knowledge base population system supported by joint entity and relation linking. *Proc. VLDB Endow.* **13**(7), 1035–1049 (2020)
17. Littlestone, N., Warmuth, M.K.: The weighted majority algorithm. *Inf. Comput.* **108**(2), 212–261 (1994)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013). [arXiv:1301.3781](https://arxiv.org/abs/1301.3781)
19. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13*, pp. 3111–3119. Curran Associates Inc. (2013)
20. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 1003–1011. Association for Computational Linguistics, Suntec, Singapore (2009)
21. Mrabet, Y., Vougiouklis, P., Kilicoglu, H., Gardent, C., Demner-Fushman, D., Hare, J., Simperl, E.: Aligning texts and knowledge bases with semantic sentence simplification. In: *Proceedings of the 2nd International Workshop on Natural Language Generation and the Semantic Web (WebNLG 2016)*, pp. 29–36. Association for Computational Linguistics, Edinburgh, Scotland (2016)
22. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR* (2019)
23. Riedel, S., Yao, L., McCallum, A., Marlin, B.M.: Relation extraction with matrix factorization and universal schemas. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 74–84. Association for Computational Linguistics, Atlanta, Georgia (2013)
24. Shimorina, A., Khasanova, E., Gardent, C.: Creating a corpus for Russian data-to-text generation using neural machine translation and post-editing. In: *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pp. 44–49. Association for Computational Linguistics, Florence, Italy (2019)
25. Simon, E., Guigue, V., Piwowarski, B.: Unsupervised information extraction: Regularizing discriminative approaches with relation distribution losses. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1378–1387. Association for Computational Linguistics, Florence, Italy (2019)
26. Smirnova A., Cudré-Mauroux, P.: Relation extraction using distant supervision: A survey. *ACM Comput. Surv.* **51**(5) (2018)
27. Yao, L., Riedel, S., McCallum, A.: Collective cross-document relation extraction without labelled data. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 1013–1023. Association for Computational Linguistics, Cambridge, MA (2010)
28. Zhao, C., Walker, M., Chaturvedi, S.: Bridging the structural gap between encoding and decoding for data-to-text generation. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2481–2491. Association for Computational Linguistics (2020)
29. Zheng, Y., Cheng, R., Maniu, S., Mo, L.: On optimality of jury selection in crowdsourcing. In: *Proceedings of the 18th International Conference on Extending Database Technology, EDBT 2015*, pp. 193–204 (2015)



Fuzzy Time Series Forecasting on the Example of the Dow Jones Index Dynamics

Ramin Rzayev^{1(✉)}, Parvin Alizada², and Tahir Mehdiyev¹

¹ Institute of Control Systems of ANAS, Vahabzadeh Str. 9, AZ1141 Baku, Azerbaijan
raminrza@yahoo.com

² Baku State University, Z. Khalilov Str. 23, AZ1148 Baku, Azerbaijan

Abstract. The paper discusses a new predictive model of a fuzzy volatile time series, in the framework of which a new approach to the fuzzification of historical data is proposed as the results of observations based on “soft measurements” of the states of a dynamic system over a certain period of time. As an example, the Dow Jones index was chosen, the readings of which are set based on the results of daily trading on the US stock exchange by the usual arithmetic averaging of contextual indicators. This allows to consider the daily readings of the Dow Jones index as weakly structured, and to interpret the dynamics of its change as a fuzzy time series. The fuzzification procedure is implemented by the fuzzy inference system that provides the values of the membership functions of the corresponding fuzzy subsets of the discrete universe covering the set of index indicators for the period from June 15, 2018 to October 10, 2019. The proposed predictive model is based on the identified internal relationships, designed as first-order fuzzy relations between evaluation criteria or fuzzy sets that describe weakly structured Dow Jones indexes. At the end of the study, the proposed model is evaluated for adequacy using the statistical criteria MAPE, MPE and MSE.

Keywords: Dow Jones Industrial Average · Fuzzy time series · Fuzzy Inference

1 Introduction

Over the past two decades, numerous publications have been devoted to the study of fuzzy time series, among which are the works of Song and Chissom [1], Kumar et al. [2], Chen [3], Cheng et al. [4], Poulsen [5]. The approaches existing in these works differ in the rules of fuzzification of historical data and defuzzification of fuzzy predicts as outputs of fuzzy models. The reliability of the final fuzzy predicts depends on how these rules adequately reflect weakly structured time series data by appropriate fuzzy sets and, accordingly, interpret fuzzy predicts in a traditional numerical manner. In [6], using the Dow Jones Industrial Average (DJIA) as an example, we proposed methods for fuzzifying historical data using a fuzzy inference system. However, the tasks of forming qualitative criteria for evaluating the indicators of the DJIA and the fuzzy model of the corresponding time series remained unresolved. Therefore, on the basis of the same example, the given paper proposes a new approach to the formation of a sufficient set of criteria for evaluating historical data and to building on this basis a predictive model of a fuzzy time series.

2 Problem Definition

The object of the study is the DJIA time series, covering the set of index indicators for the period from June 15, 2018 to October 10, 2019 (see Fig. 1) [7]. The DJIA index is established based on the results of daily trading on the US stock exchange by the usual arithmetic averaging of its constituent indicators. Therefore, each value of the DJIA $x(t)$ at time t can be considered as a weakly structured historical data, which can be interpreted by an appropriate fuzzy set A_j ($j = 1 \div J$) characterized by following tuple:

$$\{x(t) / \mu_{A_j}[x(t)]\}, \mu_{A_j}[x(t)] @ [0, 1].$$

It is necessary to develop a method for the fuzzification of the historical data of the DJIA, which would allow to adequately restore the time series in terms of fuzzy sets and, thereby, increase the reliability of its forecasting by appropriate fuzzy model.

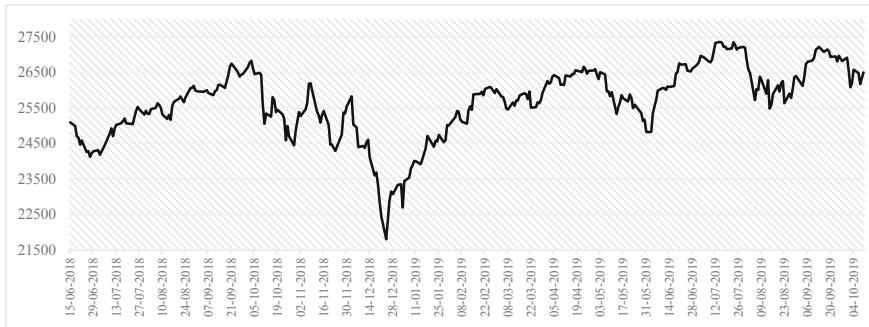


Fig. 1. DJIA time series.

3 Fuzzification of DJIA Time Series Data

To fuzzify DJIA indexes the segment $D = [D_{\min} - D_1, D_{\max} + D_2]$ is chosen as the universe, where D_{\min} and D_{\max} are the minimum and maximum values of the DJIA, respectively; $D_1 > 0$ and $D_2 > 0$ are selected based on the division of the segment D into equal intervals u_j according to the number of qualitative evaluation criteria. Suppose that such criteria are 8 terms of the linguistic variable “DJIA index value”, which are represented as the following fuzzy subsets of the discrete universe $U = \{u_1, u_2, \dots, u_8\}$:

- TOO LOW: $C_1 = \mu_{C1}(u_1)/u_1 + \mu_{C1}(u_2)/u_2 + \dots + \mu_{C1}(u_8)/u_8$,
- VERY LOW: $C_2 = \mu_{C2}(u_1)/u_1 + \mu_{C2}(u_2)/u_2 + \dots + \mu_{C2}(u_8)/u_8$,
- MORE THAN LOW: $C_3 = \mu_{C3}(u_1)/u_1 + \mu_{C3}(u_2)/u_2 + \dots + \mu_{C3}(u_8)/u_8$,
- LOW: $C_4 = \mu_{C4}(u_1)/u_1 + \mu_{C4}(u_2)/u_2 + \dots + \mu_{C4}(u_8)/u_8$,
- HIGH: $C_5 = \mu_{C5}(u_1)/u_1 + \mu_{C5}(u_2)/u_2 + \dots + \mu_{C5}(u_8)/u_8$,
- MORE THAN HIGH: $C_6 = \mu_{C6}(u_1)/u_1 + \mu_{C6}(u_2)/u_2 + \dots + \mu_{C6}(u_8)/u_8$,
- VERY HIGH: $C_7 = \mu_{C7}(u_1)/u_1 + \mu_{C7}(u_2)/u_2 + \dots + \mu_{C7}(u_8)/u_8$,
- TOO HIGH: $C_8 = \mu_{C8}(u_1)/u_1 + \mu_{C8}(u_2)/u_2 + \dots + \mu_{C8}(u_8)/u_8$,

Where $\mu_{C_i}(u_j) \in [0, 1]$ ($i, j = 1 \div 8$) are the values of the membership function of the interval u_j to the fuzzy set C_i . In other words, the index DJIA belongs to the interval u_j is determined by a qualitative criterion described by the fuzzy set C_i .

In the considered time series, which includes 333 historical data of the DJIA, $D_{\min} = 21792.2$ and $D_{\max} = 27359.2$. Choosing $D_1 = 21.2$ and $D_2 = 11.8$, a coverage in the form of the segment $U = [21771, 27371]$ is obtained. According to the chosen number of quality evaluation criteria C_i , this segment is divided into eight equal intervals of 700 units long: $u_1 = [21771, 22471]$, $u_2 = [22471, 23171]$, $u_3 = [23171, 23871]$, $u_4 = [23871, 24571]$, $u_5 = [24571, 25271]$, $u_6 = [25271, 25971]$, $u_7 = [25971, 26671]$, $u_8 = [26671, 27371]$. In this case, historical data is interpreted as fuzzy set C_j , taking into account that the interval of its localization u_j ($j = 1 \div 8$) belongs to C_j . To do this, we choose the following consistent and rather trivial statements as a basis:

e_1 : "If the DJIA index is located closer to the middle of the segment u_1 , then its value is too low";

e_2 : "If the DJIA index is located closer to the middle of the segment u_2 , then its value is very low";

e_3 : "If the DJIA index is located closer to the middle of the segment u_3 , then its value is more than low";

e_4 : "If the DJIA index is located closer to the middle of the segment u_4 , then its value is low";

e_5 : "If the DJIA index is located closer to the middle of the segment u_5 , then its value is high";

e_6 : "If the DJIA index is located closer to the middle of the segment u_6 , then its value is more than high";

e_7 : "If the DJIA index is located closer to the middle of the segment u_7 , then its value is very high";

e_8 : "If the DJIA index is located closer to the middle of the segment u_8 , then its value is too high".

The analysis of these information fragments makes it possible to identify one input characteristic in the form of a linguistic variable $x = "Data\ localization"$, the values of which are the terms: "CLOSER TO THE MIDDLE OF THE SEGMENT u_k " ($k = 1 \div 8$), and one output linguistic variable $y = "DJIA\ value"$ with terms: TOO LOW, VERY LOW, MORE THAN LOW, LOW, HIGH, MORE THAN HIGH, VERY HIGH, TOO HIGH.

The verbal assessment of the localization of the $x(t)$ on the basis of belonging to one or another segment u_j ($j = 1 \div 8$) is reflected as a fuzzy subset of the universe, consisting of all DJIA index: $U = \{x(t)\}_{t=1}^{333}$. As a membership function, the following function of the Gaussian type is chosen:

$$\mu(x) = \exp[-(x_t - u_{j0})^2 / \sigma^2] \quad (1)$$

where $x_t = x(t)$ is the DJIA index obtained as a result of the completion of trading on the stock exchange for the t -th day; u_{j0} is the middle of the interval u_j ($j = 1 \div 8$); σ^2 is a variation, which is chosen as the same for all cases, as 2500000 (see Fig. 2).

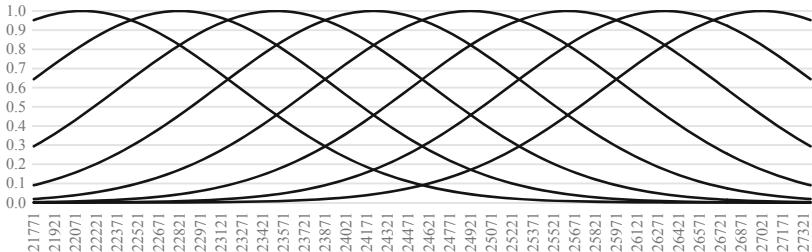


Fig. 2. Gaussian membership functions of fuzzy sets reflecting the degree of DJIA localization.

Noting that the midpoints of the segments u_j are the corresponding numbers: $u_{10} = 22121$, $u_{20} = 22821$, $u_{30} = 23521$, $u_{40} = 24221$, $u_{50} = 24921$, $u_{60} = 25621$, $u_{70} = 26321$, $u_{80} = 27021$, according to (1) features of $x(t)$ localization are interpreted as follows:

- “CLOSE TO 22121” as fuzzy set [8]:

$$X_1 = 0.952181/x_1 + 0.958630/x_2 + 0.964640/x_3 + \dots + 0.000793/x_{332} + 0.000472/x_{333};$$

- “CLOSE TO 22821” as fuzzy set:

$$X_2 = 0.643393/x_1 + 0.656883/x_2 + 0.670320/x_3 + \dots + 0.006941/x_{332} + 0.004497/x_{333};$$

- “CLOSE TO 23521” as fuzzy set:

$$X_3 = 0.293758/x_1 + 0.304145/x_2 + 0.314743/x_3 + \dots + 0.041079/x_{332} + 0.028958/x_{333};$$

- “CLOSE TO 24221” as fuzzy set:

$$X_4 = 0.090627/x_1 + 0.095155/x_2 + 0.099859/x_3 + \dots + 0.164269/x_{332} + 0.125994/x_{333};$$

- “CLOSE TO 24921” as fuzzy set:

$$X_5 = 0.018892/x_1 + 0.020116/x_2 + 0.021408/x_3 + \dots + 0.443858/x_{332} + 0.370415/x_{333};$$

- “CLOSE TO 25621” as fuzzy set:

$$X_6 = 0.002661/x_1 + 0.002873/x_2 + 0.003101/x_3 + \dots + 0.810382/x_{332} + 0.735842/x_{333};$$

- “CLOSE TO 26321” as fuzzy set:

$$X_7 = 0.000253/x_1 + 0.000277/x_2 + 0.000304/x_3 + \dots + 0.999750/x_{332} + 0.987728/x_{333};$$

- “CLOSE TO 27021” as fuzzy set:

$$X_8 = 0.000016/x_1 + 0.000018/x_2 + 0.000020/x_3 + \dots + 0.833393/x_{332} + 0.895873/x_{333}.$$

The terms of the output linguistic variable y is described by fuzzy subsets of the discrete universe $I = \{0, 0.1, 0.2, \dots; 1\}$. So, $\forall i \in I$ we have follows [8]:

- $TL = \text{TOO LOW}$, $\mu_{TL}(i) = \begin{cases} 0, & i = 1, \\ 1, & i < 1; \end{cases}$
 - $VL = \text{VERY LOW}$: $\mu_{VL}(i) = (1-i)^2$;
 - $ML = \text{MORE THAN LOW}$: $\mu_{ML}(i) = \sqrt{1-i}$;
 - $L = \text{LOW}$: $\mu_L(i) = 1-i$;
 - $H = \text{HIGH}$: $\mu_H(i) = i$;
 - $MH = \text{MORE THAN HIGH}$: $\mu_{MH}(i) = \sqrt{i}$;
 - $VH = \text{VERY HIGH}$: $\mu_{VH}(i) = i^2$;
 - $TH = \text{TOO HIGH}$, $\mu_{TH}(i) = \begin{cases} 1, & i = 1, \\ 0, & i < 1. \end{cases}$

Taking into account the introduced formalisms, the above reasoning is transformed into a fuzzy inference system, which in symbolic form looks like the following:

$$\begin{aligned} e_1 &: (x = X_1)(y = TL); e_2 : (x = X_2)(y = VL); e_3 : (x = X_3)(y = ML); \\ e_4 &: (x = X_4,)(y = L); e_5 : (x = X_5)(y = H); e_6 : (x = X_6)(y = MH); \\ e_7 &: (x = X_7)(y = VH \gg); e_8 : (x = X_8)(y = TH). \end{aligned}$$

After transforming these rules using, for example, Lukasiewicz's fuzzy implication: $\mu_W(u, i) = \min\{1, 1-\mu_X(u) + \mu_Y(i)\}$, for each pair $(u, i) \in X \times Y$ on $X \times Y$ fuzzy relations are formed in the type of the following matrices [8]:

$R_3 =$	$\begin{bmatrix} 1 & 0.9487 & 0.8944 & 0.8367 & 0.7746 & 0.7071 & 0.6325 & 0.5477 & 0.4472 & 0.3162 & 0 \\ 0.3733 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 0.9429 & 0.6267 \\ 0.4231 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 0.8932 & 0.5769 \\ 0.5734 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 0.9743 & 0.8738 & 0.7428 & 0.4266 \\ \vdots & \vdots \\ 0.0008 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 0.9589 \\ 0.0005 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 0.9710 \end{bmatrix}$
$R_4 =$	$\begin{bmatrix} 1 & 0.9 & 0.8 & 0.7 & 0.6 & 0.5 & 0.4 & 0.3 & 0.2 & 0.1 & 0 \\ 0.7390 & 1.0000 & 1.0000 & 1.0000 & 0.9610 & 0.8610 & 0.7610 & 0.6610 & 0.5610 & 0.4610 & 0.3610 & 0.2610 \\ 0.7906 & 1.0000 & 1.0000 & 1.0000 & 0.9094 & 0.8094 & 0.7094 & 0.6094 & 0.5094 & 0.4094 & 0.3094 & 0.2094 \\ 0.9122 & 1.0000 & 0.9878 & 0.8878 & 0.7878 & 0.6878 & 0.5878 & 0.4878 & 0.3878 & 0.2878 & 0.1878 & 0.0878 \\ \vdots & \vdots \\ 0.1643 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 0.9357 & 0.8357 \\ 0.1260 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 0.9740 & 0.8740 \end{bmatrix}$
$R_5 =$	$\begin{bmatrix} 0 & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 & 1 \\ 0.9886 & 0.0114 & 0.1114 & 0.2114 & 0.3114 & 0.4114 & 0.5114 & 0.6114 & 0.7114 & 0.8114 & 0.9114 & 1.0000 \\ 0.9982 & 0.0018 & 0.1018 & 0.2018 & 0.3018 & 0.4018 & 0.5018 & 0.6018 & 0.7018 & 0.8018 & 0.9018 & 1.0000 \\ 0.9807 & 0.0193 & 0.1193 & 0.2193 & 0.3193 & 0.4193 & 0.5193 & 0.6193 & 0.7193 & 0.8193 & 0.9193 & 1.0000 \\ \vdots & \vdots \\ 0.4439 & 0.5561 & 0.6561 & 0.7561 & 0.8561 & 0.9561 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ 0.3704 & 0.6296 & 0.7296 & 0.8296 & 0.9296 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \end{bmatrix}$
$R_6 =$	$\begin{bmatrix} 0 & 0.3162 & 0.4472 & 0.5477 & 0.6325 & 0.7071 & 0.7746 & 0.8367 & 0.8944 & 0.9487 & 1 \\ 0.8935 & 0.1065 & 0.4227 & 0.5337 & 0.6542 & 0.7389 & 0.8136 & 0.8811 & 0.9431 & 1.0000 & 1.0000 & 1.0000 \\ 0.8517 & 0.1483 & 0.4645 & 0.5955 & 0.6960 & 0.7808 & 0.8554 & 0.9229 & 0.9850 & 1.0000 & 1.0000 & 1.0000 \\ 0.7124 & 0.2876 & 0.6039 & 0.7348 & 0.8353 & 0.9201 & 0.9947 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ \vdots & \vdots \\ 0.8104 & 0.1896 & 0.5058 & 0.6368 & 0.7373 & 0.8221 & 0.8967 & 0.9642 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ 0.7358 & 0.2642 & 0.5804 & 0.7114 & 0.8119 & 0.8966 & 0.9713 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \end{bmatrix}$
$R_7 =$	$\begin{bmatrix} 0 & 0.01 & 0.04 & 0.09 & 0.16 & 0.25 & 0.36 & 0.49 & 0.64 & 0.81 & 1 \\ 0.5457 & 0.4543 & 0.4643 & 0.4943 & 0.5443 & 0.6143 & 0.7043 & 0.8143 & 0.9443 & 1.0000 & 1.0000 & 1.0000 \\ 0.4910 & 0.5090 & 0.5190 & 0.5490 & 0.5990 & 0.6690 & 0.7590 & 0.8690 & 0.9990 & 1.0000 & 1.0000 & 1.0000 \\ 0.3497 & 0.6503 & 0.6603 & 0.6903 & 0.7403 & 0.8103 & 0.9003 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ \vdots & \vdots \\ 0.9998 & 0.0002 & 0.0102 & 0.0402 & 0.0902 & 0.1602 & 0.2502 & 0.3602 & 0.4902 & 0.6402 & 0.8102 & 1.0000 \\ 0.9877 & 0.0123 & 0.0223 & 0.0523 & 0.1023 & 0.1723 & 0.2623 & 0.3723 & 0.5023 & 0.6523 & 0.8223 & 1.0000 \end{bmatrix}$
$R_8 =$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0.2252 & 0.7748 & 0.7748 & 0.7748 & 0.7748 & 0.7748 & 0.7748 & 0.7748 & 0.7748 & 0.7748 & 1.0000 \\ 0.1913 & 0.8087 & 0.8087 & 0.8087 & 0.8087 & 0.8087 & 0.8087 & 0.8087 & 0.8087 & 0.8087 & 1.0000 \\ 0.1160 & 0.8840 & 0.8840 & 0.8840 & 0.8840 & 0.8840 & 0.8840 & 0.8840 & 0.8840 & 0.8840 & 1.0000 \\ \vdots & \vdots \\ 0.8334 & 0.1666 & 0.1666 & 0.1666 & 0.1666 & 0.1666 & 0.1666 & 0.1666 & 0.1666 & 0.1666 & 1.0000 \\ 0.8959 & 0.1041 & 0.1041 & 0.1041 & 0.1041 & 0.1041 & 0.1041 & 0.1041 & 0.1041 & 0.1041 & 1.0000 \end{bmatrix}$

As a result of the intersection of fuzzy relations R_1, R_2, \dots, R_8 , a general functional solution is obtained in the form of the following matrix.

$$R = \left[\begin{array}{c|cccccccccccc} & 0 & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 & 1 \\ \hline x_1 = 25090.5 & 0.0114 & 0.1114 & 0.2114 & 0.3114 & 0.4114 & 0.5114 & 0.6114 & 0.5610 & 0.4610 & 0.3610 & 0.2610 \\ x_2 = 24987.5 & 0.0018 & 0.1018 & 0.2018 & 0.3018 & 0.4018 & 0.5018 & 0.6018 & 0.5094 & 0.4094 & 0.3094 & 0.2094 \\ x_3 = 24700.2 & 0.0193 & 0.1193 & 0.2193 & 0.3193 & 0.4193 & 0.5193 & 0.4878 & 0.3878 & 0.2878 & 0.1878 & 0.0878 \\ \vdots & \vdots \\ x_{332} = 26346.0 & 0.0002 & 0.0102 & 0.0402 & 0.0902 & 0.1602 & 0.1666 & 0.1666 & 0.1666 & 0.1666 & 0.1666 & 0.8357 \\ x_{333} = 26496.7 & 0.0123 & 0.0223 & 0.0523 & 0.1023 & 0.1041 & 0.1041 & 0.1041 & 0.1041 & 0.1041 & 0.1041 & 0.8740 \end{array} \right],$$

which reflects the cause-effect relations between the localization features of historical data x_t and the estimates of the DJIA value. In this case, the fuzzy interpretation of the t -th historical data A_t ($t = 1 \div 333$) is established by the compositional inference rule: $A_t = G_t \circ R$ ($t = 1 \div 333$), where G_t is a mapping of the x_t in the form of the fuzzy subset of universe U . Choosing the compositional rule in the form

$$\mu_{A_t}(i) = \max_{x \in U} \{\min(\mu_{G_t}(x), \mu_R(x))\}, \text{ and assuming } \mu_{G_t}(x) = \begin{cases} 0, & x \neq x_t; \\ 1, & x = x_t, \end{cases} \text{ as a result}$$

we have: $\mu_{A_t}(i) = \mu_R(x_t, i)$. This means that the fuzzy set A_t reflects the t -th DJIA index on the discrete universe I , the corresponding values of the membership function of which are located on the t -th row of the matrix R . In particular, the fuzzy analogue of the historical data $x_1 = 25090.5$ is the fuzzy set (1st row of matrix R): $A_1 = \{0.0114/0, 0.1114/0.1, 0.2114/0.2, 0.3114/0.3, 0.4114/0.4, 0.5114/0.5, 0.6114/0.6, 0.5610/0.7, 0.4610/0.8, 0.3610/0.9, 0.2610/1\}$.

Thus, all historical data of the DJIA are reflected in the form of corresponding fuzzy sets, which are summarized in Table 1.

The last column of Table 1 presents point estimates of fuzzy sets (PE FS) or, which is the same, defuzzified values of the corresponding FS A_t ($t = 1 \div 333$), which conditionally restore the configuration of the DJIA time series on the scale of the segment $[0, 1]$ (see Fig. 3).

PE FS are established according to the following reasoning. For a fuzzy subset of the universal discrete set, i.e. in our case, for $A \subset I$ the α -level sets are defined as following: $A_\alpha = \{i | \mu_A(i) \geq \alpha, i \in I\}$ ($\alpha \in [0, 1]$). Further, for each set A_α the corresponding cardinal number $M(A_\alpha)$ is calculated according to the formula:

$$M(A_\alpha) = \sum_{k=1}^n u_k / n, u_k \in A_\alpha \quad (2)$$

As a result, PE of the fuzzy set A is established by the formula:

$$F(A) = (1/\alpha_{\max}) \int_0^{\alpha_{\max}} M(A_\alpha) d\alpha, \quad (3)$$

where α_{\max} is the maximum value of the membership function of fuzzy sets A .

Table 1. Detailed fuzzy analogue of the DJIA time series.

Date	DJIA	FS	Values of the corresponding membership function	PE FS						
		0	0.1	0.2	...	0.8	0.9	1		
15.06.2018	25090.5	A_1	0.0114	0.1114	0.2114	...	0.4610	0.3610	0.2610	0.6062
18.06.2018	24987.5	A_2	0.0018	0.1018	0.2018	...	0.4094	0.3094	0.2094	0.5939
19.06.2018	24700.2	A_3	0.0193	0.1193	0.2193	...	0.2878	0.1878	0.0878	0.5330
...
21.12.2018	22445.4	A_{132}	0.9138	0.0412	0.0412	...	0.0412	0.0412	0.0412	0.0226
24.12.2018	21792.2	A_{133}	0.9801	0.0423	0.0423	...	0.0423	0.0423	0.0423	0.0216
...
09.10.2019	26246.0	A_{332}	0.0002	0.0102	0.0402	...	0.1666	0.1666	0.8357	0.9321
10.10.2019	26496.7	A_{333}	0.0123	0.0223	0.0523	...	0.1041	0.1041	0.8740	0.9534

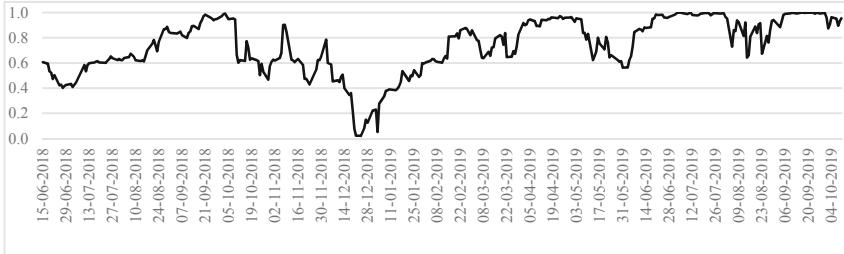


Fig. 3 DJIA time series in the notation of point estimates of fuzzy sets.

So, according to (2) for the fuzzy set A_1 as analogue of the DJIA index x_1 we have:

- for $0 < \alpha < 0.0114$, $\Delta\alpha = 0.0114$, $A_{1\alpha} = \{0, 0.1, 0.2, 0.3, \dots, 0.8, 0.9, 1\}$, $M(A_{1\alpha}) = 0.50$;
- for $0.0114 < \alpha < 0.1114$, $\Delta\alpha = 0.1$, $A_{1\alpha} = \{0.1, 0.2, 0.3, \dots, 0.8, 0.9, 1\}$, $M(A_{1\alpha}) = 0.55$;
- for $0.1114 < \alpha < 0.2114$, $\Delta\alpha = 0.1$, $A_{1\alpha} = \{0.2, 0.3, \dots, 0.8, 0.9, 1\}$, $M(A_{1\alpha}) = 0.60$;
- for $0.2114 < \alpha < 0.2610$, $\Delta\alpha = 0.0495$, $A_{1\alpha} = \{0.3, 0.4, \dots, 0.8, 0.9, 1\}$, $M(A_{1\alpha}) = 0.65$;
- for $0.2610 < \alpha < 0.3114$, $\Delta\alpha = 0.0505$, $A_{1\alpha} = \{0.3, 0.4, \dots, 0.7, 0.8, 0.9\}$, $M(A_{1\alpha}) = 0.60$;
- for $0.3114 < \alpha < 0.3610$, $\Delta\alpha = 0.0495$, $A_{1\alpha} = \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, $M(A_{1\alpha}) = 0.65$;
- for $0.3610 < \alpha < 0.4114$, $\Delta\alpha = 0.0505$, $A_{1\alpha} = \{0.4, 0.5, 0.6, 0.7, 0.8\}$, $M(A_{1\alpha}) = 0.60$;
- for $0.4114 < \alpha < 0.4610$, $\Delta\alpha = 0.0495$, $A_{1\alpha} = \{0.5, 0.6, 0.7, 0.8\}$, $M(A_{1\alpha}) = 0.65$;
- for $0.4610 < \alpha < 0.5114$, $\Delta\alpha = 0.0505$, $A_{1\alpha} = \{0.5, 0.6, 0.7\}$, $M(A_{1\alpha}) = 0.60$;
- for $0.5114 < \alpha < 0.5610$, $\Delta\alpha = 0.0495$, $A_{1\alpha} = \{0.6, 0.7\}$, $M(A_{1\alpha}) = 0.65$;
- for $0.5610 < \alpha < 0.6114$, $\Delta\alpha = 0.0505$, $A_{1\alpha} = \{0.6\}$, $M(A_{1\alpha}) = 0.60$.

Then, according to (3), the PE of the fuzzy set A_1 is the following number:

$$\begin{aligned}
 F(A) = (1/0.6114) \int_0^{0.6114} M(A_\alpha) d\alpha &= [0.0114 \cdot 0.5 + 0.1 \cdot 0.55 \\
 &+ 0.1 \cdot 0.60 + 0.0495 \cdot 0.65 + 0.0505 \cdot 0.60 + 0.0495 \cdot 0.65 \\
 &+ 0.0505 \cdot 0.60 + 0.0495 \cdot 0.65 + 0.0505 \cdot 0.60 + 0.0495 \cdot 0.65 + \\
 &+ 0.0505 \cdot 0.60]/0.6114 = 0.6062.
 \end{aligned}$$

The fuzzy sets presented in Table 1 form an overly redundant set of qualitative evaluation criteria, which does not allow the use of well-known models for fuzzy time series forecasting. Therefore, it is necessary to establish the appropriate number of fuzzy sets. To do this, we use the step-by-step procedure proposed in [9].

Step 1. Sort the DJIA index x_t ($t = 1 \div 333$) into an ascending sequence $\{x_{p(i)}\}$, where p is a permutation that sorts the DJIA values in ascending order: $x_{p(i)} \leq x_{p(i+1)}$.

Step 2. Calculation of the average value over the totality of all pairwise distances $d_i = |x_{p(i)} - x_{p(i+1)}|$ between any two consecutive values $x_{p(t)}$ and $x_{p(t+1)}$ according to the formula:

$$AD = \sum_{i=1}^{n-1} |x_{p(i)} - x_{p(i+1)}| / (n - 1), \quad (4)$$

and standard deviation according to the formula

$$\sigma_{AD} = \sqrt{\sum_{i=1}^{n-1} (d_i - AD)^2 / (n - 1)} \quad (5)$$

Step 3. Elimination of anomalies, that is, outliers that need to be removed. The values of pairwise distances that do not satisfy the condition are subject to removed:

$$AD - s_{AD} \leq d_i \leq AD + s_{AD}. \quad (6)$$

Step 4. Recalculation of AD on the set of pairwise distances remaining after their resorting and, assuming $D_1 = D_{\min} - AD$, $D_2 = D_{\max} + AD$, calculation the appropriate number m of context fuzzy sets according to the formula

$$m = (D_2 - D_1 - AD) / (2 \times AD). \quad (7)$$

Thus, guided by formulas (4) and (5), for the DJIA time series ($n = 333$) we have: $AD = 16.7681$ and $\sigma_{AD} = 43.8484$. Removing d_i that do not satisfy the condition

$$-27.0803 \gg 16.7681 - 43.8484 \leq d_i \leq 16.7681 + 43.8484 \gg 60.6165,$$

the final value of the average value for the totality of the remaining pairwise distances d_i is obtained as $AD = 9.9740$. Then, choosing the universe as segment $D = [D_{\min} - AD, D_{\max} + AD] = [D_1, D_2]$ (see [10]), where $D_1 = 21792.2 - 9.9740 = 21782.2260$, $D_2 = 27359.2 + 9.9740 = 27369.1740$, according to the formula (7) the appropriate number of FS, reflecting the qualitative criteria for evaluating the DJIA index, are established as:

$$m = (27369.1740 - 21782.2260 - 9.9740) / (2 \times 9.9740) = 279.5765.$$

Further, assuming F_1 as point estimate of FS A_1 and F_{333} as point estimate of FS A_{333} , the segment $[F_1, F_{333}]$ is divided into 280 equal segments a_k ($k = 1 \div 280$) with length $(F_{333} - F_1) / 280$. Then the FSs A_t ($t = 1 \div 333$) (see Table 1) can be distributed among the corresponding groups by the rule: "If PE of FS A_t from the interval a_k ($k = 1 \div 280$), then A_t is included in the k -th group". As a result of this distribution, 144 groups were obtained¹, within which qualitative evaluation criteria were formed and summarized in Table 2.

Table 2. Qualitative criteria for evaluating the DJIA index.

Criteria	Values of the membership function of the fuzzy subset of the universe I						
	0	0.1	0.2	0.3	0.4	0.5	0.6
C_1	0.9138	0.0412	0.0412	0.0412	0.0412	0.0412	0.0412
C_2	0.8644	0.1200	0.1200	0.1200	0.1200	0.0972	0.0472
C_3	0.8173	0.1960	0.1960	0.1960	0.1960	0.1606	0.0906
C_4	0.8115	0.2051	0.2051	0.2051	0.2051	0.1613	0.0913
C_5	0.7489	0.2985	0.2985	0.2985	0.2750	0.1830	0.1130
C_6	0.7193	0.3392	0.3392	0.3392	0.2896	0.1996	0.1296
C_7	0.6396	0.4393	0.4393	0.4393	0.3461	0.2561	0.1861
C_8	0.6379	0.4414	0.4414	0.4414	0.4414	0.3475	0.2575
C_9	0.6292	0.4514	0.4514	0.4514	0.4514	0.3545	0.2645
C_{10}	0.5875	0.4978	0.4978	0.4978	0.4978	0.3892	0.2992
...
C_{132}	0.0133	0.0233	0.0533	0.0931	0.0931	0.0931	0.0931
C_{133}	0.0185	0.0285	0.0585	0.0834	0.0834	0.0834	0.0834
C_{134}	0.0225	0.0325	0.0625	0.0737	0.0737	0.0737	0.0737
C_{135}	0.0291	0.0391	0.0684	0.0684	0.0684	0.0684	0.0684
C_{136}	0.0369	0.0469	0.0532	0.0532	0.0532	0.0532	0.0532
C_{137}	0.0440	0.0516	0.0516	0.0516	0.0516	0.0516	0.0516
C_{138}	0.0422	0.0422	0.0422	0.0422	0.0422	0.0422	0.0422

(continued)

Table 2. (*continued*)

If the group includes only one fuzzy set, then this set forms the evaluation criterion. In cases where the group includes two or more fuzzy sets, the evaluation criterion is formed in the form of their intersection using the “min” operation on the corresponding values of the membership functions. For example, the 1st group includes fuzzy sets A_{132} and A_{133} , whose PE are located in the interval [0.0216, 0.0251] (see Table 1). Information about these sets on the universe I , including the values of their membership functions and PE, calculated by formulas (2) and (3), as well as relative to the fuzzy set $C_1 = A_{132} \cap A_{133}$, as the 1st criterion for evaluating the DJIA index, are summarized in the Table 3.

Table 3. Formation of the C_1 criterion for evaluating the DJIA index.

FS	Values of membership functions of fuzzy sets from the 1st group								PE	
	0	0.1	0.2	0.3	...	0.7	0.8	0.9		
A_{132}	0.9138	0.0412	0.0412	0.0412	...	0.0412	0.0412	0.0412	0.0412	0.0226
A_{133}	0.9801	0.0423	0.0423	0.0423	...	0.0423	0.0423	0.0423	0.0423	0.0216
C_1	0.9138	0.0412	0.0412	0.0412	...	0.0412	0.0412	0.0412	0.0412	0.0226

As another example, a group of fuzzy sets is chosen (see Table 4), PE of which are located in the last 144-th interval [0.9960, 0.9995]. Here, the qualitative criterion for evaluating the DJIA index is formed as following intersection:

$$\begin{aligned} C_{144} = & A_{264} \cap A_{265} \cap A_{269} \cap A_{275} \cap A_{279} \cap A_{311} \cap A_{312} \\ & \cap A_{315} \cap A_{316} \cap A_{317} \cap A_{318} \cap A_{319} \cap A_{320} \cap A_{322} \cap A_{323} \cap A_{325}. \end{aligned}$$

Thus, in terms of fuzzy sets A_t ($t = 1 \div 333$) and C_k ($k = 1 \div 144$), the DJIA time series and its interpretation in PE notations are summarized in Table 5.

4 DJIA Fuzzy Time Series Forecasting

Identified and divided into groups, the 1st order relationships are the fuzzy relations that reflects the cause-effect relations between qualitative (fuzzy) estimates of the DJIA index as a linguistic variable and its fuzzy predict:

- *unambiguously*, in the form of the implication: “If x_t is C_i , then x_{t+1} is C_j ” ($t = 1 \div 333; i, j = 1 \div 144$),
- *ambiguously*, in the form of the implication: “If x_t is C_i , then x_{t+1} is $C_{j(1)}$ or $C_{j(2)}$ or ... or $C_{j(p)}$ ” ($t = 1 \div 333; i, j(1), j(2), \dots, j(p) = 1 \div 144$).

¹ The remaining 136 groups are empty, i.e. do not include PE of any fuzzy sets.

Table 4. Formation of the C_{144} criterion for evaluating the DJIA index.

FS	Values of membership functions of fuzzy sets from the 1st group									PE
	0	0.1	0.2	0.3	...	0.7	0.8	0.9	1	
A_{264}	0.0012	0.0012	0.0012	0.0012	...	0.0012	0.0012	0.0012	0.9509	0.9994
A_{265}	0.0039	0.0039	0.0039	0.0039	...	0.0039	0.0039	0.0039	0.9460	0.9979
A_{269}	0.0018	0.0018	0.0018	0.0018	...	0.0018	0.0018	0.0018	0.9627	0.9991
A_{275}	0.0071	0.0071	0.0071	0.0071	...	0.0071	0.0071	0.0071	0.9680	0.9963
A_{279}	0.0057	0.0057	0.0057	0.0057	...	0.0057	0.0057	0.0057	0.9670	0.9970
A_{311}	0.0050	0.0050	0.0050	0.0050	...	0.0050	0.0050	0.0050	0.9445	0.9974
A_{312}	0.0054	0.0054	0.0054	0.0054	...	0.0054	0.0054	0.0054	0.9667	0.9972
A_{315}	0.0012	0.0012	0.0012	0.0012	...	0.0012	0.0012	0.0012	0.9617	0.9994
A_{316}	0.0032	0.0032	0.0032	0.0032	...	0.0032	0.0032	0.0032	0.9646	0.9983
A_{317}	0.0063	0.0063	0.0063	0.0063	...	0.0063	0.0063	0.0063	0.9674	0.9967
A_{318}	0.0022	0.0022	0.0022	0.0022	...	0.0022	0.0022	0.0022	0.9632	0.9989
A_{319}	0.0029	0.0029	0.0029	0.0029	...	0.0029	0.0029	0.0029	0.9475	0.9984
A_{320}	0.0020	0.0020	0.0020	0.0020	...	0.0020	0.0020	0.0020	0.9492	0.9989
A_{322}	0.0010	0.0010	0.0010	0.0010	...	0.0010	0.0010	0.0010	0.9514	0.9995
A_{323}	0.0067	0.0067	0.0067	0.0067	...	0.0067	0.0067	0.0067	0.9423	0.9964
A_{325}	0.0043	0.0043	0.0043	0.0043	...	0.0043	0.0043	0.0043	0.9454	0.9977
C_{144}	0.0010	0.0010	0.0010	0.0010	...	0.0010	0.0010	0.0010	0.9423	0.9995

If in an unambiguous case everything is extremely clear, then in the case of the presence of two or more alternative fuzzy conclusions, the predict is consolidated using the logical operator “OR”. In particular, for fuzzy relations $C_{36} \Rightarrow C_{21}, C_{51}, C_{52}$ we have: “If x_t is C_{36} , then x_{t+1} is C_{21} or C_{51} or C_{52} ”, where the generalized predict is the fuzzy set $F = C_{21} \cup C_{51} \cup C_{52}$ with the membership function [8]:

$$\mu_F(u) = \mu_{C_{21} \cup C_{51} \cup C_{52}}(u) = \max\{\mu_{C_{21}}(u), \mu_{C_{51}}(u), \mu_{C_{52}}(u)\}$$

Table 6 presents generalizing fuzzy sets reflecting the consequences in the 1st order relationship groups.

Table 5. Reconstruction of the DJIA time series in terms of fuzzy sets and their PE.

SN	Date	DJIA	FS	PE	Criterion	Detailing	PE
1	15.06.2018	25090.5	A_1	0.6062	C_{55}	$A_1 \cap A_{107} \cap A_{164}$	0.6051
2	18.06.2018	24987.5	A_2	0.5939	C_{51}	$A_2 \cap A_{93}$	0.5936
3	19.06.2018	24700.2	A_3	0.5330	C_{41}	$A_3 \cap A_{18} \cap A_{150}$	0.5335
4	20.06.2018	24657.8	A_4	0.5219	C_{39}	A_4	0.5219
5	21.06.2018	24461.7	A_5	0.4723	C_{32}	$A_5 \cap A_{111} \cap A_{112}$	0.4729
6	22.06.2018	24580.9	A_6	0.5024	C_{36}	$A_6 \cap A_{92} \cap A_{156}$	0.5026
7	25.06.2018	24252.8	A_7	0.4220	C_{21}	A_7	0.4220
8	26.06.2018	24283.1	A_8	0.4280	C_{23}	$A_8 \cap A_{113}$	0.4285
9	27.06.2018	24117.6	A_9	0.4013	C_{18}	$A_9 \cap A_{127}$	0.4013
10	28.06.2018	24216.1	A_{10}	0.4154	C_{20}	$A_{10} \cap A_{148}$	0.4154
11	29.06.2018	24271.4	A_{11}	0.4256	C_{22}	A_{11}	0.4256
12	02.07.2018	24307.2	A_{12}	0.4332	C_{24}	A_{12}	0.4332
13	03.07.2018	24174.8	A_{13}	0.4089	C_{19}	A_{13}	0.4089
...
325	30.09.2019	26916.8	A_{325}	0.9977	C_{144}	$A_{264} \cap A_{265} \cap A_{269} \cap A_{275} \cap A_{279} \cap A_{311} \cap A_{312} \cap A_{315} \cap \dots \cap A_{320} \cap A_{322} \cap A_{323} \cap A_{325}$	0.9995
326	01.10.2019	26573.0	A_{326}	0.9627	C_{134}	$A_{70} \cap A_{212} \cap A_{284} \cap A_{326} \cap A_{329}$	0.9643
327	02.10.2019	26078.6	A_{327}	0.8741	C_{111}	$A_{251} \cap A_{327}$	0.8764
328	03.10.2019	26201.0	A_{328}	0.9048	C_{119}	$A_{103} \cap A_{298} \cap A_{328}$	0.9047
329	04.10.2019	26573.7	A_{329}	0.9627	C_{134}	$A_{70} \cap A_{212} \cap A_{284} \cap A_{326} \cap A_{329}$	0.9643

(continued)

Table 5. (*continued*)

SN	Date	DJIA	FS	PE	Criterion	Detailing	PE
330	07.10.2019	26478.0	A ₃₃₀	0.9513	C ₁₃₁	$A_{71} \cap A_{80} \cap A_{285} \cap A_{320} \cap A_{333}$	0.9538
331	08.10.2019	26164.0	A ₃₃₁	0.8959	C ₁₁₇	A ₃₃₁	0.8959
332	09.10.2019	26346.0	A ₃₃₂	0.9321	C ₁₂₅	$A_{204} \cap A_{322}$	0.9320
333	10.10.2019	26496.7	A ₃₃₃	0.9534	C ₁₃₁	$A_{71} \cap A_{80} \cap A_{285} \cap A_{320} \cap A_{333}$	0.9538

Table 6. Generalizing fuzzy sets reflecting the consequences in the 1st order groups.

Fuzzy predict	Membership function values							PE
	0	0.1	0.2	0.3	0.4	0.5	0.6	
F_1	0.9138	0.2051	0.2051	0.2051	0.2051	0.1613	0.0913	0.0412
F_2	0.5875	0.4978	0.4978	0.4978	0.3892	0.2992	0.2292	0.1492
F_3	0.9138	0.0412	0.0412	0.0412	0.0412	0.0412	0.0412	0.0226
F_4	0.7193	0.3392	0.3392	0.3392	0.2896	0.1996	0.1296	0.0796
F_5	0.6379	0.4414	0.4414	0.4414	0.3475	0.2575	0.1875	0.1375
F_6	0.7489	0.2985	0.2985	0.2985	0.2730	0.1830	0.1130	0.0630
F_7	0.8173	0.1960	0.1960	0.1960	0.1960	0.1606	0.0906	0.0406
...
F_{136}	0.0117	0.0217	0.0463	0.0963	0.1161	0.1161	0.1161	0.8730
F_{137}	0.0291	0.0391	0.0684	0.0684	0.0684	0.0684	0.0684	0.9215
F_{138}	0.0337	0.0337	0.0337	0.0337	0.0337	0.0337	0.0337	0.9173
F_{139}	0.0422	0.0422	0.0585	0.0834	0.0834	0.0834	0.0834	0.9800
F_{140}	0.0337	0.0337	0.0625	0.0737	0.0737	0.0737	0.0737	0.9173
F_{141}	0.0147	0.0147	0.0147	0.0147	0.0147	0.0147	0.0147	0.9423
F_{142}	0.0369	0.0469	0.0532	0.0532	0.0532	0.0532	0.0532	0.9423
F_{143}	0.0422	0.0422	0.0625	0.0737	0.0737	0.0737	0.0737	0.9800
F_{144}	0.0337	0.0337	0.0625	0.0737	0.0737	0.0737	0.0737	0.9423

Thus, in the PE notation of fuzzy sets, the predictive model built on the basis of 1st order internal relationships, or simply, the 1st order model induces defuzzified outputs (predictions) on the scale of the interval [0, 1], which are presented in Table 7. The geometric interpretation of the model in comparison with the DJIA time series in the PE notation of fuzzy sets is shown in Fig. 4.

At the end of Table 7, the values of the Mean Squared Error (MSE), the Mean Absolute Percentage Error (MAPE) and the Mean Percentage Error (MPE) are presented, which reflect the adequacy of the proposed predictive model. Errors according to these criteria are calculated by following formulas [10]:

$$\text{MSE} = \frac{1}{m} \sum_{t=1}^m (F_t - A_t)^2, \quad \text{MAPE} = \frac{1}{m} \sum_{t=1}^m \frac{|F_t - A_t|}{A_t} \times 100\%,$$

$$\text{MPE} = \frac{1}{m} \sum_{t=1}^m \frac{F_t - A_t}{A_t} \times 100\%$$

where, m is the length of the time series; A_t is the DJIA at time t ; F_t is the predict of the A_t .

The MSE criterion is most often used when choosing the optimal predictive model and highlights possible significant errors in forecasts. In our case, $\text{MSE} = 0.0020$ (see Table 7) indicates that the prediction error is too low. The MAPE shows how large the forecast errors are compared to the actual DJIA values. MPE is a more informative criterion for assessing the adequacy of the forecasting model, which determines the “bias” of the constructed predict, that is, its *перманентный* permanent underestimation or overestimation. In our case, the $\text{MPE} = -0.4391\%$ (see Table 7) reflects a slight bias of the predictive model, which does not exceed the normative 5% threshold on the left.²

5 Forecasting the DJIA Time Series in Nominal Values Based on the Use of the 1st Order Fuzzy Model

After forecasting the DJIA fuzzy time series in terms of point estimates of the fuzzy outputs (predicts) on the scale of a single segment, it's time to reflect them in nominal values. To do this, the three-layer feedforward neural network is used.³

² If there were a large negative MPE, then the constructed model would be considered “overestimating”. If the MPE indicator would reflect a large positive percentage value, i.e. beyond the 5% threshold on the right, then the model would be considered “underestimating”.

³ It is clear that by a simple mapping $x = a + t(b-a)$, where $x \in [a, b]$ (in our case, $x \in [21752.2, 27359.2]$), $t \in [0, 1]$, it is impossible to reflect point estimates of fuzzy outputs, because obviously, the relationship between them is non-linear.

Table 7. Predictive model of the DJIA time series.

№	Date	DJIA fuzzy analogue		FS	1st order relationships group	Model output (predict)		Detailing the fuzzy output
		FS	PE			FS	PE	
1	15.06.2018	A ₁	0.6062	C ₅₅	$C_{55} \Rightarrow C_{51}, C_{54}, C_{59}$	—	—	$C_{51} \cup C_{54} \cup C_{59}$
2	18.06.2018	A ₂	0.5939	C ₅₁	$C_{51} \Rightarrow C_{40}, C_{41}$	F ₅₅	0.6203	$C_{40} \cup C_{41}$
3	19.06.2018	A ₃	0.5330	C ₄₁	$C_{41} \Rightarrow C_{28}, C_{39}, C_{49}$	F ₅₁	0.5318	$C_{28} \cup C_{39} \cup C_{49}$
4	20.06.2018	A ₄	0.5219	C ₃₉	$C_{39} \Rightarrow C_{32}$	F ₄₁	0.4557	$C_{32} \Rightarrow C_{23}, C_{32}, C_{36}$
5	21.06.2018	A ₅	0.4723	C ₃₂	$C_{32} \Rightarrow C_{23}, C_{32}, C_{36}$	F ₃₉	0.4729	C_{32}
6	22.06.2018	A ₆	0.5024	C ₃₆	$C_{36} \Rightarrow C_{21}, C_{51}, C_{52}$	F ₃₂	0.4546	$C_{21} \cup C_{51} \cup C_{52}$
7	25.06.2018	A ₇	0.4220	C ₂₁	$C_{21} \Rightarrow C_{23}$	F ₃₆	0.5240	$C_{23} \cup C_{32} \cup C_{36}$
8	26.06.2018	A ₈	0.4280	C ₂₃	$C_{23} \Rightarrow C_{18}, C_{38}$	F ₂₁	0.4285	C_{23}
9	27.06.2018	A ₉	0.4013	C ₁₈	$C_{18} \Rightarrow C_{12}, C_{20}$	F ₂₃	0.4298	$C_{18} \cup C_{38}$
10	28.06.2018	A ₁₀	0.4154	C ₂₀	$C_{20} \Rightarrow C_{22}, C_{26}$	F ₁₈	0.3512	$C_{12} \cup C_{20}$
11	29.06.2018	A ₁₁	0.4256	C ₂₂	$C_{22} \Rightarrow C_{24}$	F ₂₀	0.4299	$C_{22} \cup C_{26}$
12	02.07.2018	A ₁₂	0.4332	C ₂₄	$C_{24} \Rightarrow C_{19}$	F ₂₂	0.4332	C_{24}
...

(continued)

Table 7. (continued)

№	Date	DJIA fuzzy analogue		FS	1st order relationships group	Model output (predict)		Detailing the fuzzy output	
		FS	PE			FS	PE		
328	03.10.2019	A328	0.9048	C ₁₁₉	C ₁₁₉ ⇒C ₁₀₅ , C ₁₂₂ , C ₁₃₄	F ₁₁₁	0.8872	C ₁₁₃ ∪ C ₁₁₉	
329	04.10.2019	A329	0.9627	C ₁₃₄	C ₁₃₄ ⇒C ₁₁₁ , C ₁₃₁ , C ₁₃₂	F ₁₁₉	0.8777	C ₁₀₅ ∪ C ₁₂₂ ∪ C ₁₃₄	
330	07.10.2019	A330	0.9513	C ₁₃₁	C ₁₃₁ ⇒C ₈₁ , C ₁₁₇ , C ₁₂₇ , C ₁₂₉	F ₁₃₄	0.8944	C ₁₁₁ ∪ C ₁₃₁ ∪ C ₁₃₂	
331	08.10.2019	A331	0.8959	C ₁₁₇	C ₁₁₇ ⇒C ₁₂₅	F ₁₃₁	0.8127	C ₈₁ ∪ C ₁₁₇ ∪ C ₁₂₇ ∪ C ₁₂₉	
332	09.10.2019	A332	0.9321	C ₁₂₅	C ₁₂₅ ⇒C ₁₁₆ , C ₁₃₁	F ₁₁₇	0.9320	C ₁₂₅	
333	10.10.2019	A333	0.9534	C ₁₃₁	C ₁₃₁ ⇒C ₈₁ , C ₁₁₇ , C ₁₂₇ , C ₁₂₉	F ₁₂₅	0.9055	C ₁₁₆ ∪ C ₁₃₁	
		<i>MSE</i>		0.0020					
		<i>MAPE</i>		4.4879					
		<i>MPE</i>		-0.4391					

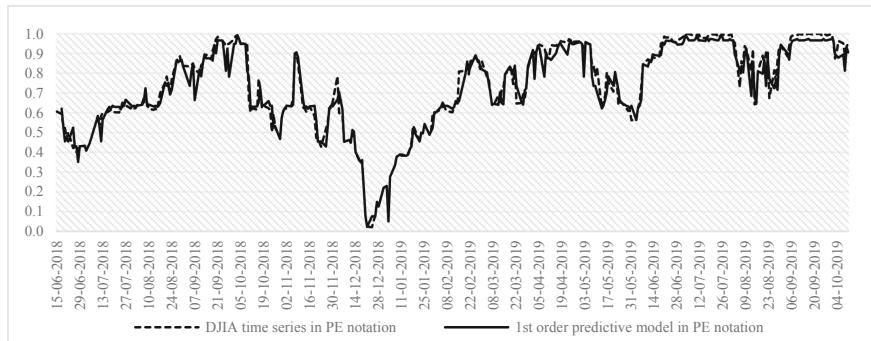


Fig. 4. 1st Order predictive model in PE notation.

To build an approximation neural network in MATLAB notation (see Fig. 5), the set of training pairs $\{(A_t^{\text{def}}, x_t)\}_{t=1}^{136}$ was chosen as a basis, where x_t is the DJIA value at time t ; A_t^{def} is the defuzzified value (PE) of the fuzzy set A_t , reflecting the x_t . After training, testing and validation (see Fig. 6), the neural network approximates a continuous function presented in tabular form (see Table 8).

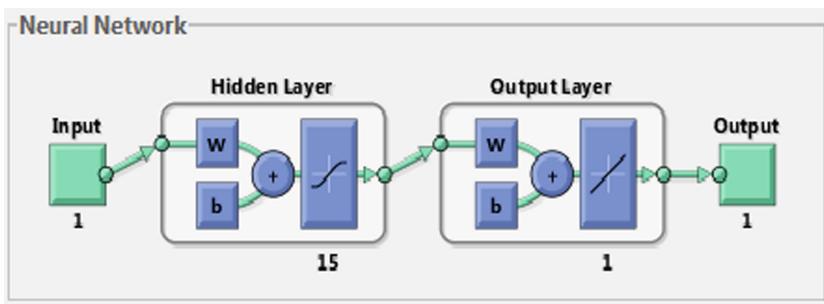


Fig. 5. Three-layer feedforward neural network in MATLAB notation.

The trained neural network induces at its output the nominal predictive values of the DJIA, corresponding to the defuzzified analogues of fuzzy outputs presented in Table 8. The predictions obtained in this way are summarized in Table 9, and the 1st order model is interpreted in Fig. 7 against the background of the original DJIA time series.

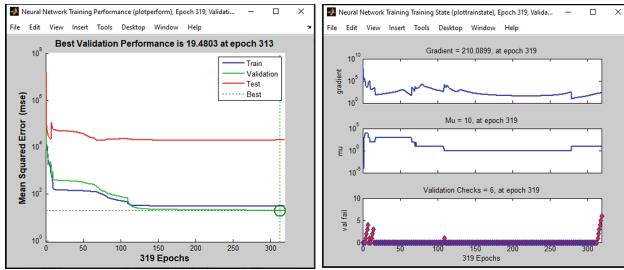
Table 8. Tabular representation of the function $x_t = f(A_t^{\text{def}})$.

T	A_t^{def}	x_t									
1	0.6062	25090.5	35	0.6392	25462.6	69	0.9835	26743.5	103	0.9025	26191.2
2	0.5939	24987.5	36	0.6467	25502.2	70	0.9615	26562.1	104	0.8462	25989.3
3	0.5330	24700.2	37	0.6736	25628.9	71	0.9529	26492.2	105	0.6264	25387.2
4	0.5219	24657.8	38	0.6614	25583.8	72	0.9384	26385.3	106	0.6202	25286.5
5	0.4723	24461.7	39	0.6474	25509.2	73	0.9465	26439.9	107	0.6051	25080.5
6	0.5024	24580.9	40	0.6204	25313.1	74	0.9489	26458.3	108	0.6205	25289.3
7	0.4220	24252.8	41	0.6143	25187.7	75	0.9712	26651.2	109	0.6314	25413.2
8	0.4280	24283.1	42	0.6213	25299.9	76	0.9870	26773.9	110	0.5844	25017.4
9	0.4013	24117.6	43	0.6125	25162.4	77	0.9921	26828.4	111	0.4733	24465.6
10	0.4154	24216.1	44	0.6569	25558.7	78	0.9687	26627.5	112	0.4730	24464.7
11	0.4256	24271.4	45	0.7003	25669.3	79	0.9475	26447.1	113	0.4286	24286.0
12	0.4332	24307.2	46	0.7517	25758.7	80	0.9523	26486.8	114	0.5173	24640.2
13	0.4089	24174.8	47	0.7827	25822.3	81	0.9452	26430.6	115	0.5457	24748.7
14	0.4448	24356.7	48	0.7383	25733.6	82	0.6641	25598.7	116	0.6237	25366.4
15	0.4709	24456.5	49	0.6924	25657.0	83	0.6021	25052.8	117	0.6221	25338.8
16	0.5530	24776.6	50	0.7677	25790.4	84	0.6220	25340.0	118	0.6533	25538.5
17	0.5835	24919.7	51	0.8656	26049.6	85	0.6159	25250.6	119	0.7845	25826.4
18	0.5330	24700.5	52	0.8699	26064.0	86	0.7716	25798.4	120	0.5991	25027.1
19	0.5844	24924.9	53	0.8862	26124.6	87	0.7230	25706.7	121	0.5881	24947.7
20	0.5981	25019.4	54	0.8454	25986.9	88	0.6254	25379.5	122	0.4531	24389.0
21	0.6034	25064.4	55	0.8378	25964.8	89	0.6359	25444.3	123	0.4622	24423.3
22	0.6090	25119.9	56	0.8337	25952.5	90	0.6225	25317.4	124	0.4482	24370.2
23	0.6150	25199.3	57	0.8412	25975.0	91	0.6145	25191.4	125	0.4892	24527.3
24	0.6034	25064.5	58	0.8485	25995.9	92	0.5030	24583.4	126	0.5065	24597.4
25	0.6027	25058.1	59	0.8211	25916.5	93	0.5936	24984.6	127	0.3992	24100.5
26	0.6011	25044.3	60	0.7978	25857.1	94	0.5298	24688.3	128	0.3448	23593.0
27	0.6173	25241.9	61	0.8398	25971.1	95	0.4673	24442.9	129	0.3600	23675.6
28	0.6316	25414.1	62	0.8495	25998.9	96	0.5752	24874.6	130	0.2192	23323.7
29	0.6513	25527.1	63	0.8914	26146.0	97	0.6086	25115.8	131	0.0785	22859.6
30	0.6371	25451.1	64	0.8936	26154.7	98	0.6255	25380.7	132	0.0226	22445.4
31	0.6218	25306.8	65	0.8693	26062.1	99	0.6189	25270.8	133	0.0216	21792.2
32	0.6318	25415.2	66	0.9147	26247.0	100	0.6391	25461.7	134	0.0821	22878.5

(continued)

Table 8. (continued)

T	A_t^{def}	x_t									
33	0.6223	25333.8	67	0.9416	26405.8	101	0.6778	25635.0	135	0.1491	23138.8
34	0.6227	25326.2	68	0.9720	26657.0	102	0.8999	26180.3	136	0.1257	23062.4

**Fig. 6.** Neural approximation of the function $x_t = f(A_t^{\text{def}})$ in MATLAB notation.

At the end of Table 9, the values of indicators $\text{MSE} = 72100.5$, $\text{MAPE} = 0.6830$ and $\text{MPE} = -0.530$ are presented. In this case, the MSE value reflects a relatively large prediction error, which is explained by insufficiently satisfactory training of the neural network ($\varepsilon = 19.4803$, see Fig. 4). Nevertheless, the MAPE indicator demonstrates an acceptable forecast error in comparison with the actual values of the DJIA time series. MPE, as a more informative criterion, reflects a slight bias of the prognostic model, which does not exceed the normative 5% threshold on the left.

Comparing the two proposed models, it is easy to see that the predictive model in nominal values of the DJIA index is significantly inferior to the predictive model of the DJIA index in PE terms of the corresponding fuzzy sets. Therefore, applying the predictive model in PE terms, as the predict of the DJIA index for the 334th day, the number $A_{334}^{\text{def}} = 0.8127$ is obtained (PE of the fuzzy predict $F_{131} = C_{81} \cup C_{117} \cup C_{127} \cup C_{129}$), which is interpreted in nominal value as 26204 using the neural network (see Fig. 7).

Table 9. DJIA time series forecasting taking into account 1st order internal relationships.

SN	Date	DJIA	Predict	SN	Date	DJIA	Predict	SN	Date	DJIA	Predict
1	15.06.2018	25090.5	112	21.11.2018	24464.7	24396	223	06.05.2019	26438.5	26430	
2	18.06.2018	24987.5	25299	113	23.11.2018	24286.0	24396	224	07.05.2019	25965.1	25821
3	19.06.2018	24700.2	24694	114	26.11.2018	24640.2	24291	225	08.05.2019	25967.3	25729
4	20.06.2018	24657.8	24400	115	27.11.2018	24748.7	24665	226	09.05.2019	25828.4	25729
5	21.06.2018	24461.7	24463	116	28.11.2018	25366.4	25329	227	10.05.2019	25942.4	25676
6	22.06.2018	24580.9	24396	117	29.11.2018	25338.8	25446	228	13.05.2019	25325.0	25329
7	25.06.2018	24252.8	24664	118	30.11.2018	25538.5	25424	229	14.05.2019	25532.1	25446
8	26.06.2018	24283.1	24285	119	03.12.2018	25826.4	25605	230	15.05.2019	25648.0	25605
9	27.06.2018	24117.6	24291	120	04.12.2018	25027.1	25676	231	16.05.2019	25862.7	25674
10	28.06.2018	24216.1	23537	121	06.12.2018	24947.7	25247	232	17.05.2019	25764.0	25824
11	29.06.2018	24271.4	24292	122	07.12.2018	24389.0	24391	233	20.05.2019	25679.9	25730
12	02.07.2018	24307.2	24308	123	10.12.2018	24423.3	24424	234	21.05.2019	25877.3	25878
13	03.07.2018	24174.8	24172	124	11.12.2018	24370.2	24372	235	22.05.2019	25776.6	25778
14	05.07.2018	24356.7	24358	125	12.12.2018	24527.3	24623	236	23.05.2019	25490.5	25694
15	06.07.2018	24456.5	24455	126	13.12.2018	24597.4	24592	237	24.05.2019	25585.7	25545
...
96	30.10.2018	24874.6	24877	207	11.04.2019	26143.1	26082	318	19.09.2019	27094.8	26598
97	31.10.2018	25115.8	25146	208	12.04.2019	26412.3	26052	319	20.09.2019	26935.1	26598

(continued)

Table 9 (*continued*)

SN	Date	DJIA	Predict	SN	Date	DJIA	Predict	SN	Date	DJIA	Predict
98	01.11.2018	25380.7	25329	209	15.04.2019	26384.8	26184	320	23.09.2019	26950.0	26598
99	02.11.2018	25270.8	25446	210	16.04.2019	26452.7	26300	321	24.09.2019	26807.8	26598
100	05.11.2018	25461.7	25424	211	17.04.2019	26449.5	26451	322	25.09.2019	26970.7	26661
101	06.11.2018	25635.0	25469	212	18.04.2019	26559.5	26451	323	26.09.2019	26891.1	26598
102	07.11.2018	26180.3	26179	213	22.04.2019	26511.1	26157	324	27.09.2019	26820.3	26598
103	08.11.2018	26191.2	26205	214	23.04.2019	26656.4	26430	325	30.09.2019	26916.8	26661
104	09.11.2018	25989.3	26093	215	24.04.2019	26597.1	26605	326	01.10.2019	26573.0	26598
105	12.11.2018	25387.2	25599	216	25.04.2019	26462.1	26447	327	02.10.2019	26078.6	26157
106	13.11.2018	25286.5	25299	217	26.04.2019	26543.3	26451	328	03.10.2019	26201.0	26129
107	14.11.2018	25080.5	25424	218	29.04.2019	26554.4	26515	329	04.10.2019	26573.7	26093
108	15.11.2018	25289.3	25299	219	30.04.2019	26592.9	26515	330	07.10.2019	26478.0	26157
109	16.11.2018	25413.2	25424	220	01.05.2019	26430.1	26447	331	08.10.2019	26164.0	25895
110	19.11.2018	25017.4	25441	221	02.05.2019	26307.8	25821	332	09.10.2019	26346.0	26342
111	20.11.2018	24465.6	24833	222	03.05.2019	26505.0	26530	333	10.10.2019	26496.7	26204
	MSE									72100.5	
	MAPE									0.6830	
	MPE									-0.2530	

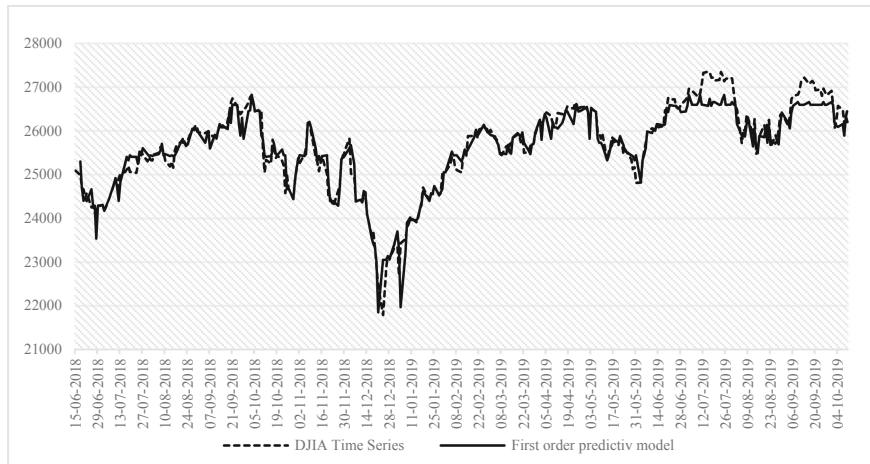


Fig. 7. 1st Order predictive model at nominal values.

6 Conclusion

On the example of the volatile DJIA time series, new method of fuzzification of historical data based on the use of the fuzzy inference system is proposed. Due to the redundancy of fuzzy interpretations of weakly structured data compiled according to the number of the DJIA index, clustering of fuzzy sets was carried out and appropriate set of corresponding evaluation criteria was formed. The application of this approach to fuzzification of historical data makes it possible to recreate the fuzzy analogue of the time series and, on its basis, apply one or another predictive model.

The article considers the predictive model based on internal relationships of the 1st order. Of course, one should also consider internal relationships of the 2nd and higher orders, which would provide additional resources for a qualitative improvement of the proposed approach. In this case, the results obtained could be compared with existing approaches, for example, with the predictive models of Q. Song and B. Chissom, N. Kumar et al. S. Chen, C. Cheng, J. Poulsen. But on account of the cumbersome calculations this is not possible.

References

1. Song, Q., Chissom, B.S.: Fuzzy time series and its models. *Fuzzy Sets Syst.* **54**, 269–277 (1993)
2. Kumar, N., et al.: Fuzzy time series forecasting of wheat production. *Int. J. Comput. Sci. Eng.* **2**(3), 635–640 (2010)
3. Chen, S.M.: Forecasting enrollments based on fuzzy time series. *Fuzzy Sets Syst.* **81**, 311–319 (1996)
4. Cheng, C.H., Chang, J.R., Yen, C.A.: Entropy-based and trapezoid fuzzification fuzzy time series approaches for forecasting IT project cost. *Technol. Forecast. Soc. Chang.* **73**, 524–542 (2006)

5. Poulsen, J.R.: Fuzzy time series forecasting—developing a new forecasting model based on high order fuzzy time series. AAUE: CIS 4, p 67 (2009)
6. Mardanov, M.J., Rzayev, R.R., Alizada, P.E.: About one approach to data fuzzification on the example of the Dow Jones index time series. Math. Mach. Syst. **1**, 108–115 (2021). (in Russian)
7. Dow Jones Industrial Index. <https://www.ru.tradingview.com/symbols/DJ-DJI/>. Last accessed 15 Nov 2022
8. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning. Inf. Sci. **8**(3), 199–249 (1975)
9. Ortiz-Arroyo, D., Poulsen, J.R.: A weighted fuzzy time series forecasting model. Indian J. Sci. Technol. **11**(27), 1–11 (2018)
10. Lewis, K.D.: Methods for forecasting economic indicators. Finance and statistics, Moscow (1986). (in Russian)



Traffic State Prediction of Perturbed and Non-perturbed Traffic Scenarios

Teck-Hou Teng¹(✉), George Rosario Jagadeesh¹, Thakkar Kunal¹,
and Chong Chee Chung²

¹ Data Analytics Strategic Technology Center (DA STC), Group Technology Office, ST Engineering Ltd., Singapore, Singapore
thteng@stengg.com

² Business Development Centre, Mobility Road, Singapore, Singapore

Abstract. Traffic state prediction is a regression problem of predicting traffic states such as travel speed, volume, occupancy and density. However, we observe that existing traffic state prediction models are often trained and tested on non-perturbed traffic scenarios. It lacks the robustness needed for handling perturbed traffic scenarios. This work addresses this gap in the stated problem statement using simulated data based on perturbed traffic scenarios. Through this work, we established the need to assess the performance of the regression models on the affected road segments rather than on the entire road network. In doing so, the performance of the regression models on the perturbed traffic scenarios has become apparent. In addition, we have confirmed that it helps to train the regression model using a mixture of traffic data from both types of traffic scenarios. For our experiments, we built a micro-simulation of a medium-sized traffic network based on the morning peak traffic scenario. We evaluated multiple regression models for multiple prediction horizons on the studied traffic scenarios. Our results based on the affected road segments reveal that Spatial-Temporal Graph Convolutional Network (ST-GCN) has the best mean absolute percentage error (MAPE) for all prediction horizons. Hence, it is established that ST-GCN is a more robust regression model for predicting traffic states under perturbed and non-perturbed traffic scenarios.

Keywords: Traffic prediction · Traffic perturbation · Supervised learning · Regression model

1 Introduction

Traffic state of road segments in a road network can be predicted to anticipate fluctuation of travel speed over time [13, 17, 19]. Apart from substantial environmental and economic values, traffic state prediction can promote better understanding of the underlying dynamics of urban traffic. It helps with the development of more efficacious transportation policies. To do that, various traffic prediction models have been developed [14, 18, 20–23]. It has enabled more fine-grained analysis of traffic patterns. However, the performance of many existing approaches is known to degrade significantly for longer

prediction horizons and perturbed traffic conditions. Special attention is given to the accuracy, scalability and robustness of traffic models used in such a context.

The performance of a traffic prediction module is a function of factors such as mapping data, traffic data, weather data and related traffic data. Mapping data comprises the detailed maps and related attributes of the road network. The traffic data from road segments where traffic states are forecasted need to be updated regularly. The traffic data comprises historical and real-time traffic information such as traffic volumes, travel time, travel speed and vehicle trajectories. Such factors can be obtained from a variety of data sources. It may originate from existing infrastructure or third-party data vendors. Weather conditions can impact traffic conditions in a tangible manner. The availability of historical, current and forecasted weather conditions can help in enhancing the prediction accuracy of the traffic states. This is because it enables the prediction model to be discriminative. Other related traffic data such as road works, traffic incidents, road closures due to planned or unplanned events can also help to enable the prediction model to be more discriminative.

Sufficient amount of a variety of high quality historical and real-time data is required for accurate forecast of traffic states. However, it is, more often than not, a luxury to have large variety and large volume of traffic data. Therefore, to address such a challenge, this work uses historical average travel speeds to predict average travel speeds of multiple road segments of a road network for fixed time intervals. The input to the prediction model is a speed matrix that is made up of historical average travel speeds at selected road segments gathered over multiple time steps. The output of the prediction model is a vector comprising the predicted average travel speed for the same set of road segments for specific prediction horizon. Mean absolute error (MAE) and mean absolute percentage error (MAPE) are the primary performance metrics for this prediction task.

Traffic scenario based on the road network of the Jurong Lake District (JLD) of Singapore is created for generating simulated traffic data to evaluate the efficacy of the traffic prediction models. The demand model of the traffic scenario is calibrated using traffic data for the morning peak period from 6 am to 9 am. The network model has 512 road segments and 48 signalized junctions. The calibrated JLD-based traffic scenario has a total traffic volume of 41,342 vehicles for the simulated 3-h period. Perturbation of the traffic scenario is introduced through random road incidents and road closures at selected locations within the road network. The prediction models are evaluated using non-perturbed as well as perturbed traffic scenarios. The prediction horizons are for 5, 10, 15 and 30 min from the moment of prediction. The average network-wide MAE for the perturbed traffic scenarios are 3.78, 3.81, 3.85 and 3.91 respectively. The averaged localized MAE for similar traffic scenarios are 8.52, 11.72, 13.69 and 16.28 respectively.

The rest of the paper is organized in the following manner. Section 2 provides a selection of the related work on the topic of traffic prediction in perturbed traffic scenarios. This is followed by the presentation of the traffic prediction models in Sect. 3. Details on the performance evaluation of the traffic prediction models are provided in Sect. 4. Finally, the summary and conclusion can be found in Sect. 5.

2 Related Work

Traffic state prediction and its variants are well-studied problems. Many related prior works predict average travel speed and average traffic volume on road segments for specific prediction horizon. Four broad approaches are known for traffic prediction [1]. They are statistical approach, conventional machine learning approach, deep learning-based approach [8, 9, 12, 14] and model-driven approach. Statistical approaches are ineffective among non-stationary traffic conditions. Most machine learning approaches are only suitable for short-term traffic prediction. Model-based approaches are good for long-term prediction with a strong assumption of model fidelity. Many of these approaches tend to overfit. They are not robust enough to deal with perturbed traffic conditions.

Using k-Nearest Neighbor (k-NN) with error feedback is observed to have better performance than recurrent neural network (RNN) and time-delay neural network (TDNN) for perturbed traffic conditions [8]. The error feedback has improved the prediction accuracy under those traffic conditions. Information capturing the input and feedback variables is known to be more important than the machine learning model for prediction accuracy. Another study on the impact of data smoothing and error feedback structures on the accuracy of short-term traffic prediction methods also agrees with [8] that k-NN gives the best performance when there is a sudden change of traffic patterns caused by abnormal traffic events [7]. Better performance is attributed to the ability to quickly detect pattern changes and the flexibility to match the best patterns with historical datasets.

The combined use of time series segmentation and clustering are investigated for its impact on the prediction accuracy of traffic conditions during incidents [14]. The proposed method aims to discover the resultant traffic pattern when there is incident. By training the prediction model using traffic data from the incident, the resultant prediction model outperforms selected traffic prediction models on benchmark datasets.

An ensemble approach integrating the gradient boosted regression trees (GBRT) and the least absolute shrinkage and selection operator (LASSO) was proposed for short-term traffic forecasting under normal and perturbed conditions [4]. Independently, GBRT performs better under normal traffic conditions while LASSO performs better under abnormal traffic conditions. Compared against the benchmark models, the combined GBRT and LASSO approach is promising and effective for short-term traffic flow prediction under various traffic conditions.

It is suggested that the problem of predicting short-term traffic flow on the freeway can also be addressed using online support vector regression (OL-SVR) [3]. Compared with Gaussian Maximum Likelihood (GML), Holt Exponential Smoothing and Artificial Neural Network, the study concluded that OL-SVR is superior at short-term traffic prediction under atypical conditions such as holidays and incidents. In addition, OL-SVR has the added advantage of having shorter runtime over the benchmark models.

Extreme traffic conditions are rare. Hence, it is challenging to build prediction model capable of good prediction accuracy for it. The Mixture Deep Long-Short Term Memory (LSTM) model is one such attempt at jointly modelling non-perturbed and perturbed traffic conditions [22]. By training it using suitable regularization, the resultant approach achieves 30–50% improvement over baselines. In addition, the model is inspected for interpretability.

Traffic prediction is considered a foundational technology of an advanced traffic management system (ATMS). Yet, many works are focused on building traffic prediction models for normal traffic conditions. The reluctance to address the issue of predicting for perturbed traffic conditions may have been due to the lack of sufficient relevant traffic data and suitable machine learning models. Despite such challenges, several works are known to have done that. This work applies on the collective wisdom of those works for specific traffic state prediction scenarios.

3 Traffic State Prediction Models

We designed our traffic prediction solution to be agnostic to the prediction model used for predicting the average travel speed. The use of CNN, LSTM and STGCN was investigated in an earlier work [17]. Building on it, this work enhances the portability and robustness of LSTM and STGCN for real world application. Subsequently, we found LSTM and STGCN to be the most promising approaches for addressing the traffic state prediction problem. The following sections shall present the essential details for these two prediction models.

3.1 Long Short-Term Memory (LSTM)

As a variant of recurrent neural network, LSTM is capable of learning long-term dependencies. The conventional error back-propagation through time approach for tuning the weights of the synaptic links of the neural network is prone to the problem of error blow up and vanishing gradient [15]. Having an architecture enforcing constant error flow through the internal states of special self-connected units, LSTM addresses these two error back-flow problems using an efficient gradient-based algorithm [11]. Though LSTM units allow gradient to flow unchanged, it can still suffer from the exploding gradient problem [2].

To make this paper self-contained, only essential details for understanding LSTM and for knowing how it can be effective for addressing the traffic prediction problem are included. The presentation of LSTM here will be made in two parts. The first part is on the architecture of LSTM. The second part is on the learning algorithm of LSTM.

3.1.1 Model Architecture

As illustrated using Fig. 1, a LSTM recurrent unit is composed of the cell state, hidden state, forget gate and input gate. The cell state is the key concept behind LSTM. It is illustrated by the horizontal line running through the top of the diagram seen in Fig. 1. The forget gate controls the passage of information using a multiplier having value between 0.0 and 1.0. The input gate is a sigmoidal layer that controls the information that need to be added to the cell state.

The main processes are to update the cell state and the hidden state. The process of updating the cell state has two steps. The first step is to multiply the results of the forget gate with the previous cell state. The second step is to add the information from the input gate to this result from the first step to get the latest cell state. The process of updating the

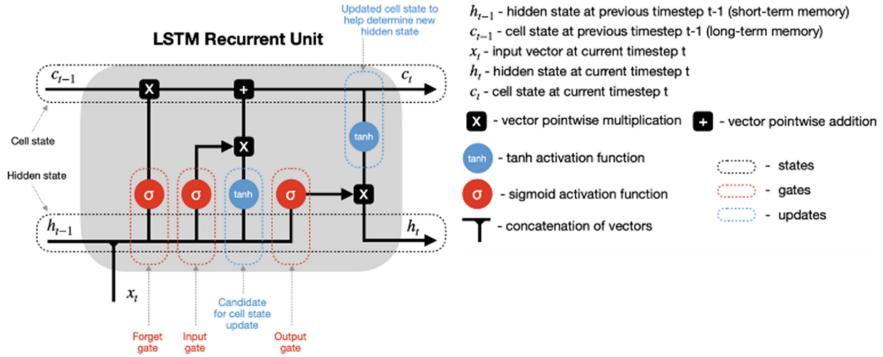


Fig. 1. An illustration of LSTM cell [6].

hidden state is the final step. The latest cell state is through the tanh activation function and multiplied by the results of the output gate. The latest cell state and hidden state are returned to the recurrent unit and the process repeats.

3.1.2 Learning Algorithm

LSTM uses a variant of the real-time recurrent learning (RTRL) which is outlined using Algorithm 1 (adopted from [10]).

Algorithm 1 Real-Time Recurrent Learning Algorithm

Parameters:

m = dimensionality of input space.

q = dimensionality of state space.

p = dimensionality of output space.

w_j = synaptic weight vector of neuron j , $j = 1, 2, \dots, q$.

Initialization:

1. Set the synaptic weights of LSTM to small values selected from uniform distribution.
2. Set the initial value of state vector $\mathbf{x}(0) = 0$.
3. Set $\Lambda_j(0) = 0$ for $j = 1, 2, \dots, q$.

Computation: compute for $n = 0, 1, 2, \dots$

$$\Lambda_j(n+1) = \Phi(n)[W_a(n)\Lambda_j(n) + U_j(n)] \quad (1)$$

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{C}\mathbf{x}(n) \quad (2)$$

$$\Delta W_j(n) = \eta C \Lambda_j(n) \mathbf{e}(n) \quad (3)$$

The definition of $\mathbf{x}(n)$, $\Lambda_j(n)$, $U_j(n)$ and $\Phi(n)$ follows below.

From Algorithm 1, $\mathbf{x}(n)$ is a q -length state vector given as below:

$$\mathbf{x}^T(n) = \left[\varphi(w_1^T \xi(n-1)), \dots, \varphi(w_j^T \xi(n-1)), \dots, \varphi(w_q^T \xi(n-1)) \right] \quad (4)$$

From Algorithm 1, $\Lambda_j(n)$ is a $(q \times (q + m + 1))$ matrix defined as the partial derivative of the state vector $\mathbf{x}(n)$ with respect to weight vector $\mathbf{W}_j(n)$:

$$\Lambda_j(n) = \frac{\partial \mathbf{x}(n)}{\partial \mathbf{W}_j(n)}, j = 1, 2, \dots, q \quad (5)$$

From Algorithm 1, $\mathbf{U}_j(n)$ is a $(q \times (q + m + 1))$ matrix whose rows are all zero except for the j^{th} row that is equal to the transpose of vector $\xi(n)$:

$$\mathbf{U}_j(n) = \begin{bmatrix} 0 \\ \xi^T(n) \\ 0 \end{bmatrix} \leftarrow j^{th} \text{ row}, j = 1, 2, \dots, q \quad (6)$$

From Algorithm 1, $\Phi(n)$ is a $(q \times q)$ diagonal matrix whose k^{th} diagonal element is the partial derivative of the activation function with respect to its argument evaluated at $(w_j^T \xi(ns))$:

$$\Phi(n) = \text{diag}\left(\varphi'(w_1^T \xi(n)), \dots, \varphi'(w_j^T \xi(n)), \dots, \varphi'(w_q^T \xi(n))\right) \quad (7)$$

The above mathematical expressions lead to the recursive equation $\Lambda_j(n+1)$ seen in Algorithm 1. Such a recursive equation describes the non-linear state dynamics of the real-time recurrent learning process. The mathematical expressions are interpreted and expanded into its native form so that it can be implemented as software routines.

3.2 Spatio-Temporal Graph Convolutional Network (ST-GCN)

ST-GCN is a type of graph neural network [16] for processing structured time series [21]. It learns hidden patterns from spatial-temporal graphs. Such data structure is applicable for traffic forecasting, driver maneuver anticipation and human action recognition [21]. It considers spatial dependency and temporal dependency at the same time. Specifically, ST-GCN composes a spatial-temporal block using a 1D convolutional layer and a spatial graph convolution layer [21]. It achieves parallelization over input with fewer parameters and faster training speed. Therefore, it can handle large networks efficiently.

Similarly, this paper is made self-contained by including only the essential details for understanding ST-GCN. It is for allowing the reader to know how it can be effective for addressing the traffic prediction problem. The presentation of ST-GCN here will be made in two parts. The first part is on the architecture of ST-GCN. The second part is on the building blocks of ST-GCN.

3.2.1 Model Architecture

The architecture of the ST-GCN is illustrated using Fig. 2. It comprises two spatio-temporal convolutional blocks (ST-Conv blocks) and a fully-connected output layer.

There are two temporal gated convolution layers and one spatial graph convolution layer to each ST-Conv block. The residual connection and bottleneck strategy are applied to each ST-Conv block. The spatial and temporal dependencies are explored coherently by the ST-Conv blocks through the uniform processing of input v_{t-M+1}, \dots, v_t .

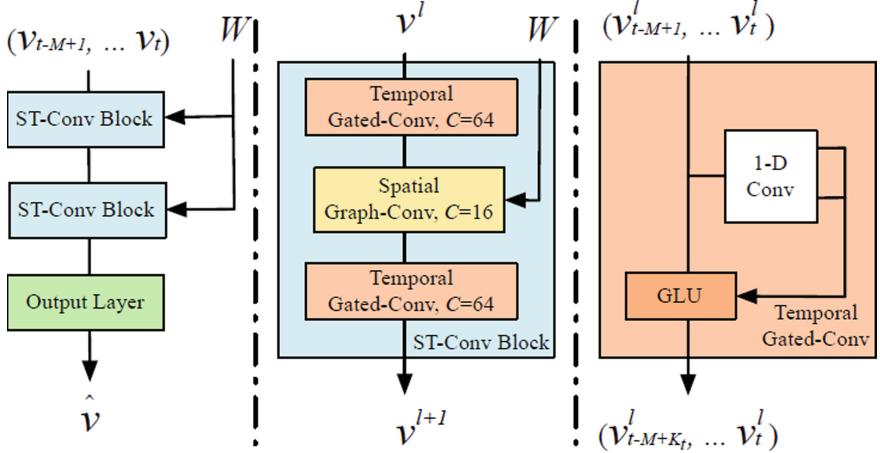


Fig. 2. An illustration of the ST-GCN architecture [21].

3.2.2 Building Blocks of STGCN

Separate algorithms are employed for extracting spatial and temporal features. Details of these algorithms can be seen in the following subsections:

Extraction of Spatial Features

Meaningful patterns and features in the space domain are extracted by applying graph convolution to the graph-structured data. Based on the *spectral graph convolution* [5], ST-GCN adopts the notion of graph convolution operator $*\mathcal{G}$. A graph signal $x \in \mathbb{R}^n$ filtered by kernel Θ and multiplied with graph Fourier transform $U^T x$ as follows:

$$\Theta * \mathcal{G} x = U \Theta(\Lambda) U^T x \quad (8)$$

where graph Fourier basis $U \in \mathbb{R}^{n \times n}$ is the matrix of eigenvectors of the normalized graph Laplacian L ; $\Lambda \in \mathbb{R}^{n \times n}$ is the diagonal matrix of eigenvalues of L and filter $\Theta(\Lambda)$ is a diagonal matrix.

Due to the multiplication with graph Fourier basis, the computation of Kernel Θ in graph convolution can be expensive. Through Chebyshev polynomial approximation and then 1st-order approximation, graph convolution can be simplified and alternatively expressed as follows:

$$\Theta * \mathcal{G} x = \theta\left(\tilde{D}^{-\frac{1}{2}} \tilde{W} \tilde{D}^{-\frac{1}{2}}\right) x \quad (9)$$

where θ is a kernel parameter, $\tilde{W} = W + I_n$ and $\tilde{D}_{ii} = \sum_j \tilde{W}_{ij}$

Extraction of Temporal Features

The temporal dynamic behaviors of traffic flows are captured using the entire convolutional structures on time axis. The resultant temporal gated convolution is defined as follows:

$$\Gamma_{*\mathcal{T}} Y = P \odot \sigma(Q) \in \mathbb{R}^{(M-K_t+1) \times C_o} \quad (10)$$

where P, Q are input of gates in GLU respectively; \odot denotes the element-wise Hadamard product; $\sigma(Q)$ is a sigmoid gate controlling which input P of the current states are relevant for discovering compositional structure and dynamic variances in time series.

3.2.3 Fusing of Extracted Spatial and Temporal Features

Spatio-temporal convolutional block (ST-Conv block) jointly processes graph-structured time series. As seen in Fig. 2, the spatial layer bridges two temporal layers to achieve fast spatial-state propagation from graph convolution through temporal convolution.

For input $v^l \in \mathbb{R}^{M \times n \times C^l}$ of block l , the output $v^{l+1} \in \mathbb{R}^{(M-2(K_t-1)) \times n \times C^{l+1}}$ is defined as follows:

$$v^{l+1} = \Gamma_1^l \text{ReLU}\left(\Theta^l * \mathcal{G}\left(\Gamma_0^l * \mathcal{T} v^l\right)\right) \quad (11)$$

where, Γ_0^l, Γ_1^l are the upper and lower temporal kernel of block l ; Θ^l is the spectral kernel of graph convolution; $\text{ReLU}(\cdot)$ denotes the rectified linear units function. Using the model final output $Z \in \mathbb{R}^{n \times c}$, the predicted speed for n node is obtained by applying a linear transformation across c -channels as follows:

$$\hat{v} = Zw + b \quad (12)$$

where, $w \in \mathbb{R}^c$ is a weight vector and b is a bias.

The L2 loss function of ST-GCN for traffic prediction can be written as:

$$L(\hat{v}; W_\theta) = \sum_t \|\hat{v}(v_{t-M+1}, \dots, v_t, W_\theta) - v_{t+1}\|^2 \quad (13)$$

where, W_θ are all trainable parameters in the model; v_{t+1} is the ground truth and $\hat{v}(\cdot)$ denotes the model's prediction.

4 Performance Evaluation

4.1 Simulation-Based Data Generation

We use a micro-simulator for simulating traffic scenarios in a medium-size road network comprising 512 links and in Singapore. The road network, shown in Fig. 3, spans an area of roughly 5×5 km comprising freeways and non-freeway roads. The traffic demand models used in the micro-simulations are developed and calibrated based on real traffic demand data and other traffic metrics obtained from a major traffic data provider.

Traffic corresponding to a 3-h period on a typical weekday morning is simulated. To account for the random nature of road traffic, each simulation run uses a different random seed. Traffic incidents are simulated by impeding the traffic flow along one or more lanes on a randomly chosen freeway link for a random duration ranging between 30 min and 45 min.



Fig. 3. The road network used for the traffic simulation.

The traffic simulator outputs high-resolution vehicle trajectory data for estimating the average travel speed at specific road segments at 5 min intervals. It is common knowledge that it is impossible to obtain trajectory data from 100% of the vehicle population at any time. To simulate such a phenomenon, a sample of the vehicle population is made. In this work, the sampling proportion is set at 0.7.

The output of the data processing module is a speed matrix containing the average travel speeds of all the links at 5-min time intervals. The resultant speed matrix containing 512 links is used as input to the traffic state prediction models. The following three datasets, in the form of speed matrices are used by the traffic prediction models in this study.

- **Normal Dataset:** This dataset contains speed data from 30 days of simulation without any traffic incidents. It is one of the two datasets used as training data.
- **Perturbed Dataset:** This dataset contains speed data from 10 days with one incident present on each day. It is used as test data. The links whose speeds are affected by each incident in the test data are manually identified.
- **Mixed Dataset:** This dataset contains speed data from 15 days without any incidents and 15 days with one incident present on each day. It is the other dataset used as training data.

Figure 4 shows an example of speed changes caused by an incident on a link on which it occurs and another link about 1 km upstream of the incident site. The incident in this example occurs in the 91st min of the simulation and lasts till the 131st min.

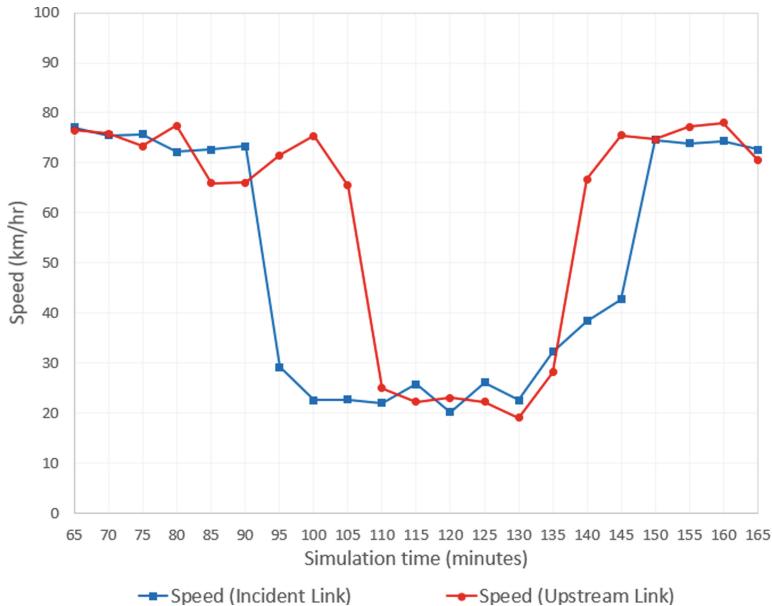


Fig. 4. An example of speed changes caused by a simulated traffic incident.

4.2 Experimental Settings

In our experiments, we use the average travel speeds from the last 30 min (i.e., 6 time steps) to predict the speeds for up to the next 30 min for all 512 links. To be specific, we consider the predictions for the next 5, 10, 15 and 30 min. It needs to be noted that the models are trained to predict only for the next time step (i.e., 5 min ahead) and predictions for k time steps ahead are generated recursively through k steps. Each input to the traffic prediction models is a 6×512 matrix that stores the speeds for the last 6 time steps for each of the 512 links. The output of the models is a 512-element vector that contains the predicted speeds for the next time step for all the links.

The efficacy of the LSTM and STGCN models are investigated in this work. The LSTM model contains 128 units and three fully connected layers. It is trained for 150 epochs with a batch size of 30 and a learning rate of 0.001. In the STGCN model, the channel sizes of the temporal gated convolution layer and the spatial graph convolution layer are set as 64 and 32, respectively. The graph convolution kernel size and temporal convolution kernel size are both set to 2. The STGCN model is trained for 150 epochs with a batch size of 50 and a learning rate of 0.001.

The training datasets used in our experiments contain 30 days of simulation data each. During each experiment, 25 of the 30 days are randomly chosen for training and the rest are used for testing the calibrated models. We use MAE and MAPE as the performance metrics for evaluating the traffic state prediction models.

4.3 Results and Analysis

Table 1 shows the test performance of the LSTM and STGCN models in terms of the average MAE for all the links in simulated road network for the duration of the traffic incidents in the test data (i.e., the Perturbed dataset). Figure 5 illustrates the corresponding MAPE values. It helps to note that each traffic incident in test data affects the speeds of only a small number of links. Most of the links in the network experience normal traffic flow. Therefore, the results presented on the localized evaluation seen in Table 2 and Fig. 6 are more representative of the robustness of the traffic state prediction models on perturbed traffic scenarios.

Table 1. Network-wide performance comparison (MAE).

Prediction model	Training dataset	Prediction interval (minutes)			
		5	10	15	30
LSTM	Normal	3.78	3.81	3.85	3.91
	Mixed	3.78	3.82	3.86	3.93
STGCN	Normal	3.97	4.68	4.88	4.81
	Mixed	3.36	3.90	4.28	4.96

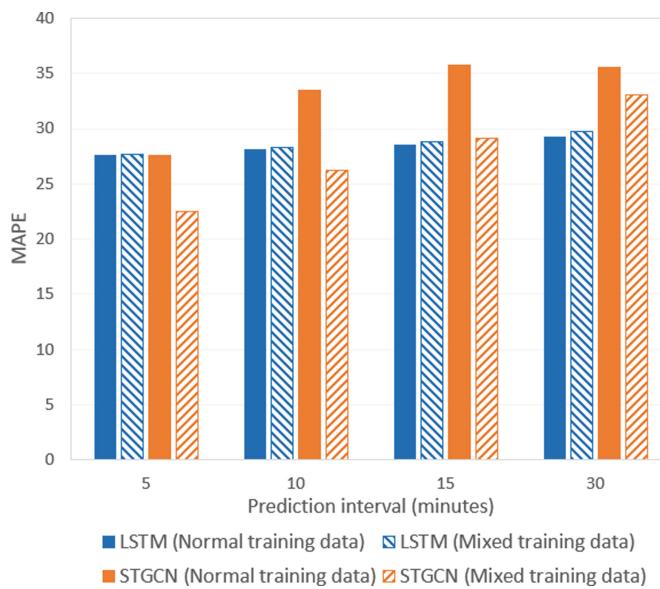


Fig. 5. Network-wide comparison of MAPE.

To specifically examine the prediction performance of the models in perturbed traffic conditions, we perform a localized evaluation in which we consider the predictions for

only the links affected by each traffic incident in the test data for the duration of the incident. Table 2 shows the MAE from the localized evaluation of the LSTM and STGCN models for the incident-affected links.

It can be seen from Figs. 5 and 6 that the use of the Mixed dataset, which includes perturbed traffic data, for training produces a significant performance improvement for STGCN compared to the use of Normal dataset. However, no such improvement is observed for LSTM. It could be hypothesized that the presence of perturbation in the training data allows STGCN to better model the spatial dependencies leading to better performance.

Table 2. Localized performance comparison (MAE) for the only the incident-affected links.

Prediction model	Training dataset	Prediction interval (minutes)			
		5	10	15	30
LSTM	Normal	16.27	16.19	16.12	16.06
	Mixed	16.15	16.08	16.03	16.05
STGCN	Normal	15.14	16.99	17.68	17.28
	Mixed	8.52	11.72	13.69	16.28

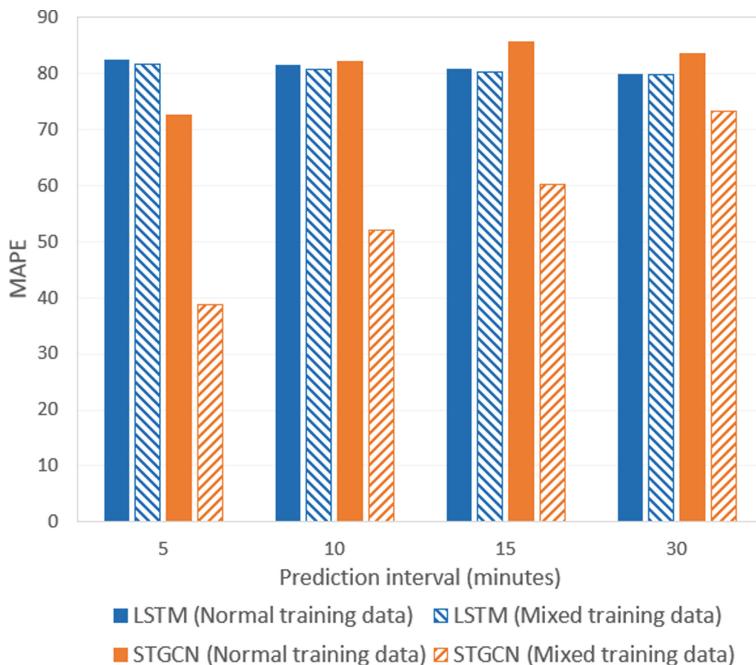


Fig. 6. Localized comparison of MAPE for the incident-affected links.

The results from the network-wide performance comparison indicate that LSTM outperforms STGCN in large part of the non-perturbed traffic scenario with the exception of a very short prediction interval. This is because LSTM is known to be highly effective for assimilating relatively long temporal dependencies. The better performance of LSTM observed in Fig. 5 may be attributed to the non-perturbed traffic conditions at most parts of the simulated road network.

The results from the localized comparison of prediction performance for the perturbed links indicate that STGCN performs substantially better than LSTM in those traffic conditions. The robustness of STGCN to perturbations is more evident for short prediction intervals. In contrast, the MAPE of LSTM remains high and largely unchanged for all prediction intervals. It is apparent that STGCN's ability to capture spatial dependencies yields better results in the case of links experiencing abnormal traffic speeds due to incidents. A model such as LSTM that relies only on temporal dependencies can be ineffective at handling deviation from non-perturbed temporal patterns.

Figure 7 shows the 5-min-ahead predictions made by LSTM and STGCN for a link severely affected by an incident downstream. The speeds predicted by STGCN closely follow the ground truth speed with a time lag equal to the prediction interval of 5 min. On the other hand, the severe drop in the link speed caused by the incident is not tracked by the predictions made by LSTM. The predicted average travel speeds remain high throughout the duration of the incident. The speeds predicted by LSTM are close to the typical speeds experienced by the link in a non-perturbed traffic scenario. Figure 8 shows the 30-min-ahead predictions for the same incident-affected link. In this case too, the predictions by STGCN tracks the abrupt drop in the average travel speed caused by the incident but with a longer time lag equal to the prediction interval of 30 min. The reason for the observed phenomenon is that only the historical average travel speed is the input to the prediction models at the moment of prediction. Without any indication of the perturbation in the input data, it is impossible for the prediction models to know that there is a perturbation impacting the travel speeds in the next time step of the prediction horizon. Hence, there is the lag for the same amount of time of the prediction horizon.

An interesting observation made in Fig. 8 that STGCN is able to predict the recovery from the incident with a time lag shorter than the prediction interval. This is an interesting phenomenon that must be studied in greater detail before any conclusion can be made about it.

The ability of traffic prediction models that leverage spatial and temporal dependencies to better adapt to abrupt speed changes has been previously highlighted by other researchers [21]. However, they considered abrupt speed changes that regularly occur during rush hours. The temporal dynamics of such regular rush-hour speed changes can generally be learned from the training data even by models that rely solely on temporal characteristics. On the other hand, it is a lot more challenging for traffic prediction models to adapt to speed changes caused by unplanned and unanticipated incidents. The presented results indicate that models such as STGCN are more capable of adapting to incident-induced traffic perturbations compared to purely temporal models.

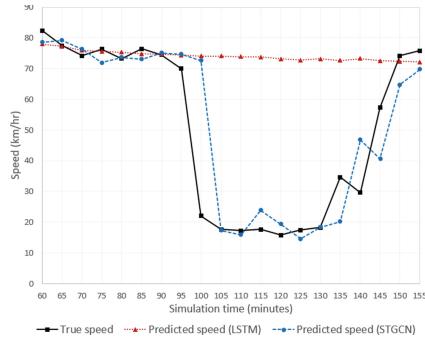


Fig. 7. Five-minutes-ahead prediction for an incident-affected link.

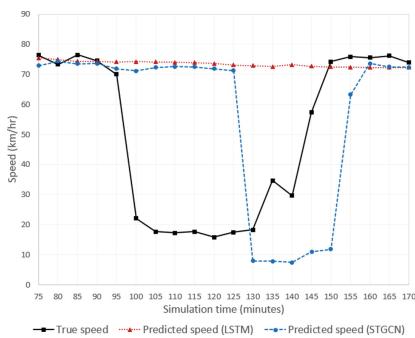


Fig. 8. Thirty-minutes-ahead prediction for an incident-affected link.

5 Summary and Conclusion

A significant amount of time and effort had been devoted towards ascertaining the robustness of two regression models under perturbed and non-perturbed traffic conditions. Traffic data from simulated traffic conditions from a micro-simulation based on a medium-sized road network is used. Several experiments were performed to investigate the robustness issues of the regression models. Our experience from this work informed us that it is much more targeted to evaluate the performance of the regression model on the segments affected by the incident. We have also learned that it helps to train the regression models on a mixture of perturbed and non-perturbed traffic scenarios. Our experiment results reveal to us that ST-GCN to be the most robust regression model for perturbed and non-perturbed traffic scenarios.

This work can be continued in several ways. One of the ways is to push for adoption by our business unit in one of their commercial projects. It can be delivered as a part of an advanced traffic management system (ATMS) for the land transport authority. In addition, more recent and advanced traffic prediction models shall be explored for addressing the traffic state prediction problem. As pointed out in [19], there remains strong research interest for the traffic prediction problem. From our own survey, it is found

that there is still very little work investigating the effect of perturbed traffic scenarios on the performance of the prediction models. Also there are many possible future directions [19] to advance the state-of-the-art of traffic prediction models.

Acknowledgment. This work is a continuation of the research collaboration between ST Engineering and National University of Singapore (NUS). The authors are grateful for the valuable contributions made by members of the research team responsible for the research deliverables on which this work is based on.

References

1. Barros, J., Araujo, M., Rossetti, R.J.: Short-term real-time traffic prediction methods: a survey. In: Proceedings of the models and technologies for intelligent transportation systems, pp. 132–139. (2015)
2. Calin, O.: Deep learning architectures: A mathematical approach, 1st edn. ISBN 3030367207. Springer, (2020)
3. Castro-Neto, M., Jeong, Y.-S., Jeong, M.-K., Han, L.D.: Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert Syst. Appl.* **36**(3), 6164–6173 (2009)
4. Chen, X., Zhang, S., Li, L.: Multi-model ensemble for short-term traffic flow prediction under normal and abnormal conditions. *IET Intel. Transport Syst.* **13**(2), 260–268 (2019)
5. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in neural information processing systems, pp. 3844–3852, (2016)
6. Dobilas, S.: LSTM recurrent neural networks—how to teach a network to remember the past, Neural networks, towards data science. Accessed February 2022
7. Guo, F., Krishnan, R., Polak, J.W.: A novel three-stage framework for short-term travel time prediction under normal and abnormal traffic conditions. In: Transportation research board annual meeting. (2014)
8. Guo, F., Polak, J.W., Krishnan, R.: Comparison of modelling approaches for short term traffic prediction under normal and abnormal conditions. In: Proceedings of 13th International IEEE conference on intelligent transportation systems, pp. 1209–1214. IEEE, (2010)
9. Guo, F., Polak, J.W., Krishnan, R.: Predictor fusion for short-term traffic forecasting. *Transp. Res. Part C: Emerg. Technol.* **92**, 90–100 (2018)
10. Simon, H.: Neural networks—A comprehensive foundation, 2nd Edn. Prentice Hall, (1999)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **8**(8), 1735–1780 (1997)
12. Salamanis, A., Margaritis, G., Kehagias, D.D., Matzoulas, G., Tzovaras, D.: Identifying patterns under both normal and abnormal traffic conditions for short-term traffic prediction. *Transp. Res. Procedia*, 665–674 (2017)
13. Sun, S., Chen, J., Sun, J.: Traffic congestion prediction based on GPS trajectory data. *Int. J. Distrib. Sens. Netw.*, (2019)
14. Tedjopurnomo, D.A., Bao, Z., Zheng, B., Choudhury, F.M., Qin, A.K.: A survey on modern deep neural network for traffic prediction: trends, methods and challenges. *IEEE Trans. Knowl. Data Eng.*, **34**(4), 1–20 (2022)
15. Werbos, P.J.: Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, (1988)

16. Zonghan, W., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **32**(1), 4–24 (2020)
17. Yang, L., Chattopadhyay, R., Teng, T.H., Yang, X.: Performance estimation of machine learning models in predicting short-term travel speeds using microscopic traffic simulation data. In: Proceedings of the 4th international symposium on multimodal transportation, pp. 1–14. (2021)
18. Yao, H., Tang, X., Wei, H., Zheng, G., Li, Z.: Revisiting spatial-temporal similarity: a deep learning framework for traffic prediction. In: The Proceedings of the Thirty-Third AAAI conference on artificial intelligence (AAAI-19), pp. 5668–5675. (2019)
19. Yin, X., Genze, W., Wei, J., Shen, Y., Qi, H., Yin, B.: Deep learning on traffic prediction: methods, analysis and future directions. *IEEE Trans. Intell. Transp. Syst.* **23**(6), 4927–4943 (2021)
20. Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., Yin, B.: Deep learning on traffic prediction: methods, analysis, and future directions. *IEEE Trans. Intell. Transp. Syst.*, **23**, 4927–4943 (2022)
21. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: Proceedings of IJCAI, pp. 3634–3640. (2018)
22. Yu, R., Li, Y., Shahabi, C., Demiryurek, U., Liu, Y.: Deep learning: A generic approach for extreme condition traffic forecasting. In: Proceedings of the 2017 SIAM international conference on data mining, pp. 777–785. Society for Industrial and Applied Mathematics, (2017)
23. Zheng, C., Fan, X., Wang, C., Qi, J.: GMAN: A graph multi-attention network for traffic prediction. In: Proceedings of the 34th AAAI conference on artificial intelligence, pp. 1234–1241, (2020)



“Seeing Sound”: Audio Classification Using the Wigner-Ville Distribution and Convolutional Neural Networks

Christonasis Antonios Marios¹✉, Stef van Eijndhoven¹, and Peter Duin²

¹ Engineering Doctorate Data Science, Technical University of Eindhoven (TU/E), Eindhoven, The Netherlands

antwnchris@gmail.com

² Dutch National Police, Eindhoven, The Netherlands

<https://www.linkedin.com/in/amchristonasis/>

Abstract. With big data becoming increasingly available, IoT hardware becoming widely adopted, and AI capabilities becoming more powerful, organizations are continuously investing in sensing. Data coming from sensor networks are currently combined with sensor fusion and AI algorithms to drive innovation in fields such as self-driving cars. Data from these sensors can be utilized in numerous use cases, including alerts in safety systems of urban settings, for events such as gun shots and explosions. Moreover, diverse types of sensors, such as sound sensors, can be utilized in low-light conditions or at locations where a camera is not available. This paper investigates the potential of the utilization of sound-sensor data in an urban context. Technically, we propose a novel approach of classifying sound data using the Wigner-Ville distribution and Convolutional Neural Networks. The Wigner-Ville distribution has not been considered in similar applications in literature. In this paper, we report on the performance of the approach on open-source datasets. The concept and work presented is based on my doctoral thesis, which was performed as part of the Engineering Doctorate program in Data Science at the University of Eindhoven, in collaboration with the Dutch National Police. Additional work on real-world datasets was performed during the thesis, which is not presented here due to confidentiality.

Keywords: Audio classification · Sound classification · Wigner-Ville distribution · Spectrogram · Convolutional neural networks · Deep learning · Machine learning · Time-Frequency analysis · Time-Frequency distributions · VGG · Imagenet · Audioset

1 Introduction

In the past years, Deep Learning has made enormous steps in Image Recognition. Open-source image datasets are abundant and architectures like Convolutional Neural Networks have been used to achieve performance that can even surpass human accuracy [1].

However, the same is not the case for sound-sensor data. While sensor networks, IoT hardware, and 5G are becoming increasingly part of our urban settings, research on the application of AI and sensor fusion on such settings has been more scarce. Open-source datasets are much more limited and techniques to classify this data are still in research phase. Datasets like AudioSet [2] and techniques using the log-mel spectrogram and Convolutional Neural Networks [3] are currently the state of the art. That being said, there is still a lot of untapped potential, both in terms of research, as well as real-world applications.

In this paper, we provide a collection of open-source sound datasets that we encountered in our research, some of which proved useful in assessing the performance of our approach. Moreover, we propose a novel approach to classify sound clips, using the Wigner-Ville time-frequency distribution and Convolutional Neural Networks, that we have not seen anywhere else in the literature. The advantage of the Wigner-Ville distribution compared to log-mel spectrograms are the more detailed resolution it can provide in the time-frequency spectrum and a more accurate estimate of the instantaneous frequency [4].

Following, we present relevant open-source datasets, a brief literature review, an introduction to time-frequency distributions and the Wigner-Ville distribution, present our technical approach, report results on several datasets, provide some conclusions, and finally lead into possible future work.

2 Open-Source Datasets

A vital and challenging step of this research was the identification and collection of relevant open-source datasets.

It was interesting to observe that open-source datasets in this area of research were limited, especially when compared to many available open datasets containing structured image data. Despite this lack of public datasets containing acoustical sound signals, we were able to identify open-source datasets that could be used for training sound classifiers. The most notable are presented as follows.

2.1 AudioSet

AudioSet [2] is open sourced by Google Research. It contains around two million audio clips with a duration of 10 s; they are fetched from YouTube videos and labelled semi-automatically. The dataset is weakly labeled, meaning that we only know the presence of a sound class in each sound clip, but we do not know exactly when during the ten seconds the sound of that class is apparent. Finally, the dataset is suited for a multi-label classifier, since more than one sound can be present in each sound clip, whereas for this research we aimed to design a multi-class classifier [5].

From the two million sound clips, two structured subsets were made available by Google Research for training and validation. The training set contains around 20,000 clips and the validation set around 15,000 clips. The limitation with AudioSet is that Google Research provides only the 128-dimensional feature vector they computed for each clip based on log-mel spectrograms [6]. For that reason, someone must scrape the

raw audio clips from YouTube, if they want to apply a different transformation to the sound signal, as was the case in this research. That is a very time-consuming process and, even more, several YouTube videos from the original datasets have been removed by YouTube. As a result, there is a small reduction in the number of audio clips someone can scrape, compared to the original datasets.

2.2 UrbanSound8K

The UrbanSound8K dataset [7] is open sourced by researchers from the New York University. The clips were scraped from the freesound.org website and were labelled manually. The dataset contains around 8000 clips with a duration ranging from 4 to 30 s. The clips belong to 10 classes corresponding to common urban sounds. The dataset is provided in the form of raw audio clips. In addition, each audio clip contains one distinct sound, which makes it suitable for training multi-class classifiers.

2.3 Urban-SED

The Urban-SED dataset [8] is suitable for building sound segmentation algorithms. It contains ten thousand ten-second clips which are created as follows. Background noise combined with smaller audio clips from the UrbanSound8K dataset is present during each 10-s clip. Annotations for the sound classes and their start and end times are available.

2.4 Environmental Sound Classification-50 (ESC-50) Dataset

The ESC-50 dataset [9] contains four-second clips from fifty different classes of environmental sounds, ranging from animal, to human, to machine sounds. Each class features forty available clips. In total, the dataset contains 20000 audio clips. This dataset has been used for benchmarking many different sound classification algorithms. It is useful for creating multi-class classifiers in the context of environment understanding.

2.5 Montevideo Audio Visual Dataset (MAVD-Traffic)

MAVD-Traffic [10] is another dataset that is unique and was recently open-sourced. The dataset comes from the Montevideo University in Uruguay. The creators recorded both audio and video from different locations near roads in Montevideo city. They also provided annotations for the different vehicles that pass by with their start and end times. The annotations are provided at different levels based on a taxonomy they defined. In total, the audio and video recordings last around four hours.

The downside of the Montevideo recordings is that cars are passing by most of the time, whereas other vehicles pass by with a much lower frequency. The result is a very unbalanced dataset that is also quite noisy in terms of traffic noise. Furthermore, multiple vehicles can pass by at the same time, which results in sounds getting mixed up. Finally, the annotations are weak, in the sense that they are not detailed as to exactly when each individual vehicle passes by the recorders. That being said it is a unique dataset and very close to a real-life scenario.

2.6 Military Vehicle Dataset

Last but not least, the military vehicle dataset was collected in November 2001 in the city of Twenty-Nine Palms in California [11]. The authors collected data from sound, seismic, and infrared sensors, while two military vehicles were driving around an area of $900 \times 300 \text{ m}^2$, where a wireless sensor network of the abovementioned sensors was deployed. The military vehicles were a dragon wagon (209 entries) and an assault amphibian vehicle (180 entries). All the recordings ranged from 1.5 s to around 38 s with an average of around 13 s. The acoustic data was recorded at a rate of 4960 Hz. Each sound recording was annotated with the name of the vehicle that passed by the sensor node.

3 Literature Review

Approaches that are currently considered state-of-the-art follow the below-mentioned general classification strategy:

1. Transformation of the sound signals to images (two-dimensional arrays) with the use of time-frequency analysis techniques, such as the spectrogram and the mel-frequency cepstral coefficients (MFCCs).
2. Classification with the use of image classification techniques. The state of the art here is neural network architectures that make use of convolutional layers.

Several research papers have been published in the past few years in which approaches similar to the abovementioned strategy are proposed. For the case of the current research, the most influential papers were the AudioSet [2] and VGGish [3] papers from Google Research. In the AudioSet paper, the authors describe the creation of a generic dataset that can become an equivalent to ImageNet for sound research. More details about AudioSet can be found in the previous section. In the technical approach part, the VGGish paper was the most influential in our approach. In this paper, the authors propose a neural network architecture based on convolutional layers that is inspired by the popular VGG architecture [12] for image classification.

One shortcoming of approaches in literature is that almost none consider the Wigner-Ville distribution as a time-frequency distribution, despite its enhanced resolution. In this paper, we examine its feasibility and prove that it is a perfectly viable time-frequency distribution to be used.

4 Time-Frequency Distributions and the Wigner-Ville Distribution

Time-Frequency Distributions are a very useful tool to perform analyses of signals. They provide a way to analyze signals both at the temporal and frequency domain at the same time. Essentially, they show the dominant frequencies in each signal as a function of time.

There exist several time-frequency distributions that have been researched in the context of many interesting applications. The most notable ones include:

- The spectrogram and its variations, such as the mel-spectrogram.

- The mel-frequency cepstral coefficients.
- The Wigner-Ville distribution.

The Wigner-Ville distribution essentially computes the Fourier transform of the signal’s Ambiguity Function (AF), where:

$$AF(\tau; t) = x(t + \tau/2)x^*(t - \tau/2) \quad (1)$$

The AF is a general representation of the signal’s autocorrelation function. An advantage of the Wigner-Ville distribution is that it does not suffer from leakage effects, in contrast to the spectrogram. It is a quadratic transform and not a linear one, so that it creates cross terms when the signal is comprised of multiple dominant frequency components. The issue of cross terms is solved with an extended form of the Wigner-Ville distribution, the pseudo Wigner-Ville distribution. The latter is the one we used in this research [13].

The Wigner-Ville distribution was chosen for three reasons. First, it provides more details in the time-frequency domain than the mainstream techniques like the spectrogram. Second, it was interesting to investigate its potential use for sound classification, since our literature research showed that most approaches neglected the Wigner-Ville distribution in favor for the spectrogram and MFCCs. Third, there exist papers that showcase the potential of the distribution for many interesting use cases. Some interesting examples include arrhythmia detection [14], classification of bird-song syllables [15], image segmentation [16], and seismic applications [17].

5 Technical Approach

Based on the motivations and ideas described above, we formulated a processing plan that is comprised of the following steps:

1. Averaging the sound signals in case more than one sources of sound are available.
2. Filtering and subsequently down-sampling the signals using frequency filters. This step helps in focusing the analysis on the most relevant frequency range and reduces the computational time and power needed.
3. Transforming the signal to its analytical format, using the Hilbert transform. This step helped in removing the aliasing of frequencies that would be present without performing it.
4. Transforming the signals to one-channel images in the time-frequency domain using the pseudo Wigner-Ville distribution.
5. Classification of the Wigner-Ville arrays with the use of a convolutional neural network of our own design that makes use of single-channel convolutions, motivated by the classification of grayscale images.

6 Results

The results in this section are divided according to what dataset was used for training and testing the classification model, as well as the associated use case.

6.1 UrbanSound8K

We built a classifier for the sounds present in the UrbanSound8K dataset. As mentioned in the previous chapter, the UrbanSound8K is a dataset open-sourced by the New York University, which contains around 8700 audio clips from 10 different categories of urban sounds. Most of the sound clips have a duration of about 4 s. Interestingly, the researchers mention in their paper [7] that, according to their findings, a duration of 4 s is enough to classify sound excerpts. In Fig. 1, the number of audio clips available per class is depicted. From the figure, we see that most classes comprise more than 1000 sound samples, with siren, car horn, and gunshot being the only ones with a number lower than 1000, at a minimum of around 390 sound clips for gun shot.

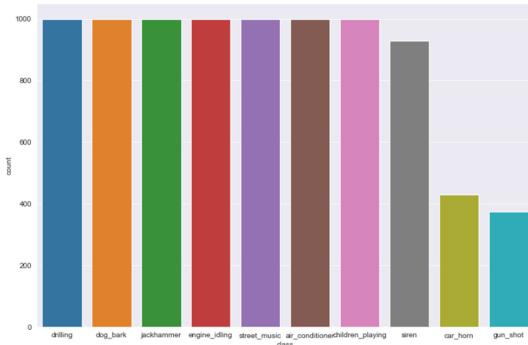


Fig. 1. Number of sound clips per class in the UrbanSound8K dataset.

For this dataset we used the preprocessing steps as described in the Technical Approach section. Regarding the specifics of the implementation, for the down-sampling, following an approach of trial and error, we chose a sampling rate of 4 kHz, essentially focusing on the frequencies up to 2 kHz. Using the technique of signal down-sampling, we also reduced the computation time of the Wigner-Ville distribution. Finally, after we transformed the sound signal to the Wigner-Ville distribution representation, we down-sized the 2D image arrays to a dimension of 300x300 pixels using interpolation. The reason for that was that the original dimension of the arrays proved to be computationally intensive to serve as input for the Neural Network.

A successful architecture for the Neural Network is depicted in Fig. 2. As described in the Technical Approach section, the architecture is motivated by the VGG architecture for image classification; hence, it makes use of Convolutional and MaxPooling layers for feature learning and a fully-connected part for the final classification, as well. Moreover, incremental experimentation resulted in optimal hyperparameters for numbers of layers, filter kernels, depth of layers, and weights of dropout layers that are used to reduce overfitting. This experimentation started with an architecture that is documented to be successful in literature and gradually changed according to the results we observed during our experiments.

Examples of the arrays of each class that served as input for the training of the neural network are shown in Fig. 3. The distinctive patterns of the different classes of sounds are

```

Net(
    (conv1): Conv2d(1, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (conv2): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (conv3): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (fc1): Linear(in_features=87616, out_features=500, bias=True)
    (fc2): Linear(in_features=500, out_features=10, bias=True)
    (dropout): Dropout(p=0.25, inplace=False)
)

```

Fig. 2. A neural network architecture used for the classification of the UrbanSound8K dataset.

obvious to the naked eye. A promising initial signal is that transforming to Wigner Ville distribution and performing classification based on that can be a successful strategy.

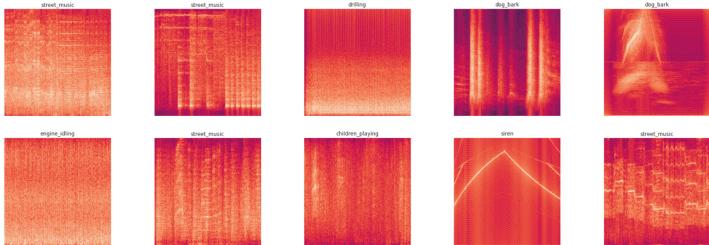


Fig. 3. Preprocessed Wigner-Ville arrays that serve as input to the neural network architecture—the different patterns of different classes of sound are evident with a naked eye—the most prominent frequencies for each audio clip can be seen in yellow.

The data was split in an 80/20 ratio, thus 80% of the data for training and 20% of the data for testing. The results of the validation on the test set are shown in Fig. 4. Most of the sound excerpts were classified correctly by the Neural Network. It is interesting to note that classes such as drilling and jackhammer showed misclassifications, which makes sense, since their time-frequency distributions are very similar.

In Fig. 5, the accuracy metrics for all the classes are shown. Overall, the neural network achieves an accuracy of 76%, a successful result in a 10-class random-guessing baseline of 10%.

Finally, we experimented with using the trained model to perform inference on an artificially-created streaming sound. The sound begins with children playing, continues to a dog barking, and finally returns to children playing. The signal is divided into 4-s overlapping windows with a stride of 1 s (first row of images in Fig. 6), then transformed individually to the Wigner-Ville distribution (second row of images in Fig. 6), which serve as input to the trained neural network that performs the inference successfully, as depicted in the third row of images in Fig. 6.

6.2 Military Vehicle Classification

The goal of the classification of the two military vehicles in the military vehicle dataset was to create a proof of concept for classifying sounds of the engines coming from different types of vehicles. We split the data into an 80/20 train-test split, resulting in 311 entries in the training and 78 entries in the test set.

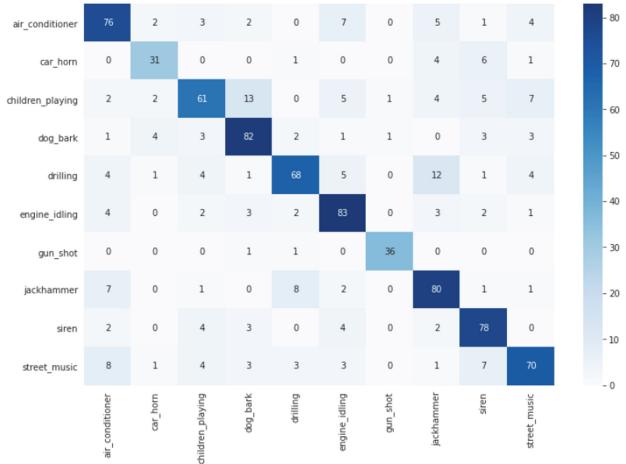


Fig. 4. Confusion matrix for the test set of an 80/20 split of the data—the majority of classes is correctly classified by the neural network.

	precision	recall	f1-score	support
0	0.73	0.76	0.75	100
1	0.76	0.72	0.74	43
2	0.74	0.61	0.67	100
3	0.76	0.82	0.79	100
4	0.80	0.68	0.74	100
5	0.75	0.83	0.79	100
6	0.95	0.95	0.95	38
7	0.72	0.80	0.76	100
8	0.75	0.84	0.79	93
9	0.77	0.70	0.73	100
accuracy			0.76	874
macro avg	0.77	0.77	0.77	874
weighted avg	0.76	0.76	0.76	874

Fig. 5. Accuracy metrics for all the classes of the urbansound8k dataset—an overall accuracy of 76% is achieved, a very successful result in a 10-class random-guessing baseline of 10%.

We followed the same steps explained before to preprocess the data for neural network training. In this dataset, we did not use down-sampling, since the data was already recorded at a low sampling rate of 4960 Hz. Figure 7 shows examples of preprocessed signals that served as input for the training of the neural network. The most prominent frequencies for each signal are depicted in yellow. On top of each image, the type of vehicle that passed by the sound sensor is indicated.

Built with a similar architecture as the one shown in Fig. 8, using the neural network we could classify the images, and thus the corresponding sounds, with an accuracy of 86% after 60 training epochs, a considerable improvement to a 50% baseline of random guessing between the two classes of vehicles.

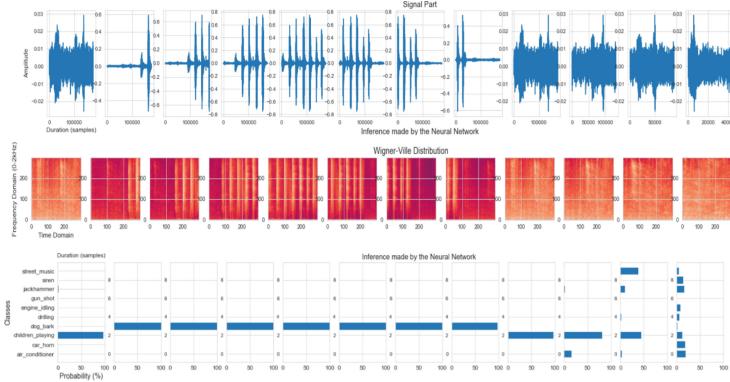


Fig. 6. Example of inference on streaming sound—the signal begins with children playing, passes to a dog barking, and finally returns to children playing. The signal is divided into 4-s overlapping windows with a stride of 1 s (first row of images), then transformed individually to the Wigner-Ville distribution (second row of images), which serve as input to the trained neural network that performs the inference successfully, as depicted in the third row of images.

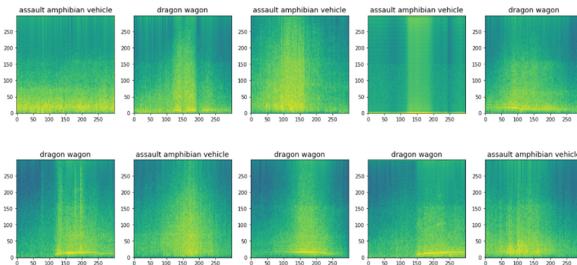


Fig. 7. Military vehicle classification dataset—examples of preprocessed signals that served as input for the training of the neural network—the most prominent frequencies for each signal are shown in yellow.

6.3 Esc-50

We applied the same technical approach to classify the 50 classes of common sounds present in the ESC-50 dataset. The goal was to identify if there is potential in building a classifier with our approach on this broad dataset. Data was down-sampled at a sampling rate of 4 kHz, essentially focusing on the frequency range from 0 to 2 kHz.

The dataset is already split in 5 folds for the purpose of cross-validation. Based on this, we trained models on 4 folds and tested them on the fifth fold that was left out from the training dataset at each point. This resulted in an average accuracy of 25% on the unseen folds, a significant improvement to random guessing 50 classes at 2% accuracy.

6.4 Note on Maximization of Accuracy

At this point, it is important to note that the optimization efforts made for the maximization of accuracy of the different models mentioned above was minimal. The focus was

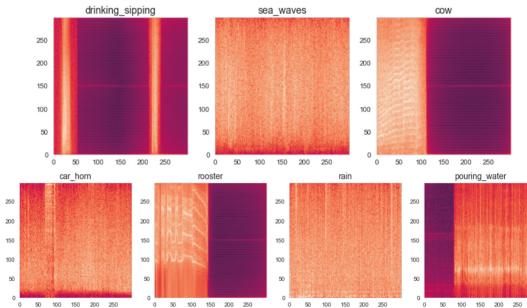


Fig. 8. Preprocessed signals from the esc-50 dataset that served as input for training the neural network.

geared towards building proof of concepts fast that showcase the possible application of the approach in different interesting use cases. The accuracy has potential to increase if further efforts are made in the context of dataset creation/collection and hyperparameter tuning. This kind of optimization was not relevant at this phase of the research.

7 Conclusions

Our conclusions are summarized as follows:

- The use of sound sensors to detect events is possible with such high accuracy that it can be used for practical purposes.
- We gathered extensive documentation on open-source sound datasets and state-of-the-art research and technical approaches. It is important to note that the quality and number of open-source datasets in the context of urban sounds is limited, especially when compared to image and other structured datasets.
- Sound sensors are GDPR-compliant and less intrusive than sensors such as cameras.
- We designed a technical approach which transforms the sound signals to the Wigner-Ville time-frequency distribution. These transformed signals are treated as images and classified with the use of Convolutional Neural Networks.
- We validated positively the technical approach through several proof-of-concepts based on open-source datasets.

8 Future Work

There is undeniable potential in the classification of sound signals using AI techniques. What is missing currently and should drive future work is the application of such techniques on real-world datasets and applications, as well as the dissemination of such work in the form of open-source datasets and research papers, to further foster research in the field.

Acknowledgments. I would like to thank my supervisors Stef van Eijndhoven and Peter Duin for their continuous support and inspiration, as well as my colleagues at the Jheronimus Academy of Data Science (JADS) for the fruitful and engaging working environment.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105. (2012)
2. Gemmeke, J.F., Ellis, P.D., Freedman, D., Jansen, A., Lawrence, W., Moore, C.R., Ritter, M.: Audio set: An ontology and human-labeled dataset for audio events. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP), p. 5. IEEE, New Orleans, LA, USA
3. Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B., Slaney, M.: CNN architectures for large-scale audio classification. Weiss, Kevin Wilson Google, Inc., New York, NY, and Mountain View, CA, USA
4. Moss, J.C., Hammond, J.K.: A comparison between the modified spectrogram and the pseudo-Wigner-Ville distribution with and without modification
5. Scikit-learn.: Multiclass and multilabel algorithms. Retrieved from Scikit-learn Documentation: <https://scikit-learn.org/stable/modules/multiclass.html>
6. Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, F. J., Jansen, A., Moore, C. R., Wilson, K.: CNN architectures for large-scale audio classification. In: 2017 IEEE International Conference on acoustics, speech and signal processing (ICASSP), p. 5. IEEE, New Orleans, LA, USA (2017)
7. Salamon, J., Jacoby, C., Bello, J.P.: A dataset and taxonomy for urban sound research. In: MM’14: Proceedings of the 22nd ACM international conference on Multimedia, pp. 1041–1044. Association for Computing Machinery New York NY United States, Orlando Florida USA (2014)
8. Salamon Justin, MacConnell, D., Cartwright, M., Li, P., Bello, J.P.: Scaper: A library for soundscape synthesis and augmentation. In: 2017 IEEE workshop on applications of signal processing to audio and acoustics (WASPAA), pp. 344–348. IEEE, New Paltz, NY (2017)
9. Piczak, K.J.: ESC: Dataset for environmental sound classification. In: MM ‘15: Proceedings of the 23rd ACM international conference on Multimedia, pp. 1015–1018. Association for Computing Machinery New York NY United States, Brisbane Australia, (2015)
10. Zinemanas, P., Cancela, P., Rocamora, M.: MAVD: A dataset for sound event detection in urban environments. In: 4th Workshop on detection and classification of acoustic scenes and events (DCASE 2019). (2019)
11. Duarte, M.F., Hu, Y.H.: Vehicle classification in distributed sensor networks. J. Parallel Distrib. Comput., 826–838 (2004)
12. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CORR. (2015)
13. Discussion on Time-Frequency Distributions. (2013). Retrieved from Research Gate: https://www.researchgate.net/post/Can_anyone_explain_what_is_Wigner-Ville_distribution_and_how_the_time_frequency_analysis_performed_by_WVD
14. Sanjit, D.K., Sasibhusan, G.R.: Arrhythmia detection using wigner-ville distribution based neural network. In: International Conference on Computational Modelling and Security (CMS 2016), pp. 806–811. (2016)
15. Sandsten, M., Brynolfsson, J.: Classification of bird song syllables using. In: 2017 25th European signal processing conference (EUSIPCO), vol. 5. (2017)
16. Hormigo, J., Cristobal, G.: Image segmentation using the Wigner-Ville distribution. Adv. Imaging Electron Phys., **14**, (2003)
17. Boualem, B., Whitehouse, J.H.: Seismic applications of the Wigner-Ville distribution, vol. 4. (1986)



Application and Performance Improvement of Transfer Learning on ICBHI Lung Sound Dataset

Mohan Xu^{1(✉)} and Lena Wiese^{1,2}

¹ Fraunhofer Institute for Toxicology and Experimental Medicine, Hannover,
Germany

mohan.xu@item.fraunhofer.de, lena.wiese@item.fraunhofer.de

² Institute of Computer Science, Goethe University Frankfurt, Frankfurt a. M.,
Germany

Abstract. Chronic respiratory diseases are one of the leading causes of morbidity and mortality worldwide. How to prevent the disease or to diagnose and treat it effectively in the early stage has always been a focused medical research area. In this paper, a neural network that was pre-trained based on a large audio event dataset called AudioSet is transferred and applied in the training and testing of the Respiratory Sound database ICBHI; in addition, various methods are used in data preprocessing, neural network configuration and post-processing to improve the performance of the transfer learning model. The final model can not only converge quickly, but also use the accuracy calculation method provided by ICBHI Challenge to reach 81.1% in the four classification tasks containing normal, crackle, wheeze and both respiratory sounds, which is superior to the previous methods. This paper also analyzes the unbalanced distribution of the respiratory cycle dataset based on demographic data on the binary classification task (normal and abnormal). The binary classification model scored 85.5% and 81.1% on the female test group and the male test group, respectively. To address the above differences due to the unbalanced dataset, we used a restricted mixup approach to successfully reduce the difference between the male and female test groups to 0.82%.

Keywords: ICBHI dataset · Respiratory sounds classification · Neural network · Transfer learning

1 Introduction

According to a report released by the World Health Organization (WHO) [1], in 2016, the number of deaths from non-communicable diseases worldwide accounted for 71% of the total number of deaths. Chronic respiratory disease ranks third among the four leading causes of death from non-communicable diseases. More than 40% of countries have fewer than 10 doctors per 10,000 people.

How to prevent a chronic lung disease or take effective treatment in the early stage of disease is an important research topic in modern medicine. Developing and verifying novel digital health approaches to support the effective detection and diagnosis of lung diseases are hence an important step in order to improve the outcomes for affected patients.

Stethoscopes are widely used worldwide as a non-invasive method for the analysis of lung sounds. They enable medical staff to diagnose possible diseases by auscultation of the lungs (and potentially other organs as for example the heart). The use of stethoscope for medical diagnosis is not only suitable for a wide range of people, but also can quickly obtain test results, thereby winning precious time for patient treatment. However, the use of a stethoscope is very dependent on the doctor's individual hearing abilities and clinical experience. If the diagnosis results are not accurate enough, it may lead to incalculable consequences. Therefore, it is our research purpose to apply machine learning methods on the existing respiratory system sound database to obtain more accurate results in the future diagnosis of respiratory diseases. This cannot only provide general practitioners with machine-aided analysis of diagnostic data from experienced doctors, but may also provide a valuable digital health tool for patients in remote areas or in home environments due to the feasibility of conducting a diagnosis remotely.

1.1 Our Contribution

This work uses a transfer learning approach to apply Wavegram Logmel CNN [13] trained based on Audioset [14] to the ICBHI dataset [2] and save snapshots [15] of the model at different stages as the learning rate changes. In the pre-processing phase of the data, we provide several data augmentation methods: splitting and padding, *nlpAug* library [11], *rollAudio* and *mixup* [12] and compare their effects on the model performance to select the best model configuration. In a 4-classification task containing respiratory sounds of normal, crackle, wheeze and both, the prediction results of the ensemble model obtained by our transfer learning system after 10-fold cross validation achieved a score of 81.1%, outperforming previous methods. Based on the demographic information provided by ICBHI [2], we investigated the distribution of the dataset in terms of gender, age and BMI. The model scores of the female test group and the male test group based on normal/abnormal respiratory cycle were 85.5 and 81.1%, respectively. To reduce discrimination in the male test group due to the uneven distribution of the dataset, we used a restricted mixup approach to reduce the difference in model scores between the male and female test groups to 0.82%.

1.2 Outline of this Paper

The paper is outlined as follows. Section 2 describes research work on the classification task of the ICBHI respiratory sound database. The proposed methodology, introduction of ICBHI dataset, official evaluation methods, preprocessing, transfer learning and ensembling steps are depicted in Sect. 3. Section 4 presents the experiments and results on the proposed transfer learning system. We conclude this work and identify future directions in Sect. 5.

2 Related Work

In recent years, machine learning has been widely used in the classification of the respiratory system sound database ICBHI [2]. The sound data enters the classifier after different preprocessing methods, and the prediction of the category to which the data belongs is realized through the learning of parameters. The boosted tree model proposed by [3] takes all the features as input and performs multiple iterations to achieve the prediction of breathing cycle. The LungBRN model [4] simultaneously receives the features obtained by the two preprocessing methods, trains them in the Resnet network respectively, and finally multiplies them in the fully connected layer. Demir et al. [5] combines transfer learning (VGG16 pretrained model) and SVM algorithm. In addition, Recurrent Neural Networks (RNNs) have also been used in some works [6–8] for respiratory cycle classification problems.

In contrast to research works [4,5,10] that used neural networks pre-trained on the large-scale visual database Imagenet [9], the proposed transfer learning system applies the neural network pre-trained on the large-scale audio dataset google audioset [14] containing 527 sound categories [13], thus enabling the downstream task to learn audio features faster. In the preprocessing stage, preprocessing methods that have performed well in other tasks such as [6,17,20] are used in our respiratory sound classification task. Table 3 shows the comparison between our work and some research works with the same dataset division ratio. In addition, this work investigates the distribution of the dataset on gender, age and BMI, and tests the performance differences of the neural network model on uneven datasets. As an example of uneven datasets due to the gender of the subjects, we propose a restricted mixup approach to reduce the resulting discrimination of the model against the male test group. To the best of our knowledge, this work is the first research on the respiratory sound database ICBHI to analyse network model performance based on demographic information.

3 Methodology

This section begins with a detailed description of the ICBHI dataset and the corresponding evaluation criteria. The general approach of our system based on google audioset’s pre-trained model Wavegram Logmel CNN [13,14] to achieve classification prediction of respiratory cycles on normal, crackle, wheeze and both. The proposed system is shown in Fig. 1, including different preprocessing methods for the respiratory cycles, transfer learning of Wavegram Logmel CNN [13] and taking model snapshots at different local minima. The details of the transfer learning system are described in the following subsections.

3.1 Dataset

The ICBHI Scientific Challenge database [2] is a publicly available respiratory sound database that is tested through scientific challenges, thus enabling digital

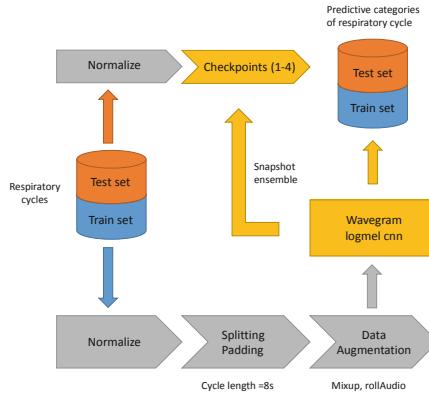


Fig. 1. Workflow of training and testing process on the ICBHI dataset.

auscultation based on its usability in terms of data and algorithms. This respiratory sound database consists of 920 annotated audio samples from 126 subjects with data categories labeled by respiratory experts as: normal, wheeze, crackle, and both (wheeze and crackle). Each audio sample can be divided into multiple respiratory cycles based on annotation and may contain multiple categories. The database contains a total of 6898 respiratory cycles, of which 1864 contain crackles, 886 contain wheezes, 506 contain both, and the rest are normal. In addition, a large number of samples in the database contain noise, which makes the data classification problem closer to a real-life scenario. Figure 2 shows mel-spectrograms of four representative respiratory cycles from the same subject, belonging to four data categories (normal, wheezing, crackles and both).

3.2 Evaluation Method and Criteria

The evaluation method in this work uses the widely used officially proposed criteria. For the four classification (normal (N), crackle (C), wheeze (W) and both (B)) problems, the three measures Sensitivity (S_e), Specificity (S_p) and Score (S_c) are defined as follows,

$$S_e = \frac{C_{correct} + W_{correct} + B_{correct}}{C_{total} + W_{total} + B_{total}} \quad (1)$$

$$S_p = \frac{N_{correct}}{N_{total}} \quad (2)$$

$$S_c = \left(\frac{S_e + S_p}{2} \right) * 100 \quad (3)$$

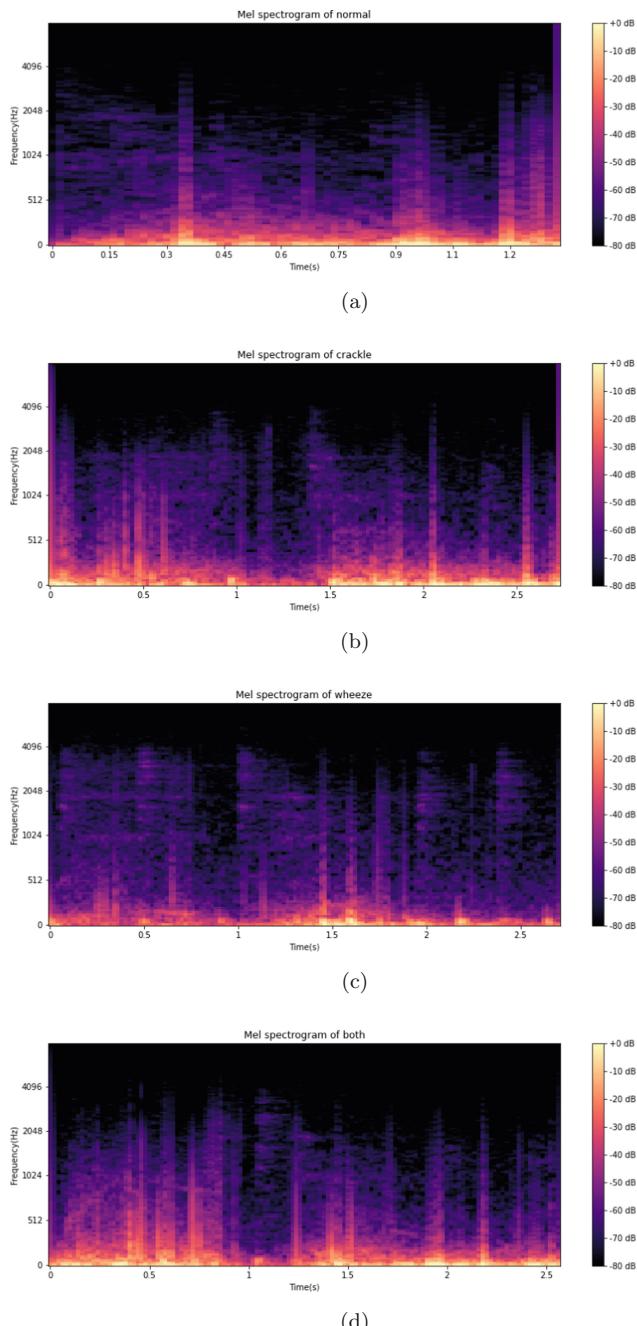


Fig. 2. Mel-Spectrograms of Different Classes of Respiratory Cycles. **a** Normal
b Crackle **c** Wheeze **d** Both

where $i_correct$ and i_total denote the number of correctly classified respiratory cycles and the total number of respiratory cycles contained in the class when $i \in \{N, C, B, W\}$.

3.3 Preprocessing

To facilitate the preprocessing of continuous audio signals in the neural network workflow, the librosa library [16] reads sound files and samples them as discrete audio signals at 16 kHz. The sampled data are divided into different respiratory cycles according to the corresponding annotated data and identified to the category (0–3) they belong to. The data for each respiratory cycle are classified according to “data”, “start”, “end”, “label”, “cycle” and “filename” column names (Table 1). The first four columns were used for the whole experiment, i.e., the classification task of the respiratory cycle and the difference in model performance on test groups by gender. The last column was used for the second part of the experiment, where the subject index was identified by the filename corresponding to that respiratory cycle, and thus the subject’s gender information was obtained from the official demographic information provided.

Table 1. Part of the respiratory cycle information contained in a sound file (filename.wav).

Cycle	Start (s)	End (s)	Crackle	Wheeze	Label
1	1.778	4.032	0	0	0
2	4.032	6.319	1	0	1
3	6.319	8.239	0	1	2
4	8.239	10.075	1	1	3

Before starting the formal neural network training, length alignment, data augmentation and normalization operations are required. The following section describes in detail the various data preprocessing methods currently in use.

Splitting and Padding: The respiratory cycle lengths in the ICBHI dataset ranged from 0.2 to 16.1 s, while the input shape of the neural network is fixed. Considering the length of the pre-trained dataset in [14,17] and the performance of the ICBHI dataset trained on different cycle lengths, we set the length of each cycle of the input neural network to 8s.

When the length of a respiratory cycle is greater than 8s, the respiratory cycle will be divided into multiple sub-respiratory cycles, which can be achieved by the framing function of the librosa library [16]. The frame length is set to 8s and the hop length of the frame is 4s. The frame will be displaced in the direction of the respiratory cycle length until the remaining respiratory cycle length does not satisfy the frame length, thus different frames (sub-respiratory cycles) can be obtained.

When the length of a certain respiratory cycle is less than 8s, that respiratory cycle will be repeatedly spliced along the length direction until it is greater than or equal to 8s.

Data Augmentation: In the data preparation phase, a randomly ordered data augmentation combination is formulated, which contains the *NoiseAug*, *SpeedAug*, *LoudnessAug*, *VtlpAug* and *PitchAug* methods from the *nlpaug* library [11]. The augmented data are processed according to the *rollAudio* method in Table 2. A random index is generated based on the cycle length, and the data is rolled from this index until it returns to the previous bit of this index.

Table 2. Comparison of arbitrary respiratory cycle before and after rollaudio operation (β is the generated random index)

Before	2	3	5	7	11	13	17	19
After	7	11	13	17	19	2	3	5
Index	0	1	2	3	4	5	6	7

mixup [12] augments the dataset by mixing random respiratory cycles over a certain respiratory cycle according to $mixingproportion = (\lambda : 1 - \lambda)$ over the range of the dataset. λ is generated by beta distribution, a set of continuous probability distributions (Eq. 4) defined on the $(0, 1)$ interval, by setting alpha and beta and thus controlling the interpolation intensity between two respiratory cycles. For $\alpha, \beta \in (0, \infty)$

$$\lambda \propto Beta(\alpha, \beta), \quad (4)$$

Since this probabilistic event mixup occurs on a pair of respiratory cycles, the interpolated intensity distributions of the two respiratory cycles in mixup are equal in the absence of constraints such as data weights. The beta distribution is uniform only when $\alpha = \beta$ and λ is symmetric on $x = 0.5$. Referring to the experiments of [12], we set both α and β in the beta distribution equal to 0.2 so that the λ used in Eqs. (5) and (6) can be calculated.

$$\tilde{x} = \lambda x_i + (1 - \lambda)x_j \quad (5)$$

$$\tilde{y} = \lambda y_i + (1 - \lambda)y_j \quad (6)$$

(x_i, y_i) are the inputs and targets of the original cycle, respectively, and the inputs and targets of the random cycles involved in the mixup operation are (x_j, y_j) , respectively. The results returned by Eqs. (4) and (5) will be used as the inputs of the neural network.

Normalization: If the original data is used directly, the data of different orders of magnitude will have different effects on the analysis results. The data of the larger order of magnitude will weaken the effect of the data of the

smaller order of magnitude in the analysis, which is not the result we want to see. The data processed by Z-score normalization have a mean of 0 and a variance of 1, which are on the same order of magnitude.

The mean and standard deviation corresponding to each respiratory cycle are calculated separately and averaged over the entire dataset. For any data point x we obtain,

$$x' = \frac{x - \mu}{\delta} \quad (7)$$

where μ is the mean of all sample data, and δ is the standard deviation of all sample data.

3.4 Transfer Learning

Transfer learning can apply the trained model to a new but related field, thus the convergence of the model can be completed faster with less training cost in the absence of annotated data. The transfer learning model used in this paper is loaded with the parameters of the Wavegram Logmel CNN [13] trained on Audioset [14] and fine-tuned. After activation function *log softmax* the prediction results of respiratory cycle on four classifications normal, crackle, wheeze and both are obtained. The input original waveform is trained separately on the two branches and the merged result goes through 5 dropout layers and 5 block operations containing *conv2d*, *batchNorm2d*, *relu* and *avg_pool2d*.

More precisely, the branches proceed as follows:

Branch 1: After the data is processed by Conv1d with a kernel size of 11 and BatchNorm1d, it goes through three blocks containing *conv1d*, *batchNorm1d*, *relu* and *max_pool1d*.

Branch 2: The 8s audio data is subjected to Fourier short-time transform and 64 Mel bins to generate a 701×64 Mel spectrogram. The spectrum is output after a block operation including *conv2d*, *batchNorm2d*, *relu* and *avg_pool2d*.

3.5 Snapshot Ensemble

Snapshot Ensemble [15] is a way to get an ensemble of models from a single training session without additional training cost. It sets the learning rate as shown in Fig. 3. Whenever the learning rate restarts, the model starts exploring other local optima and takes model snapshots at different local minima.

Equation 8 shows the mathematical representation of Fig. 3, where l_{init} is the initial learning rate, e_i is the i -th epoch in the current cycle, and c is the cycle length. The function f calculates the learning rate for each epoch as follows:

$$f(e_i) = \frac{l_{init}}{2} \cdot (\cos(\frac{\pi \cdot i}{c}) + 1) \quad (8)$$

Figure 3 sets the initial learning rate to 0.01, 50 epochs as a cycle, and a total of 4 cycles. Considering the memory problem of a single gpu, the test process

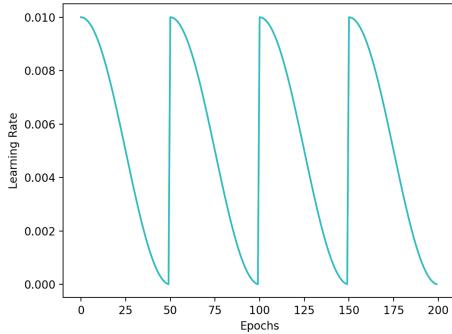


Fig. 3. Learning Rate with the Current Epoch Used in the Training Process.

can load model snapshots on several gpus and the results obtained are averaged on the same gpu.

4 Experiments and Results

The transfer learning system proposed in this paper is used to solve three tasks: (1) To investigate the effect of different experimental setups on model scores and to compare our work with other works on four classifications of respiratory cycles (normal, crackles, wheezes and both crackles and wheezes); (2) the unbalanced distribution of the dataset in terms of gender, age and BMI was analyzed with demographic information provided by ICBHI and the differences in model scores were validated on the male and female test groups; (3) we proposed a restricted mixup to reduce the differences in model performance between the male and female test groups found in the second task.

4.1 Performance Comparison

First, respiratory cycles with a sampling frequency of 16kHz are divided into trainingset and testset according to a ratio of 8:2. We set the baseline of the transfer learning system without using `nlpAug` library, `rollAudio` and `mixup` for data augmentation in the data preprocessing. By filling respiratory cycles with 0 (Zeropadding) or extracting only the first 8s of them we can increase or decrease the length of respiratory cycles to 8s. The baseline uses the SGD optimizer and cross entropy loss.

4.1.1 Hyperparameters We evaluate the scores of baselines using different learning rates and batchsizes on the ICBHI dataset. Referring to the batchsize in [13] we compared its model scores on 16, 32 and 64 (Fig. 4). When the batch size is 32, the model score is 2.0% and 2.9% higher than other 2 options. Figure 5 shows that the model scores at learning rate 0.01 are much higher than those

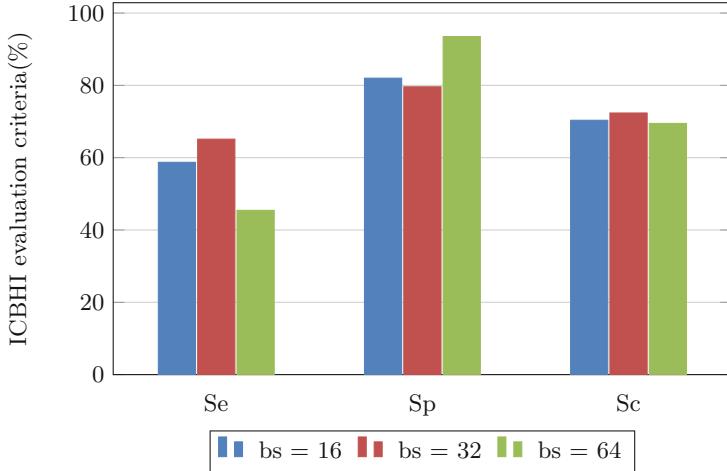


Fig. 4. Comparison of baselines with different batchsize on the testset ($epochs = 50$, $lr = 0.01$).

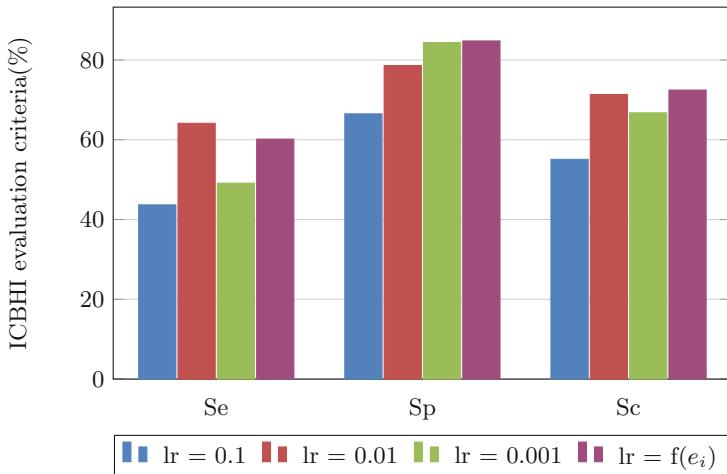


Fig. 5. Comparison of baselines with different learning rates on the testset ($epochs = 50$, $bs = 32$).

at learning rates 0.1 and 0.001. And the model scores in the learning rate based on Eq. 8 are higher than the model scores when the learning rate is constant at 0.01. So we will apply a batchsize of 32 and a learning rate based on Eq. 8 to the baseline in the following performance comparison.

4.1.2 Data Preprocessing We focus on the effect of different data preprocessing methods on the model training and loss functions in Figs. 6 and 7. Setting the epoch to 200, we add the splitting and padding method to the baseline model identified in the previous subsection, resulting in a 4.3% increase in the model score. The combination of splitting and padding, mixup has only 0.3% difference

in model score but the loss function has dropped by 29.2%. We found that using the splitting and padding, mixup and data augmentation combination in the data preprocessing stage resulted in the smallest loss function and the highest model score of 80.9%.

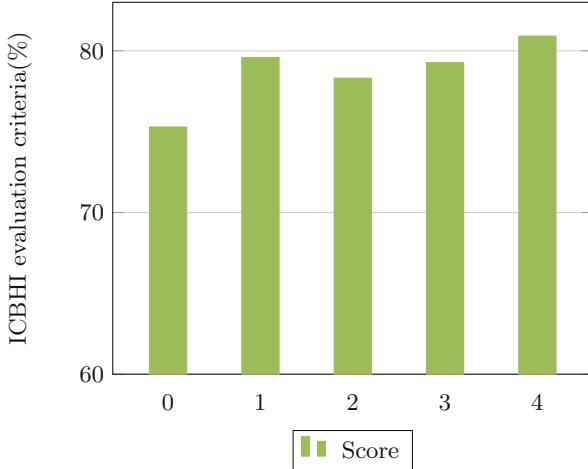


Fig. 6. Comparing model performance with different data preprocessing methods (0: Baseline, 1: SP, 2: SP+rollaudio, 3: SP+mixup, 4: SP+mixup+DA Group) on the test set ($epochs = 200$, $bs = 32$).

4.1.3 Snapshot Ensemble From Figs. 7 and 8 we see the effectiveness of Snapshot Ensemble in the respiratory cycle 4-classification task. Also, we verify that the data preprocessing scheme with the best performance in the previous subsection still has the highest model score (82.0%) and the smallest loss function value (0.513) after Snapshot Ensemble.

4.1.4 Comparison to other Works The most suitable configuration is selected by comparing the performance of our transfer learning system on hyperparameters, data preprocessing and snapshot ensemble in Sect. 4.1. This section compares the model scores of the transfer learning system after a 10-fold cross validation against other work.

The results in Table 3 show that our transfer learning system outperforms competitors on all three evaluation criteria. The RNN-based end-to-end model architecture adopted in [7] detects abnormal sounds in the respiratory cycle through masking of noise. the research [17] adopted the data processing method of device specific fine-tuning, concatenation-based augmentation, blank region clipping, and smart padding and realized the respiratory cycle classification task based on a simple CNN. The study [18] used STFT and wavelet to extract features and input them into the support vector machine. The authors in [6] established a learning framework based on recurrent neural networks to discover time-dependent patterns from sound data. Both [8, 19] completed training on a

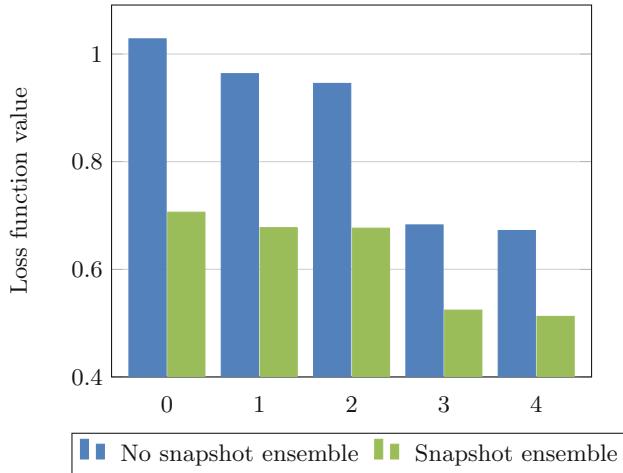


Fig. 7. Compare the loss function values of different data preprocessing methods (0: Baseline, 1: SP, 2: SP+rollaudio, 3: SP+mixup, 4: SP+mixup+DA Group) before and after snapshot ensemble.

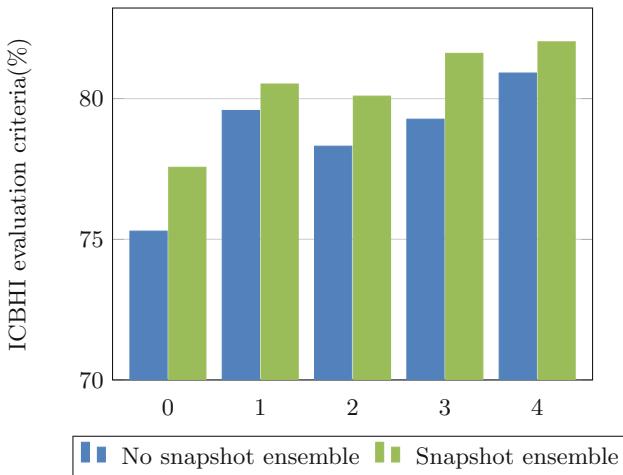


Fig. 8. Compare the model scores of different data preprocessing methods (0: Baseline, 1: SP, 2: SP+rollaudio, 3: SP+mixup, 4: SP+mixup+DA Group) before and after snapshot ensemble.

hybrid neural network: the former added a Bi-LSTM layer for learning temporal features to CNN, and the latter integrated multi-branch temporal convolutional network and squeeze-and-excitation network. The research [20], which is the closest to us in terms of experimental methods and experimental results, also uses the combined method of CNN and snapshot ensemble, but since our transfer

Table 3. ICBHI challenge comparison (On four categories).

Method	<i>Se</i> (%)	<i>Sp</i> (%)	<i>Sc</i> (%)
NMRNN [7]	56.0	73.6	64.8
RespireNet [17]	53.7	83.3	68.5
STFT+Wavelet [18]	55.3	83.3	69.3
Hybrid CNN-RNN [8]	56.9	86.7	71.8
LSTM [6]	62.0	85.0	74.0
MBTCNSE [19]	65.3	86.1	75.7
CNN(snapshot ensemble) [20]	69.4	87.3	78.4
Our System	70.5	91.7	81.1

learning system uses more diverse data augmentation methods in data processing (such as random combination of augmentation methods and mixup), using pre-trained network parameters on the large-scale sound dataset Google audioset [14]. Our transfer learning system achieves a score of 81.1% on the test set when compared with other state-of-the-art systems, which is obviously more advantageous.

4.2 Demographic Data

In this section, we analyze the imbalanced distribution of the respiratory cycle dataset based on demographic data on the binary classification task (normal and abnormal). Referring to the labels in Table 1, the normal class includes all normal respiratory cycles, and the abnormal class includes crackle, wheeze and both respiratory cycles. The demographic data provided by ICBHI [2] for each participant were arranged in order of age, gender, BMI, child weight and child height. Considering that there are different classification criteria for children's BMI [21, 22], we excluded respiratory cycles belonging to children and only analyzed the imbalance of the ICBHI dataset on the demographic data age, gender and BMI of adults. We also excluded information on some subjects due to missing age, gender or BMI. After the above two rounds of filtering the dataset contains 6004 respiratory cycles

Table 4. Distribution and overall percentage of subjects' gender on binary classification task (Normal and abnormal) corresponding to the respiratory cycle dataset.

Normal		Abnormal	
Female	Male	Female	Male
929	2018	1065	1992
15.47%	33.61%	17.74%	33.18%

Table 4 summarizes the data distribution of the respiratory cycle dataset based on different genders when performing the binary classification task. In the normal and abnormal data distributions, males have 18.14 and 15.44% higher than females in the dataset, respectively.

Table 5. Distribution and overall percentage of subjects' age on binary classification task (Normal and abnormal) corresponding to the respiratory cycle dataset.

Normal		Abnormal	
Adult	Senior	Adult	Senior
500	2447	448	2609
8.33%	40.76%	7.46%	43.45%

We classified subjects aged greater than or equal to 18 years but less than 60 years and those aged greater than or equal to 60 years as adults and seniors, respectively. In Table 5, it is shown that in the respiratory cycle judged as normal, the data for the senior were 4.89 times higher than the data for adults; in the respiratory cycle judged as abnormal, the data for the senior were 5.82 times higher than the data for adults.

Table 6. Distribution and overall percentage of subjects' bmi on binary classification task (Normal and abnormal) corresponding to the respiratory cycle dataset.

Normal				Abnormal			
Under	Normal	Over	Obesity	Under	Normal	Over	Obesity
135	767	1393	652	596	728	1143	590
2.25%	12.77%	23.20%	10.86%	9.93%	12.13%	19.04%	9.83%

According to the BMI we divided the subjects into underweight ($BMI < 18.5$), normalweight ($18.5 \leq BMI < 25$), overweight ($25 \leq BMI < 30$) and obesity ($BMI \geq 30$). Table 6 shows that the underweight and overweight groups had the smallest (2.25%) and the largest (23.20%) data distribution in the normal category; the obesity and overweight groups had the smallest (9.83%) and the largest (19.04%) data distribution in the abnormal category, respectively.

From Tables 4, 5 and 6, we can see that the subgroups based on demographic data age, gender and BMI, respectively, are unevenly distributed in the binary-classes dataset. Table 7 is based on the summary of Tables 4, 5 and 6. The dataset containing 6004 respiratory cycles can be divided into 32 different attributes based on the three demographic variables gender, age, BMI, and respiratory cycle labels. In studying the differences between female and male transfer learning system, we divided the data corresponding to each attribute into the training-, validation-, and testsets sequentially in the ratio of 72:8:20 to avoid the uneven distribution of the dataset aggravated by the random division.

Table 7. In the binary classification task (Normal and abnormal), the uneven distribution of the respiratory cycle dataset on the three demographic variables of gender, age and bmi, each row corresponds to underweight, normalweight, overweight and obesity

	Normal				Abnormal			
	Female		Male		Female		Male	
	Adult	Senior	Adult	Senior	Adult	Senior	Adult	Senior
Underweight	7	80	0	48	0	427	0	169
Normalweight	225	115	76	351	238	37	105	348
Overweight	16	259	56	1062	15	142	2	984
Obesity	0	227	120	305	0	206	88	296

4.3 Comparison of Model Performance based on Subject Gender

In this section, three experiments will be conducted based on no mixup, using global mixup and using restricted Mixup, so as to compare the difference in scores of the transfer learning system on the female test group and the male test group (Fig. 9) and to ameliorate the discriminatory effect on a particular gender due to training the model on an unbalanced dataset.

In the first experiment, the transfer learning system inherits the best configuration of the model from Sect. 4.1, but does not use mixup for data augmentation in the data preprocessing phase. As known from Fig. 9, the model scored 4.44% lower on the male test group than on the female test group. Thus, respiratory cycles from males are less likely to be correctly classified than those from females.

Based on the findings of the first experiment we used mixup in the global scope to perform online data augmentation. Referring to Eqs. 5 and 6, we randomly selected respiratory cycle x_j across the entire dataset and fused it with the original respiratory cycle x_i in a ratio of $(1 - \lambda)$ to λ , where λ is from the beta distribution. The transfer learning system with the addition of global Mixup increased the score on the male testset by 0.86%, but still scored 3.53% lower than the female testset.

Considering the positive impact of the second experiment, we continued mixup's method in the third experiment but added the restriction. For $G_1, G_2 \in \{\text{female}, \text{male}\}$ there are the following formulas,

$$\tilde{x} = \lambda x_{G_1,i} + (1 - \lambda)x_{G_2,j} \quad (9)$$

$$\tilde{y} = \lambda y_{G_1,i} + (1 - \lambda)y_{G_2,j} \quad (10)$$

$x_{G_1,i}$ and $y_{G_1,i}$ are the input and target of the original respiratory cycle from the subject with gender G_1 ; the input and target of the random cycle for mixup are $x_{G_2,j}$ and $y_{G_2,j}$ from the subject with gender G_2 . The two subjects differ in gender ($G_1 \neq G_2$). The results returned by Eqs. (9) and (10) will be used as the input to the neural network.

The transfer learning system with the addition of gender discrimination-aware mixup scored 2.6% higher on the male testset than in the first experiment,

and the difference in scores with the female testset was reduced to 0.82%. This shows that adding the restriction to mixup can effectively reduce the difference in model performance on the male and female testsets due to the unbalanced dataset and reduce the discriminatory effect of the model on the male testset.

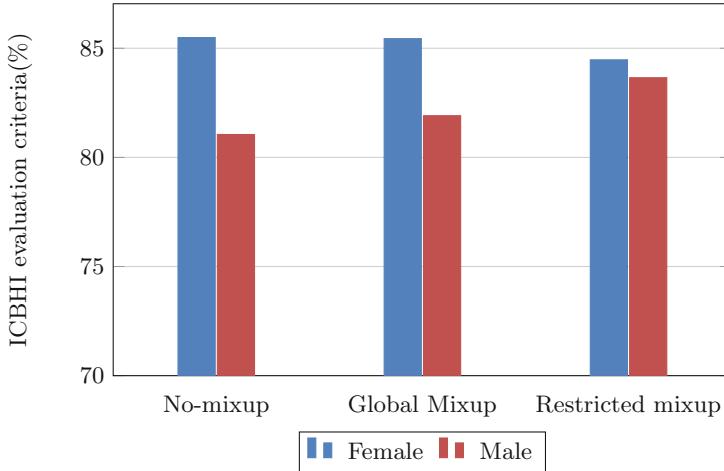


Fig. 9. Comparison of the transfer learning system scores on the female test group and the male test group under different mixup settings.

5 Conclusion and Future Work

Our proposed transfer learning system implements the respiratory cycle 4-classification task (normal, crackle, wheeze or both) in the Respiratory Sound database ICBHI [2] and achieves a score of 81.1% using the evaluation method provided by ICBHI Challenge. After comparing the performance of the transfer learning system based on the pre-trained model Wavegram Logmel CNN [13] with different hyperparameters, data preprocessing methods and model ensembling, we selected the best model configuration that outperformed almost all state-of-the-art systems. In addition, we also discuss the uneven distribution of the respiratory cycle dataset on the gender, age and BMI of the subjects according to the 2-classification task (normal or abnormal respiratory cycle) and validate the 4.44% difference in their scores on the female testset compared to the male testset. For the discriminatory effect of the transfer learning system on the male testset, we explored different mixup methods in the data preprocessing phase and proposed a restricted mixup method to reduce this difference to 0.82%.

In the future work, how to utilize the demographic information corresponding to the respiratory cycle will be the focus of our research. Therefore, the

following questions are worth for further exploration in this study. (1) Considering the inclusion of subject demographic information (gender, age, and BMI) in the training workflow, how will these information affect the model performance? (2) Due to the uneven distribution of demographic information (gender, age, and BMI) in the respiratory cycle dataset (Table 7), whether the model performance also varies across age and BMI groups? What is the effect of the proposed restricted mixup method in the third experiment on the model performance trained on different age or BMI groups? (3) Whether the proposed restricted mixup method can improve the classification accuracy of the entire dataset.

References

1. World Health Organization. World health statistics 2020 (2020)
2. Rocha, B.M., Filos, D., Mendes, L., et al.: A respiratory sound database for the development of automated classification. In: International Conference on Biomedical and Health Informatics, pp. 33–37. Springer, Singapore (2017)
3. Chambres, G., Hanna, P., Desainte-Catherine, M.: Automatic detection of patient with respiratory diseases using lung sound analysis. In: 2018 International Conference on Content-Based Multimedia Indexing (CBMI), pp. 1–6. IEEE (2018)
4. Ma, Y., Xu, X., Yu, Q., et al.: LungBRN: A smart digital stethoscope for detecting respiratory disease using bi-resnet deep learning algorithm. In: IEEE Biomedical Circuits and Systems Conference (BioCAS), pp. 1–4. IEEE (2019)
5. Demir, F., Sengur, A., Bajaj, V.: Convolutional neural networks based efficient approach for classification of lung diseases. *Health Inf. Sci. Syst.* **8**(1), 1–8 (2020)
6. Perna, D., Tagarelli A.: Deep auscultation: Predicting respiratory anomalies and diseases via recurrent neural networks. In: IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS), pp. 50–55. IEEE (2019)
7. Kochetov, K., Putin, E., Balashov, M., et al.: Noise masking recurrent neural network for respiratory sound classification. In: International Conference on Artificial Neural Networks, pp. 208–217. Springer, Cham (2018)
8. Acharya, J., Basu, A.: Deep neural network for respiratory sound classification in wearable devices enabled by patient specific model tuning. *IEEE Trans. Biomed. Circuits Syst.* **14**(3), 535–544 (2020)
9. Russakovsky, O., Deng, J., Su, H., et al.: Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**(3), 211–252 (2015)
10. Nguyen, T., Pernkopf, F.: Lung sound classification using co-tuning and stochastic normalization. *IEEE Trans. Biomed. Eng.* (2022)
11. Edward, M.A.: NLP Augmentation. <https://github.com/makcedward/nlpaug>
12. Zhang, H., Cisse, M., Dauphin, Y.N., et al.: mixup: Beyond empirical risk minimization (2017). [arXiv:1710.09412](https://arxiv.org/abs/1710.09412)
13. Kong, Q., Cao, Y., Iqbal, T., et al.: Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Trans Audio, Speech, Lang Process* **28**, 2880–2894 (2020)
14. Gemmeke, J.F., Ellis, D.P.W., Freedman, D., et al.: Audio set: An ontology and human-labeled dataset for audio events. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 776–780. IEEE (2017)
15. Huang, G., Li, Y., Pleiss, G., et al.: Snapshot ensembles: Train 1, get m for free (2017). [arXiv:1704.00109](https://arxiv.org/abs/1704.00109)

16. McFee, B., Raffel, C., Liang, D., et al.: librosa: Audio and music signal analysis in python. In: Proceedings of the 14th Python in Science Conference, vol. 8, pp. 18–25 (2015)
17. Gairola, S., Tom, F., Kwatra, N., et al.: Respirenet: A deep neural network for accurately detecting abnormal lung sounds in limited data setting. In: 43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 527–530. IEEE (2021)
18. Petmezas, G., Cheimariotis, G.A., Stefanopoulos, L., et al.: Automated lung sound classification using a hybrid CNN-LSTM network and focal loss function. Sensors **22**(3), 1232 (2022)
19. Zhao, Z., Gong, Z., Niu, M., et al.: Automatic respiratory sound classification via multi-branch temporal convolutional network. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 9102–9106. IEEE (2022)
20. Nguyen, T., Pernkopf, F.: Lung sound classification using snapshot ensemble of convolutional neural networks. In: 42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 760–763. IEEE (2020)
21. Rolland-Cachera, M.F.: European childhood obesity group. Childhood obesity: current definitions and recommendations for their use. Int. J. Pediatr. Obes. **6**(5–6), 325–331 (2011)
22. Niederer, I., Kriemler, S., Zahner, L., et al.: BMI group-related differences in physical fitness and physical activity in preschool-age children: a cross-sectional analysis. Res. Q. Exerc. Sport **83**(1), 12–19 (2012)



A Computational Situationally Self-controlled Brain and Mind Interface Under Uncertainty

Ben Khayut^(✉), Lina Fabri, and Maya Avikhana

Intelligence Decisions Technologies Systems at IDTS, 7774711 Ashdod, Israel
ben_hi@hotmail.com

Abstract. The modern computational Interface of Artificial Intelligence cannot independently, continuously and without reprogramming it by Human Intelligence, think, understand, be conscious, aware, cognize, infer, self-learn, and self-develop under uncertainty and changing environmental objects and situations over time. The article explores the model and implementation method of Computational Situationally Self-Controlled Brain and Mind Interface Under Uncertainty (CSS-CBMIUU), representing the plausibility of Human Intelligence in the form of the next generation Artificial Intelligence computing system, using the Computational Memory and modeling of the Computing Systemic Thinking, Awareness, Consciousness, Cognition, Intuition, and Wisdom of Computational Brain and Computational Mind under uncertainty and changing environmental objects and situations in time. In doing so, the perceived objects are computationally identified, interpreted, classified, structured and stored in Computational Memory in the form of psycholinguistic and cognitive values of categories, features, and images in the revealed domain based on processing data, information, knowledge and images, stored in computational repositories. For investigation of the above-mentioned possibilities of the proposed Interface, have been applied the principles of Situational Control, Fuzzy Logic, Psycholinguistics, Informatics, and Data Science.

Keywords: Self-controlled · Brain · Mind interface

1 Introduction

1.1 Conception

By the Computational Situationally Self-Controlled Interface of the Brain and Mind, we mean a reasonable system, based on computational knowledge, with functionality and mind, that is plausible to the human intelligence, capable of perceiving the current situation and objects of the environment in real time, as well connect and interact with people, devices, software and/or hardware applications, with the aim to provide them the intellectual assistance in solving their tasks.

The modern neurobiological studies in the domain of creation Brain-computer interface are associated by analyzing the results of the impact on the brain of a living individual

by implanting a computing device into it. Research in this direction is carried out by the Neuralink Company, founded by Elon Mask in 2016.

One of the modern approaches to human-machine interaction is the approach to create specific subject area Machine Learning (ML) applications, related to Artificial Narrow Intelligence (ANI) by using Generative Pre-trained Transformer (GPT), where the professional programmers use the ML technology for periodic to retraining (reprogramming) and orientations of a model that has lost its actuality to the changed state of its object in a specific application's subject area, reflecting the relevance of answers to users questions using certain and known in advance labels-hints that users introduce when interacting with the GPT application. Due to the fact that today none of the neurobiologists knows how the human brain and mind actually work, research in the field of creating a Brain-computer interface is carried out on the basis of the need to model their functionality by applying of probabilistic and statistical linear models of Machine Learning. In this regard (Fig. 1) presents Traditional Control User Interface (UI) and Situational Fuzzy Self-Controlled Multi-agent Brain and Mind Intelligent User Interface (IUI) and their joint functioning for the aim of intellectualization of UI through of IUI and their interaction with humans and devices to solve their tasks in the environment and changing situations in it.

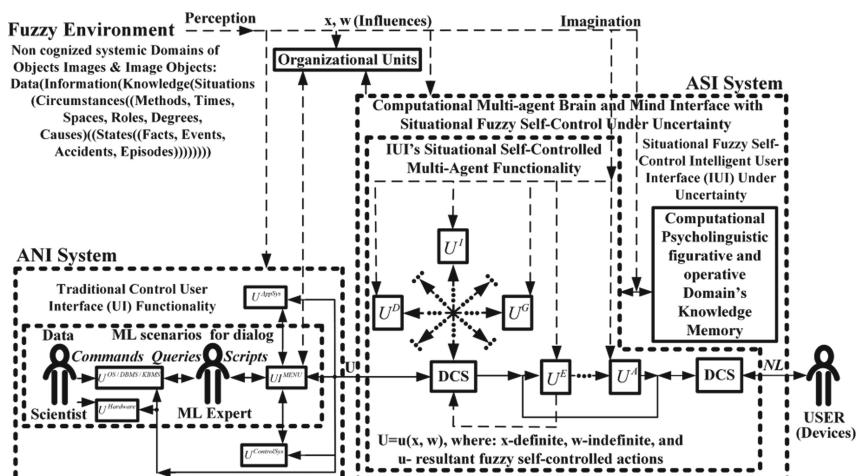


Fig. 1. Traditional control UI and situational fuzzy self-controlled multi-agent brain and mind IUI.

The article proposes a method and model for creating CSSCBMIUU (Fig. 2), which ensure the functioning of the Computational Brain (CB) and Computational Mind (CM) with applying Fuzzy Logic [1, 2], Linguistics, Psychology, Situational Control [3], Cognitive Science, Data Science, Informatics, and the proposed by us below functional models and methods for its implementation. The CSSCBMIUU is represented by set of functionality interconnected and situationally self-controlled computing Agents, that ensure the implementation of the detection of objects and situations of the surrounding reality under conditions of their uncertainty and change in time, realizing its functionality by

computational processes of their Perception, their classification and structuring, the formation their Imaginations, based on their identified categories, properties, features, their images, storing them in memory, and organizing the Analysis process using the Agents of Representation, Generalization, and Explanation of Knowledge, Consciousness, Awareness, Cognition, Systems Thinking, Decision Making, Intuition, Wisdom, and others. In doing so, the computational mental action of the CM carries out the target elementary functional action, organized by it to activate the computational system agent of thinking, carried out by the computational agent of the Situational fuzzy self-control of the CB, which activates the functional computational agents of Consciousness, Awareness, Cognition, Intuition, Wisdom, and other agents of CB for the purpose of performing this mental action of CM. Computational Understanding in CSSCBMIUU is carried out by a reasonable and unambiguous psycholinguistic situational-fuzzy self-controlling process of mental action of the CM, organized by the functionality of CB agents, that using the imaginary image of an object from Computational Memory (CMM) for determination its identity with the imaginary of an object from reality.

The computational mental representations are presented in the article as symbolic representations that have a psychologically, linguistically, and domain computational meanings of the perceived, identified, connected, integrated and nested each to other computational symbolic values of properties, features and categories (images, knowledge), data, information, and imaginary images, that form integrated whole in the CMM. The computational semantics of a language is presented in the article as symbolic, sound, visual, and other representations of the computationally meaningful, related, nested each to other, and have been computed syntactic, grammatical, semantic, pragmatic, and domain properties, features, and categories, that are integrated into symbol meanings (images, knowledge) and perceived by an imaginary set of elementary and psychologically, linguistically, and domain-generalized objects of words, phrases, sentences, and texts in form of one imaginary whole in CMM. The meaning of a word and a sentence depends, respectively, on grammatical, syntactic, semantic, and pragmatic left, conceptual, and right compatibility of the letters, words, phrases, sentences, and texts reflecting, respectively, an elementary mental image of the essence (knowledge) of an object and a generalized mental image of the essence (knowledge) of its constituent objects in the computational act of language-thought link.

Thus, the CM is implemented by functional agents of the CB, which situationally, continuously and purposefully self-activated and fuzzy-self-controlled by the computational agents of the Situational Fuzzy Self-Controlled and Computational Nervous System of the CB, which functioning under uncertainty of the surrounding environment (Fig. 3).

The CM's synthesized computational Structures of the Data, Information, Knowledge, and the Models, which have been implemented by means of CB's functionality are given, respectively, on Figs. 4, 5, 6, 7, 8 and 9, and (1), (2), (3), (4).

In the proposed article, we explore our own idea of what, in our opinion, should be the capabilities of human intelligence are inferior to Reasonable Artificial Intelligence under uncertainty. In this regard, the article proposes the implementation of the following computationally plausible to human intelligence functionality of CSSCBMIUU using computer psycholinguistics and cognitive science in subject areas: (a) Environmental

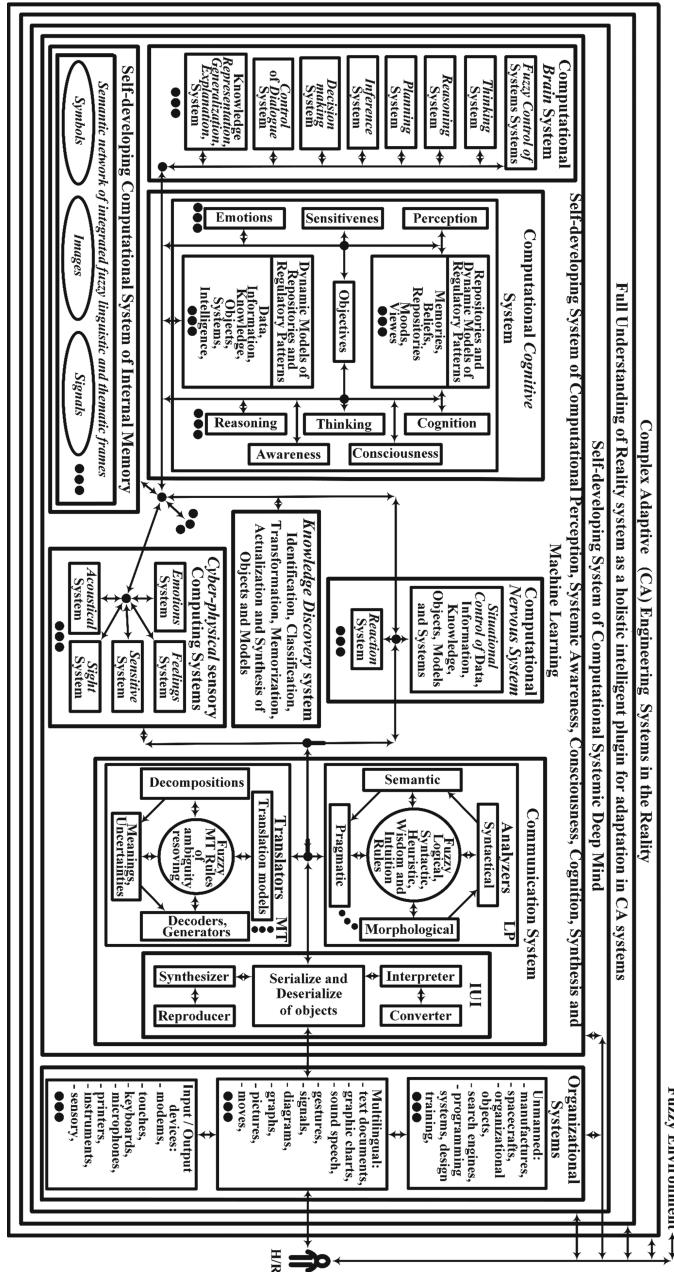


Fig. 2. A functional scheme of CSSCBMIUU.

Objects' Perception, (b) Event Reaction activity (Fig. 2), (c) Computing Memory [5], (d) Computing Inference [6], (e) Making Decisions [7], (f) Planning [7], (g) Situational

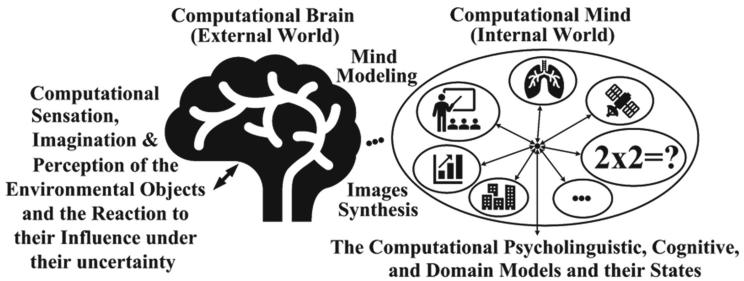


Fig. 3. The scheme of synthesizing states and models of the CM using of the CM's functionality.

Domain model (P)								
Language								
Morphological model			Semantic model	Intelligent model	Psychological (mental) model			%
Lexis	Syntax	Grammar	Meanings Concepts	Knowledge	Mind			
Categories	Categories	Categories			Knowledge			
Properties	Properties	Properties			Knowledge			
Terms	Terms	Terms			Knowledge			
Values	Values	Values			Knowledge			

Fig. 4. The CMM's psychological, cognitive and thematic models of data, information and knowledge.

Fuzzy Control [8], (h) Reasoning [9], (i) System Thinking [9], (j) Awareness [10], (k) Consciousness [11], (l) Cognition [12], (m) Intuition, and Wisdom [13], and (n) Fuzzy Control of Data, Information, and Knowledge [14].

2 The Brain Memory Model in CSSCBMIUU

2.1 The Psycholinguistic and Domain Model of Computational Brain Memory

The Figs. 4, 5 and 6, [5, 12] represent structures of psychological, linguistical, cognitive, and domain data, Information, and Knowledge, organized in Computing Memory of the CSSCBMIUU. The semantic meanings of these data, information, and knowledge are interconnected and organized in the form of a semantic network of generalized frames, reflecting the semantic mental, linguistic, cognitive, and domain relationships used by the CSSCBMIUU to analyze and implement its functionality.

The presented FLV and FMF (Figs. 5, 6 and 8) are, respectively, the values of Fuzzy Linguistic Variables and Fuzzy Membership Functions of Fuzzy Logic. They are used in models (1), (2), (3), (4) to implement functional algorithms of CSSCBMIUU.

2.2 The Fuzzy Situational Computing Model of the Self-Controlled Brain

The all CSSCBMIUU's processes operate on basis of generalized model (1), the description of which is given in [14]. the sub model (2) of model (1) displays a nested set of computational functional modules (agents) interacting with each other under the control of the computational module (agent) of the Situational Psycholinguistic and Cognitive

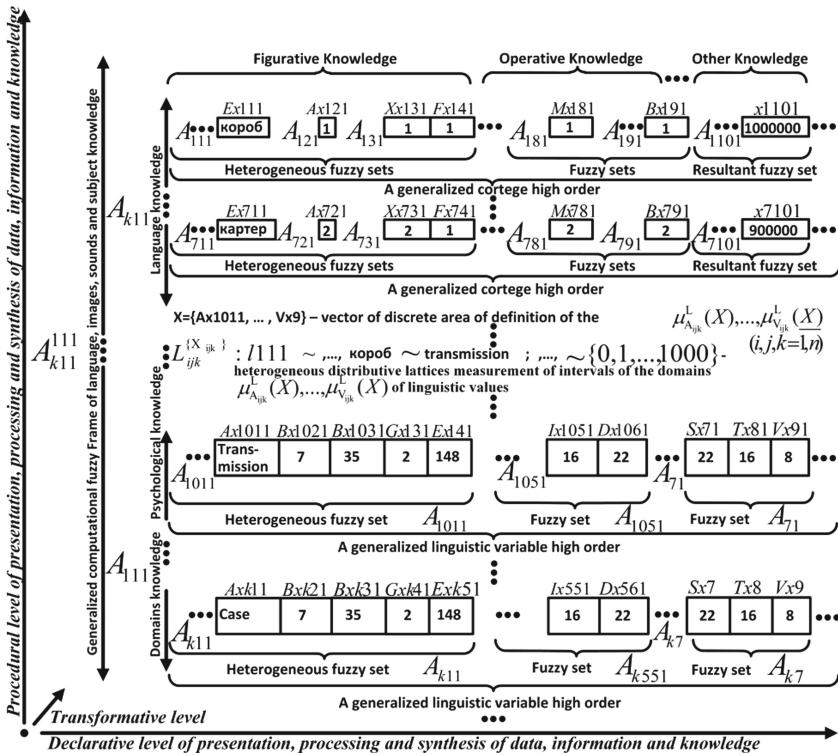


Fig. 5. The CB's memory model of frame semantic network of linguistic, psychological, and thematic repositories of objects.

Fuzzy Control, the aim of which is computing the FMF values using the corresponding FLV, Memory's knowledge, Rules, and Logical Operations of Disjunctions of Conjunctions of all optimal computed values of FMFs, which satisfying to the generated aim in the current situation and under environmental uncertainty. In doing so, the model (3), presents the model of Fuzzy Situational Control (FSC) in CSSCBMIUU which, when it used, implements the control action (decision) I_i which transfers the FSC to the new situation S_i , that represents its new state Q_i , after the state Q_j has passed to Q_i " [14].

The modules in model (2) are generating the Situational Psychological, Linguistical, Cognitive, and Thematic CSSCBMIUU's functionality [14], (Figs. 2 and 9), where: μ_R^Y —a CM's module [5], μ_R^F —a target generation of compositional rules [8], μ_R^Z —a module of computing Knowledge Presentation, Generalization and Explanation [9], μ_R^D —a module for organizing FSC by Dialog and Communication between users and systems [11], μ_R^M —a module of computational Decision making and Planning decisions [7], μ_R^T —a module of computing Reasoning [9], μ_R^H —a module of computing Systemic Thinking [9], μ_R^C —a computing Cognition module [12], μ_R^O —a computing Consciousness module [11], μ_R^Z —a computing Awareness module [10], μ_R^H —a computing Intuition module [13], μ_R^W —a computing Wisdom module [13], μ_R^L —a computing Fuzzy

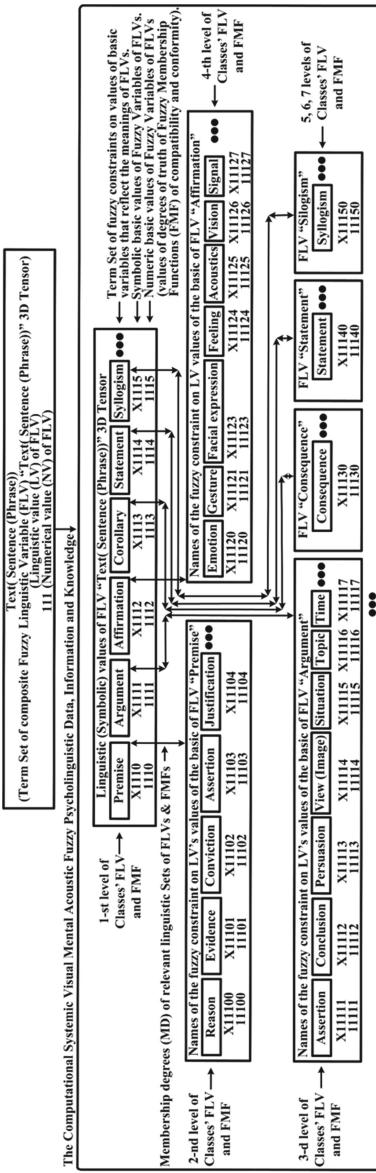


Fig. 6. The CB's and CM's model of frame semantic network of generalized linguistic, psychological, and thematic repositories of objects.

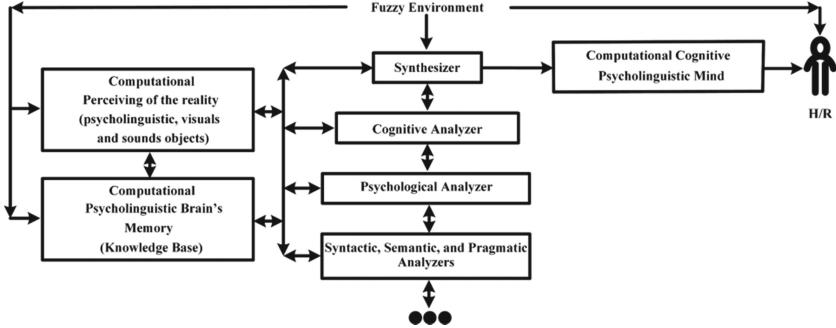


Fig. 7. The general scheme of linguistic psychological and thematic processing and synthesizing of mind in CSSCBMIUU.

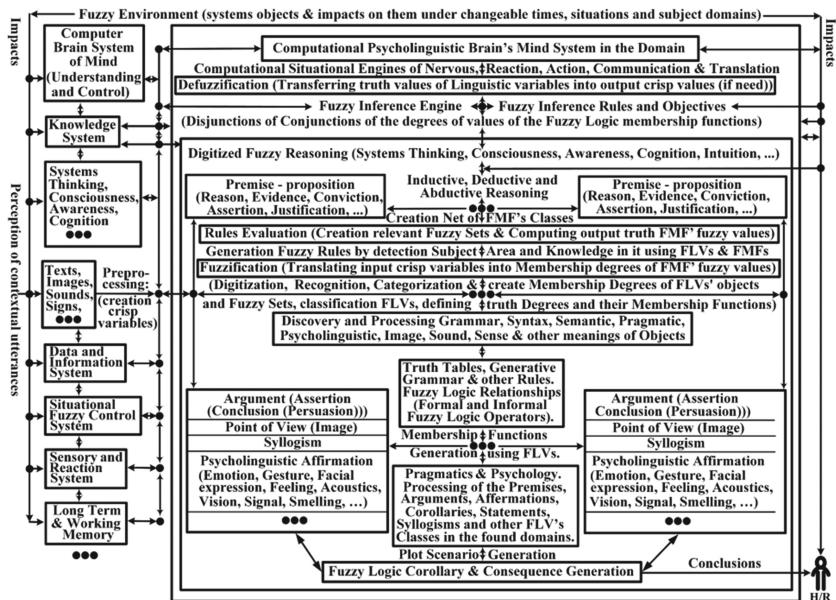


Fig. 8. The detailed functional scheme of linguistic psychological cognitive analysis and synthesis of mind in CSSCBMIUU.

Situational Inference module [6, 14].

$$\mu = \langle A_n^S, K_n^S, F_n^S, M_B^R(x), N_B^R(x) \rangle \quad (1)$$

where:

$$M_B^R(x) = (\mu_R^Y(\mu_R^F(\mu_R^Z(\mu_R^D(\mu_R^M(\mu_R^T(\mu_R^H(\mu_R^O(\mu_R^Z(\mu_R^H(\mu_R^W(\mu_R^L))))))))))))))) \quad (2)$$

$$N_B^R(x) = (S_i : Q_j \xrightarrow{x,u,w} Q_l : I_i) \quad (3)$$

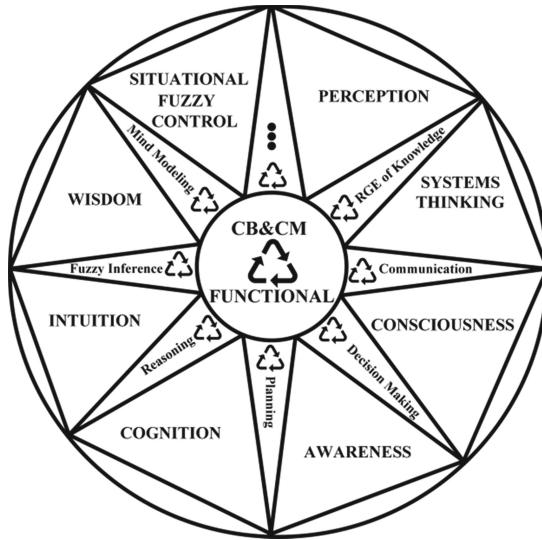


Fig. 9. The CB's and CM's functional in CSSCBMIUU. The Fuzzy Self-Controlled Brain's Model in CSSCBMIUU

In doing so:

S_i —the full , Q_j —current, and Q_l —new situations of states of psychological, linguistical, cognitive, and domain objects of data, information, knowledge [14] in the CSSCBMIUU,

I_i —the created CSSCBMIUU's computational psychological, linguistical, cognitive, and domain rules and actions,

A_n^s —the CSSCBMIUU's Model of Memory of the computational psychological, linguistical, cognitive, and domain objects of data, information, knowledge [14] (Figs. 4, 5 and 6),

K_n^s —the CSSCBMIUU's functionality Model [14] (Figs. 5 and 6),

$M_B^R(X)$ —the CSSCBMIUU's functional Modules' Models (Fig. 2).

The (4) is a CSSCBMIUU's target processes model:

$$F_N^S = u(x, w) \quad (4)$$

where: x , w , and u —are, respectively, CSSCBMIUU's *definite*, *indefinite*, and *residual* fuzzy self-controlled actions for implementing of the functional processes under uncertainty [14].

3 The CSSCBMIUU's Computational Mind Modeling

3.1 The Modeling of Computational Mind via Situational-Fuzzy Self-controlled Brain's Functionality Under Uncertainty

Since all the CSSCBMIUU's functionalities of the modules, presented in model (1) are involved in the modeling of each of its functionality, therefore, the process and the model (1) of modeling of the CSSCBMIUU are similar and unified for any functionality [14], mentioned in the model (1).

In doing so, the CSSCBMIUU's Mind modeling process is described in [12], (Figs. 8 and 10) and are using its functionality (Fig. 9), the models (1), (2), (3), (4), *measures of opportunities* (5), mapping *rules* (6), where:

α_n^{Poss} , β_n^{Poss} —are, respectively, the mappings, defined by input and output generalized multidimensional matrices of psychological, linguistic, cognitive, thematic, and relevant numerical *estimates* (meanings) of fuzzy membership functions $\mu_b^q(X)$, $\mu_a^q(X)$ and relationships R_b^X , R_a^X (Figs. 5 and 6).

A_n^s , A'_n^s —are *current states* of the CSSCBMIUU's models, respectively, previous and after their actualizing.

B_n^s , B'_n^s —are *mapped states* of the CSSCBMIUU's models, respectively, previous and after their actualizing, that are related to the states A_n^s , A'_n^s .

T , Q —are transforms' operators of the CSSCBMIUU's states, that shifts their *current states* A_n^s , B_n^s to their *new states* B'_n^s , A'_n^s .

The mapping *rules* (5) are described in [14] and they are used for mapping of the resultant values of Fuzzy Logic membership functions μ_b^q using their μ_a^q input values and *measures of opportunities* φ_l^q by *fuzzy matching* of fuzzy sets in the fuzzy *relations* [14] R_b^X , R_a^X :

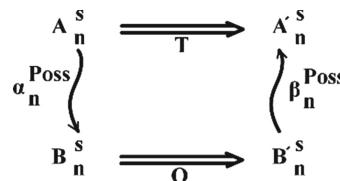


Fig. 10. The diagram of modeling of computational brain and mind functional processes in CSSCBMIUU.

$$\mu_b^q(X) = \mu_a^q(X) \circ \varphi_l^q \quad (5)$$

The matching of base fuzzy sets is implemented by using of compositional mapping rules $\mu_{B_{ijk}}^q(X) = \mu_{A_{L_{ijk}}}^q(X) \circ \Phi_{L_{ijk}}^q$, where: $\mu_{B_{ijk}}^q(X)$, $\mu_{A_{L_{ijk}}}^q(X)$ are, respectively, the *resultant* and *initial* CSSCBMIUU's *membership functions* in the considered generalized fuzzy relation. $R_{ijk}^{X_{ijk}}$, $X = \{x_{ijk}\}$ is a vector of the discrete domain of definition

of resultant and initial CSSCBMIUU's membership functions, \circ —is a *sign* of the computational mapping, $\Phi_{L_{ijk}}^q$ is a fuzzy *matching* in the procedure of CSSCBMIUU, $A_{L_{ijk}}$, $B_{L_{ijk}}$ —are the various multi-dimensional CSSCBMIUU's *fuzzy sets*, L_{ijk} —are the various distributive *lattices* of measurement intervals of the domains of definitions of considered membership functions, q—determine the *levels*, respectively, on *sublevels* of *modeling* of CSSCBMIUU's Semantic Frames, l, k, m = (1, n).

The presented computational CSSCBMIUU's *mapping rules* in model (6) are described in [14], and used for modeling of the CSSCBMIUU, where A, B are Fuzzy Sets.

The rule (7) represents an example of one of the psychological, linguistic, thematic, and cognitive mapping rules that removes the multilingual semantic ambiguity [14].

$$M_B^R(x) = \bigvee (\bigwedge M_A^R(x) \circ \text{Poss } a / a') \quad (6)$$

$$\text{IF}((C'_{xyz} = L'_{xyz}) \text{ AND } (C^*_{xyz} = R^*_{xyz})) \text{ THEN } ((C_{xyz} = R'_{xyz}) \text{ AND } (C^*_{xyz} = L^*_{xyz})) \quad (7)$$

The rule (7) is a conditional logical expression, that was realized in the algorithm for removing semantic ambiguity in sentences and texts of bidirectional machine translation from one natural language to another. This rule contains the meanings of concept codes, left and right compatibility codes in relation to concept codes of words in a bilingual translation situation. The computational codes for identified word groups in multilingual communication, presented in the Neural Network of CSSCBMIUU's Memory, means the following: C'_{xyz} , C^*_{xyz} —the *Conceptual compatibility Codes* of the words, respectively, for input and output languages; L'_{xyz} , L^*_{xyz} —the Codes of the words of *Left compatibilities* in relation to codes of words with *Conceptual compatibilities*, respectively, for the input and output languages; R'_{xyz} , R^*_{xyz} —codes of words of *Right compatibilities* in relation to codes of words with *Conceptual compatibilities*, respectively, for the input and output languages.

4 Conclusion

The creation of a Computational multi-lingual and multi-domain Situationally Self-Controlled Brain and Mind IUI through of application on Cognitive Psychology, Linguistics, Fuzzy Logic, Situational Control, Data Science, and Computer Science leads to novelty in the field of realization of the AI, functioning without of use of human intelligence for reprogramming models, that have lost their relevance under uncertainty and continuously changing in time objects, situations, and models in their surrounding environment. In doing so, the modern IUIs, which are using Machine Learning technology (that function on base of statistic and predictable data) not capable how computationally, systemically, and situationally to think, cognize, infer, self-learn, be aware, possess consciousness and self-control in the unexpected situations, that leads to a doubt about their implementation in complex (non-linear) AI systems. The paper explores the concept, method, and model for creating a Computational Intelligent Multi-Agent Brain and

Mind Interface with Fuzzy Self-Control Under Uncertainty, that has functionality and intelligence plausible to human intelligence, capable of perceiving the current situation and objects in the surrounding environment in time, by interacting with these objects (humans, devices) in order to help them solve their tasks by connecting to this IUI as plug-in.

References

1. Zadeh, L.: Fuzzy sets. *Inf. Control* **8**, 338–353 (1965)
2. Zadeh, L.: The concept of a linguistic variable and its application to approximate reasoning. *Inf. Sci.* **14**, 141–164 (1995)
3. Jones, N., Ross, H., Lynam, T., Perez, P., Leitch, A.: Mental models: an interdisciplinary synthesis of theory and methods. *Ecol. Soc.* **16**(1), (2011)
4. Khayut, B., Fabri, L., Avikhana, M.: Modeling of Computational Systemic Deep Mind Under Uncertainty. In: 8th International Conference on Complex Adaptive Systems, pp. 253–258. USA, (2016)
5. Khayut, B.: Modeling of Fuzzy Logic Inference in decision-making system. Modeling system, Institute of Mathematics of the Moldavian Academy of Science **110**, 134–143 (1989)
6. Khayut, B., Fabri, L., Avikhana, M.: Modeling, planning, decision-making, and control in fuzzy environment. In: Advance Trends in Soft Computing Conference., vol. 312, pp. 137–143. Springer, USA, (2013)
7. Khayut, B., Fabri, L., Avikhana, M.: Intelligent multi-agent fuzzy control system under uncertainty. *J. Comput. Sci. Inf. Technol.*, **4**(18), 369–380 (2014). USA
8. Khayut, B., Fabri, L., Avikhana, M.: Knowledge representation, reasoning and system thinking under uncertainty. In: 16th International conference on computer modeling and simulation, pp 119–128. Cambridge, UK, (2014)
9. Khayut, B., Fabri, L., Avikhana, M.: A self-developing computational system of full awareness and understanding of reality. In: ISAE-MAICS Conference, pp. 37–42. Spokane, USA, (2018)
10. Khayut, B., Fabri, L., Avikhana, M.: Toward general AI: Consciousness computational modeling under uncertainty. In: 2nd International Conference on mathematics and computers in science and engineering (MACISE), pp 90–97. Madrid, Spain, (2020)
11. Khayut, B., Fabri, L., Avikhana, M.: A computational intelligent cognition system under uncertainty. In: Sixth International Congress on Information and Communication Technology (ICICT 2021), vol. 235, pp. 127–136, Springer, Singapore, (2021)
12. Khayut, B., Fabri, L., Avikhana, M.: A computational Psycholinguistic System of Intuition and Wisdom Under Uncertainty. In: Sixth International congress on information and communication technology, lecture notes in networks and systems, vol. 235, pp. 127–136. Springer Nature Singapore, (2022)
13. Pospelov, D.: Situational Control. **288**, (1986)
14. Khayut, B., Fabri, L., Avikhana, M.: Computational system of psycholinguistic fuzzy inference under uncertainty. In: 2022 International conference on innovations in intelligent systems and applications (INISTA), (2022)



Ethical Concerns About Personhood, Responsibility, and Privacy in Active and Passive Brain-Computer Interfaces

Ronja Rönnback^(✉), Fenna Blom, and Maryam Alimardani

Tilburg University, Warandelaan 2, 5037 Tilburg, AB, The Netherlands
r.g.i.ronnback@tilburguniversitu.edu

Abstract. Brain Computer Interfaces (BCIs) are intelligent systems that enable direct communication between the human brain and machines. While BCI systems are promising for future medical and non-medical applications, studies concerning their ethical considerations are growing. However, no previous study has examined how the public's ethical perception of the BCI technology is affected by the particular BCI type in question. This study thus considered whether the public experienced active and passive BCIs differently in the prominent ethical domains of personhood, responsibility and privacy. Results suggest that active BCIs induce a higher ethical concern regarding personhood, and that women experienced privacy to be more concerning in passive BCIs. There were no other significant differences between the two BCI types in the examined ethical domains. A regression analysis also indicated that a person's general ethical concern for BCIs was unaffected by their demographical information. This study provides preliminary insights for the development of ethically informed BCI systems.

Keywords: Brain-Computer Interface (BCI) · Ethics · Active and passive BCIs · Personhood · Responsibility · Privacy

1 Introduction

Brain-Computer Interfaces (BCIs) are an emerging technology that allow for direct communication between the human brain and external machines. They achieve this by recording brain activity (often via electroencephalography) and decoding the signals into corresponding output or commands [1]. For example, it is possible to translate brain signals associated with a user's motor intentions to movements of a robot [4, 6] or link the perception of errors (the so-called Error-Related Potential, ErrP) to a change in system behavior [1]. The technology is being steadily developed, and BCIs have in the recent years been applied to numerous domains such as healthcare [6, 7], education [29] and art [33] to only name a few.

In recent years, BCI ethics has received a lot of attention. Particular focus has been levied on ethical concerns related to privacy, responsibility and personhood [9, 36], which are briefly defined for the context of this paper in the Table 1.

Previous studies have examined how groups such as BCI experts [15, 24], software developers [22], rehabilitation professionals [23], BCI users [15, 19] and the general public [27, 28] perceive BCIs in such various ethical domains. However, these topics of investigation all relate to how various user groups perceive BCIs quite generally, but make no distinction based on the type of BCI.

BCI systems can be divided into various types depending on their functionality, neuroimaging technique or the level of user's control [37]. For instance, invasive and non-invasive BCIs are one such division, referring to whether or not the brain activity for the BCI system is recorded from within the brain or skull, requiring risky surgical installation [1]. The ethical perceptions of specifically invasive BCIs in rehabilitation have previously been researched, and while the study mentioned privacy concerns, it mainly focused on other aspects such as information transfer, participant selection, user motivation, distress stemming from training sessions and BCI illiteracy, negative effects of BCI-induced brain plasticity (which somewhat roughly relates to our concept of personhood), end of study problems, and security of the system [15].

However, another categorization of BCI systems that has been overlooked in the past ethics studies is the division between active, reactive and passive BCIs. This division relates to a user's control on their brain signals and hence the BCI output. In short: **Active BCIs** produce output based on the user's voluntary modulation of brain activity to control an external device [37]. A common active BCI application is motor imagery BCI which involves a user imagining the movement of a specific body part, e.g., left arm, to generate recognizable brain patterns for the BCI system and move a left robotic arm [4]; **Passive BCIs** decode spontaneous cognitive states such as sleep stage, stress level, fatigue or emotions without the user performing a specific BCI task [3, 37]. A passive BCI could be used to unobtrusively monitor an individual's mental states during daily activities, for example, evaluate how stressed a doctor is during an operation, or to detect if a driver is falling asleep; **Reactive BCIs** use spontaneous brain signals generated as a reaction to external stimuli [37]. A common reactive BCI is the P300-speller using which the user can select a target letter on a digital keyboard by focusing their visual attention [3].

Active BCI systems are generally used for controlling robotic prostheses or exoskeletons to perform movements [1, 6, 16], calling into question personhood and agency [9, 36] as well as responsibility over actions [8]. Passive BCIs' use cases center around continuous monitoring of mental states, and thus the possibility for "mind-reading" poses severe privacy issues [9, 36]. However, Reactive BCI paradigms are commonly used as communication tools [1] or as mere sub-components of active or passive BCI systems [17, 18, 23, 35]. Thus, these systems may be considered as more benign compared to active or passive paradigms, especially since the application areas for active and passive BCIs are more susceptible to ethical concerns and deemed as high-risk according to the European Commission's Artificial Intelligence Act's Annex 3 [10]. Such concerns are particularly present in the three ethical domains of interest to this study (personhood, responsibility and privacy), as is discussed in the following section.

Although the ethical perception of the BCI subtypes, e.g., active vs. passive, has not been systematically investigated, past literature has theorised that the public's ethical concerns may differ: Schmid et al. [28], in their survey of public BCI perception, posited that the increasing adoption of passive BCIs could raise further discussion about personhood and responsibility. Burwell et al. [9] note the diverging worries surrounding privacy and the potential for “mind-reading” using BCIs. Furthermore, Bublitz et al. [8] suggest that actions executed through a passive BCI do not constitute as legally liable and responsible actions because they are not willfully controlled. Thus, unique ethical challenges arise for active and passive subtypes in particular.

Additionally, it is essential to consider an individual's background and demographics when examining their perception of new technologies. Sample et al. [27], for example, found that women generally reported higher concerns for BCI ethical domains than men. Such an effect was, however, not researched in the study of Schmid et al. [28]. The results of these two studies are discussed in detail in the next section, but confirming whether such gender effect remains persistent for different BCI types calls for further investigation.

The current paper focuses exclusively on the ethical implications of active versus passive BCIs and how individuals' demographics and technical background could impact their perception of ethical concerns associated with this novel technology. Ensuring that users are comfortable with and able to trust BCI applications is critical for their mainstream adoption and continued use in the future. Additionally, awareness of individual differences in technology perception could assist future developers when designing BCI interactions as well as policymakers when assessing the needs and concerns of different user groups. This latter one is particularly important for effective legislation and regulation of emerging neurotechnology, as regulation that is either too cautious or overly permissive may either stifle innovation or allow for the proliferation of potentially harmful technologies [11, 14]. Accordingly, the central research question of this study is:

- **RQ: Do ethical concerns differ between active and passive BCIs?**

In addition, two sub-questions were investigated:

- *Sub-RQ1: Is there a gender difference in ethical concern for each BCI type?*
- *Sub-RQ2: Can demographics and BCI type be used to predict the ethical concern?*

2 Literature Review

2.1 Ethical Domains

The three ethical domains chosen as focus points in this project are personhood, responsibility and privacy. These were selected due to their overarching prevalence in the relevant literature. For example, they were reported by Burwell et al. [9] as the primary ethical concerns related to BCI usage, and also appeared

Table 1. Definitions of ethical domains of interest with respect to BCI technology.

Ethical domain	Description
Personhood	Personhood encompasses discussions about how much BCIs integrate with the human body and how the user's humanity might be impacted by the usage of BCIs [9].
Responsibility	Responsibility relates to how users (or other stakeholders) are morally and legally responsible for the BCI modulated actions and where the burden of liability lies should something go awry [9, 36].
Privacy	Privacy concerns the usage of personal data collected through BCIs and the possible dangers of its misuse [9, 36]. Due to the novelty of BCIs, many worry about the prospect of mind-reading, although it is uncertain whether this is at all possible and to what extent if so.

in previous public attitude screenings [27, 28], although sometimes with slightly different names [36]. An exhaustive definition for each of these ethical domains is outside the scope of this paper, however a brief description and what they mean in the context of neurotechnology and BCI systems are presented in Table 1.

Previous literature has commonly measured concerns about such ethical concepts through interviews [15, 19, 22–24] or questionnaires [2, 24, 27, 28]. Previous public opinion surveys [27, 28] exclusively use questionnaires. Notably, Sample et al. [27] conducted a survey examining public opinion of BCIs from an ethical perspective, as well as the effect of demographics. Demographics of main interest were gender, religiosity, age, education, country of residence, disability and previous BCI experience. Subjects were provided a short general description about BCIs, but no direct BCI interaction. The briefing used examples of active BCIs only. Users then rated different statements with relevant examples on a scale from “not concerning” to “extremely concerning”. The statements covered a wide variety of ethical domains, ranging from privacy to cyborg-related concerns. Results indicated that there was a moderately high level of concern about BCI ethical issues held by the general public, in particular by female respondents and individuals who self-reported high levels of religiosity.

Another public opinion survey by Schmid et al. [28] assessed their sample’s affinity for technology, previous knowledge or experience with BCI as well as ethical attitudes towards BCIs, measured through Likert-scale items on a scale from “completely disagree” to “completely agree.” Participants were informed through a short general video about BCI technology and its use. They specifically saw examples of active BCI use cases. Regrettably, the study did not investigate whether demographics influenced ethical opinions, making it difficult to estimate concerns for particular user groups. However, Schmid et al. [28] hypothesized at the end of their paper that, due to general focus on active BCIs in previous ethical public opinion surveys, ethical considerations may differ when involving passive BCIs.

2.2 Ethical Concerns in Active and Passive BCIs

Previous research has only partially explored or mentioned how active and passive BCIs differ in terms of ethical perception by public, but the literature already points to some recurrent issues surrounding these BCI subtypes. Regarding personhood, concerns of “cyborgization” are common for both BCI types [9, 15, 23, 27, 36]. Studies involving active BCIs seem to note concerns about neurological changes in identity specifically brought about through repeated use of a BCI system [9, 15, 24].

Perceptions of responsibility seems to differ as well. Previous research determined that a motor imagery BCI (an active BCI which uses imagined body movement as a signal) can give rise to the illusion of ownership over a prosthetic hand by aligning intentions to move and visual outcome (said hand either moving or not moving), indicating that active BCIs can indeed significantly influence our sense of self, agency and responsibility over BCI-induced actions [4]. With respect to passive BCIs, however, Bublitz et al. [8] renounce responsibility for actions caused by passive BCIs altogether, as such outputs are the result of spontaneous activity and not voluntary modulations of brain patterns. Indeed, previous research contends that users will tend to feel an increased sense of responsibility when intentions, sensory-motor cues, and outcomes match [31]. This may occur in active BCIs, but passive BCIs do not involve such explicit intentions—users cannot necessarily control their mental state. These previous studies raise the possibility that each subtype would have a different effect on notions of responsibility.

Privacy is a major concern in passive BCIs in particular [24]. Nijboer et al. [24] note that the use of passive BCIs might raise more alarming privacy issues, as there are many scenarios where information captured by passive BCIs is sensitive. For instance, a controversial case took place in a Chinese school, where BCI headsets were employed to track students’ attention levels and shared a summary of it with teachers [29, 34]. Students reported feeling pressure if their parents saw low concentration levels in school reports [34]. Schmid et al. also note that current BCI experiments commonly focus on motor imagery paradigms (an active BCI application), and hence “do not directly interact with more intimate elements of a person, such as their emotions, their moral values and religious beliefs [28, p. 309]. These aspects may thus make passive BCIs more threatening to privacy than active BCIs. Whereas active BCIs inherently involve voluntary directed action by a user, passive BCIs function as monitors, collecting information such as emotional state, stress level or workload [20, 21]. This spontaneous activity is, naturally, more difficult for a user to control or conceal. Therefore, active and passive BCIs may result in different perceptions of threats to privacy.

In sum, it is crucial to examine whether a distinction is made by naive users from general population between active and passive BCIs. To date, no research has systematically compared these two BCI subtypes in terms of ethical concerns. Therefore, the current study aimed to examine users ethical perceptions in the domains of personhood, privacy and responsibility between active and passive BCIs.

3 Methods

3.1 Subjects

Data were collected from 34 university students in the Netherlands. The sample included 17 males and 17 females with no previous BCI experience, between 19 and 36 years old ($M = 25.3$, $SD = 3.9$). Of these participants, 26 reported not having heard of BCIs before.

3.2 Materials

The general population's perception of ethical concerns toward active and passive BCIs was evaluated using a Qualtrics questionnaire, focusing on the main three ethical domains: privacy, personhood, and responsibility.

The questionnaire consisted of a subset of items from four different questionnaires used in different studies with a 5-point Likert scale. These were the Users Information Privacy Concerns (UIPC) questionnaire [2], the Public Attitudes Towards BCIs (PATBCI) [27], the Frankenstein Syndrome Questionnaire (FSQ) [30] and the “Bevölkerungsumfrage zum Thema Gehirn-Computer-Schnittstellen” [Public survey on the topic of Brain Computer Interfaces] (PSBCI) [28]. The subset of items used from these questionnaires can be seen in Table 2.

Table 2. Collection of questionnaire items selected for each ethical domain per source. Items are also cited in full in Appendix A

	UIPC	PATBCI	FSQ	PSBCI
Personhood		M, P, Q, R, T, U, W	1, 4, 8	Table 5: 1, 3
Responsibility		F	12	Table 5: 2, Table 4: 1, 2
Privacy	PS1, PS3, PV1, PV3, UIPC1-2, UIPC4-5	A, E		Table 6: 3

Additionally, participants general attitude toward technology (Affinity for Technology Interaction (ATI)) [13] was collected, similarly to metrics of affinity for technology collected in [28]. As recommended by Franke et al. [13], a reliability test was done, wherein the mean results from the ATI test indicated a good reliability of the ATI measure in our sample ($M = 3.4$, $SD = 0.6$, $\alpha = 0.77$).

3.3 Procedure

Participants were briefed on the procedure and asked to sign a consent form, after which they received an introductory text about BCIs. They were then provided explanations of either active or passive BCIs, presented in randomised order, followed by questions targeting the three ethical domains, also in random order (see Fig. 1).

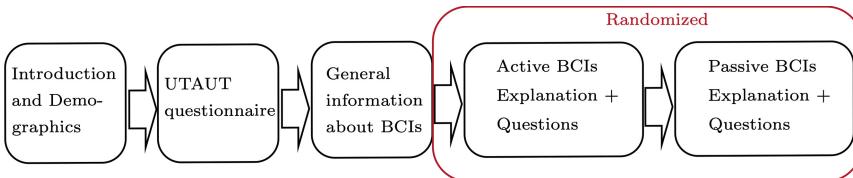


Fig. 1. Schematic outline of the survey procedure.

3.4 Analysis

All analyses were done in R [26]. Different exclusion criteria were used for data cleaning. Participants could not have any previous experience with BCIs, should not have given the same responses for all questions within a category, and should not have taken longer than the average response time ($M = 935$ s) plus three times the standard deviation ($SD = 697.1$). Incomplete trials were also excluded.

To obtain a numeric score representing “ethical concern” for each ethical domain, responses were aggregated by summing items in each ethical domain, followed by a min-max normalization. These values were used in all subsequent analyses.

The two conditions (active and passive BCI) were thusly compared per ethical domain using a paired-samples t-test or a Wilcoxon test if the data distribution was not normal. This comparison was recomputed in males and females separately in order to examine if a gender difference was found in our sample. This helped in answering the first sub-question.

To further investigate the relationships in our data, a linear mixed-effects model was run using the lme4 package in R [5]. This examined whether gender, ATI, previous knowledge of BCI, and the BCI type can predict ones general ethical concern for the BCI technology. This regression was implemented to answer the second sub-question. No interaction effects between the independent variables were considered. The general concern was defined as the summed ethical concern of all three categories, and is referred to as “General Ethical Concern” henceforth. Because the goal of this analysis was to study the effect of BCI type on the General Ethical Concern (Sub-RQ2) and not the concern per domain, the data from ethical domains were aggregated and considered as one dependent variable.

4 Results

The level of concern for active and passive BCIs was compared using paired-samples t-tests for all ethical domains except for personhood values which were non-normally distributed ($V = 0.9$, $p < 0.05$), and thus were analysed with a paired Wilcoxon test. The results indicated a significant difference in concerns regarding personhood, with active BCIs being rated as slightly more concerning ($M = 21.7$, $SD = 5.7$) than passive BCIs ($M = 20.2$, $SD = 5.5$); $V = 349.5$, $p < .05$. None of the other ethical domains were significantly different (Fig. 2).

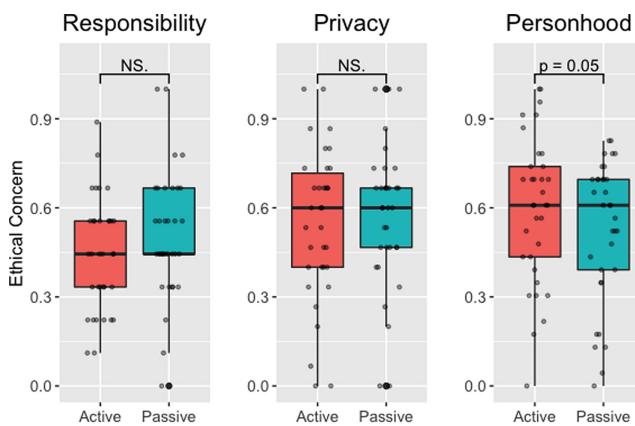


Fig. 2. Boxplots of the normalized total score per ethical domain, for each BCI subtype. A significant difference between the two subtypes was found for personhood only.

The subsequent analysis involved the same comparison, but focused on either female or male participants. Only among female participants, privacy in passive BCIs ($M = 19.4$, $SD = 3.9$) was seen as more concerning than active BCIs ($M = 18$, $SD = 4.4$); $t(16) = -2.4$, $p < .05$ (Fig. 3). No significant differences were found in other ethical domains, for neither males nor females.

The linear mixed-effects model was fitted, predicting the general ethical concern as a function of gender, ATI, previous knowledge of BCIs, and BCI type. The random effect of the model was the respondent id. None of the coefficients were significant (see Table 3).

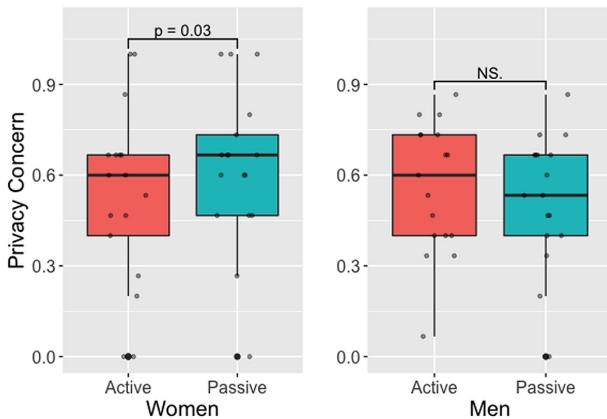


Fig. 3. Boxplots of the privacy concerns for men and women. There was a significant difference found for women only, with passive BCIs presenting a higher privacy concern.

Table 3. Results of the linear mixed-effects model predicting general ethical concern.

Predictor	b	se	t	p
Gender Male	1.1	2.3	0.48	0.64
ATI	1.71	1.92	0.89	0.38
BCI heard: Yes	-2.97	2.8	-1.06	0.3
Type: Passive	-0.06	1.11	-0.06	1.0

5 Discussion

This study investigated the differences in ethical concerns about active and passive BCIs. More specifically, it focused on three different categories of concerns: personhood, responsibility, and privacy. This was measured with a questionnaire, which was distributed among university students in the Netherlands.

Though the difference in ethical concerns between active and passive BCIs was not as evident as initially expected, a distinction could be made between the two subtypes when personhood was concerned. Active BCI applications were found to be slightly more concerning than their passive counterparts on this particular topic, indicating some general worry from the population echoed in previous literature [9, 15, 23, 24, 27, 36], validating the main research question.

The second analysis indicated that there was some difference between male and female participants, corroborating sub-RQ1. Women in particular were more concerned about privacy in passive BCIs than active BCIs, whereas men made no significant difference between the two. These results mirror previous studies

which found that women were more worried about ethical issues in BCIs [27] than men, an effect also found in other, more general settings [12]. This has been attributed to different social and cultural factors facing the two genders and has previously been extended to race as well [12]. Although this difference is interesting, threats to privacy through BCIs must be strongly addressed regardless of gender or other demographic characteristics. It does, however, highlight the need for a diverse subject pool when estimating public acceptance of a BCI and technology in general. Failing to properly represent groups with different concerns might, in the worst case, lead to serious apprehensions being neglected, thereby deterring some users from adopting the technology.

As opposed to the analyses on differences per ethical domain, no significant results were found with the linear mixed-effects model. Although it is possibly caused by the low sample size in this study, the data currently do not provide statistically significant evidence that gender, ATI, BCI knowledge, and BCI type can predict ethical concern. Thus, for now, the sub-RQ2 is refuted.

Overall, the findings of the current study suggest that the two types of BCIs might require somewhat different considerations for the public's apprehensions. Concerning active BCIs, it seems that the presence of a new communication channel, as provided by an active BCI, raises concerns for personhood. Indeed, it adds a completely new dimension to interacting with and influencing our environment. Generally, our use of technology is known to significantly influence and change ourselves as well [32], of which the public seems aware and weary. This threat of "cyborgization" may thus be a part of why active BCIs in particular are seen as threatening to personhood.

Interestingly, privacy and personhood generally raised more concerns than responsibility. This is in line with results of Schmid et al. [28], which suggested that personhood was one of the top concerns of their participants, but that respondents seemed more uncertain or ambivalent about responsibility. Despite these mixed results, their participants did indicate a strong desire for regulations. Contrarily to these results, Sample et al. [27] found that responsibility was firmly amongst the more concerning ethical issues, higher than personhood.

Our mean values for responsibility were roughly centred around the "neither agree nor disagree" option, suggesting a similar ambivalence or hesitation in our sample as in that of Schmid et al. [28]. The exact reason for this is difficult to determine from our results alone, and would benefit from a future study focused on this topic. It is presumable that it may stem from a general lack of knowledge of the intricacies of the subject, making users hesitant to affiliate themselves to one or the other. More analyses and discussion with a well-informed public was recommended by Schmid et al. [28] and now by us, as it is crucial for future regulations to align with the publics conceptions of appropriate responsibility and consequently justice.

A limitation of our study is that, while our results indicate some interesting starting points for the development of ethically informed BCIs, there are still many facets we did not investigate. Concerns about justice, user safety, stigma, autonomy and research ethics were cited as other key ethical issues in the field

[9], but are not all explored here. This might be an interesting opening for future studies. Additionally, we used a combination of questionnaire items from multiple studies. A future study might benefit from the creation of a custom questionnaire for more accurate and representative results, though we did restrict ourselves to previously validated questionnaires. Our sample was also restricted to a relatively small sample of university students. It is likely that respondents of different age groups might differ in their judgements on the ethical domains covered here, as this has been the case in some other related studies on acceptance of new technologies in different age or demographic groups [25].

All in all, these results highlight some particular topics that need to be addressed when developing future, human-centric BCIs. Concerns about personhood in active BCIs and privacy in passive BCIs ought to be taken seriously. On the other hand, it seems that issues of responsibility are still unclear to the public, or that they are as of now ambivalent. This highlights a need for more investigation, especially for the creation of appropriate ethical codes or regulations. Future studies could examine the uncertainty about responsibility, explore other noteworthy ethical concerns or, importantly, what practical solutions could be used to address said concerns of the public. Interdisciplinary studies between BCI and Human-Centred Design could be an important avenue to explore this last question, as there is already some research done into practical guidelines of human-centric technology development. It is important to develop and propose concrete suggestions for ethical BCI implementations which are informed by normative analyses. Currently, there seem to be few papers with such practical guidelines specifically for BCIs.

6 Conclusion

While there is a growing discussion about ethical BCI concerns, previous literature restricted itself to covering BCIs quite generally. This study analysed differences in ethical concerns about active versus passive BCI technology. Three categories of ethical concerns were selected due to their proliferation in the literature, namely personhood, responsibility and privacy. The results suggest that active and passive BCIs are viewed differently and may require separate ethical considerations for mainstream adoption by the public. A higher concern was observed for personhood and privacy than responsibility, and women reported a higher concern for privacy in passive BCIs. This study provides preliminary insights for the development of ethically informed BCI systems. Future studies are required to expand these findings within larger and more diverse samples, and also investigate other ethical challenges in BCIs, such as justice, safety, and stigma.

Appendix A: Questionnaire Items

Table 4. Questionnaire items used in the Users' Information Privacy Concerns (UIPC) questionnaire [2]. Some items were slightly modified to fit the current study, as is shown by the text within brackets.

Item ID	Full text
PS1	Losing information privacy through [BCIs] would pose serious problems for me.
PS3	Misuse of personal information available [through BCIs] would pose serious problems for me.
PV1	I could potentially suffer from malicious online security issues (e.g., privacy intrusions, virus attacks, etc.) [through BCIs].
PV3	I feel it is not safe to share personal information [to BCIs].
UIPC1	I am concerned that my personal information [through BCIs] could be used for wrong purposes.
UIPC2	I am concerned that my personal information [through BCIs] could be accessed by unknown parties.
UIPC4	I feel BCIs are collecting excessive personal information.
UIPC5	I am concerned that my personal information [through BCIs] could be used in a manner I am unaware of.

Table 5. Questionnaire items used in the Frankenstein Syndrome Questionnaire (FSQ) [30]. Some items were slightly modified to fit the current study, as can be seen in brackets.

Item ID	Full text
1	[BCIs] will make us forget what it is like to be human.
4	[BCIs] may make us lazier.
8	[BCIs] will encourage less interaction between humans.
12	If [BCIs] cause accidents or trouble, persons and organizations related to development of them should give sufficient compensation to the victims.

Table 6. Questionnaire items used in the Public Attitudes Towards BCIs (PATBCI) questionnaire [27]. Some items were slightly modified to fit the current study.

Item ID	Full text
A	Malicious persons, companies, or governments may target BCIs to harm the user.
E	BCIs may provide unprecedented access to things that are normally private and record them.
F	It is currently unclear if and when BCI users will be held responsible for his or her actions.
M	Using a BCI might lead to noticeable changes in ones personality and self-understanding.
P	BCI users may seem to lose their humanity.
Q	BCIs could change the very idea of humanity.
R	Persons who choose to use a BCI might find themselves being treated differently by family, friends, and strangers.
T	BCI devices may increase the stigma of disability.
U	A BCI user may wonder is this really me?.
W	BCI users might seem to be a mix of machine and person.

Table 7. Questionnaire items used in the public survey on the topic of BCIs (PS-BCI) [28]. Some items were slightly modified to fit the current study, as can be seen in brackets.

Item ID	Full text
Table 4: 1	You should need a license for BCI use.
Table 4: 2	The use of BCIs should be regulated by law.
Table 5: 1	A BCI-modulated action is still a human action.
Table 5: 2	The user is responsible for BCI-modulated actions.
Table 5: 3	BCI users are a mixture between man and machine.
Table 6: 3	My thoughts could be read by a BCI.

References

1. Abiri, R., Borhani, S., Sellers, E.W., Jiang, Y., Zhao, X.: A comprehensive review of eeg-based brain-computer interface paradigms. *J Neural Eng* **16**(1), 011001 (2019)
2. Adhikari, K., Panda, R.K.: Users' information privacy concerns and privacy protection behaviors in social networks. *J. Glob. Mark.* **31**(2), 96–110 (2018)
3. Alimardani, M., Hiraki, K.: Passive brain-computer interfaces for enhanced human-robot interaction. *Front. Robot. AI* **7**, 125 (2020)
4. Alimardani, M., Nishio, S., Ishiguro, H.: Humanlike robot hands controlled by brain activity arouse illusion of ownership in operators. *Sci. Rep.* **3**(1), 1–5 (2013)
5. Bates, D., Mächler, M., Bolker, B., Walker, S.: Fitting linear mixed-effects models using lme4 (2014). [arXiv:1406.5823](https://arxiv.org/abs/1406.5823)

6. Benabid, A.L., Costecalde, T., Eliseyev, A., Charvet, G., Verney, A., Karakas, S., Foerster, M., Lambert, A., Morinière, B., Abroug, N., et al.: An exoskeleton controlled by an epidural wireless brain-machine interface in a tetraplegic patient: a proof-of-concept demonstration. *Lancet Neurol* **18**(12), 1112–1122 (2019)
7. Bockbrader, M.A., Francisco, G., Lee, R., Olson, J., Solinsky, R., Boninger, M.L.: Brain computer interfaces in rehabilitation medicine. *PM&R* **10**(9), S233–S243 (2018)
8. Bublitz, C., Wolkenstein, A., Jox, R.J., Friedrich, O.: Legal liabilities of bci-users: Responsibility gaps at the intersection of mind and machine? *Int. J. Law Psychiatry* **65**, 101399–101399 (2018)
9. Burwell, S., Sample, M., Racine, E.: Ethical aspects of brain computer interfaces: a scoping review. *BMC Med. Ethics* **18**(1), 1–11 (2017)
10. European Commission.: Proposal for regulation of the european parliament and of the council - laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts. European Commission (2021)
11. European Commission. Regulatory framework proposal on artificial intelligence (2022)
12. Finucane, M.L., Slovic, P., Mertz, C.K., Flynn, J., Satterfield, T.A.: Gender, race, and perceived risk: The 'white male' effect. *Health, Risk Soc.* **2**(2), 159–172 (2000)
13. Franke, T., Attig, C., Wessel, D.: A personal resource for technology interaction: development and validation of the affinity for technology interaction (ati) scale. *Int. J. Hum.-Comput. Interact.* **35**(6), 456–467 (2019)
14. Glauner, P.: An assessment of the ai regulation proposed by the european commission. In: *The Future Circle of Healthcare*, pp. 119–127. Springer, Berlin (2022)
15. Grübler, G., Al-Khodairy, A., Leeb, R., Pisotta, I., Riccio, A., Rohm, M., Hildt, E.: Psychosocial and ethical aspects in non-invasive eeg-based bci research-a survey among bci users and bci professionals. *Neuroethics* **7**(1), 29–41 (2014)
16. Guger, C., Harkam, W., Hertnaes, C., Pfurtscheller, G.: Prosthetic control by an eeg-based brain-computer interface (bci). In: *Proceeding aaate 5th European Conference for the Advancement of Assistive Technology*, pp. 3–6. Citeseer (1999)
17. Kaongoen, N., Jo, S.: A novel hybrid auditory bci paradigm combining assr and p300. *J. Neurosci. Methods* **279**, 44–51 (2017)
18. Khan, M.J., Hong, K.-S., Naseer, N., Raheel Bhutta, M.: Hybrid eeg-nirs based bci for quadcopter control. In: *2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 1177–1182 (2015)
19. Kögel, J., Jox, R.J., Friedrich, O.: What is it like to use a bci?-insights from an interview study with brain-computer interface users. *BMC Med. Ethics* **21**(1), 1–14 (2020)
20. Krol, L.R., Zander, T.O.: Passive bci-based neuroadaptive systems. In: *GBCIC* (2017)
21. Lotte, F., Roy, R.N.: Brain-computer interface contributions to neuroergonomics. In: *Neuroergonomics*, pp. 43–48. Elsevier (2019)
22. Merrill, N., Chuang, J.: From scanning brains to reading minds: Talking to engineers about brain-computer interface. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–11 (2018)
23. Astobiza, A.M., Arias-Vailhen, D.R., Ausín, T., Toboso, M., Aparicio, M., López, D.: Attitudes about brain-computer interface (bci) technology among spanish rehabilitation professionals. *AI & SOCIETY*, pp. 1–10 (2021)

24. Nijboer, F., Clausen, J., Allison, B.Z., Haselager, P.: The asilomar survey: Stakeholders' opinions on ethical issues related to brain-computer interfacing. *Neuroethics* **6**(3), 541–578 (2013)
25. Nomura, T., Sugimoto, K., Syrdal, D.S., Dautenhahn, K.: Social acceptance of humanoid robots in Japan: a survey for development of the frankenstein syndrome questionnaire. In: 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012), pp. 242–247. IEEE (2012)
26. R Core Team.: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria (2021)
27. Sample, M., Sattler, S., Blain-Moraes, S., Rodríguez-Arias, D., Racine, E.: Do publics share experts' concerns about brain-computer interfaces? a trimnational survey on the ethics of neural technology. *Sci Technol Hum Values* **45**(6), 1242–1270 (2020)
28. Schmid, J.R., Friedrich, O., Kessner, S., Jox, R.J.: Thoughts unlocked by technology—a survey in germany about brain-computer interfaces. *NanoEthics* **15**(3), 303–313 (2021)
29. Standaert, M.: Chinese primary school halts trial of device that monitors pupils' brainwaves. *The Guardian*
30. Syrdal, D., Nomura, T., Dautenhahn, K.: The frankenstein syndrome questionnaire—results from a quantitative cross-cultural survey. In: *Social Robotics*, pp. 270–279. Springer International Publishing (2013)
31. Tausen, B.M., Miles, L.K., Lawrie, L., Macrae, C.N.: The role of perspective in self-perceptions of responsibility: It wasn't me. *Conscious. Cogn.* **63**, 89–98 (2018)
32. Verbeek, P.-P.: *What Things Do, In What Things Do*. Penn State University Press (2021)
33. Wadeson, A., Nijholt, A., Nam, C.S.: Artistic brain-computer interfaces: state-of-the-art control mechanisms. *Brain-Comput. Interfaces* **2**(2–3), 70–75 (2015)
34. Wang, Y., Hong, S., Tai, C.: China's efforts to lead the way in ai start in its classrooms. *Wall Str. J.*
35. Wang, Z., Yang, Yu., Ming, X., Liu, Y., Yin, E., Zhou, Z.: Towards a hybrid bci gaming paradigm based on motor imagery and ssvep. *Int. J. Hum.-Comput. Interact.* **35**(3), 197–205 (2019)
36. Yuste, R., Goering, S., Bi, G., Carmena, J.M., Carter, A., Fins, J.J., Friesen, P., Gallant, J., Huggins, J.E., Illes, J. et al.: Four ethical priorities for neurotechnologies and ai. *Nature* **551**(7679), 159–163 (2017)
37. Zander, T.O., Kothe, C.: Towards passive brain–computer interfaces: applying brain–computer interface technology to human–machine systems in general. *J. Neural Eng.* **8**(2), 025005 (2011)



The VesselAI Methodology for AI-Powered Decision Support Systems for the Maritime Industry

Christos Kontzinos¹(✉), Spiros Mouzakitis¹, Carlos Agostinho², Paulo Figueiras², and Dimitris Askounis¹

¹ National Technical University of Athens, 10682 Athens, Greece
ckon@epu.ntua.gr

² Centre of Technology and Systems, UNINOVA, 2829-516 Caparica, Portugal

Abstract. The maritime industry is a major contributor in the European economy, employment, and shipping. However, the industry is facing various challenges that concern the need to increase performance and efficiency, optimise costs, obey the strict regulations imposed and increase its overall sustainability when it comes to fuel consumption and other environmental concerns. Such challenges can be resolved with the help of innovative, emerging technologies such as big data, AI, and HPC, the combination of which can lead the way for the development of the next generation of maritime applications. The use of these technologies has the potential to greatly enhance the performance and competitiveness of the maritime industry, but any such initiative must take into account current maritime challenges and needs, as well as the heterogeneity of the maritime environment, processes, and data in order to develop solutions that tackle real problems and correspond to the actual needs of maritime stakeholders. This can only be achieved through cooperation with maritime and other stakeholders and requires comprehensive methodologies that can guide the development of a solution from the early phases of knowledge generation and requirement elicitation to the latter stages of technical development and testing. As such, the current publication presents the VesselAI methodology for AI-powered decision support systems for the maritime industry. VesselAI is an EU-funded project that aims to combine innovative technologies, mainly big data, AI, and HPC to develop the next generation of maritime applications.

Keywords: Maritime · Methodology · Artificial intelligence · Decision support · Big data

1 Introduction

The European maritime industry is a major contributor to the region's economy, with seaborne trade playing a key role in the movement of goods and resources. In 2021, 3.5 billion tonnes of freight were handled in European Union (EU) ports [1], while EU shipping contributes almost 150 billion euro to the European Gross Domestic Product

(GDP) and accounts for almost 2 million jobs [2]. The industry encompasses a wide range of activities, including shipbuilding, cargo handling, maritime services, and offshore energy production [3]. The maritime industry faces several challenges, including strict regulations, environmental concerns, and competition from other modes of transportation [4]. Another major challenge is the need to improve efficiency and reduce costs, as the industry faces increasing pressure to operate more sustainably and profitably [5]. These challenges are complex and interrelated, and addressing them requires the cooperation and innovation from all involved stakeholders.

The aforementioned challenges, in combination with recent technological advances in the fields of big data, artificial intelligence (AI), distributed computing and extreme-scale analytics have paved the way for several projects and initiatives at the European level, aiming to address challenges, solve problems, and redesign maritime processes with the help of innovative technologies, with big data and AI being at the forefront [6]. Specifically, big data and AI can be used to improve the efficiency of operations, such as by optimizing routes and schedules for ships, or by predicting maintenance needs and reducing downtime [7]. They can also help to improve safety and security, by enabling the tracking and monitoring of ships and their cargo, and by providing early warning of potential hazards [8]. Additionally, big data and AI can be used to support decision-making, by providing real-time analysis of large amounts of data and by identifying patterns and trends that might not be immediately apparent [7]. Overall, the use of these technologies has the potential to greatly enhance the performance and competitiveness of the maritime industry.

There are a number of challenges involved in developing the next generation of maritime applications [9]. One of the key challenges is the need to integrate a wide range of technologies and systems; including sensors, communication networks, and data management tools [10]. This requires a deep understanding of the maritime industry, as well as the ability to design and implement complex technical solutions. Additionally, the development of maritime applications must also take into account the unique challenges of the maritime environment, such as the effects of weather and sea conditions on the operation of equipment [11]. Other challenges include the need to ensure the security and reliability of maritime applications, and the need to comply with a range of regulations and standards that govern the use of technology in the maritime industry [12]. Overall, developing the next generation of maritime applications is a complex and demanding task that requires a combination of technical expertise, industry knowledge, and problem-solving skills, as well as comprehensive end-to-end methodologies that can guide such a process holistically from the requirement elicitation phase to the development of the technical solution.

Stemming from the above, this publication presents the high-level methodology of VesselAI. VesselAI is a European research project that aims to combine innovative technologies, mainly big data, AI, and High-Performance Computing (HPC) to develop the next generation of maritime applications [13]. As such, it incorporates four different pilots that represent different organisations and companies of the industry, with different needs, challenges, geographical locations, and processes. The applications developed for the four VesselAI pilots deal with global vessel traffic monitoring, optimal design of a ship's energy system, management of autonomous ships in short sea transport, weather

routing, and fleet intelligence respectively. The publication is structured as follows: Sect. 1 provides the introduction and the scope of the publication. Section 2 presents the VesselAI concept and methodology and finally, Sect. 3 concludes the document and presents next steps.

2 The VesselAI Methodology

The VesselAI methodology is a comprehensive, end-to-end process that elaborates all phases from the stakeholders' requirement elicitation to the description of the methodology that facilitates the development of the VesselAI technical offering. As such, it starts by eliciting the user requirements and turning them into user stories which in turn helped shape the high-level usage scenarios. The above helped distil the VesselAI technical requirements, which were then grouped based on specific components into high-level functionalities. The functionalities were then prioritized by the project pilots to assess the VesselAI Minimum Viable Product (MVP). The last part of the methodology concerned the creation of the VesselAI platform methodology. All the phases will be described in the following subsections.

2.1 User Requirements and MVP Definition

The VesselAI user requirements extraction was carried out through the application of an Agile methodology, driven by the end-users in close collaboration with the technical partners [14]. This type of methodology is comprised by multiple steps, in which the main focus is the definition of the use cases and the end-user stories that will derive the technical requirements. It is important to note that the user requirements approach was implemented since the beginning of the project and produced for each pilot the Use Cases, which are high-level descriptions of the stakeholders' viewpoints, where the main goals are identified through descriptions of the *as-is* and the *to-be* scenarios, including the challenges, the benefits and Key Performance Indicators (KPIs) of those scenarios, as well as the User stories (USs), which are the smallest unit in the Agile framework, where the main focus is the definition of the goal and what benefits it would bring to the end-user. The tool used to write the user stories was the Trello online platform [15].

The next step of the methodology was the user story mapping in which user stories were collected, and then collaboratively organised in vertical and horizontal maps, where the vertical view represents the priority of the requirement/functionality, and the horizontal view is connected to the user journey (workflow) of all the requirements. Using the user story mapping, it was possible to include the plan of releases, in particular the MVP, following a voting process between the project pilots.

After analysing the documented use case descriptions and the user stories, the outlines of scenarios were chosen as representative of all the main VesselAI users, in terms of user needs to be covered. The final use case scenarios correspond to real life problems in data analysis related to the maritime domain. Each of the scenarios describe the actor, the overview and provide a detailed description to highlight the inherent need for novel, HPC and AI powered services among maritime stakeholders.

The last phase of the VesselAI methodology, described in the next section, comprises the description of the platform methodology which is structured in phases that will guide the design and implementation of the VesselAI solution.

2.2 VesselAI Platform Methodology

The VesselAI Platform Methodology aims at expressing the relationships among the tools responsible for data collection, semantic enrichment, AI model creation, HPC, and service offering. VesselAI's high-level goal is to enable the effortless consumption of (big) maritime data and the efficient exploitation of HPC in the creation of maritime services. To accomplish this, the project partners, taking into account the requirements and the MVP preparatory activities, as well as input from a detailed landscape analysis on big data methodologies and tools, the work implemented on the identification of stakeholders and the use cases descriptions designed a platform methodology that aims to provide a well-constructed and meaningful workflow that will guide the development of the VesselAI solution and will ensure that alignment on VesselAI offerings is reached among all VesselAI partners. The methodology pinpoints seven main components as seen in Fig. 1.

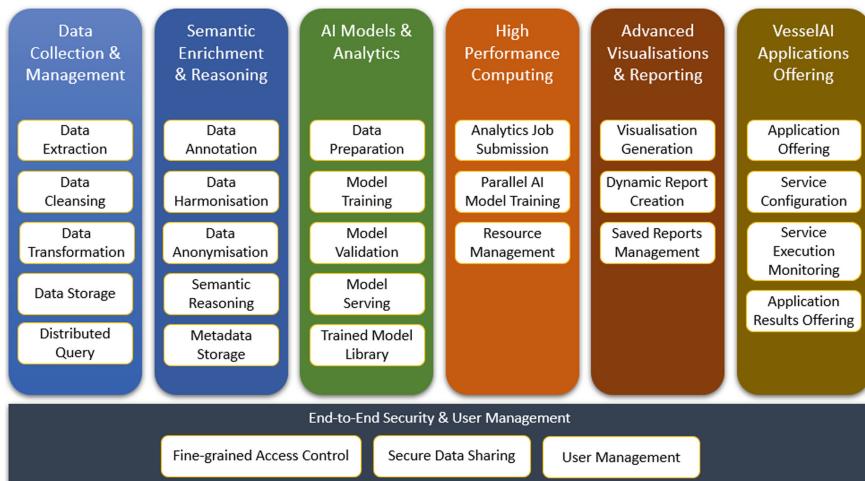


Fig. 1 VesselAI methodology.

The VesselAI Methodology is structured in phases which are not strictly sequential but tightly interconnected, as each phase depends on the outcomes of others and adjustments that might be made in one phase can affect the workflows running through all phases. Such phases are based on the challenges mentioned above, adapting them to the VesselAI context.

The seven components as well as their interconnections and dependencies have also helped in the design of the VesselAI high-level Architecture, which can be seen in Fig. 2.

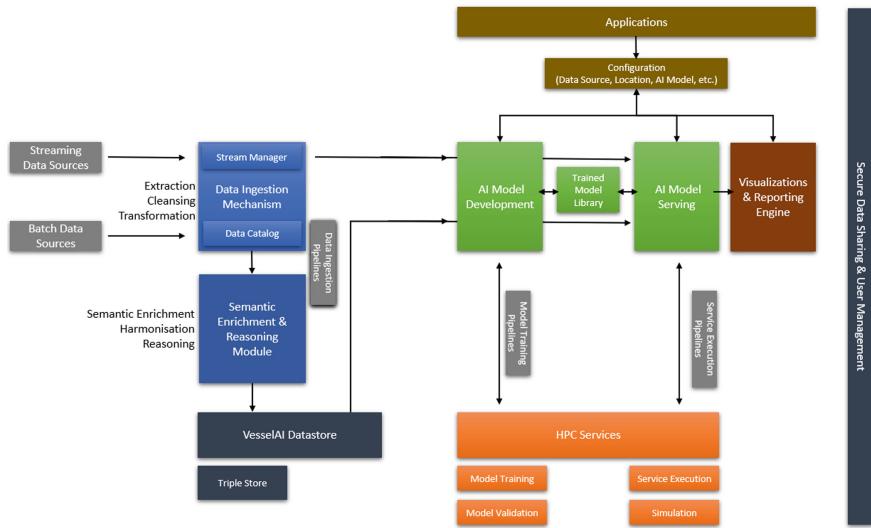


Fig. 2 VesselAI high-level architecture.

Phase 1: Data Collection and Management

The first phase concerns the identification and development of a series of ETL (extract, transform, load) and data management techniques and pipelines that should be implemented to facilitate the data ingestion process and more specifically the operations of data extraction, cleansing, transformation, provenance, storage and efficient and scalable querying of both batch and streaming data. As such, the data collection and management phases aim at developing the architecture of the pipelines to facilitate certain operations on data which are needed for later stages of the data services.

The data pipeline and architecture of VesselAI consists of two data pipelines, one for streaming and one for batch data, which require different approaches for their efficient management. In addition, several data transformation processes are included which entail extraction, cleansing, anonymisation, harmonisation, aggregation and filtering of streaming or batch datasets among others. The transformation functions for a specific dataset are selected and applied based on the technological components that will use the dataset as input.

Phase 2: Semantic Enrichment and Reasoning

The Semantic Enrichment and Reasoning phase aims to represent data in VesselAI using a common semantic model that enables interlinking with data coming from various data sources. To realize this goal, a set of methods and techniques is employed with the following main modules: (1) an ontology for providing the common semantic model, (2) data transformation to RDF (Resource Description Framework), in accordance with the model defined by the ontology, for enabling semantic representations, data interoperability and sharing, (3) link discovery using complex spatial and spatio-temporal relations, and (4) complex event recognition on maritime data for annotating movement data with complex events.

Phase 3: AI Models and Analytics

The AI models and analytics phase focuses on the implementation of AI models and their workflows for the maritime use cases specified by the VesselAI's pilots. The development of the AI models will utilize integrated and fused (both historical and real-time) data from several diverse sources. Integration of diverse data sources, such as AIS (Automatic Identification System) and meteorological data, will help to improve the precision of the AI models. In VesselAI, based on the pilot requirements, the AI models that are being developed concern the following analytical tasks: route/position/trajectory forecasting, traffic flow forecasting, moving speed prediction, collision detection, ship energy system conceptual design optimisation, route planning, anomaly detection, and weather routing. Additional models will be developed in the second phase of the project.

The high-level workflow approach for development and deployment of AI models and analytics consists of steps categorised as follows: (1) data preparation and validation, which is the transformation of data considering the algorithm that will be used to train them, (2) model training, which applies AI algorithms on retrieved and prepared datasets using open-source AI frameworks (3) model evaluation based on multiple criteria, (4) model registry/library, which stores the validated models and (5) model serving that pertains to the implementation of services to deploy selected AI models to maritime applications. To deal with extreme scale analytics and ensure continuous, federated learning of AI models, the AI workflows need to manage and exploit both online and offline data workflows.

Phase 4: High Performance Computing

The design of the HPC structure adopts a co-design approach to converge towards the requirements in terms of performance, computing power, memory/storage resources and exploitation methodology derived not only from the analysis of the project pilots but also from the investigation of related topics in the literature. In this approach, the bottom-up design of the HPC cluster is incremental, i.e., a preliminary generic architecture is refined progressively to converge towards a solution adapted to the needs of the applications of the project.

Based on the technical needs of the project and its pilots, the VesselAI HPC infrastructure must include powerful CPUs (Central Processing Units) and GPUs (Graphics Processing Units), sufficient memory (hundreds of terabytes) as storage, connection nodes for different networks, various software for acceleration and submission of analytical jobs, big data management, and AI training among others.

Phase 5: Advanced Visualisations and Reporting

Advanced visualisation and reporting tools will be deployed to support the visual analysis, data comprehension and value extraction on the large volumes of data gathered, annotated, integrated, modelled, and extracted from data analytics and AI processes. These tools should meet several non-functional requirements, such as scalability, performance, portability, user-friendliness, availability and spatiotemporality. Based on these requirements, Apache Superset [16] was selected as the main tool as it can be distributed across several nodes to cope with large volumes or fast streams of data. Focusing on

portability, the visualisation and reporting tool will be deployed as a containerized solution, whether using Docker, Kubernetes, or any other containerization environment, while it is user-friendly and can handle, integrate and present linked visualisations over Big Spatiotemporal data. The various steps of the advanced visualisation and reporting engine are depicted in Fig. 3.

The first step corresponds to the selection and configuration of the dashboarding/reporting/linked visualisation methods, i.e., the selection of specific visualisation methods and their setup, defining the way they are linked together, and considering the data's spatiotemporal dimensions, the format of reports or the arrangement of dashboards. The second step is the actual data selection to be visualised and comprises data access through data adapters, connecting to external Web Services and other data sources, data querying directly to the VesselAI data storage component and data stream access, via message queues or publish-subscribe mechanisms. Next, the data transformation step comprises all processes to integrate and summarise data. These techniques are data filtering, or the selection of specific data ranges to be presented, for instance by spatial region (spatial filtering) or by temporal range (temporal filtering), data integration, i.e., joining and integrating different data types and filtering out unnecessary parameters, and data summarisation, which corresponds to filtering out redundant data points without losing the meaning of the underlying information. Finally, the data visualisation step provides several types of visualisations, from visual analytics dashboards, simple visualisations, or reports to fully-fledged spatiotemporally linked views.

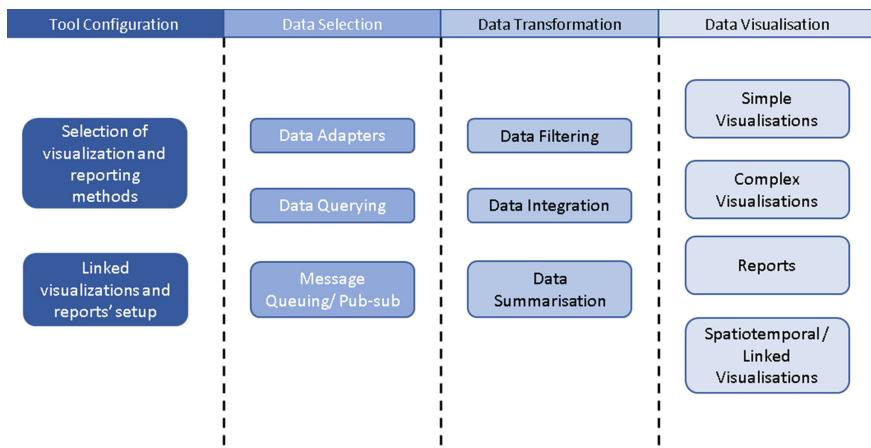


Fig. 3. Advanced visualisation and reporting engine high-level methodology.

Phase 6: VesselAI Service and Applications Offering

In the context of VesselAI, maritime-related big data will be turned into value for the maritime stakeholders, through the offering of meaningful and innovative applications/services. The core service offering of VesselAI will be based on the utilisation of well-proven HPC, Deep Learning and Big Data Analytics frameworks in combination

with the implementation of domain-specific algorithms. These services will constitute the core functionality of the platform. An initial set of such services is going to be developed by the technical partners in close connection with the pilot partners, to effectively demonstrate the benefits obtained by the proposed solution. A prime methodology for the suggested process is illustrated in Fig. 4.

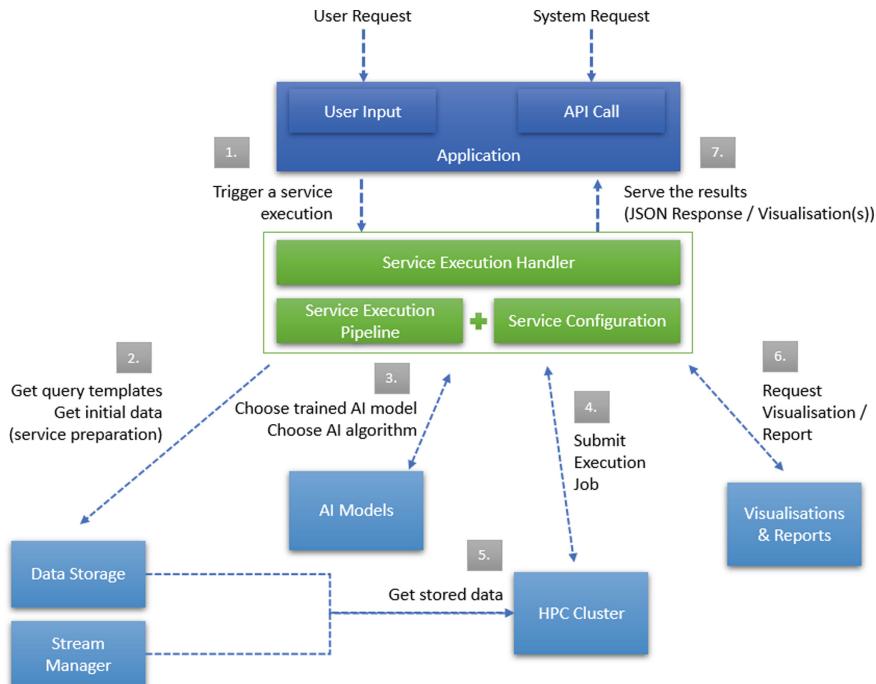


Fig. 4. VesselAI application offering methodology.

The VesselAI application is planned to be closely integrated with the pilot systems to maximise the produced impact. At the same time, an independent environment will be created through which the end-users can find a set of applications, along with details about them, as an additional demonstration channel of the project. Using this environment, the end-user will have the ability to access an application, provide a configuration as input through a user-friendly interface, trigger an execution and get a visual representation of the results.

Each VesselAI application, is going to be related with a Service Object and an initial request is what triggers the Service execution. This request could either come from an external system or even from a scheduled task. The execution request should contain all the required input that the service expects to be configured and executed. A component responsible for handling these requests, will offer both dedicated endpoints and a user interface. Each VesselAI Service will define all the information regarding the required inputs, the execution logic, and the expected outputs. The execution logic is going to be

represented in the form of workflows/pipelines that encapsulate all the required algorithms and analytical processes to provide the desired results. These pipelines consist of multiple well-defined steps/tasks that communicate with the different software modules and utilise their offering, i.e., retrieve data from the data storage, do data pre-processing tasks, trigger an AI model training, etc.

When an execution request is initiated, the system will create a new Service Execution Instance, gather the input given and then configure and trigger the execution of the service's related pipeline. The Service Execution Handler will be a software module responsible for accepting the requests, creating the Service Execution Instance, configuring, and executing the Service's pipeline, monitoring, and orchestrating the execution, and provide the results as a response. A high-level example of a Service pipeline is displayed in Fig. 5.

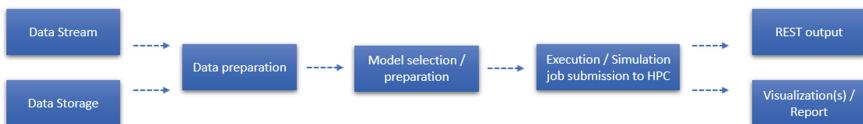


Fig. 5 VesselAI service execution pipeline.

Phase 7: Secure Data Sharing/User Management

The Secure Data Sharing and User Management phase addresses how the data can be shared among the different stakeholders as well as which permissions should be granted to the different users concerning the access to data and VesselAI offerings and the actions that they will be allowed to take. Secure Data Sharing involves the appropriate techniques to ensure that information and data stored, being processed, and exchanged both between VesselAI architectural components and between VesselAI components and external actors are adequately protected and accessed. Secure Data Sharing encapsulates the to-be developed end-to-end security framework covering all the layers and architectural components of the envisaged solution. In this respect, this phase involves the identification of measures concerning the transmission of the data, both stream and batch, utilising secure channels (e.g., TLS/SSL protocols), the storage of the data, utilising, if needed, encryption and anonymisation techniques, the assurance of data privacy and workload isolation while executing workloads in the HPC infrastructure, the secure development and deployment of the services and applications exposing the underline to-be developed AI models, the prevention of unauthorised access to the assets through all components included the HPC platform.

User Management is directly connected with the Secure Data Sharing since it enables the adoption of appropriate authentication and authorisation protocols thus ensuring that only identified users will have access and only on the allowed/predefined resources. Therefore, appropriate access control mechanisms will be considered and defined implementing access policies that will be agreed between stakeholders and describe the level of access of each one of them to the data and computing resources. User Management phase will guarantee that all VesselAI resources (data, models, services, infrastructure)

will be accessed only by the authorised users and will provide the mechanism to manage the users' privileges and permissions.

3 Conclusions and Next Steps

The present publication presented a comprehensive methodology that guides the VesselAI project in achieving all its goals and successfully incorporate big data, AI, and HPC into maritime processes and applications. Leveraging these technologies in maritime has the potential to greatly improve the efficiency and safety of the industry. By using advanced algorithms to analyse vast amounts of data, ships can be more accurately navigated, potential hazards can be detected earlier, and overall performance can be optimized. However, there are also potential drawbacks to consider, such as the potential for job loss and the need for robust security measures to protect against cyber threats. Overall, the integration of HPC, AI and big data in the maritime industry holds great promise, but it must be approached carefully and with consideration of all potential implications, needs, challenges, and requirements of the projected users.

In the VesselAI project, the methodology presented has guided the initial development of the first releases of the VesselAI platform. Future work includes the development of all software components according to the methodology and implementation plan. Additional focus will be put on the integration and offering of the AI models, the communication between the software components and the integration with maritime applications, according to the pilots' needs. Finally, thorough testing and evaluation of the overall technical solution will be performed to ensure its proper function.

Acknowledgment. This work has been co-funded by the European Union's Horizon 2020 research and innovation programme under the VesselAI project, Grant Agreement No 822404.

References

1. Maritime freight and vessels statistics: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Maritime_freight_and_vessels_statistics, last accessed 2022/12/14
2. Oxford Economics: The Economic Value of the EU Shipping Industry, e European Community Shipowners' Associations (ECSA), (2020)
3. Langen, P.D.: Clustering and performance: the case of maritime clustering in The Netherlands. In *Maritime Policy & Management* **29**(3), 209–221 (2002)
4. Koilo, V.: Sustainability issues in maritime transport and main challenges of the shipping industry. In *Environmental Economics* **10**(1), 48–65 (2019)
5. Lee, P.T.W., Kwon, O.K., Ruan, X.: Sustainability Challenges in Maritime Transport and Logistics Industry and Its Way Ahead. In *Sustainability* **11**, 1331 (2019)
6. Liang, T.P., Liu, Y.H.: Research landscape of business intelligence and big data analytics: a bibliometrics study. *Expert. Syst. Appl.* **111**, 2–10 (2018)
7. Munim, Z.H., Dushenko, M., Jimenez, V.J., Shakil, M.H., Imset, M.: Big data and artificial intelligence in the maritime industry: a bibliometric review and future research directions. *Marit. Policy & Manag.* **47**(5), 577–597 (2020)
8. Heilig, L., Lalla-Ruiz, E., Voß, S.: Digital transformation in maritime ports: analysis and a game theoretic framework. In *Netnomic* **18**, 227–254 (2017)

9. Aslam, S., Michaelides, M.P., Herodotou, H.: Internet of ships: A survey on architectures, emerging applications, and challenges. *IEEE Internet Things J.* **7**(10), 9714–9727 (2020)
10. Xia, T., Wang, M.M., Zhang, J., Wang, L.: Maritime internet of things: Challenges and solutions. *IEEE Wirel. Commun.* **27**(2), 188–196 (2020)
11. Oliveira-Pinto, S., Stokkermans, J.: Marine floating solar plants: An overview of potential, challenges and feasibility. In: Proceedings of the Institution of Civil Engineers-Maritime Engineering **173**(4), 120–135 (2020)
12. Kechagias, E.P., Chatzistelios, G., Papadopoulos, G. A., Apostolou, P.: Digital transformation of the maritime industry: A cybersecurity systemic approach. *Int. J. Crit. AI Infrastruct. Prot.*, **37**, (2022)
13. Mouzakitis, S., Kontzinos, C., Kapsalis, P., Kanellou, I., Kormpakis, G., Tsapelas, G., Askounis, D.: optimising maritime processes via artificial intelligence: the vesselai concept and use cases. In: 13th international conference on information, intelligence, systems & applications (IISA), pp. 1–5. IEEE, (2022)
14. Paton, J.: User story mapping: discover the whole story, build the right product. O'Reilly Media, (2014)
15. Trello Homepage: <https://www.trello.com/>. Last accessed 2022/12/14
16. The Apache Software Foundation.: Apache Superset. <https://superset.apache.org/>. Last accessed 2022/12/14



An Energy-Efficient Reconfigurable Autoencoder Implementation on FPGA

Murat Isik^(✉), Matthew Oldland, and Lifeng Zhou

Electrical and Computer Engineering, Drexel University, Philadelphia, USA
mci38@drexel.edu

<https://github.com/Muratcanisik4/CNN-Autoencoder.git>

Abstract. Autoencoders are unsupervised neural networks that are used to process and compress input data and then reconstruct the data back to the original data size. This allows autoencoders to be used for different processing applications such as data compression, image classification, image noise reduction, and image coloring. Hardware-wise, re-configurable architectures like Field Programmable Gate Arrays (FPGAs) have been used for accelerating computations from several domains because of their unique combination of flexibility, performance, and power efficiency. In this paper, we look at the different autoencoders available and use the convolutional autoencoder in both FPGA and GPU-based implementations to process noisy static MNIST images. We compare the different results achieved with the FPGA and GPU-based implementations and then discuss the pros and cons of each implementation. The evaluation of the proposed design achieved 80% accuracy and our experimental results show that the proposed accelerator achieves a throughput of 21.12 Giga-Operations Per Second (GOP/s) with a 5.93 W on-chip power consumption at 100 MHz. The comparison results with off-the-shelf devices and recent state-of-the-art implementations illustrate that the proposed accelerator has obvious advantages in terms of energy efficiency and design flexibility. We also discuss future work that can be done with the use of our proposed accelerator.

Keywords: Convolutional autoencoder · Reconfigurable architecture · FPGA · Energy efficiency

1 Introduction

Autoencoders are an unsupervised neural network that aims to learn how to compress input data and then reconstruct the data to match the input data with the least amount of loss [1]. This is done using encoder and decoder methods, where the encoder compresses the data and the decoder reconstructs the data [1–3, 7]. There are many types of autoencoders that can be used in machine learning; some examples are simple, convolutional, removing noise, and sparse autoencoders. These different autoencoders can then be used for different processing needs, like data compression before storage, image classification, and removing noise from images.

This paper shows the use of convolutional autoencoders and how effective they are at removing noise from images. Convolutional autoencoders use convolutional neural networks for the layers of encoding and decoding to help compress and reconstruct the image respectively. Convolution neural networks use different filters to pull out features from the data. These filters are designed to pull out specific features and are dependent on what the feature needed is. The filters are designed to scan through the data and then create a map of how well the data matched the filter by creating scores, where a higher score is a better match [4]. The filter strides across the image based on how the layer is set up by the user. These maps are then used in the next layer, which is either a pooling layer or an up-sample layer. The pooling layer looks at a window of the data and will combine the window into one sample. The pool can use either a max of the data, the sum of the data, or the average of the data to create the new sample [4]. The pooling layer is used in the encoding process to help compress the data and store the important features. While the upsampling process is used in the decoding process to help with reconstruction. There are many different ways to implement an autoencoder using different hardware. This paper compares two methods, one is an implementation of an FPGA using VHDL and the other is a GPU-based implementation using python.

While developing the rest of the framework, machine vision framework developers and respectability might become engrossed in deciding which of these steps to use. The organize option is commonly chosen when prototyping a system for the first time. A prototype application must be tested to determine how many sections it must prepare every second or how many outlines it must handle per second of the live video. The power consumption of embedded systems is an important consideration for most applications, and researchers strive to develop energy-efficient design methodologies [14, 15].

The applications require for handling enormous data sets of high-resolution images preparing applications requests faster, configurable, long throughput frameworks with superior vitality productivity [16, 17]. FPGAs (Field-Programmable Gate Arrays) can play an important role since they provide configurability, adaptability, and parallelism to coordinate the necessary throughput rates of the application under consideration [13]. Real-world applications can be implemented on FPGA devices thanks to their execution capabilities. The development of FPGAs has taken hardware flexibility, in general, one step further. In the end, the toolchain for developing applications on these devices has also advanced significantly, allowing these devices to be used by a larger building community [18]. FPGAs are typically used in applications that demand concurrency, high transfer speeds, and re-programmability.

FPGAs have achieved rapid acceptance and growth over the past decade because they can be used in a very wide range of applications. The algorithms that extract features are time-consuming, which is a huge drawback when developing real-time applications. One solution to this problem is the use of dedicated hardware for the algorithms, such as the FPGA, which can provide dedicated functional blocks that perform complex image processing operations in parallel. FPGAs have achieved rapid acceptance and growth over the past decade because they can be used in a very wide range of applications [19, 20].

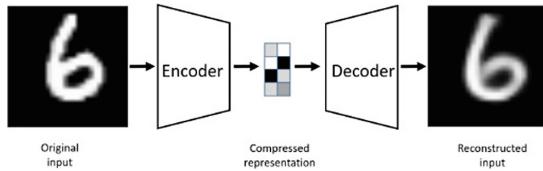


Fig. 1. Example autoencoder showing input image compression and reconstruction [1].

We designed the convolutional autoencoder in both FPGA and GPU-based implementations to process noisy static MNIST images. Results show that the proposed design is both area- and power-efficient.

2 Autoencoder Types

Autoencoders are a type of unsupervised neural network and are used for various processing applications. They are trained to encode and compress input data then reconstruct and decode the data back to try to match the input data. There are three main parts to autoencoders [3]:

- Encode and Compress.
- Bottleneck.
- Decode and reconstruct.

The encode and compress part of the autoencoder is composed of layers of neural networks that work to compress the data down to a set size. This encoded and compressed data is considered the bottleneck part since it is the smallest part of the system before getting reconstructed. The data could then be stored if needed, which allows for less required storage capacity. The compressed data is then used in the next part of the system, the decoding, and reconstruction part. This part works to decode the encoded data and reconstruct the data back to match the input data that was passed in. It does this by using neural networks that are designed the same as in the encode and compress part, however instead of compressing the data it is up-sampled instead. The number of layers used in both the compression and reconstruction parts is set by the user based on the required implementation needed. Figure 1, below, shows an example of an input image being compressed and reconstructed to an output image that matches closely [2]. There are many different types of autoencoders that are used within machine learning. These autoencoders have the same flow and parts between them but the layers used within the encoding and decoding change. The sections below cover some of the different types of autoencoders that can be used.

2.1 Simple Autoencoder

Simple autoencoders are the basic implementation of an autoencoder and are a simple feed-forward network that can have multiple layers within the encode and decode parts. One implementation had three layers and the layers were fully connected to each other [1]. This implementation also had the input and out of the autoencoder as the same size which forces the autoencoder to learn the compressed data with no information loss [1].

2.2 Convolutional Autoencoder

Convolutional autoencoders are similar to simple autoencoders but change the layers to convolution layers using a convolution neural network. The convolution neural network layers are set up with several parts [4]:

- Convolution.
- Rectified Linear Unit.
- Pooling (Encode) or Up-Sample (Decode).

The convolution is performed by forming features out of the data and creating feature maps using convolution window filters. These filters are used to maintain the relationship between the pixels and the input image [4]. The resulting features create high and low scores that are representative of how well the pixels are in relation to the filter; the better the match the higher the score. The convolution has two parameters to it, the padding and the stride length [4]. The padding is so that feature has the same amount to work on no matter the input data size. The stride parameter is set to determine how many pixels the filter shifts over the input data when creating the feature.

The Rectified Linear Unit (ReLU) part is used to make sure that there are no negative numbers going between layers. This is done by replacing any negative number found with zeros to help the math work out. The pooling part is the compression of the image. A window size is selected to combine parts of the filtered features into one pixel of the new image to be passed to the next layer. In the pooling window, either the max value, average value, or summed value within the window is passed into the next pixel of the new image. The decoding and reconstruction layers will use an up-sample part instead of the pooling part to increase the size of the image as it gets reconstructed back to match the input size [6–10].

2.3 Denoising Autoencoder

Denoising autoencoders [2] use Gaussian noise on the input data to disrupt the data then the autoencoder is trained to reconstruct the data with the noise removed. The goal of the denoising autoencoder is to be used for error correction of data that has been corrupted. The effectiveness of the autoencoder is measured by the reconstruction loss between the output data and the input data.

2.4 Sparse Autoencoder

Sparse Autoencoders use sparsity in the layers to help with keeping the reconstruction error low and have a meaningful bottleneck image in the middle. In [2], the sparsity was enforced by applying L₁ regularization in the layer. This changes the autoencoder optimization objective to:

$$\operatorname{argmin}_{A,B} E[\Delta(x, B \circ A(x))] + \lambda \sum_i |a_i|, \quad (1)$$

Table 1. Layer construction for convolutional autoencoder.

Layer	Type	Window size	Stride
1	Convolution	4	1
2	Max pool	4	1
3	Convolution	4	1
4	Max pool	4	1
5	Convolution	4	1
6	Max pool	4	1
7	Convolution	4	1
8	Up-sample	4	1
9	Convolution	4	1
10	Up-sample	4	1
11	Convolution	4	1
12	Up-sample	4	1
13	Convolution	4	1

where, a_i is the activation at the i th hidden layer and i iterates over all hidden activations [2]. Another method covered in [2], is the KL-divergence method. This method measures the distance between two probabilities by changing the activation a_i to a Bernoulli variable with a probability of p [2]. The method tweaks the probability p by taking the measured probability for a batch, finding the difference, and then applying that as the regularization factor.

3 Method

We consider an image has been corrupted by noise and requires the noise to be removed for processing later on. A convolutional autoencoder was designed to perform image noise reduction on the corrupted input images. The convolutional autoencoder is designed using three convolution layers and three pooling layers, interleaved, for encoding and compression. The decoding and reconstruction used four convolution layers and three up-sampling layers that were interleaved. The pooling layers were constructed to use the maximum pooling operation when operating over the windows of the images.

Table 1 shows the layer construction of the convolutional autoencoder for encoding and decoding.

The convolution layers were designed using convolution neural networks to train the weights and perform feature extraction on the image before the pooling or up-sampling layers. A 2-D convolution was used as the filter to create the features from the pixels. The general 2-D function equation used is given by

$$f = \sum_{a=-y}^y \sum_{b=-z}^z x_i - a, j - b * k_{a,b}. \quad (2)$$

- x is the input image, with pixel values $x_{i,j}$ at position (i, j)
- k is the filter (kernel), with coefficients, $k_{a,b}$ at position (a, b)
- y and z are the half-widths of the filter, which determine the size of the convolution operation. For example, if $y = z = 1$, then the filter is a 3×3 matrix centered on each pixel of the input image.
- The summations over a and b iterate over all positions of the filter centered on each pixel of the input image. For each position, the corresponding input pixel and filter coefficient are multiplied together, and the results are summed up to give the output pixel value.

After the convolutional filters are run over the pixels a ReLU function is performed to make sure that no negative numbers are passed to the next layer. This allows for the math to always come back correct and not have errors when performing calculations. The ReLU function used is given by

$$y = \max(0, x), \quad (3)$$

where, x is the filtered pixel feature. The output of the ReLU function is then passed to the next layer of either pooling or up-sampling. The pooling function was a max function of window size of the filtered pixel features and is given by

$$y = \max(x_1, 1, \dots, x_{i,j}), \quad (4)$$

where, i and j are the window size chosen for the layer.

4 Results

4.1 Dataset

This dataset is to be a noisy variation of the MNIST digit set, with the noise coming from an internal sense of style. The training set receives 80% of the whole data, whereas the validation set receives 20% of the total data. The actual training data contains 60,000 photos in all. Since the dataset, we require for the task at hand requires noisy images, we define functions to add random Gaussian noise to the dataset elements. The new dataset now consists of noisy images and can be visualized.

Our noisy dataset is shown Fig. 2. Training data contains 60k photos in all. We use the random Gaussian noise function to make the MNIST dataset into a noisy dataset.

Gaussian noise is a type of noise that is in the form of Gaussian distribution, such as the white noise commonly encountered. It is random-valued and in impulses.

4.2 Training Details

We eventually settled on the 20-epoch training time and the training loss and validation loss are displayed at the end of each epoch. Training time for each model amounted to about 100 s each due to the input images being of small size and hence not requiring too much processing power. Although it can occasionally reduce the visibility of already visible numerals, it does a better job of cleaning out the number and making it far more visible in more challenging circumstances like the first image. For a modest number of training parameters, the results are reasonable, with the greatest accuracy of 80.83%. The output of our autoencoding process is shown in Fig. 3.



Fig. 2. Project noisy dataset.



Fig. 3. Denoised image.

4.3 Experimental Setup

We focused on two approaches to physically implementing the convolutional autoencoder; one using a GPU and the other using an FPGA hardware implementation. The GPU implementation was done using Python and Google Colaboratory. Google Colaboratory allowed for running the machine learning python executable on an NVIDIA Tesla K80. We implement the same design on the XCZU7EV FPGA chip having a 504K lookup table (LUT) and used 2021.1 software version of Vivado.

4.4 FPGA Implementation

The encoded images are further analyzed with a convolutional autoencoder architecture comprising of matrix operation, channel distributor, encoding module, round-robin arbitration, and output controller blocks whose parts are shown in Figs. 4 and 5.

Matrix Operation: The boundary information during the convolution operation ($28 * 28 * 1$).

Channel Distributor: Matrix is sent to the channel distributor module, which contains a channel controller and first-in-first-out memories. The channel controller interleaves channel data points and then sends them to the FIFOs in sequence. The FIFOs buffer the interleaved data and then divide them into eight channels.

Encoding and Decoding Module: First, the convolutional encoding module does the convolution process, activation, and then max-pooling. Decoding is performed by up-sampling rather than max-pooling.

Round Robin Arbitration: The output data of the convolutional encoding module are orderly packaged and then sent to the output controller. Round-Robin arbitrator is giving priority to requestors.

Output Controller: This module is used to judge whether the data stream is processed completely.

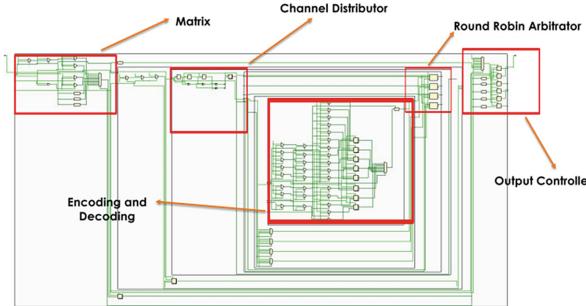


Fig. 4. RTL design of the implementation.

4.5 Comparison

The implementation of Neural Networks (NNs) has been done using CPUs, GPUs, and ASICs. CPU/GPU-based NNs consume a lot of power and have a limited throughput due to limited memory bandwidth. Many researchers have developed custom ASICs for accelerating network inference workloads in order to achieve the best performance and energy efficiency. In spite of their attractiveness, ASICs cannot provide sufficient flexibility to accommodate the rapid development of Neural Networks, and FPGAs function as programmable devices that can construct unique logic, alleviating constraints on neural network implementation. As a result, one of the current research hotspots involves the development of hardware systems supporting NN inference based on FPGA to achieve high throughput and power efficiency.

As shown in Table 2, we compare our CNN autoencoder implementation with other CNN autoencoder models presented in the literature. The evaluation of the proposed design achieved 80% accuracy and our experimental results show that the proposed accelerator achieves a throughput of 21.12 Giga-Operations Per Second (GOP/s) with a 5.93 W on-chip power consumption at 100 MHz. The comparison results with off-the-shelf devices and recent state-of-the-art implementations illustrate that the proposed accelerator has obvious advantages in terms of energy efficiency and design flexibility.

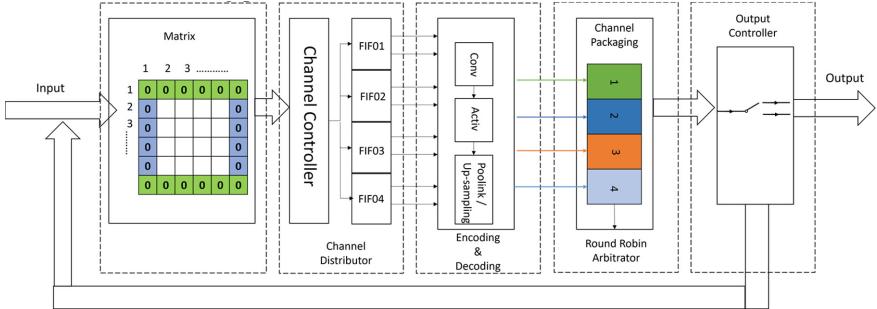


Fig. 5. Block diagram of implementation.

Table 2. Autoencoder on different heterogeneous devices.

Platform	Technology	Operation	Frequency power (W)	Latency (ms)	Throughput (GOP/s)	Energy efficiency (GOP/s)
NVIDIA K80	ASIC (28nm)	1.48 GHz	30	1137.62	22000	733.33
NVIDIA GTX 1080 TI	GPU (16nm)	1.48 GHz	250	6.15	235.7	0.94
Chakradhar et al. [12]	FPGA (28nm)	200 MHz	15	—	16	1.06
Gokhale et al. [11]	FPGA (28nm)	150 MHz	8	4.50	23.18	2.90
Zhang et al. [5]	FPGA (28nm)	100 MHz	18.61	21.61	61.62	3.31
Ours	FPGA (16nm)	100 MHz	5.93	2.91	21.12	3.5

5 Conclusion and Future Work

In a confusion matrix, the model determined which classes it is accurately forecasting and which classes it is forecasting inaccurately. More training parameters and deeper layers in the autoencoder could improve the model's performance. However, in the context of this study, the model performs admirably in reassembling denoised photos. It is shown that the use of convolutional layers is very important in the case of denoising images using deep learning based on a distinction between encoder and decoder architectures. The model is implemented for hardware acceleration with various heterogeneous devices and resulting in an energy-efficient, reconfigurable system on the latter. Therefore, a deeper model and a more suitable latent space dimension can be used for more complex denoising involving RGB images. Future work scenarios could be removing noise from

pictures received from cameras attached to autonomous robots before image processing, as the next step of this research. We are seeking to integrate our model into robots so that a response can be made by cleaning images received wirelessly from different robots if the transition is not clear before detecting position.

References

1. Zhang, Y.: A better autoencoder for image: Convolutional autoencoder. ICONIP17-DCEC. Available online http://users.cecs.anu.edu.au/Tom.Gedeon/conf/ABCs2018/paper/ABCs2018_paper_58.pdf. (Accessed on 23 March 2017). (2018)
2. Bank, D., Noam, K., Raja, G.: Autoencoders. arXiv preprint [arXiv:2003.05991](https://arxiv.org/abs/2003.05991). (2020)
3. Badr, W.: Auto-encoder: What is it? And what is it used for? (Part 1). Medium, (2021). <https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726>
4. Dataman., Kuo, C.: Convolutional autoencoders for image noise reduction. Medium, (2022). towardsdatascience.com/convolutional-autoencoders-for-image-noise-reduction-32fce9fc1763
5. Zhang, C., et al.: Optimizing FPGA-based accelerator design for deep convolutional neural networks. In: Proceedings of the 2015 ACM/SIGDA international symposium on field-programmable gate arrays. (2015)
6. Xu, J., Duraisamy, K.: Multi-level convolutional autoencoder networks for parametric prediction of spatio-temporal dynamics. Comput. Methods Appl. Mech. Eng. **372**, 113379 (2020)
7. Turchenko, V., Chalmers, E. and Luczak, A.: A deep convolutional auto-encoder with pooling-unpooling layers in caffe. arXiv preprint [arXiv:1701.04949](https://arxiv.org/abs/1701.04949). (2017)
8. Chen, M., et al.: Deep feature learning for medical image analysis with convolutional autoencoder neural network. IEEE Trans. Big Data **7**(4), 750–758 (2017)
9. Koushik, J.: Understanding convolutional neural networks. arXiv preprint [arXiv:1605.09081](https://arxiv.org/abs/1605.09081). (2016)
10. Wang, W., et al.: Generalized autoencoder: A neural network frame-work for dimensionality reduction. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. (2014)
11. Gokhale, V., et al.: A 240 g-ops/s mobile coprocessor for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. (2014)
12. Chakradhar, S., et al.: A dynamically configurable coprocessor for convolutional neural networks. In: Proceedings of the 37th annual international symposium on Computer architecture. (2010)
13. Woods, R., et al.: FPGA-based implementation of signal processing systems. John Wiley & Sons, (2008)
14. Camposano, R., Wilberg, J.: Embedded system design. Des. Autom. Embed. Syst. **1**(1), 5–50 (1996)
15. Polzer, A., et al.: Managing complexity and variability of a model- based embedded software product line. Innov. Syst. Softw. Eng. **8**(1), 35–49 (2012)
16. Kattenborn, T., et al.: Review on convolutional neural networks (CNN) in vegetation remote sensing. ISPRS J. Photogramm. Remote. Sens. **173**, 24–49 (2021)
17. Benzigar, M.R. et al.: Advances on emerging materials for flexible supercapacitors: current trends and beyond. Adv. Funct. Mater. **30**(40), 2002993 (2020)

18. Sass, R., Schmidt, A.G.: Embedded systems design with platform FPGAs: principles and practices. Morgan Kaufmann, (2010)
19. Trimberger, S.M.S.: Three ages of fpgas: a retrospective on the first thirty years of fpga technology: this paper reflects on how moore's law has driven the design of fpgas through three epochs: the age of invention, the age of expansion, and the age of accumulation. *IEEE Solid-State Circuits Mag.* **10**(2), 16–29 (2018)
20. Isik, M., et al.: A design methodology for fault-tolerant computing using astrocyte neural networks. In: Proceedings of the 19th ACM international conference on computing frontiers. (2022)



Convergence of the Mini-Batch SIHT Algorithm

Saeed Damadi^(✉) and Jinglai Shen

Department of Mathematics and Statistics, University of Maryland, Baltimore County (UMBC), Baltimore, MD 21250, USA
{sdamadi1,shenj}@umbc.edu

Abstract. The Iterative Hard Thresholding (IHT) algorithm has been considered extensively as an effective deterministic algorithm for solving sparse optimizations. The IHT algorithm benefits from the information of the batch (full) gradient at each point and this information is a crucial key for the convergence analysis of the generated sequence. However, this strength becomes a weakness when it comes to machine learning and high dimensional statistical applications because calculating the batch gradient at each iteration is computationally expensive or impractical. Fortunately, in these applications the objective function has a summation structure that can be taken advantage of to approximate the batch gradient by the stochastic mini-batch gradient. In this paper, we study the mini-batch Stochastic IHT (SIHT) algorithm for solving the sparse optimizations. As opposed to previous works where increasing and variable mini-batch size is necessary for derivation, we fix the mini-batch size according to a lower bound that we derive and show our work. To prove stochastic convergence of the objective value function we first establish a critical sparse stochastic gradient descent property. Using this stochastic gradient descent property we show that the sequence generated by the stochastic mini-batch SIHT is a supermartingale sequence and converges with probability one. Unlike previous work we do not assume the function to be a restricted strongly convex. To the best of our knowledge, in the regime of sparse optimization, this is the first time in the literature that it is shown that the sequence of the stochastic function values converges with probability one by fixing the mini-batch size for all steps.

Keywords: Stochastic sparse optimization · Stochastic hard thresholding

1 Introduction

We consider the following sparse optimization problem:

$$(P) : \begin{aligned} \min f(\mathbf{x}, \Xi) &:= \frac{1}{N} \sum_{i=1}^N f^{(i)}(\mathbf{x}, \xi^{(i)}) \\ \text{s.t. } \mathbf{x} &\in C_s \end{aligned} \tag{1}$$

Algorithm 1 The Mini-Batch Stochastic Iterative Hard Thresholding

Require: $\mathbf{x}^0 \in C_s$ such that $\|\mathbf{x}^0\|_0 \leq s$, a stepsize $0 < \gamma < \frac{1}{L_s}$, and $1 \leq S_B \in \mathbb{N}$ such that

$$S_B \geq \frac{N}{1 + \frac{1 - L_s \gamma}{1 + L_s \gamma} \frac{N-1}{\frac{c}{N} - 1}}$$

for some $c > 0$.

- 1: **for** $k = 0, 1, \dots$ **do**
- 2: Construct B^k by selecting S_B elements from $\{1, \dots, N\}$ uniformly without replacement such that $|B^k| = S_B$.
- 3: Calculate the stochastic mini-batch gradient as

$$\mathcal{G}(\mathbb{X}^k, \Xi, B^k) = \frac{1}{S_B} \sum_{i \in B^k} \nabla f^{(i)}(\mathbb{X}^k, \xi^{(i)}).$$

- 4: $\mathbb{X}^{k+1} \in H_s(\mathbb{X}^k - \gamma \mathcal{G}(\mathbb{X}^k, \Xi, B^k))$.

- 5: **end for**
-

where $f^{(i)} : \mathbb{R}^n \times \Xi \rightarrow \mathbb{R}$ for $i = 1, \dots, N$, $\Xi = \{\xi^{(1)}, \dots, \xi^{(N)}\}$, and

$$C_s = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\|_0 \leq s\}$$

(sparsity constraint) is the union of finitely many subspaces whose dimension is less than or equal to the sparsity level s such that $1 \leq s < n$. The importance of the Problem (P) is due to the fact that finding a sparse network whose accuracy is on a par with a dense network amounts to solving a bi-level, constrained, stochastic, nonconvex, and non-smooth sparse optimization problem [7]. Thus finding efficient algorithms that solve Problem (P) can be beneficial for addressing compression of deep neural networks.

Among algorithms for solving sparse optimization the Iterative Hard Thresholding (IHT) algorithm has been a very successful one due to the simplicity of its implementation. The IHT algorithm not only has been practically efficient, but also shows theoretical promising results. It was originally devised for solving compressed sensing problems in 2008 [4,5]. Since then, a large body of literature has been studying it from different perspectives. For example, [2,15,16,19,29] consider convergence of iterations, [9,14] study the limit of the objective function value sequence, [13,30] address duality, [26,28] extend it to Newton's-type IHT, [3,10,23,25] address accelerated IHT, and [1,24] solve logistic regression problem using the IHT algorithm. Recently [8] introduced the concepts of HT-unstable stationary points (saddle points in the sense of sparse optimization) and showed the escapability property of the HT-unstable stationary points as one of the crucial properties of the IHT algorithm. Also, they showed Q-linearly convergence of the IHT algorithm towards strictly HT-stable stationary points. However, these desirable properties, requires to compute the batch (full) gradient at each iteration which is computationally expensive or impractical with current GPUs.

On the other hand, almost all training for deep neural networks are done using the mini-batch stochastic gradient which is a combination of the stochastic approximation [20] implemented by the backpropagation algorithm [21]. By taking the mini-batch stochastic approximation, we consider solving Problem (P) using the mini-batch Stochastic Iterative Hard Thresholding algorithm outlined in Algorithm 1. Similar to practice where the mini-batch size is fixed beforehand, we fix the mini-batch size at the beginning which is different from previous work [27] in this area. Also, for showing our theoretical results we directly use the mini-batch stochastic gradient and derive our theoretical results which is different from previous works [6, 11] where the batch (full) gradient is used to show the theoretical results. As opposed to other works where restricted strong convexity is necessary for deriving convergence results [12, 27], here the only assumption we make is the restricted strong smoothness on the objective function not on each individual one. Also, we assume that the objective function is a bounded below function which is the case for objective functions used in machine learning applications. Similar to practice where the mini-batch size is fixed beforehand, we fix the mini-batch size at the beginning which is different from previous works [27].

1.1 Summary of Contributions

By considering the mini-batch SIHT Algorithm 1 for Problem (P), we develop the following results:

- We establish a new critical sparse stochastic gradient descent property of the hard thresholding (HT) operator that has not been found in the literature.
- For a given step-size $0 < \gamma < \frac{1}{L_s}$, we find a lower bound on the size of the mini-batch that guarantees the expected descent of the objective value function after hardthresholding.
- Using the sparse stochastic gradient descent property we show that the sequence generated by the mini-batch SIHT algorithm is supermartingale and converges with probability one.
- We show that for a certain class of functions in Problem (P) where $f(\mathbf{x}, \xi^i) := f^{(i)}(\mathbf{V}_{i\bullet}\mathbf{x})$ $f^{(i)} : \mathbb{R}^n \rightarrow \mathbb{R}$, the sum of norm squared of individual gradients restricted to a set of some elements \mathcal{J} , i.e., $\sum_{i=1}^N \|\nabla_{\mathcal{J}} f^{(i)}\|_2^2$, evaluated at every point is proportionate to the norm of the batch gradient $\|\nabla_{\mathcal{J}} f\|_2^2$ where the proportionality constant only depends on the data. Moreover, dependency of the proportionality constant on the data is restricted to the set of \mathcal{J} not the entire data.

2 Related Work

In order improve computational efficiency of the IHT algorithm, algorithms based on stochastic hard thresholding try to use the finite-sum structure of problem (P) [11, 18, 22]. The StoIHT algorithm is introduced in [18] where at

each iteration a random element from the sum in Problem (P) is drawn and the associated gradient is calculated. Basically, the gradient is approximated by a mini-batch stochastic gradient with size one. The StoIHT algorithm defines a sparse subspace and then projects the updated vector into that. To show the theoretical results in [18], the restricted strong smoothness condition for each individual function in Problem (P) is required as well as the restricted strong convexity for the objective function. In addition, the StoIHT algorithm needs the restricted condition number be to $4/3$ which is hard to meet in practice. The stochastic variance reduced gradient hard thresholding (SVRG-HT) algorithm [11, 22] tries to mitigate the variance with a cost of calculating the (batch) full gradient at some stages. This information of the batch gradient is the key for reducing the variance. Similar to the StoIHT algorithm, the SVRG-HT algorithm requires the restricted strong smoothness condition for each individual function in Problem (P) as well as the restricted strong convexity for the objective function. The Accelerated Stochastic Block Coordinate Gradient Descent with Hard Thresholding (ASCDHT) algorithm in [6] is a randomized version of the StoIHT algorithm which suffers the drawbacks of the StoIHT algorithm, i.e., calculating the full gradient and requirement of the restricted strong conditions. The Hybrid Stochastic Gradient Hard Thresholding (HSG-HT) algorithm in [27] is a variant of stochastic IHT algorithms that uses a mini-batch stochastic gradient at each step. However, from the theoretical perspective, the size of a mini-batch has to increase as the algorithm progresses. This makes the algorithm almost deterministic in calculating the gradient and defeats the purpose of using the mini-batch stochastic gradient. The stochastically controlled stochastic gradients (SCSG-HT) algorithm in [12] uses mini-batch stochastic gradients with large batch size as opposed to the SVRG-HT and the ASCDHT algorithms to reduce the variance with less computation, i.e., not calculating the batch gradient at some steps. We present the mini-batch stochastic IHT algorithm and show that the stochastic sequence of the function value is a supermartingale sequence and it converges with probability one. To show our result, we assume the objective function has the restricted strong smoothness property and is bounded below which is the case for objective functions used machine learning applications. Also, to the best of our knowledge, in the regime of sparse optimization, this is the first time in the literature that it is shown that the sequence of the stochastic function values converges with probability one by fixing the mini-batch size for all steps.

3 Definitions

We provide some definitions that will be used throughout the paper.

Definition 1. (*Restricted Strong Smoothness (RSS)*) A differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be restricted strongly smooth with modulus $L_s > 0$ or is L_s -RSS if

$$\begin{aligned} f(\mathbf{y}) &\leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{L_s}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 \\ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n \text{ such that } &\|\mathbf{x}\|_0 \leq s, \|\mathbf{y}\|_0 \leq s. \end{aligned} \tag{2}$$

Definition 2. (*The HT operator*) The HT operator $H_s(\cdot)$ denotes the orthogonal projection onto multiple subspaces of \mathbb{R}^n with dimension $1 \leq s < n$, that is,

$$H_s(\mathbf{x}) \in \arg \min_{\|\mathbf{z}\|_0 \leq s} \|\mathbf{z} - \mathbf{x}\|_2. \quad (3)$$

Claim. The HT operator keeps the s largest entries of its input in absolute values.

For a vector $\mathbf{x} \in \mathbb{R}^n$, $\mathcal{I}_s^{\mathbf{x}} \subset \{1, \dots, n\}$ denotes the set of indices corresponding to the first s largest elements of \mathbf{x} in absolute values. For example $H_2([1, -3, 1]^{\top})$ is either $[0, -3, 1]^{\top}$ or $[1, -3, 0]^{\top}$ where $\mathcal{I}_2^{\mathbf{x}} = \{2, 3\}$ and $\mathcal{I}_2^{\mathbf{y}} = \{1, 2\}$, respectively. Therefore, the output of it may not be unique. This clearly shows why HTO is not a convex operator and why there is an inclusion in (3) not an inequality.

Definition 3. (*Convergence with probability one*) A random sequence $(\mathbf{x}^k \in \mathbb{R}^n)$ in a sample space Ω converges to a random variable \mathbf{x}^* with probability one if

$$\mathbb{P}\left[\omega \in \Omega : \lim_{k \rightarrow \infty} \|\mathbf{x}^k(\omega) - \mathbf{x}^*\| = 0\right] = 0.$$

4 Results

We consider solving Problem (1) using the mini-batch SIHT Algorithm 1 and develop results that guarantee the convergence of the sequence of function values generated by the SIHT Algorithm. To do so, we present our results in two separate subsections. The first part provides stochastic results characterizing expectation of functions involving the sample average of given vectors. Then, in the subsequent subsection we use the aforementioned results to show Theorem 3 which establishes a stochastic gradient result that is the foundation for the convergence of the function value sequence.

4.1 Stochastic Results for Sample Average

In this subsection, we consider a sample average whose elements are drawn uniformly and without replacement. Then, we prove Lemma 2 that calculates the expected value of the norm squared of the sample average based on the covariance matrix of a random vector whose elements are Bernoulli random variable determining elements of the sample average. Next, in Corollary 1 using Lemma 2 we calculate the expected value of the squared distance between the sample average and the overall average. This result is extended in Theorem 1 where the expected value is calculated so that one is able to find the mentioned expectation based on each individual vector and the overall average. We start with the following well-known lemma.

Lemma 1. ([17]) Let $\Lambda \in \mathbb{R}^{n \times n}$ be a deterministic matrix and $\xi \in \mathbb{R}^n$ be a random vector that is distributed according to some probability distribution \mathcal{P} . Then,

$$\mathbb{E}_{\xi} [\xi^{\top} \Lambda \xi] = \text{trace}(\Lambda \text{Cov}(\xi)) + \mathbb{E}_{\xi}^{\top} [\xi] \Lambda \mathbb{E}_{\xi} [\xi]. \quad (4)$$

To invoke the above lemma, notice that one can define a random vector whose elements are Bernoulli random variables determining whether the associated vector is in the sample average or not. Thus we prove the following lemma.

Lemma 2. Let $\mathbf{g}^{(1)}, \dots, \mathbf{g}^{(N)} \in \mathbb{R}^n$ be N deterministic vectors and $B \subseteq \{1, \dots, N\}$ be a random set. Let $\bar{\mathbf{g}} := \frac{1}{N} \sum_{i=1}^N \mathbf{g}^{(i)}$, $\mathcal{G}(B) := \frac{1}{|B|} \sum_{i \in B} \mathbf{g}^{(i)}$,

$$\mathbf{G} := \begin{bmatrix} \mathbf{g}^{(1)} & \dots & \mathbf{g}^{(N)} \end{bmatrix} \in \mathbb{R}^{n \times N},$$

and $\mathbf{z}(B) = [z_1(B), \dots, z_N(B)]^\top$ where $z_i(B)$ is a Bernoulli random variable such that $z_i(B) = 1$ if $i \in B$ otherwise $z_i(B) = 0$ for $i = 1, \dots, N$. Assume $\mathbb{E}_B[\mathcal{G}(B)] = \bar{\mathbf{g}}$, then for any random set B with fixed size $|B|$, the following holds:

$$\mathbb{E}_B[\|\mathcal{G}(B)\|^2] = \frac{1}{|B|^2} \text{trace}(\mathbf{G}^\top \mathbf{G} \text{Cov}(Z(B))) + \|\bar{\mathbf{g}}\|^2. \quad (5)$$

Once the above result is established, it is straightforward to show the following by observing the fact that the sample average is an unbiased estimator of the overall average, i.e., $\mathbb{E}_B[\mathcal{G}(B)] = \bar{\mathbf{g}}$.

Corollary 1. Assume all the assumptions in Lemma 1 hold. Then for any random set B with fixed size $|B|$, the following holds:

$$\mathbb{E}_B[\|\mathcal{G}(B) - \bar{\mathbf{g}}\|^2] = \frac{1}{|B|^2} \text{trace}(\mathbf{G}^\top \mathbf{G} \text{Cov}(Z(B))) \quad (6)$$

Finally, we use the above results to prove the following which calculates the expected squared distance between the sample average and the overall average based on individual vectors and the overall average. The following result is critical because later we will see that Eq. (7) connects the mini-batch stochastic gradient, the batch gradient, and individual gradients in Problem (P).

Theorem 1. Assume all the assumptions in Lemma 1 hold. If elements of the random set B are drawn uniformly and without replacement, then

$$\begin{aligned} \mathbb{E}_B[\|\mathcal{G}(B) - \bar{\mathbf{g}}\|^2] &= \frac{N - |B|}{|B|N(N - 1)} \left(\sum_{i=1}^N \|\mathbf{g}^{(i)}\|_2^2 - N\|\bar{\mathbf{g}}\|^2 \right) \\ &= \frac{N - |B|}{|B|N} \frac{1}{N - 1} \sum_{i=1}^N \|\mathbf{g}^{(i)} - \bar{\mathbf{g}}\|_2^2. \end{aligned} \quad (7)$$

4.2 Stochastic Results for Hard Thresholding Operator

The goal of this subsection is to show the random sequence $(f(\mathbf{x}^k)_{k \geq 1})$ generated by the mini-batch SIHT algorithm converges with probability one. To show this we prove that the random sequence of the function value is a supermartingale sequence so the expected value of the function value sequence is decreasing. To achieve our goal, we prove the following lemma that provides an upper bound on the function value evaluated at a thresholded vector. Notice that the following result does not require the input be an updated vector by the gradient.

Lemma 3. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be in C^1 and Ls-RSS. Then for a fixed $\mathbf{x} \in C_s$ with any $\mathcal{I}_s^\mathbf{x}$, any $0 < \gamma \leq \frac{1}{L_s}$, and any given vector $\mathbf{g} \in \mathbb{R}^n$, either of the following holds for any $\mathbf{y} \in H_s(\mathbf{x} - \gamma\mathbf{g})$ with any $\mathcal{I}_s^\mathbf{y}$:

$$f(\mathbf{y}) \leq f(\mathbf{x}) - \frac{\gamma}{2}(1 - L_s\gamma)\|\mathbf{g}_{\mathcal{I}_s^\mathbf{y}}\|_2^2 - \frac{\gamma}{2}\|\mathbf{g}_{\mathcal{I}_s^\mathbf{x}}\|_2^2 + \gamma\langle\delta_{\mathcal{I}_s^\mathbf{y}}, \mathbf{g}_{\mathcal{I}_s^\mathbf{y}}\rangle + \gamma\langle\delta_{\mathcal{I} \setminus \mathcal{I}_s^\mathbf{y}}, \mathbf{x}_{\mathcal{I} \setminus \mathcal{I}_s^\mathbf{y}}\rangle \quad (8)$$

where $\mathcal{I} = \mathcal{I}_s^\mathbf{x} \cup \mathcal{I}_s^\mathbf{y}$ and $\delta = \mathbf{g} - \nabla f(\mathbf{x})$.

Observe that in the above lemma the vector \mathbf{g} can be any vector in \mathbb{R}^n . It need not be the gradient nor the mini-batch gradient. However, in the following lemma we prove that if \mathbf{g} is designated to be an unbiased stochastic approximation of the gradient at an arbitrary point, then the following result holds.

Lemma 1. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be in C^1 and Ls-RSS. Assume $\mathbf{g}(\mathbf{x}, \omega)$ be an unbiased stochastic approximation of the gradient at $\mathbf{x} \in \mathbb{R}^n$ where $\omega \sim D$ for some distribution D , i.e., $\mathbb{E}_\omega[\mathbf{g}(\mathbf{x}, \omega)] = \nabla f(\mathbf{x})$. Then for a fixed $\mathbf{x} \in C_s$ with any $\mathcal{I}_s^\mathbf{x}$ and $0 < \gamma \leq \frac{1}{L_s}$, either of the following holds for any $\mathbf{y}(\omega) \in H_s(\mathbf{x} - \gamma\mathbf{g}(\mathbf{x}, \omega))$ with any $\mathcal{I}_s^{\mathbf{y}(\omega)}$:

$$\begin{aligned} \mathbb{E}_\omega[f(\mathbb{Y}(\omega))] &\leq f(\mathbf{x}) - \frac{\gamma}{2}(1 - L_s\gamma)\mathbb{E}_\omega[\|\mathbf{g}_{\mathcal{I}_s^{\mathbb{Y}(\omega)}}(\mathbf{x}, \omega)\|_2^2] \\ &\quad - \frac{\gamma}{2}\|\nabla_{\mathcal{I}_s^\mathbf{x}} f(\mathbf{x})\|_2^2 + \gamma\mathbb{E}_\omega[\|\delta_{\mathcal{I}_s^{\mathbb{Y}(\omega)}}(\omega)\|_2^2] \end{aligned} \quad (9)$$

where $\mathcal{I}(\omega) = \mathcal{I}_s^\mathbf{x} \cup \mathcal{I}_s^{\mathbb{Y}(\omega)}$ and $\delta(\omega) = \mathbf{g}(\mathbf{x}, \omega) - \nabla f(\mathbf{x})$.

The following Theorem is the climax of our technical results because it establishes a stochastic gradient descent property for the expectation of the function value. Later we will see how Inequality (11) is used in Theorem 3 to show the sequence of the function values generated by the mini-batch SIHT is a supermartingale sequence.

Theorem 2. Let $f^{(i)} : \mathbb{R}^n \times \Xi \rightarrow \mathbb{R}$ be in C^{11} for $i = 1, \dots, N$ and $\Xi = \{\xi^{(1)}, \dots, \xi^{(N)}\}$ be a given set such that $f(\mathbf{x}, \Xi) = \frac{1}{N} \sum_{i=1}^N f^{(i)}(\mathbf{x}, \xi^{(i)})$ be an Ls-RSS function. Assume there exists a $c > 0^2$ such that

$$\mathbb{E}_{\mathcal{J}} \left[\sum_{i=1}^N \|\nabla_{\mathcal{J}} f^{(i)}(\mathbf{x}, \xi^{(i)})\|_2^2 \right] \leq c \mathbb{E}_{\mathcal{J}} \left[\|\nabla_{\mathcal{J}} f(\mathbf{x}, \Xi)\|_2^2 \right] \quad (10)$$

for all $\mathbf{x} \in \mathbb{R}^n$ and any random index set $\mathcal{J} \subseteq \{1, \dots, n\}$ with $|\mathcal{J}| \leq s$. Let $\mathcal{G}(\mathbf{x}, \Xi, B) = \frac{1}{|B|} \sum_{i \in B} \nabla f^{(i)}(\mathbf{x}, \xi^{(i)})$ be the mini-batch stochastic gradient at any $\mathbf{x} \in \mathbb{R}^n$ where $B \subseteq \{1, \dots, N\}$ be a random set whose elements are drawn randomly and uniformly from $\{1, \dots, N\}$ without replacement and its size is $|B|$. For a fixed $0 < \gamma < \frac{1}{L_s}$, assume the size of B is fixed such that $|B| \geq$

¹ The class consisting of all differentiable functions whose derivative is continuous.

² In Remark 1, we explain why such a c always exist for widespread objective functions in machine learning applications.

$N/\left(1 + \frac{1-L_s\gamma}{1+L_s\gamma} \frac{N-1}{\frac{c}{N}-1}\right)$ and let $\zeta := \frac{N-|B|}{|B|(N-1)}$ for $N \geq 2$. Then for a fixed $\mathbf{x} \in C_s$ with any $\mathcal{I}_s^{\mathbf{x}}$ the following holds for any $\mathbb{Y}(B) \in H_s(\mathbf{x} - \gamma \mathbf{g}(\mathbf{x}, \Xi, B))$ with any $\mathcal{I}_s^{\mathbb{Y}(B)}$:

$$\begin{aligned} \mathbb{E}_B \left[f(\mathbb{Y}(B), \Xi) \right] &\leq f(\mathbf{x}, \Xi) - \frac{\gamma}{2} \|\nabla_{\mathcal{I}_s^{\mathbf{x}}} f(\mathbf{x})\|_2^2 \\ &\quad - \frac{\gamma}{2} (1 + L_s \gamma) \zeta \left(1 - \frac{c}{N} + \frac{1 - L_s \gamma}{1 + L_s \gamma} \frac{1}{\zeta} \right) \mathbb{E}_{\mathcal{I}_s^{\mathbb{Y}(B)}} \left[\|\nabla_{\mathcal{I}_s^{\mathbb{Y}(B)}} f(\mathbf{x}, \Xi)\|^2 \right] \end{aligned} \quad (11)$$

where $1 - \frac{c}{N} + \frac{1 - L_s \gamma}{1 + L_s \gamma} \frac{1}{\zeta} \geq 0$.

A crucial assumption for proving the results in Theorem (11) is the assumption made in Inequality (10). In the following Claim we show that for a certain class of functions $c > 0$ always exists and it does not depend on the function. We will prove that for these special classes of functions the value of c only depends on the data.

Claim. Let the given set Ξ in Problem (P) be defined such that $\Xi := \{\mathbf{V}_{1\bullet}, \dots, \mathbf{V}_{N\bullet}\}$ where each $\mathbf{V}_{i\bullet}$ is the i -th row of a given matrix $\mathbf{V} \in \mathbb{R}^{N \times n}$. Then the objective function in Problem (P) can be defined as $f(\mathbf{x}, \Xi) := \frac{1}{N} \sum_{i=1}^N f^{(i)}(\mathbf{V}_{i\bullet} \mathbf{x})$ $f^{(i)} : \mathbb{R}^n \times \Xi \rightarrow \mathbb{R}$ and the following holds:

$$\sum_{i=1}^N \|\nabla_{\mathcal{J}} f^{(i)}(\mathbf{V}_{i\bullet} \mathbf{x})\|_2^2 \leq \frac{N^2}{\sigma_{min}^2(\mathbf{V} \mathbf{I}_{\mathcal{J}\bullet}^\top \mathbf{I}_{\mathcal{J}\bullet} \mathbf{V}^\top)} \left(\max_{r=1, \dots, N} \left\{ \|(\mathbf{V}_{r\bullet}^\top)_{\mathcal{J}}\|_2^2 \right\} \right) \|\nabla_{\mathcal{J}} f(\mathbf{x}, \mathbf{V})\|_2^2 \quad (12)$$

where $\mathcal{J} \subseteq \{1, \dots, n\}$ with $|\mathcal{J}| \leq s$, $\mathbf{I}_{\mathcal{J}\bullet} \in \mathbb{R}^{|\mathcal{J}| \times n}$ is a restriction of the identity matrix whose rows are associated with indices in \mathcal{J} , $\mathbf{V} \mathbf{I}_{\mathcal{J}\bullet}^\top \mathbf{I}_{\mathcal{J}\bullet} = \sum_{i=1}^{|\mathcal{J}|} \mathbf{V}_{\bullet i} \mathbf{V}_{\bullet i}^\top$, $\sigma_{min}(\cdot)$ is the smallest singular value, $\mathbf{V}_{\bullet i}$ is the i -th column of \mathbf{V} , and $(\cdot)_{\mathcal{J}}$ is a vector restricted to indices in \mathcal{J} .

Remark 1. The above claim shows that for a class of functions $f(\mathbf{x}, \Xi) := \frac{1}{N} \sum_{i=1}^N f^{(i)}(\mathbf{V}_{i\bullet} \mathbf{x})$ the constant $c > 0$ in Theorem 3 always exists and it does not depend on the value of \mathbf{x} or its gradient whether it is batch (full) gradient or individual one. For an example of functions belonging to this class one can think of the mean square error loss used for linear regression as follows:

$$f(\mathbf{x}, \mathbf{V}) = \frac{1}{N} \|\mathbf{V}\mathbf{x} - \mathbf{y}\|^2 = \frac{1}{N} \sum_{i=1}^N (\mathbf{V}_{i\bullet} \mathbf{x} - y_i)^2$$

where $\mathbf{V} \in \mathbb{R}^{N \times n}$, $\mathbf{V}_{i\bullet}$ is the i -th row of \mathbf{V} , $\mathbf{x} \in \mathbb{R}^n$ is the optimization variable, and $\mathbf{y} \in \mathbb{R}^N$ is the target. Also, the logistic regression loss (binary cross entropy) is a function for which $c > 0$ in Inequality (12) always exists since it can be written as follows:

$$f(\mathbf{x}, \mathbf{V}) = \frac{1}{N} \sum_{i=1}^N \left(-y^{(i)}(\mathbf{V}_{i\bullet} \mathbf{x}) + \log(1 + e^{\mathbf{V}_{i\bullet} \mathbf{x}}) \right)$$

where $\mathbf{V} \in \mathbb{R}^{N \times n}$ whose last column is all one, $\mathbf{V}_{i\bullet}$ is the i -th row of \mathbf{V} , $\mathbb{R}^n \ni \mathbf{x} = [\mathbf{w}, b]^\top$ such that $\mathbf{w} \in \mathbb{R}^{n-1}$ and $b \in \mathbb{R}$ are the optimization variables, and $y^{(i)} \in \{0, 1\}$ for $i = 1, \dots, N$.

Now we can provide a result showing that by fixing a sparse point, one can use the stochastic mini-batch gradient with a fixed mini-batch size determined in Theorem 3 and decrease the function value in expectation.

Theorem 3. *Assume all the assumptions in Theorem 2 hold. Then for a fixed $\mathbf{x} \in C_s$ with any $\mathcal{I}_s^{\mathbf{x}}$ the following holds for any $\mathbb{Y}(B) \in H_s(\mathbf{x} - \gamma \mathcal{G}(\mathbf{x}, \Xi, B))$:*

$$\mathbb{E}_B \left[f(\mathbb{Y}(B), \Xi) \middle| \mathbf{x} \right] \leq f(\mathbf{x}, \Xi) - \frac{\gamma}{2} \|\nabla_{\mathcal{I}_s^{\mathbf{x}}} f(\mathbf{x})\|_2^2. \quad (13)$$

The above result is the analogue result to [8, Corollary 1].

Theorem 4. *Assume all the assumptions in Theorem 2 hold. Let f be a bounded below differential function and $(\mathbb{X}^k \middle| \mathbb{X}^{k-1})_{k \geq 0}$ be the stochastic IHT sequence. Then, $\left(f(\mathbb{X}^k, \Xi, B) \middle| \mathbb{X}^k \right)_{k \geq 1}$ is a supermartingale sequence and converges to a random variable f^* with probability one.*

5 Conclusion

We showed the stochastic sequence generated by the mini-batch stochastic IHT is a supermartingale sequence converging with probability one. To show this result we used the stochastic gradient descent property that we derived where we utilized the property of the mini-batch stochastic gradient as the sample sum of a finite sum.

References

1. Bahmani, S., Raj, B., Boufounos, P.T.: Greedy sparsity-constrained optimization. *J. Mach. Learn. Res.* **14**, 807–841 (2013)
2. Beck, A., Eldar, Y.C.: Sparsity constrained nonlinear optimization: Optimality conditions and algorithms. *SIAM J. Optim.* **23**(3), 1480–1509 (2013)
3. Blumsath, T.: Accelerated iterative hard thresholding. *Signal Process.* **92**(3), 752–756 (2012)
4. Blumsath, T., Davies, M.E.: Iterative thresholding for sparse approximations. *J. Fourier Anal. Appl.* **14**(5), 629–654 (2008)
5. Blumsath, T., Davies, M.E.: Iterative hard thresholding for compressed sensing. *Appl. Comput. Harmon. Anal.* **27**(3), 265–274 (2009)
6. Chen, J., Gu, Q.: Accelerated stochastic block coordinate gradient descent for sparsity constrained nonconvex optimization. In: UAI (2016)
7. Damadi, S., Nouri, E., Pirsavash, H.: Amenable sparse network investigator (2022). [arXiv:2202.09284](https://arxiv.org/abs/2202.09284)

8. Damadi, S., Shen, J.: Gradient properties of hard thresholding operator (2022). [arXiv:2209.08247](https://arxiv.org/abs/2209.08247)
9. Jain, P., Tewari, A., Kar, P.: On iterative hard thresholding methods for high-dimensional m-estimation. *Adv. Neural Inf. Process. Syst.* **27** (2014)
10. Khanna, R., Kyriolidis, A.: Iht dies hard: Provable accelerated iterative hard thresholding. In: International Conference on Artificial Intelligence and Statistics, pp. 188–198. PMLR (2018)
11. Li, X., Arora, R., Liu, H., Haupt, J., Zhao, T.: Nonconvex sparse learning via stochastic optimization with progressive variance reduction (2016). [arXiv:1605.02711](https://arxiv.org/abs/1605.02711)
12. Liang, G., Tong, Q., Zhu, C., Bi, J.: An effective hard thresholding method based on stochastic variance reduction for nonconvex sparse learning. *Proc. AAAI Conf. Artif. Intell.* **34**, 1585–1592 (2020)
13. Liu, B., Yuan, X.-T., Wang, L., Liu, Q., Metaxas, D.N.: Dual iterative hard thresholding: From non-convex sparse minimization to non-smooth concave maximization. In: International Conference on Machine Learning, pp. 2179–2187. PMLR (2017)
14. Liu, H., Foygel Barber, R.: Between hard and soft thresholding: optimal iterative thresholding algorithms. *Inf. Inference: J. IMA* **9**(4), 899–933 (2020)
15. Zhaosong, L.: Iterative hard thresholding methods for l_0 regularized convex cone programming. *Math. Program.* **147**(1), 125–154 (2014)
16. Lu, Z.: Optimization over sparse symmetric sets via a nonmonotone projected gradient method. *Optim. Control* (2015)
17. Mathai, A.M., Provost, S.B.: Quadratic Forms in Random Variables: Theory and Applications. Dekker (1992)
18. Nguyen, N., Needell, D., Woolf, T.: Linear convergence of stochastic iterative greedy algorithms with sparse constraints. *IEEE Trans. Inf. Theory* **63**(11), 6869–6895 (2017)
19. Pan, L., Zhou, S., Xiu, N., Qi, H.: A convergent iterative hard thresholding for sparsity and nonnegativity constrained optimization. *Pac. J. Optim.* **13**(2), 325–353 (2017)
20. Robbins, H., Monro, S.: A stochastic approximation method. *Ann. Math. Stat.* 400–407 (1951)
21. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. *Nature* **323**(6088), 533–536 (1986)
22. Shen, J., Li, P.: A tight bound of hard thresholding. *J. Mach. Learn. Res.* **18**(1), 7650–7691 (2017)
23. Vu, T., Raich, R.: Accelerating iterative hard thresholding for low-rank matrix completion via adaptive restart. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2917–2921. IEEE (2019)
24. Wang, R., Xiu, N., Zhou, S.: Fast newton method for sparse logistic regression (2019). [ArXiv:1901.02768](https://arxiv.org/abs/1901.02768)
25. Fan, W., Bian, W.: Accelerated iterative hard thresholding algorithm for l_0 regularized regression problem. *J. Global Optim.* **76**(4), 819–840 (2020)
26. Zhao, C., Xiu, N., Qi, H., Luo, Z.: A lagrange-newton algorithm for sparse nonlinear programming. *Math. Program.* 1–26 (2021)
27. Zhou, P., Yuan, X., Feng, J.: Efficient stochastic gradient hard thresholding. *Adv. Neural Inf. Process. Syst.* **31** (2018)
28. Zhou, S., Pan, L., Xiu, N.: Subspace newton method for the l_0 -regularized optimization (2020). [ArXiv:2004.05132](https://arxiv.org/abs/2004.05132)

29. Zhou, S., Xiu, N., Qi, H.-D.: Global and quadratic convergence of newton hard-thresholding pursuit. *J. Mach. Learn. Res.* **22**(12), 1–45 (2021)
30. Zhu, W., Dong, Z., Yuanlong, Yu., Chen, J.: Lagrange dual method for sparsity constrained optimization. *IEEE Access* **6**, 28404–28416 (2018)



Graph Autoencoder-Based Detection of Unseen False Data Injection Attacks in Smart Grids

Abdulrahman Takiddin^{1(✉)}, Muhammad Ismail², Rachad Atat³,
Katherine R. Davis¹, and Erchin Serpedin¹

¹ Texas A&M University, College Station, TX 77843, USA

abdulrahman.takiddin@tamu.edu, katedavis@tamu.edu, eserpedin@tamu.edu

² Tennessee Tech University, Cookeville, TN 38505, USA

mismail@tnstate.edu

³ Texas A&M University at Qatar, Doha, Qatar

rachad.atat@qatar.tamu.edu

Abstract. A major concern in smart power grids is when malicious or manipulated data is injected into measurement data due to malicious activities. Several approaches have been investigated to counter such false data injection attacks (FDIAs). However, such data-driven detectors present two major limitations. First, they neglect capturing the grid's spatial characteristics. Second, they offer limited attack identification to familiar types of FDIAs since they are present within the model's train sets. To conquer such limitations, we propose the use of an artificial intelligence-based graph autoencoder (GAE) for FDIAs detection. Our proposed detector offers three main advantages compared to existing detectors. First, it employs the operation of graph convolution to apprehend the grid's spatial characteristics. Second, it offers an unsupervised autoencoder-based anomaly detection that requires only benign samples under normal operation for training. Third, it outperforms existing detectors by 16–47% in FDIAs detection rate (DR) when tested against unseen FDIAs on an IEEE 39-bus system.

Keywords: Cyberattacks · Graph neural network · Machine learning · Smart grid

1 Introduction

The decision making within smart power grids is highly dependent on measurement data collected from several components among the power grid for proper operation [1]. Therefore, the integrity of the collected data is critical to ensure the reliability of the system and for stable operation. However, malicious entities may carry out stealthy attacks (e.g., false data injection attacks (FDIAs)) to

This work was supported by NSF EPCN Awards 2220346 and 2220347.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
K. Arai (Ed.): IntelliSys 2023, LNNS 822, pp. 234–244, 2024.

https://doi.org/10.1007/978-3-031-47721-8_16

manipulate measurement data from sensors and hence jeopardize the integrity of the power grid data [2]. As a result, decision making will be based on inaccurate measurement values, which might lead to instabilities or overload in the system [3]. Unfortunately, such stealthy attacks can bypass existing bad data detectors [4]. Thus, more complex attempts have been proposed to detect such FDIs employing multiple data-driven-based approaches.

1.1 Related Work and Limitations

Several approaches have been investigated to counter such FDIs. We divide these approaches into three main categories, namely, shallow machine learning (ML), deep learning (DL), and graph-based models. Next, we report the performance of relevant studies along with their limitations.

Shallow ML Models Relevant shallow ML-based FDIs detection schemes employ the following. Support vector machines (SVMs) provided 82% in F1-Score [5]. A decision trees model offered an F1-Score of 88% [6]. A random forest model reported an attack detection rate (DR) of 93% [7]. Nevertheless, such shallow models do not apprehend the patterns and spatial characteristics of the data [8]. They also present supervised learning that offers detection limited to the familiar attacks that are seen the models' train sets. Hence, they are susceptible to new types of attacks (i.e., not present in the train sets).

DL Models DL-based detectors have been proposed to apprehend the pattern characteristics within the data [9]. To achieve this, DL-based detectors employ the following models. A feedforward neural network (FNN) model provided an accuracy (ACC) score of 90% [10]. A convolutional neural network (CNN) model offered an ACC score of 93% [11]. A recurrent neural network (RNN) model offered a DR of 96% [12]. Although these DL-based detectors are able to apprehend the data patterns that are sophisticated using deep neural networks, they still fail to detain the system's spatial characteristics [4]. Also, they still offer limited detection performance against new types of attacks that are not present in the train sets [13].

Graph Models Graph-based detectors have been proposed to capture the grid spatial information. In particular, graph-based detectors employ graph signal processing (GSP) and graph neural network (GNN) models. GSP models employ spectral filters that are manually designed [14–16] and provided DRs of nearly 90%. However, the custom design of the filter limits the scalability of the model [4]. To overcome this, GNN models have been proposed. Specifically, a convolutional GNN (CGNN) model that incorporates the GSP operation automatically and utilizes undirected graphs offered DRs of 83–96% [4]. Despite the provided advantages, existing GNN-based FDIs detection schemes still offer attack detection only against seen attack types that are part of their training

sets due to their supervised learning nature. However, in practice, the system might encounter new unseen attack types (i.e., zero-day attacks) that are different than the types the detector has been trained on.

According to the presented limitations, there is a need to improve the attack identification performance of existing state-of-the-art models. This could be achieved by proposing an artificial intelligence-based robust detection strategy that apprehends the system's sophisticated patterns as well as the spatial characteristics while offering robust identification against new types of attacks that are not present in the train sets.

1.2 Contributions

We conquer the drawbacks of existing FDIA detection schemes by proposing a graph-based unsupervised anomaly detector. The proposed detector employs a graph autoencoder (GAE) providing three major benefits. The advantages of the proposed detector are highlighted by comparing it to various data-driven detection strategies. Specifically, the GAE model offers the following.

- It presents a deep structure with stacked graph encoder and decoder layers that detain the complex patterns of the measurement data. It is also able to detain the grid's spatial characteristics as it employs a graph Chebyshev convolution operation.
- It offers detection of new FDIA types (i.e., types that are not present during training) as it represents an unsupervised anomaly detection scheme that relies only on benign data during training. During testing, it marks unseen malicious samples of under-attack operation according to the presented dissimilarity from the features of normal operation (benign samples) that were learned during the training stage.
- It offers a superior DR of 90.2% against unseen attacks in an IEEE 39-bus system, providing DR enhancements of 16–47% compared to a comprehensive list of benchmarks including shallow, deep, and graph-based detectors.

The layout of the paper is as follows. Section 2 describes the data preparation and the investigated attack types. Section 3 presents the architecture of the GAE model. Section 4 introduces the benchmark detectors and reports the detection performance. Section 5 concludes the outcomes of this work.

2 Data Preparation

For the training and testing of the studied models, we employ an IEEE 39-bus system. To generate malicious samples replicating the under-attack system operation, we acquire three FDIA types [17].

2.1 System Model

In this work, we detain the spatial and temporal characteristics of the power system (i.e., IEEE 39-bus system). Specifically, we model the system via an undirected graph. Within the graph, buses are depicted by vertices (nodes) whereas power lines are denoted by edges. Figure 1 shows the adopted IEEE 39-bus system modeled as an undirected graph. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ denote the undirected graph with vertices \mathcal{V} , edges \mathcal{E} , and weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$. In \mathcal{G} , when bus i is connected to bus j , a weight W_{ij} is allied to an edge $e = (i, j)$ according to the line admittance.

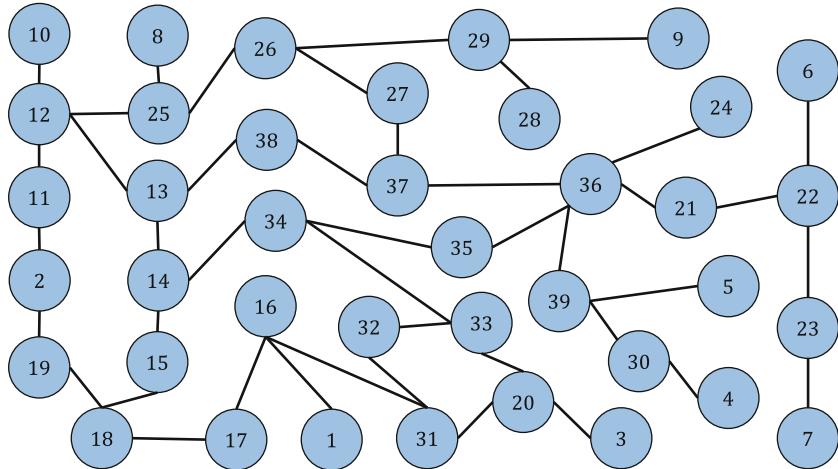


Fig. 1. IEEE 39-bus system graph illustration.

In addition to the spatial aspects, temporal characteristics referring to power injections and flows are also captured where \mathcal{V} and \mathcal{E} are accompanied with features. To detain such features, we adopt an analysis of the power flow through Newton's method via MATLAB MATPOWER toolbox [18]. This is carried out to establish the flows of real and reactive power in the system. Specifically, the features of \mathcal{V} comprise the active power (i.e., real power demand) P_i and reactive power demand Q_i in MW and MVAr, respectively. The features of \mathcal{E} comprise the real power flow P_{ij} from bus i to bus j in MW and the reactive power flow Q_{ij} from bus i to bus j in MVAr.

2.2 Benign Data

We adopt the feature values discussed above to represent measurement data of normal operation, which results in generating benign samples denoted as $\mathbf{x}_b(t, i)$ at bus i and timestamp t . Specifically, over a period of half a year, each hour, we report 4 power dynamics timestamps, which leads to approximately 17,000 timestamps in total.

2.3 Malicious Data

To constitute the malicious data representing the under-attack system operation, we adopt three FDIA functions, namely, direct, replay, and general attacks. These attacks are applied to $\mathbf{x}_b(t, i)$ and bypass existing bad data detectors since they present similar data patterns as benign samples [19]. The generated malicious samples are denoted as $\mathbf{x}_m(t, i)$ at bus i and timestamp t . The three FDIA types are described next.

The direct attack applies specific perturbations bounded by a scaling factor $|\alpha| \leq 0.05$ that are injected into benign samples. For instance,

$$\mathbf{x}_m(t, i) = \mathbf{x}_b(t, i) + \alpha \cdot \mathbf{x}_b(t, i). \quad (1)$$

The replay attack generates malicious samples throughout a false repetition of readings from a prior timestamp $t - 1$. As a result, the true reading of a present timestamp t is replaced as follows

$$\mathbf{x}_m(t, i) = \mathbf{x}_b(t - 1, i). \quad (2)$$

The general attack [20] uses a true measurement value interval along with a binary β and uniform random $0 \leq \gamma \leq 1$ variables to create malicious samples where

$$\mathbf{x}_m(t, i) = \mathbf{x}_b(t, i) + (-1)^\beta \alpha \cdot \gamma \cdot \text{Range}(\mathbf{x}_b(t, i)), \quad (3)$$

2.4 Dataset Splitting

The generated sample types (benign \mathbf{x}_b and malicious \mathbf{x}_m) are equal in number. Supervised models are required to be trained and tested on both sample types (\mathbf{x}_b and \mathbf{x}_m). On the other hand, unsupervised models necessitate to be trained on \mathbf{x}_b , but they are still tested on both samples types. To carry out the experiments, we split the samples into three sets where the training \mathbf{X}_{TR} , validation \mathbf{X}_{VA} , and testing \mathbf{X}_{TS} sets represent 80, 10, and 10% of samples, respectively. To avoid any bias, all sets have equal numbers of samples of both types.

3 GAE-Based Detector

To detect FDIA, we put forward an unsupervised GAE-based anomaly detector. For training, the proposed detector relies on graph expressions of data during normal operation (benign samples) [21]. Hence, it offers identification ability against new types of attacks that are not present in the train sets.

3.1 GAE Model Architecture

The GAE model employs an autoencoder that utilizes graph encoding and decoding layers. Such layers help when it comes to apprehending the graph expressions of normal operation (i.e., benign data) via a data reconstruction process [22], [23]. The overall process is illustrated in Fig. 2.

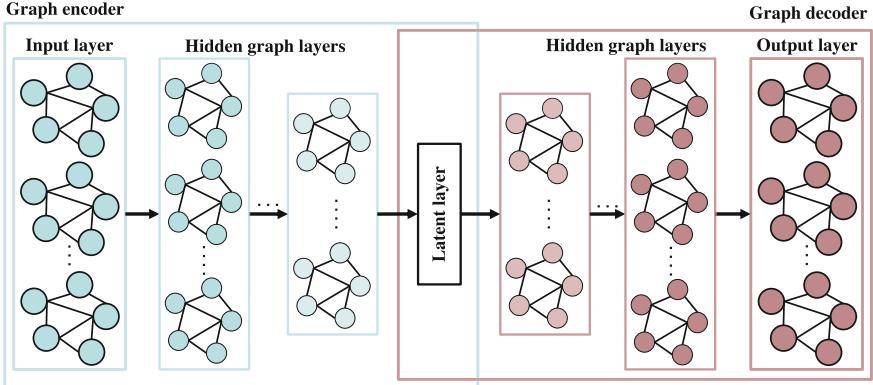


Fig. 2. Proposed unsupervised GAE model architecture.

The GAE model operates as follows. First, it takes benign samples' \mathbf{x}_b with $[P_i, Q_i] \in \mathbb{R}^{n \times 2}$ measurements as input. Then, following the input layer, the encoder \mathbf{E} with \mathcal{L}_E hidden graph encoding convolution layers are placed. Applying the convolution operation to graph signals is essential when retaining the grid's spatial characteristics. The graph encoding block is responsible for compressing the data. On the graph encoder side, the number of channels depicting what is fed to the convolution layers is denoted by c_{l_E} in a hidden encoding layer l_E . The input and output of l_E are $\mathbf{X}^{l_E-1} \in \mathbb{R}^{n \times c_{l_E-1}}$ and $\mathbf{X}^{l_E} \in \mathbb{R}^{n \times c_{l_E}}$, respectively. The presence of the encoding layers helps in capturing the spatial characteristics as well as constructing the output tensor, which is expressed next

$$\mathbf{X}^{l_E} = \text{ReLU}(\boldsymbol{\theta}^{l_E} *_{\mathcal{G}} \mathbf{X}^{l_E-1} + \mathbf{b}^{l_E}). \quad (4)$$

In (4), $\boldsymbol{\theta}^{l_E} \in \mathbb{R}^{K \times c_{l_E-1} \times c_l}$, $\mathbf{b}^{l_E} \in \mathbb{R}^{c_{l_E}}$, and $*_{\mathcal{G}}$ depict the Chebyshev coefficients, bias, graph convolution operator. The role of the added bias and ReLU activation function is to improve the model's non-linearity ability [24].

The encoding block is followed by a latent layer, which is responsible for holding the representations of the compressed data throughout the encoding process. Thus, the presence of the latent layer enhances the features' learning process and helps in learning simpler data representations. The graph decoder block is placed after the latent layer. The role of the decoder \mathbf{D} is decompressing the data and reconstructing the input. The graph decoder consists of \mathcal{L}_D hidden decoding graph Chebyshev convolution layers with c_{l_D} channels. Each graph decoding layer l_D has the input and output of $\mathbf{X}^{l_D-1} \in \mathbb{R}^{n \times c_{l_D-1}}$ and $\mathbf{X}^{l_D} \in \mathbb{R}^{n \times c_{l_D}}$, respectively. Finally, $\tilde{\mathbf{X}}$ denotes the reconstructed input by the graph decoder.

3.2 Training and Testing the GAE Model

Our GAE model is trained on benign samples and tested on both, benign and malicious samples. Specifically, it recognizes malicious samples of under-attack

operation based on the presented dissimilarity from the learned normal patterns during training. Since the model is familiar with the patterns of normal operation during training, the dissimilarity is expected to be small during testing. This means that under-attack operation samples are expected to present higher dissimilarity during testing. Marking samples during testing is carried out based on a reconstruction error ζ of the data regeneration procedure. The graph encoder and decoder are denoted as $E = f_\Phi(\mathbf{X})$ and $D = g_\Phi(\mathbf{X})$, respectively, where Φ depicts the GAE model parameters, which are expressed as follows

$$\min_{\Phi} C(\mathbf{X}, g_\Phi(f_\Phi(\mathbf{X}))), \quad \mathbf{X} \in \mathbf{X}_{\text{TR}}. \quad (5)$$

In (5), $C(\mathbf{X}, g_\Phi(f_\Phi(\mathbf{X})))$ represents a mean squared error (MSE) cost function that imposes a penalty on $g_\Phi(f_\Phi(\mathbf{X}))$ for the presented dissimilarity from \mathbf{X} . In other words, (5) estimates the MSE between the original input \mathbf{X} and the reconstructed output $\tilde{\mathbf{X}}$. The proposed GAE model is trained with the goal of identifying parameters Φ with the aim of optimizing (5). Using the iterative gradient descent approach, the minimization of (5) is accomplished where we divide the training samples $\mathbf{X} \in \mathbf{X}_{\text{TR}}$ into small batches. Following (5), the value of ζ , which indicates the level of the model's familiarity against $\mathbf{X} \in \mathbf{X}_{\text{TST}}$, is anticipated to be small and large for benign and malicious samples, respectively. When the value of ζ becomes higher than a threshold value ψ , a malicious sample \mathbf{x}_m reflecting an attack is flagged with $y = 1$, otherwise, the sample is considered benign \mathbf{x}_b with a $y = 0$ label.

4 Experimental Results

This section assesses the performance of the GAE model compared to several benchmark detectors. Also, we present the used hyperparameters for each model that are selected based on a grid-search selection process. We then analyze the performance of the examined detectors.

4.1 Benchmark Detectors

For an exhaustive analysis, we include multiple data-driven benchmark detectors with shallow, deep, and graph models that are either supervised (trained on benign samples only and tested on benign and attack samples) or unsupervised (trained and tested on benign and attack samples). The adopted shallow models are listed next. The unsupervised auto-regressive integrated moving average (ARIMA) model is trained to predict future data patterns [25]. The supervised SVM model classifies samples using a hyperplane that separates both sample types [5]. The adopted deep models include the FNN, which is a supervised model that employs feedforward layers that are fully-connected to classify samples [10]. The supervised RNN model exploits temporal correlations via utilizing recurrent cells [26]. The supervised CNN model performs the convolution operation to classify samples [11]. We also adopt a classical stacked autoencoder

(SAE), which is an unsupervised model that identifies samples using a reconstruction process using fully-connected feedforward layers [22] without employing a graph convolution operation. Finally, we adopt a graph-based detector, which is a supervised CGNN model that utilizes vertices and edges for modeling the spatial aspects of the data [4].

4.2 Hyperparameter Selection

To select the most suitable hyperparameters for each of the adopted models, we utilize a grid-search selection process that is carried out on multiple stages. The best hyperparameter option is picked from a pool of options according to the offered DR calculated against \mathbf{X}_{VA} by that value. The selected hyperparameters are listed next. ARIMA uses 1 and 0 as the differencing and moving averages, respectively. SVM uses scale and sigmoid as the kernel and gamma values, respectively. FNN employs 4 layers, 32 units, Adamax optimizer, and ELU activation without dropout. RNN employs 3 layers, 16 units, Adam optimizer, and ReLU activation with a dropout rate of 0.2. CNN employs 4 layers, 32 units, 5 neighborhood order, Adam optimizer, and ReLU activation. SAE employs 3 encoding layers with (32, 16, 8) units, 3 decoding layers with (8, 16, 32) units, Adam optimizer and Simoid activation without dropout. CGNN employs 4 layers, 32 units, 3 neighborhood order, Adam optimizer, and ReLU activation. The proposed GAE-model employs 3 encoding layers with (32, 16, 8) units, 3 decoding layers with (8, 16, 32) units, 4 neighborhood order, Rmsprop optimizer, and ReLU activation.

4.3 Evaluation Metrics

To evaluate the models, we adopt the following assessment metrics. First, $DR = TP / (TP + FN)$ reflects the model's ability to correctly mark malicious samples, where TP and FN denote true positive and false negative, respectively. Second, false alarm rate ($FAR = FP / (FP + TN)$) reflects the quantity of benign samples that the model incorrectly marks as malicious, where FP and TN depict false positive and true negative, respectively. Third, $ACC = (TP + TN) / (TP + TN + FP + FN)$ reflects the model's ability to mark both sample types.

4.4 Detection Performance

Tables 1 and 2 present the results of the investigated detectors. Table 1 reports the evaluation results when the supervised models (SVM, FNN, RNN, CNN, and CGNN) encounter seen attacks and when unsupervised detectors (ARIMA, AE, and GAE) encounter unseen (new) attacks. The GAE model demonstrates superior DR by 33.1–36.3%, 17.8–29.2%, and 2.5% compared to the graph, deep, and shallow ML-based detectors, respectively. The reason behind the superior performance of the proposed detector is that it learns the graph representations and captures spatial aspects of normal operation of the power system without

the need of being trained on malicious samples. Hence, unlike existing supervised detectors, the offered attack identification results by the GAE model are not limited to a set of FDIA, which highlights its superiority.

Table 1. Evaluation results against FDIA (%) .

Detector	DR	FAR	ACC
ARIMA	53.9	53.6	53.1
SVM	57.1	45.8	56.1
FNN	61	39.3	60
RNN	66.6	32.8	65.4
CNN	71.6	25.9	71.3
SAE	72.4	24.7	72
CGNN	84.4	13.6	83.5
GAE	90.2	9.3	89.8

In real-life, attackers might launch unseen new FDIA types. These attack types might not be present during training stage of the model. To reflect such a scenario, in Table 2, we present the evaluation results of supervised models when encountering new unseen attacks compared to the unsupervised GAE model. The detection performances of such detectors significantly degrade when they are tested against unseen FDIA types. Specifically, the DRs of shallow, deep, and graph-based detectors deteriorate by 13.9%, 10.1–12.7%, and 9.8% compared to encountering seen attacks. This means that the proposed GAE-based detector provides an increase of 47–15.6% in DR compared to supervised benchmark detectors against unseen FDIA types. The reason behind such enhancements is that the proposed GAE-based detector offers unsupervised training that learns the graph representations of benign behavior through the encoder-decoder benign data reconstruction process, which increases its robustness against unseen FDIA types.

Table 2. Evaluation results against unseen FDIA (%) .

Detector	DR	FAR	ACC
SVM	43.2	59.9	42.2
FNN	48.3	52.1	47.9
RNN	54.8	44.6	53.6
CNN	61.5	36.5	61.7
CGNN	74.6	24.3	73.7
GAE	90.2	9.3	89.8

5 Conclusions

This work proposed adopting an artificial intelligence-based GAE unsupervised anomaly detector that provides three major advantages compared to existing FDIA detectors. First, it employs Chebyshev graph convolution operation. Thus, it captures the grid's spatial characteristics. Second, it offers an unsupervised learning strategy using an autoencoder that relies only on benign samples of normal operation for training and hence offers robust detection of unobserved types of FDIA that do not take part of the training process. Third, due to its structure that is equipped with stacked graph layers and its unsupervised learning nature, employing the proposed detector leads to a superior detection performance as opposed to benchmark detectors by 16–47% in DR against unseen FDIA types.

References

1. An, D., et al.: Data integrity attack in dynamic state estimation of smart grid: Attack model and countermeasures. *IEEE Trans. Autom. Sci. Eng.* **19**(3), 1631–1644 (2022). Jul
2. Zhang, Z., et al.: Cyber-physical coordinated risk mitigation in smart grids based on attack-defense game. *IEEE Trans. Power Syst.* **37**(1), 530–542 (2022). Jan
3. Huang, K., et al.: False data injection attacks detection in smart grid: A structural sparse matrix separation method. *IEEE Trans. Netw. Sci. Eng.* **8**(3), 2545–2558 (2021). Jul.
4. Boyaci, O., et al.: Graph neural networks based detection of stealth false data injection attacks in smart grids. *IEEE Syst. J.* **16**(2), 2946–2957 (2022). Jun
5. Esmalifalak, M., et al.: Detecting stealthy false data injection using machine learning in smart grid. *IEEE Syst. J.* **11**(3), 1644–1652 (2017). Sept
6. Lu, X., et al.: False data injection attack location detection based on classification method in smart grid. In: International Conference on Artificial Intelligence and Advanced Manufacture (AIAM), pp. 133–136. Manchester, United Kingdom (2020)
7. Wang, D., et al.: Detection of power grid disturbances and cyber-attacks based on machine learning. *J. Inf. Secur. Appl.* **46**, 42–52 (2019). Jun.
8. Musleh, A.S., et al.: A survey on the detection algorithms for false data injection attacks in smart grids. *IEEE Trans. Smart Grid* **11**(3), 2218–2234 (2020). May
9. Takiddin, A., et al.: Robust electricity theft detection against data poisoning attacks in smart grids. *IEEE Trans. Smart Grid* **12**(3), 2675–2684 (2021). May
10. Xue, D., et al.: Detection of false data injection attacks in smart grid utilizing ELM-Based OCON framework. *IEEE Access* **7**, 31 762–31 773 (2019)
11. Wang, S., et al.: Locational detection of the false data injection attack in a smart grid: A multilabel classification approach. *IEEE Internet Things J.* **7**(9), 8218–8227 (2020). Sept
12. Wang, Y., et al.: Kfrnn: An effective false data injection attack detection in smart grid based on kalman filter and recurrent neural network. *IEEE Internet Things J.* **9**(9), 6893–6904 (2022). May
13. Takiddin, A., Ismail, M., Serpedin, E.: Robust data-driven detection of electricity theft adversarial evasion attacks in smart grids. *IEEE Trans. Smart Grid* **14**(1), 663–676 (2023). Jan.

14. Drayer, E., et al.: Detection of false data injection attacks in power systems with graph fourier transform. In: IEEE Global Conference on Signal and Information Processing, pp. 135–140. Anaheim, CA, USA (2018)
15. Drayer, E., Routtenberg, T.: Detection of false data injection attacks in smart grids based on graph signal processing. *IEEE Syst. J.* **14**(2), 1886–1896 (2020). Jun
16. Ramakrishna, R., et al.: Detection of false data injection attack using graph signal processing for the power grid. In: IEEE Global Conference on Signal and Information Processing (GSIP). Ottawa, ON, Canada (2019)
17. Takiddin, A., et al.: Detection of electricity theft false data injection attacks in smart grids. In: 30th European Signal Processing Conference (EUSIPCO)), pp. 1541–1545. Belgrade, Serbia (2022)
18. Zimmerman, R.D., et al.: Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Trans. Power Syst.* **26**(1), 12–19 (2011). Feb
19. Takiddin, A., et al.: A graph neural network multi-task learning-based approach for detection and localization of cyberattacks in smart grids. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023), pp. 1–5. Rhodes Island, Greece (2023)
20. Hasnat, M., et al.: Detection and locating cyber and physical stresses in smart grids using graph signal processing (2020). [arXiv:2006.06095](https://arxiv.org/abs/2006.06095)
21. Stamile, C., et al.: Graph Machine Learning: Take Graph Data to the Next Level by Applying Machine Learning Techniques and Algorithms. Packt Publishing, Birmingham, United Kingdom (2021)
22. Takiddin, A., et al.: Deep autoencoder-based anomaly detection of electricity theft cyberattacks in smart grids. *IEEE Syst. J.* **16**(3), 4106–4117 (2022). Sept
23. Wu, L., et al.: Graph Neural Networks: Foundations, Frontiers, and Applications. Springer, Singapore (2022)
24. Takiddin, A., et al.: Generalized graph neural network-based detection of false data injection attacks in smart grids. *IEEE Trans. Emerg. Top. Comput. Intell.* **7**(3), 618–630 (2023). Jun
25. Krishna, V., et al.: ARIMA-Based modeling and validation of consumption readings in power grids. In: Critical Information Infrastructures Security, pp. 199–210. Springer Intl. Publishing
26. Takiddin, A., et al.: Data-driven detection of stealth cyber-attacks in dc microgrids. *IEEE Syst. J.* **16**(4), 6097–6106 (2022). Dec



Causal Analysis of Artificial Intelligence Adoption in Project Management

Egor Sarafanov, Omid Fatahi Valilai, and Hendro Wicaksono^(✉)

School of Business, Social & Decision Sciences, Constructor University, Campus Ring 1, 28759 Bremen, Germany
{ofatahivalilai,hwicaksono}@constructor.university

Abstract. Artificial intelligence (AI) technologies have great potential to improve decision-making and automation processes in various sectors, including project management. AI technologies could significantly contribute to overcoming the complexity of project management through process automation, cognitive insight, and engagement. However, the adoption of AI technologies still faces many challenges due to technical, human resource-related, organizational, and legal issues. Our research identified the potential factors that lead to the willingness of people and organizations to adopt AI technologies in project management. This paper proposes a causal model describing multivariate causal relationships between the driving factors and the willingness to adopt AI. The causal model is a set of hypotheses evaluated through a survey and causal analysis using the structural equation modeling (SEM) technique. The analysis focused on six factors influencing the willingness to adopt AI in project management, i.e., performance effectiveness, price, previous experience, feedback, complexity, and complementary technologies. Our research found that the perception of the high effectiveness of AI technologies leading to higher profits and overall the state of the project is the main factor influencing the willingness to adopt AI technologies in project management.

Keywords: Artificial intelligence · Project management · Causal model · Causal analysis · Structural equation modeling

1 Introduction

Technological progress affects many aspects of the modern world. Nowadays, technologies evolve rapidly, causing many technological and non-technological fields to transform. Things that looked unmanageable and impossible came to be due to this fast technological growth. Complex issues became straightforward with the help of new techniques, machinery, software, and others. For instance, humanity has gained the ability to operate vehicles remotely, program machinery to conduct monotonous industrial activities without human interaction, analyze consumer behavior and advise related products or services [15]. Nevertheless, many tasks and problems require autonomous solutions [8].

Companies adopt different technologies, methods, and approaches to optimize production, adjust costs, and ensure good revenue. At the same time, firm management has to be dependable, secure, and acquiescent [9]. Cooperation of human physical and intellectual labor and technical systems has led to increased performance and an improved environment in agriculture, medicine, and business [33]. Computer-human cooperation in project management is a successful example of that [38]. For example, Artificial Intelligence (AI) technologies have proved to manage risks arising during project activities effectively, deliver valuable insights and allocate physical and intellectual resources [3].

In particular, AI technologies, tools, and methods have been beneficial for managing various projects and tasks. AI helps to surpass human capabilities. This paper discusses artificial intelligence applications in project management and the benefits of their integration [20].

AI applies different methods and tools which can assist people in project management. For example, AI can effectively assign project roles based on specific factors. Moreover, it helps to keep track of the deadlines and agenda [30]. Another essential feature is reducing unnecessary steps, which could increase project duration. Besides, AI is accurate and precise with computations and sufficiently reduces errors, and mistakes [30]. Finally, AI offers managers alternative methods and strategies to optimize and complete the project.

One of the core components of the 4th Industrial Revolution or Industry 4.0 is AI technology capable of combining data and computation abilities created to function autonomously [23]. AI has been useful in many spheres, from production to business analytics [24]. It has significantly optimized various operational processes, automated production, and prevented and reduced errors and mistakes [3]. In particular, it has significantly affected the project management set of activities that require organization and control of all resources to achieve specific goals. 20 Companies all over the globe have started to apply AI on different scales. In the next few years, the AI market is expected to reach a \$ 190.61 billion market value, and by 2021, only 7% of companies do not use any AI technology [34].

Thus, the value of AI is inevitable, and its role is significant in many fields. Nevertheless, as said before, some enterprises are redundant in introducing AI tools or methods in their project activities. Some factors could explain such redundancy. Regarding various challenges, project managers are running into known and unknown issues: the futility of AI in the project, the complexity of the application, organizational and legal issues, and the lack of professionals. Those are why enterprises tend to stay with traditional methods of managing their projects [19].

This paper discusses AI, its application in project management, existing challenges during its application, and boundaries that prevent companies from inheriting AI technologies. We structure the paper as follows. In Sect. 2, we report the literature review on the role of AI in project management and the challenges while applying AI in project management. From the literature review, we also identified the factors influencing the adoption of AI in project management.

Section 3 elaborates a set of hypotheses that assumes the correlations among identified factors in the literature. We then explain the survey design to test the hypotheses in Sect. 4. Section 5 describes the targeted samples and the method to analyze the survey data. After the data gathering and analysis, we present the analysis results that confirm or reject the hypotheses in Sect. 6. Section 7 discusses the findings from the course of our research, and finally, Sect. 8 presents our research conclusions.

2 Literature Review

2.1 AI in Project Management

According to Gartner's "Top 10 Strategic Trends for 2017", AI stands at the number one position for technology companies want to adopt. Wauters (2016) has predicted a significant increase in AI investments in the upcoming years. AI technologies will optimize the working procedure, reduce recourse and time consumption, and decrease the number of faced issues and errors [35]. According to Belharet, Artificial Intelligence brings three essential concepts that simplify, arrange and help solve major and minor issues in Projects: Business Insights, Risk Management, and Resource Allocation [3]. AI helps to produce insights regarding possible outcomes. Technology provides those business insights by removing unnecessary data from large data tables and creating links between data, results, and trends. That lets managers focus on the most valuable things. One example of Business Insight technologies is virtual assistant [21].

Risk management is crucial for the proper functioning of any project, and AI can constantly access data and avoid undesired choices. For example, companies use machine learning to adjust project activities in near real-time regimes to bypass errors and failures [37].

The last concept is resource allocation. AI calculates best resource allocation, assigns specific skills for specific tasks, and provides feedback about employee performance and behavior. Even though some employees fear that the rising application of AI in PM may threaten their job position, AI will not reduce employee demand and automate controlling and organizational activities [3]. That will provide managers with a more significant amount of time to focus on crucial and complicated tasks. Besides, AI will reduce overall risks during management [17].

When introducing the technology into project management procedure, there are three main ways AI can contribute to projects: process automation, cognitive insight, and cognitive engagement [6]. Process automation is the most applied AI function and requires a minor financial investment among all three. An example of process automation is transferring data from one source and stashing it into records or automatically replacing lost credit cards [37].

Cognitive insight detects paternal data and interprets it. For example, trends in one's behavior, personalization of data online, digital fraud, and quality assessment [13]. Finally, cognitive engagement focuses on automated software that can interact with users and other programs. Examples would be automated customer

services, bots, and recommendation systems [32]. AI software in project management generates vast amounts of data regarding project states, key indicators, and project life cycles. This perspective supports more reasonable project decisions and helps compute the most fitting solutions. Such data is also crucial for correctly assessing and managing risks to mitigate risks and ensure unrealistic solutions are avoided [28].

Compared to the traditional tools and approaches to project management, AI has various superior traits: robust predictive tools, regular and thorough automated project reviews, decisive actions done according to the latest analysis and independent audit of project performance improvement, ability to make predictions based on trends of behavior [29]. Besides, AI function ability can be helpful directly for the team. AI can track team members' behavior and project habits and act according to those frames and constraints. In addition, the software chooses the best strategy to avoid possible conflicts, and complex solutions [25].

Before complete involvement in project management, AI will go through three stages (generations), according to Atlassian. The first generation of AI in PM applies project assistants for a narrow area, limited tasks in one defined area but not the project as a whole. For example, some companies use smart assistants for estimates and budgets. Tools automate tasks and provide insights. The second generation introduces new qualities, functions, and performance. That happens because AI has existed in projects for some time, and companies have gained experience and knowledge regarding its applications. Assistants will expand their understanding of tasks, introduce new metric systems and increase overall quality and performance. The third generation introduces even better quality and broader analytical skills. With the application's background in hundreds of projects, AI can make meaningful decisions. With the gained experience, AI can make assumptions regardless of insufficient data. Moreover, AI will create new levels of metadata to represent successful and valuable concepts that are not created yet [24].

In order to get more transparency about the application of AI in PM, this paper gives some common examples of these applications. The healthcare sphere has been accumulating vast amounts of essential data. AI in healthcare enables faster, in-depth analysis of scans, prescriptions, and recommendations. AI technologies have already helped doctors give diagnoses faster and more accurately. Predictions show that the application of AI will reduce costs 25% of costs by 2023 [29].

Transport nowadays can navigate and move through various routes without human intervention. Besides, AI assures higher passenger safety and reduces accidents, carbon emissions, and financial expenses. By 2023, the market for AI transportation technologies expects to reach 3.5 billion dollars [18]. Finance, banking, and insurance focus mainly on the experience that customers undergo. AI chatbots can improve this experience with faster and smarter responses based on personalized data. Automated processes, voice recognition, and Natural language Processing are the core AI features applied in this field [26].

2.2 Challenges While Applying AI in Project Management

Challenges are expected to arise in the different stages of the project while introducing AI into project management various issues. Based on Belharet's report on AI's impact on PM, there are five risk dimensions for AI when companies apply it in PM: security, privacy, autonomy, data quality/availability, and employment [3]. In terms of security, AI can be unsuitable for the standards that companies hold and thus can threaten projects and possibly lead to failure. Due to the lack of expertise, AI might be unable to distinguish between public and restricted data such as unauthorized recordings, personal data collection, and violation of privacy rights [21].

The autonomy given to the AI can dominate the project environment leading to project managers losing control and track of the processes and progress. A lack of valid and accurate data leads to invalid decisions and conclusions. Managing projects with unpredictability will fail to approach and solve challenges efficiently [24]. The presence of AI will reduce the number of low-skilled jobs and thus negatively affect the job market [21].

Scott Middleton has identified some significant challenges faced by AI in project management: data inconsistency, lack of creativity, hiring and retention issues [24]. First, data inconsistency arises when different formats of the same data conflict with each other on the value level. Such kind of data exists in all areas where project management is applied. Data inconsistency worsens predictive capabilities and leads to unwanted decisions. Thus, AI needs to assess the quality of input data constantly. Second, machines cannot make creative decisions based on structural analysis and previous experience. AI can mimic human behavior by copying some of the already existing ideas but lacks human "originality". However, AI can be taught some of the parameters of human creativity and develop a "parody" of it. Lastly, due to the relative novelty of AI in the PM sphere, enterprises might have to hire into their employees' professional development to operate with AI software successfully [3].

Even though AI is a popular and promising solution, there could be a chance that a given task does not require an AI application. Instead, it requires more straightforward and reliable methods that enterprises have established for a while in the sphere. Article published by Wired shed light on plagiarism in using AI technologies. It became clear that there is a chance of plagiarizing the ML (machine learning) model, especially when ML runs on public API (Application Programming Interface). Thus, there has to be a well-structured prevention strategy for any upcoming hacker attacks [38].

Another significant issue is a talent shortage. Since AI is a relatively new concept, the market is mainly empty of experts specializing in a particular AI tool or method. Besides, the competition level among companies is high, and specialists are hired as soon as they appear. The complexity of the field makes specialists high-paid, and scarce [17]. Thus, there is a need for educational institutions to invest and create new high-qualified specialists. One of the most significant issues is dealing with legal and ethical challenges regarding AI in PM. When dealing with AI technology, project manager has to understand the origin of

this technology and the utilization right that the company has. From the ethical perspective, the company needs to ensure that AI does not create biases and discriminates against any group [3].

2.3 Factors Influencing the Adoption of AI

Enterprises' willingness to adopt AI into their PM can vary according to certain factors. Nik Dawson, in his article, defines six primary variables which define the rate at which a firm adopts AI.

The first variable is competition. According to McKinsey Global Institute, the rate of rivalry within market branches affects AI adoption the most. This statement overlaps with game theory since the readiness of the company to introduce AI correlates to the number of companies that have already adopted it [4]. In other words, when a relatively small number of companies apply AI to their project, they receive competitive superiority compared to the others. When more firms apply AI, the incentive to adopt it decreases since AI provides fewer advantages than competitors. However, the decision to introduce AI solutions is made with imperfect background, as it is practically impossible to know what competitors are using due to the privacy of this information [7].

The second variable is firm characteristics. The scale of firm activities, revenue, and industry type affect the rate of AI adoption. In other words, companies with good financial foundations and income usually introduce AI earlier than smaller and less financially successful enterprises. Besides, firms working with finances tend to apply AI in their activities more often than, for example, agricultural companies due to the sort of activities performed [14].

The third factor is workforce skill capabilities. The competency level at which employees can work and operate with AI technologies directly affects the company's adoption rate. Therefore, the more companies hire a professional workforce with knowledge of AI technologies, the greater the chance companies will introduce AI [7].

The fourth variable is the digital maturity of the company. Several studies show that decision to introduce a new AI tool or software method depends on the previous adoption of digital technologies or the experience with another AI-related entity. For example, broadband infrastructures support and allow more complex digital tech adoption. Another example shows that companies which had experience with web technologies are more likely to adopt AI solutions [27].

The fifth variable is expected return on investments in AI. Firms will invest more in more promising technologies. Thus, firms who are optimistic about AI applications (due to previous experience or recommendation) are more likely to adopt AI. Also, vice versa: the more the company is uncertain about AI solutions, the less chance it will adopt AI. The sixth and last variable is AI complements. If the firm has several tools or technologies of a particular AI type or branch, it will be more likely to invest in another piece from the same branch. An example would be a retailer implementing computer vision to identify inventory previously. This retailer has already adopted automation processes to receive and distribute stock [7].

Moreover, the UTAUT model also defines some behavioral patterns influencing the desire to implement AI into PM: (1) performance expectancy (PE), defined as the rate at which technology gives benefits in performing specific activities; (2) effort expectancy (EE) measures the ease of use of the technology; (3) social influence (SI) is the perception of how others people should apply particular technology; (4) facilitation conditions (FC) shows consumers' perception of which resources are available with the given technology; (5) hedonic motivations (HM), or enjoyment of using the technology; (6) price value (PV) is the value perceived by the user in comparison to the price that is paid for the technology; (7) habit (HT) is measured by how often the technology is used [5].

The literature review above found that the role of AI in project management is becoming more crucial. However, the challenges in implementing AI still exist due to several causes, such as talent shortage and technological limitations. However, research on the willingness to adopt AI in project management is still lacking. Our research addresses it by identifying the potential factors influencing the willingness of project managers to adopt AI and validating the factors through a survey.

3 Hypothesis Development

3.1 Selected Factors Influencing the Use of AI

AI has proved useful in PM and activities requiring specific and highly sophisticated tools and methods. Digital automation, risk management, and error identification are parts of AI and are highly applied in various projects [29]. All that and more simplifies project work letting project managers focus on other primary and complicated tasks [28].

This paper demonstrates that AI improves project management. In order to show that, we conducted a literature review and a survey asking for human opinion. In order to attain analytical data survey uses factors (variables) identifying human willingness to implement AI, which would be factors used in both Dawson's article and the UTAUT (User acceptance of information technology) model.

Firstly, the UTAUT model discusses certain variables influencing the desire to adopt AI into PM. Those factors are performance expectancy, effort expectancy, social influence, facilitation conditions, hedonic motivations, price value, and habit [5].

Secondly, variables that define the rate at which enterprise, firm, or project adopts AI: competition, firm characteristics, workforce skill capabilities, company's maturity, expected ROI, and AI complements.

According to Dawson, companies will adopt technology more likely if the technology is profitable. Hence, more efficient technology will likely be introduced [7]. That correlates with a performance expectancy [5]. Thus, the efficiency of the technology directly affects the desire to implement such technology.

The first variable for the survey is the efficiency of the technology. Besides, the desire to adopt particular technology correlates strongly with the price. Some

enterprises are incapable of covering the cost of particular AI technologies. Vice versa: lower price ensures higher chances of application [5]. Thus, the second important factor for the survey is price.

Moreover, companies introduce technologies only if workforce capabilities are appropriate [7]. That correlates to effort expectancy and complexity of use of specific method or tool [5]. Therefore, the third factor is the complexity of use. The fourth factor is previous experience; companies tend to apply the technology they are familiar with and know its strengths and weaknesses. The fifth factor that is related to the previous experience is feedback. According to Dawson, companies will be more likely to introduce a technology if there is positive feedback from the market regarding the technology [7]. UTAUT identifies one variable with a similar function, i.e., social influence [5].

The final variable is complementary to the technology. Companies will implement AI technology with higher chances if there are complementary technologies [7]. UTAUT defines that as facilitation conditions. Companies are more likely to use a particular method or tool if they have previously introduced technologies of a similar field or scale. Thus, those technologies will complement each other [5].

3.2 Hypothesis Formulation

According to Dawson, one of the primary reasons to adopt technology is its efficiency and ability to manage given tasks. In other words, expected positively affects the rate at which the company introduces AI into project [5]. For project managers, highly effective technology results in higher profits and overall the state of the project [7]. Therefore, the corresponding hypothesis is:

Hypothesis 1. (H1): *Performance effectiveness will positively influence the willingness to adopt AI.*

Another factor influencing the willingness is price. Certain enterprises cannot afford AI technologies due to weak financial backgrounds or can carry out tasks with lower costs without AI. Accordingly, the lower price of AI increases the willingness to implement AI [29]. Thus the hypothesis can be formulated:

Hypothesis 2. (H2): *The higher price will negatively influence the willingness to adopt AI.*

The third variable responsible for the adoption of AI is previous experience. Previous AI applications make companies familiar with the technical aspects of AI, its benefits, and its drawbacks. However, if companies apply technology, first-time challenges and errors will probably arise more often [7]. Thus, the hypothesis is:

Hypothesis 3. (H3): *Previous experience applying particular AI technology will positively impact the willingness to adopt AI.*

Moreover, the willingness to adopt AI can be related not only to one's personal experience. Other users can advertise tools or methods, and their positive feedback increases the AI adoption rate. On the other hand, negative feedback decreases the rate of AI adoption [5]. Therefore, the hypothesis can be formulated as follows:

Hypothesis 4. (*H*4): *Positive external feedback on applying particular AI technology will positively impact the willingness to adopt AI.*

The fifth variable is AI's complexity of use. Both people and enterprises strongly base their implementation choice on how intuitive and difficult it is to use the technology. Besides, some firms do not have the high-level machinery required by the technology [5]. Thus the formed hypothesis is:

Hypothesis 5. (*H*5): *The high complexity of the application will negatively influence the willingness to adopt AI.*

The final variable affecting the introduction of AI is complementary to the chosen methods or tools. The existence of other technologies that ease or add to the application of the technology increases the chances firm will implement this technology [5]. Thus, the hypothesis is:

Hypothesis 6. (*H*6): *The existence of complementary technologies will positively influence the willingness to adopt AI.*

The framework (see Fig. 1) is developed to demonstrate the relationships between previously discussed influence factors and the willingness to implement or use AI in project management. The willingness to adopt AI is placed on the right side of the framework, and the variables influencing it are on the left. All variables are connected to the "willingness" box, demonstrating their relationships. The main goal of the framework is to calculate the significance of each variable to the willingness factor.

All six hypotheses mentioned above are tested through a survey. The results will be analyzed using partial least squares (PLS) path modeling using structural equation modeling (SEM).

4 Survey Design

We designed the survey based on a six-point Likert scale model. The Likert scale gives a range of possible answers by which people indicate how much they agree or disagree with a particular claim or show how much they are to a particular statement. For the statement, the first option to the far left is "Definitely disagree," and the last to the far right is "Definitely agree". All options in between were left blank as "the extremes only scale format performs better than the fully labeled scale format in terms of criterion validity" [36]. We chose the six-point Likert model to eliminate neutral answers since participants usually tend to pick that answer most often to reduce the time of survey completion [22].

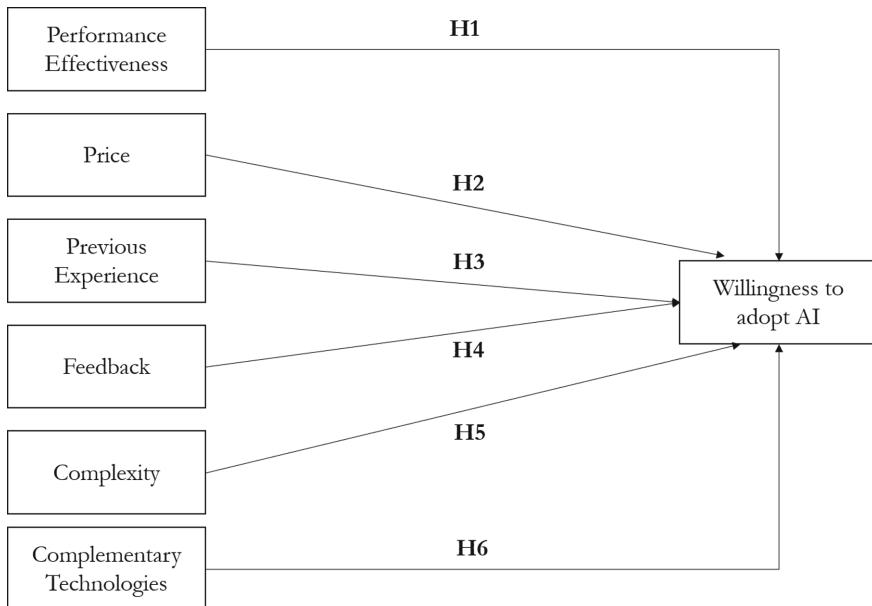


Fig. 1. Hypothesis framework of factors influencing the willingness to adopt AI

First, the survey introduces the research concept and a short definition of AI and PM. Then, it follows with demographic questions such as age, gender, and nationality.

5 Sample Description and Analysis Method

5.1 Sample Description

The total number of participants is 124 people. Figure 2 summarizes the survey participant demographics. 54.5% of participants were females, 42.3% males, 2.4% other, and only one participant (0.8) decided not to identify gender. Regarding age, 54.8% of participants belong to the age group of 18–20 years, 29% belong to the 21–25 age group, 6.5% to 26–30, 5.6% to 31–45, and the rest 4% belong to the age group of 45 and above. Most of the participants are from Europe 67.5%, 17.9% are from Asia, 10.6% are from North America, 3.3% are from Africa, and only one participant (0.8%) is from South America. However, due to software limitations, only 99 out of 124 participants are used for the analysis.

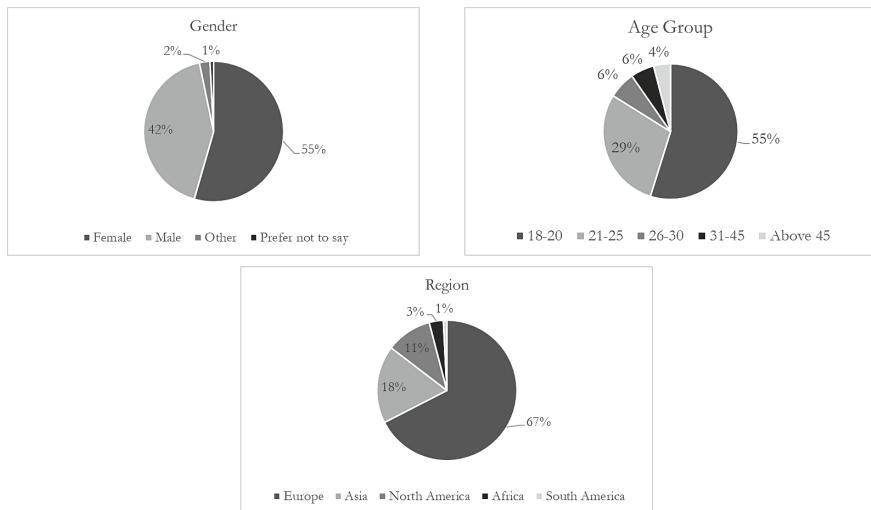


Fig. 2. Respondent demographics.

5.2 PLS-SEM

PLS-SEM or PLS path modeling is a statistical model that “incorporates unobservable variables measured indirectly by indicator variables” [16]. Henseler describes PLS-SEM as a part of the second-generation statistical technique called Structural Equation Modelling (SEM). SEM can be classified into two groups: (1) covariance-based SEM (CB-SEM) and (2) partial least squares SEM (PLS-SEM).

CB-SEM is used for the rejection or confirmation of developed theories. However, PLS-SEM is applied to develop theories [12]. The primary PLS-SEM application is for survey analysis based on developed hypotheses.

The PLS-SEM model comprises two parts: the inner model and the outer model. The outer model shows relationships between the latent variables and indicators [16]. Latent variables are defined as variables derived from the accumulated data [31]. In our research, latent variables are effectiveness, price, feedback, previous experience, complexity, complementary technologies, and willingness to adopt AI. The indicators are defined as data gained through the survey, particularly by receiving responses to statements. According to Henseler, “indicators represent all the relevant dimensions of the latent variables” [16]. Moreover, the inner model demonstrates the connections between latent variables and their relations.

PLS-SEM is capable of analyzing two types of models: reflective model and formative model. In the reflective model, each indicator represents an error-afflicted latent variable measurement. Besides, variables in the reflective model are exchangeable, and each variable can do the same task as others [2]. However,

interchanging one latent variable for another in a formative model could change the meaning of the whole model [16].

Furthermore, SEM is constructed of five parts: measurement, measurement scale, coding, data distribution, and composite variables. Measurement is described as a value of indicator variables, and measurement scale defines closed-ended answers. Coding is a part where all the responses are split into categories such as strongly disagree or strongly agree. That is why we chose the Likert scale for the survey formation. Lastly, composite variables demonstrate the weighting process of different variables developed through the overarching hypothesis [12].

One chose PLS-SEM due to its ability to create and predict theories that do not yet exist [2]. Besides, SEM “allows the simultaneous modeling of relationships among multiple independent and dependent constructs” [11].

6 Analysis Results

The survey is analysed through SmartPLS software (Version 3) in order to build and assess PLS-SEM. Due to the unavailability of unlimited version of the SmartPLS software, only 100 responses could be assigned to the sample. The analysis is split into three parts. Firstly, the reliability of the model will be assessed as measurement model. Secondly, the effect of latent variables on each other will be analysed through structural model analysis. Finally, the significance of hypothesis will be identified through bootstrapping method.

6.1 Measurement Model

In order to analyze the model, two methods can be applied: confirmation factor analysis and exploratory factor analysis. Exploratory factor analysis is applied to understand the relationships between latent variables, and confirmation factor analysis tests the hypotheses by rejecting or accepting the factor structures of the variables [10]. Since this paper has to confirm or reject the formulated hypotheses, the confirmation factor analysis is applied. The following sections discuss indicator reliability, construct reliability, and model fit. In order to assess indicator reliability, the paper revises outer loadings. Only outer loadings are described since the model is reflective.

In Fig. 3, the outer loadings are shown as arrows between latent variables (oval shapes) and the indicators (square shapes). The outer loadings represent the reliability of the indicator to its latent variable [10]. A value of 0.7 and above shows that the indicator is well fit to its latent variable. Such a value means that half of the variances related to the indicator are greater than the error variances. However, it is only recommended to eliminate indicators with outer loadings between 0.4 and 0.7 when elimination increases the reliability and validity of the model [12].

In the model, one of the latent constructs is not reliable. However, it was not removed since it worsened reliability and validity after the elimination.

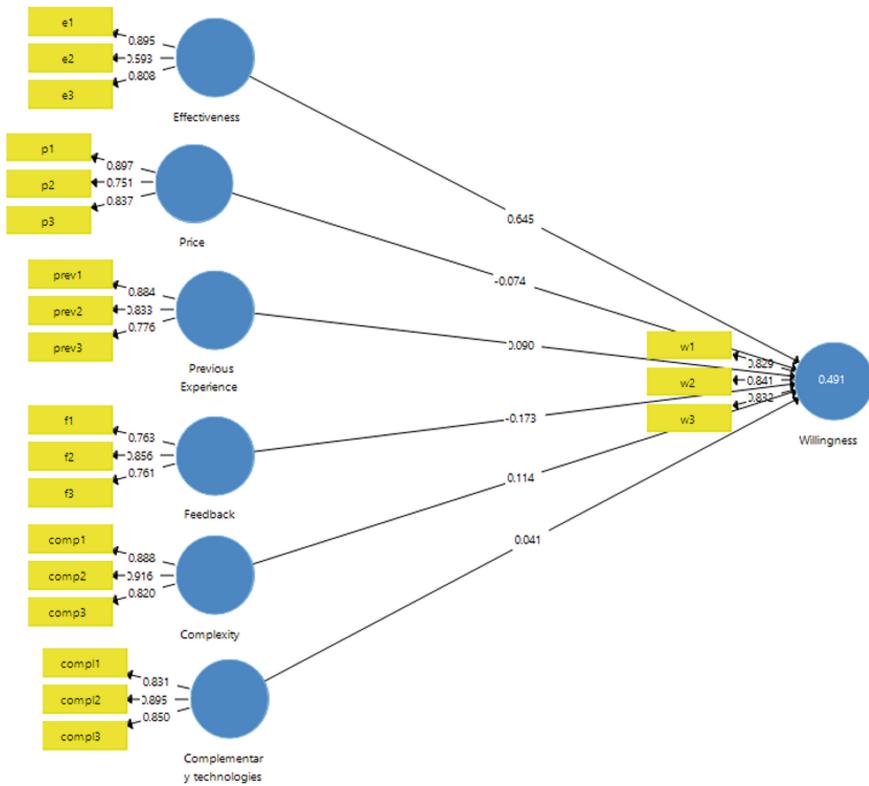


Fig. 3. Path coefficients, R-Squared values, and outer loadings for the willingness to implement AI based on certain factors before removing non-fitting indicators.

Figure 4 shows the changes after removing non-fitting indicators. Most outer loadings have decreased or have not changed significantly after the elimination. However, none of the loadings decreased significantly below the threshold, and all the loadings still fit the model. Thus, we can conclude that the neighboring indicator has a minor positive or minor negative influence on other indicators, and the original model does not require the elimination of any indicators.

Composite reliability (CR) is another essential factor. CR identifies “reliability and consistency of the indicators of corresponding latent construct” [10]. The latent construct is consistent with latent variables and indicators and ranges from 0 to 1. Higher CR values indicate higher reliability. The required value for the reliable model is 0.6, 0.7 is sufficient, and 0.8 is considered good [10]. For the given model, CR values of 0.7 and larger are taken into account.

Another essential factor is the average variance extracted (AVE). AVE is defined as the validity of a latent construct and defines the average percentage of variation of the indicators [1]. If AVE is bigger than 0.5, indicators support half of the construct variance and prove the model's validity.

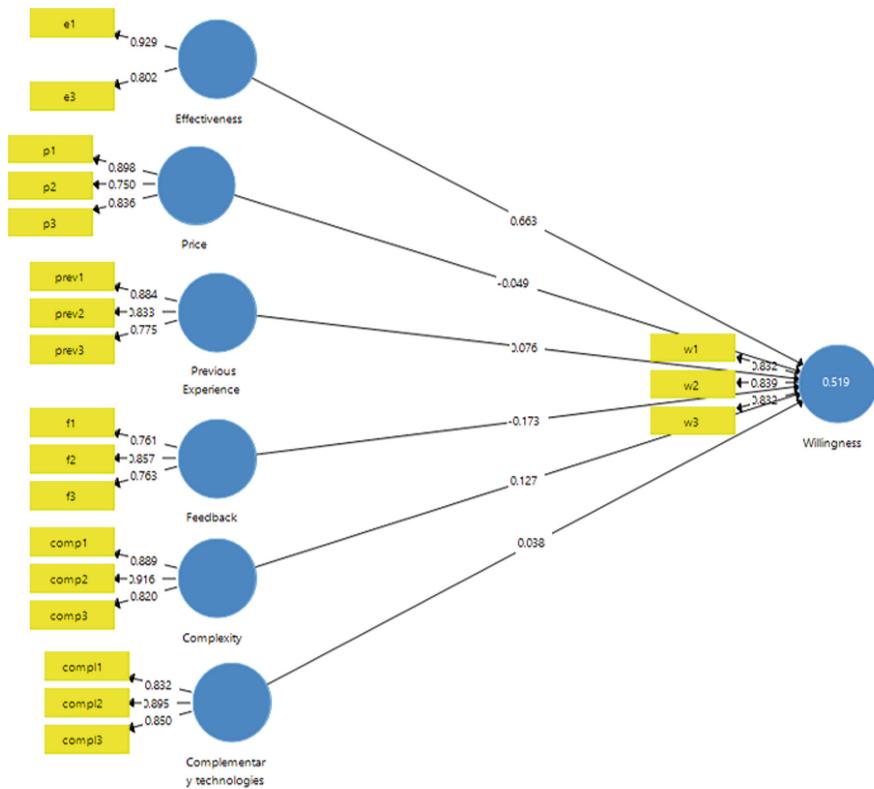


Fig. 4. Path coefficients, R-Squared values, and outer loadings for the willingness to implement AI based on certain factors after removing non-fitting indicators.

As well as for indicator reliability, the elimination of not fitting indicator did not improve the CR and AVE values but also decreased them (see Tables 1 and 2). Standardized root mean square residual measures the average difference between the observed and estimated model. A value of 0.08 and less is defined as an appropriate fit [10].

The difference between the saturated model and the estimated model for the given case equals zero. Thus, the model has a good fit. Both numbers for the saturated and unsaturated models for SRMR equal 0.089.

6.2 Structural Model, Inner Model

Path coefficients of the structural model define the effect from endogenous and exogenous latent variables [10]. The effect varies from 1 to +1; values closer to the absolute of 1 are described as strong paths and closer to 0 as weak paths [10]. Values of the path coefficients are shown in Fig. 4.

F-square defines the value of the magnitude of unexplained variance after removal of exogenous latent variable [10]. An F-square value of 0.02 represents

Table 1. Measurement model determinants before indicator removal.

	CR	AVE
Effectiveness	0.815	0.602
Price	0.869	0.690
Previous experience	0.871	0.692
Feedback	0.837	0.631
Complexity	0.908	0.767
Complementary technologies	0.894	0.739
Willingness	0.873	0.696

Table 2. Measurement model determinants after indicator removal.

	CR	AVE
Effectiveness	0.859	0.753
Price	0.869	0.689
Previous experience	0.871	0.692
Feedback	0.837	0.632
Complexity	0.908	0.767
Complementary technologies	0.894	0.739
Willingness	0.873	0.696

a weak effect, 0.15 medium effect, and 0.35 high effect. Notably, the positive or negative effect of the variable is not considered only the absolute value.

Based on the F-square values of the given model (see Table 3), the high effect is assigned to effectiveness to willingness. It has a positive sign and is the highest value compared to the rest. The medium effect is assigned to feedback to the willingness variable, and other variables have either a minor or a negative effect.

R-square shows the effect on exogenous latent variables from the related endogenous latent variables. In other words, R-square determines the percentage of variance in exogenous latent variables [10]. A weak effect is characterized by 0.19, medium by 0.33, and strong 0.69. The value of R-square for the current model is demonstrated in Fig. 4. The model has a moderate effect of 49% of variance in the willingness variable. Nevertheless, the variance value in the adjusted model (Fig. 4) is slightly higher 52%.

Bootstrapping is a resampling method to calculate the significance of the PLS coefficient to measure the structural model [10]. The probability value (p-value) of 5% (p-value = 0.05) and t statistic value above 1.96 define indicators and variables as significant. In order to effectively apply to bootstrap, 5000 subsamples are used (at least 500 is a requirement) [10].

The measurement model demonstrates that all statements are significant to the related variable varying from 0.0 to 0.015. The p-values are demonstrated in Table 4.

Table 3. Structural model determinants.

	F-Square	R-Square
Effectiveness	0.608	0.645
Price	0.009	-0.074
Previous experience	0.010	0.090
Feedback	0.033	-0.173
Complexity	0.015	0.114
Complementary technologies	0.002	0.041

According to Table 4, effectiveness is the only variable significant towards willingness. Therefore, the hypothesis regarding this factor is supported. That means people tend to implement AI technologies if the technology is effective.

On the other hand, the hypothesis regarding positive feedback has a t-statistic value of 1.535 and a p-value of 0.125. Possibly, the feedback hypothesis would be supported if the survey contained more responses and the PLS software did not limit the data for its free version.

Table 4. Hypotheses evaluation.

	Hypothesis	Original sample	Sample mean	Standard dev.	T statistics	p-value	Conclusion
H1	Effectiveness → Willingness	0.645	0.640	0.070	9.191	0.000	Supported
H2	Price → Willingness	-0.074	-0.030	0.115	0.647	0.518	Rejected
H3	Previous experience → Willingness	0.090	0.059	0.093	0.967	0.334	Rejected
H4	Feedback → Willingness	-0.173	-0.098	0.113	1.535	0.125	Rejected
H5	Complexity → Willingness	0.114	0.106	0.102	1.116	0.264	Rejected
H6	Complementary technologies → Willingness	0.041	0.025	0.086	0.480	0.631	Rejected

7 Discussion

From the attained data and conducted analysis, the hypothesis regarding the positive effect of efficiency on the willingness to implement AI technologies is confirmed, and other hypotheses are rejected.

In particular, bootstrapping showed no significant relationship between the complementary technologies and willingness variables. The F-square value is insufficient, as well as the path coefficient. Therefore, the availability of complementary technologies has no direct emphasis on adopting AI, and the hypothesis is rejected. Nevertheless, it is clear that the built model is successful, and all statements are significant for the latent variables. It might mean that participants are familiar with complementary technologies but are indifferent toward them while applying AI technologies.

A similar situation happens with the previous experience. Bootstrapping has not identified any significant relationship between the previous use of technology and the willingness to adopt AI. The path coefficient and F-square are not significant. As well as for complementary technologies, previous experience fits well into the model, and sub-statements are significant. That might mean that people might apply the technology they have previously used.

Moreover, the lack of expertise in AI technologies explains the insufficiency of the experience variable. Participants could have had no previous experience with any AI entity and thus could not appropriately respond to the given statements [19]. Thus, the hypothesis stating that the willingness to implement AI is positively affected by previous experience is rejected.

The hypothesis regarding external feedback is associated with the result that seems unclear. It has a moderate path coefficient. Even though its F-square value and p-value are insufficient for supporting the data (see Table 4), they are closer to the fitting values than others. Therefore, the hypothesis that external feedback positively affects AI adoption is rejected. However, more data may support the hypothesis and prove it right.

Lastly, two hypotheses regarding the negative effect of complexity and price on the implementation of AI are rejected. Bootstrapping technique showed that p-values and F-square of both complexity and price are insufficient to support the hypotheses. For the complexity, the path coefficient has an insignificant positive value. That can be explained by the fact that the complexity of technology could make participants view particular technology as highly sophisticated and, therefore, highly effective [5]. However, the value is still insignificant, and the hypothesis is rejected. The price variable has an insignificant negative path coefficient; thus, the hypothesis is rejected. Nevertheless, the p-value of each sub-statement regarding those two variables are significant, and those two variable fit well into the model. That could mean that participants might be influenced by the price and complexity variables. Nevertheless, the influence would be insignificant due to unfamiliarity with the AI [19].

After evaluating each sub-hypothesis, the overarching hypothesis that certain factors have either positive or negative effects on the desire to implement AI technology is partially supported. Since the effectiveness hypothesis was proven correct, the hypothesis regarding external feedback can be potentially proven with more data. In conclusion, complexity, price, previous experience, and complementary technologies have no significant positive or negative effect on the willingness to adopt AI technologies in project management.

Our study can contribute to companies, project managers, and project management communities who plan to adopt AI in managing their projects. The performance effectiveness of AI technologies is essential to motivate companies and project managers to adopt them. As a consequence, the explicit description of the AI technology features and information sharing on success stories in adopting AI in project management would increase the willingness of companies and project managers to adopt AI. Project management communities and platforms, such as Project Management Institute (PMI) and Association for Project

Management (APM), must make information on AI features and success stories available on their portals.

Our research has some limitations, especially the limited number of respondents. Furthermore, the origin of the respondents is imbalanced with the significant number of European respondents. Most of the respondents were younger project managers in their early twenties. Our research can be improved by increasing the size and diversity of the sample. Future research will focus on project management in some specific sectors, such as manufacturing, construction, or healthcare since there could be additional factors playing essential roles in adopting AI.

8 Conclusions

This paper aims to study the adoption of AI in project management, constraints, and factors that influence the adoption of AI. Therefore, a literature study was conducted to identify all the previously mentioned aspects. Then, we developed a hypothesis framework consisting of six hypotheses. Each hypothesis was tested through an online survey. We analyzed the survey data by applying PLS-SEM with the SmartPLS software.

Generally, our research has achieved the goal of identifying and validating the factors influencing the willingness to adopt AI in project management. However, according to the survey results, most described factors have no significant impact on adopting AI technologies. Therefore, the broader introduction of AI into different aspects of human lives and project management can increase the knowledge regarding AI and influence their desire to implement it.

According to the attained results, effectiveness and potential feedback strongly impact the implementation of AI. Therefore, companies can emphasize the quality assessment of technology more and work on broader promotion via various media. Moreover, other factors such as price, complexity, complementary technologies, and experience also have to be assessed since they still have an insignificant influence on the willingness to adopt AI. Besides, companies can start applying more AI technologies to see what factors improve the application experience and influence the adoption.

References

1. Ahmad, S., Nur Ain Zulkurnain, N., Khairushalimi, F.I.: Assessing the validity and reliability of a measurement model in structural equation modeling (sem). *Br. J. Math. Comput. Sci.* **15**(3), 1–8 (2016)
2. Avkiran, N.K., et al.: Measuring the systemic risk of regional banks in Japan with pls-sem. *Theor. Econ. Lett.* **8**(11), 2024 (2018)
3. Belharet, A.: Report on the impact of artificial intelligence on project management
4. Bughin, J., Seong, J., Manyika, J., Chui, M., Joshi, R.: Notes from the ai frontier: Modeling the impact of ai on the world economy. *McKinsey Glob. Inst.* **4** (2018)

5. Cabrera-Sánchez, J.-P., Ramos-de Luna, I., Carvajal-Trujillo, E., Villarejo-Ramos, A.F.: Online recommendation systems: Factors influencing use in e-commerce. *Sustainability* **12**(21), 8888 (2020)
6. Reza Davahli, M.: The last state of artificial intelligence in project management (2020). [arXiv:2012.12262](https://arxiv.org/abs/2012.12262)
7. Dawson, N.: 6 major factors affecting ai adoption and diffusion in firms (2019)
8. Elrajoubi, S.: Artificial intelligence in project management. In: Seminar Paper Interactive Online Ph.D. Course: Project Management. LIGS University USA, pp. 9–12
9. Fama, E.F.: Agency problems and the theory of the firm. *J. Polit. Econ.* **88**(2), 288–307 (1980)
10. David Garson, G.: Path analysis. Statistical Associates Publishing Asheboro, NC, (2013)
11. Haenlein, M., Kaplan, A.M.: A beginner's guide to partial least squares analysis. *Underst. Stat.* **3**(4), 283–297 (2004)
12. Hair, J.F., Ringle, C.M., Sarstedt, M.: Partial least squares structural equation modeling: Rigorous applications, better results and higher acceptance. *Long Range Plan.* **46**(1–2), 1–12 (2013)
13. Halina, M.: Insightful artificial intelligence. *Mind Lang.* **36**(2), 315–329 (2021)
14. Hall, B.H., Khan, B.: Adoption of New Technology (2003)
15. Hancock, P.A.: Some pitfalls in the promises of automated and autonomous vehicles. *Ergonomics* **62**(4), 479–495 (2019)
16. Henseler, J., Hubona, G., Ray, P.A.: Partial least squares path modeling: Updated guidelines. In: Latan, H., Noonan, R. (eds.) *Partial Least Squares Path Modeling: Basic Concepts, Methodological Issues and Applications* (2017)
17. Project Management Institute. Project management job growth and talent gap 2017–2027. Project Management Institute Newtown Square, PA (2017)
18. Joshi, N.: How AI Can Transform the Transportation Industry (2019)
19. Kashyap, R., Kumar, A.V.: Challenges and Applications for Implementing Machine Learning in Computer Vision (2019)
20. Kutschchenreiter-Praszkiewicz, I.: Application of artificial intelligence in project management under risk condition. *Appl. Comput. Sci.* **5**(1) (2009)
21. Lahmann, M.: Ai will transform project management. Are you ready? (2018)
22. Matell, M.S., Jacoby, J.: Is there an optimal number of alternatives for likert-scale items? effects of testing time and scale properties. *J. Appl. Psychol.* **56**(6), 506 (1972)
23. Matt, D.T., Modrák, V., Zsifkovits, H.: Industry 4.0 for Smes: Challenges, Opportunities and Requirements (2020)
24. Middleton, S.: 3 ways ai will change project management for the better (2017)
25. Mullainathan, S., Thaler, R.H.: Behavioral Economics (2000)
26. Noelle, C.: Why Ai is the Future of Finance (2019)
27. OECD. Recommendation of the council on artificial intelligence (2019)
28. Chust, A.P., Heras, S., Gil Pérez, A.: Intelligent project management tools. In: Proceedings from the 24th International Congress on Project Management and Engineering = Comunicaciones presentadas al XIV Congreso Internacional de Dirección e Ingeniería de Proyectos, celebrado del 7 al 9 de julio de 2020, pp. 1860–1870. Asociación Española de Dirección e Ingeniería de Proyectos (AIEPRO) (2020)
29. Prieto, B.: Impacts of artificial intelligence on management of large complex projects. *PM World J.* **8**(5), 1–20 (2019)

30. Reikhanova, I.V., Ilyina, L.A., Kuklev, S.E., Zakharov, A.A.: Ai as a subject of staff: Management and team-building. In: 13th International Scientific and Practical Conference-Artificial Intelligence Anthropogenic nature versus Social Origin, pp. 345–351. Springer, Berlin (2020)
31. Sarstedt, M., Cheah, J.-H.: Partial Least Squares Structural Equation Modeling Using Smartpls: A Software Review (2019)
32. Sion, G.: Smart educational ecosystems: Cognitive engagement and machine intelligence algorithms in technology-supported learning environments. *Anal. Metaphys.* **17**, 140–146 (2018)
33. Taddeo, M., Floridi, L.: How ai can be a force for good. *Science* **361**(6404), 751–752 (2018)
34. Todorov, G.: 65 artificial intelligence statistics for 2021 and beyond (2021)
35. Wauters, M., Vanhoucke, M.: A comparative study of artificial intelligence methods for project duration forecasting. *Expert Syst. Appl.* **46**, 249–261 (2016)
36. Weijters, B., Cabooter, E., Schillewaert, N.: The effect of rating scale format on response styles: The number of response categories and response category labels. *Int. J. Res. Mark.* **27**(3), 236–247 (2010)
37. Wu, D.D., Chen, S.-H., Olson, D.L.: Business intelligence in risk management: Some recent progresses. *Inf. Sci.* **256**, 1–7 (2014)
38. Zujus, A.: Ai project development—how project managers should prepare (2018)



Comparative Lightweight Scheme for Individual Identification Through Hand-Vein Patterns

Mateo Mejia-Herrera^{1,2}, Juan S. Botero-Valencia²,
and Ruber Hernández-García³(✉)

¹ Grupo de Sistemas de Control y Robótica, Facultad de Ingenierías, Instituto Tecnológico Metropolitano, Medellin, Colombia

² Grupo de Investigación en Sistemas Informáticos, Institución Universitaria de Envigado, Envigado, Colombia

³ Research Center for Advanced Studies of Maule (CIEAM), Universidad Católica del Maule, Talca, Chile
rfernandez@ucm.cl

Abstract. Biometric characterization systems are generally used in safety-related applications because they allow the identification or verification of individuals based on human body traits. In recent years hand veins have become an attractive biometric trait due to their advantages compared with other classical biometric traits (i.e., fingerprints, iris, face). However, due to the number of possible architectures for feature extraction and individual identification, different combinations between such methods should be evaluated to give a baseline for further vein biometrics development. This work presents a comparative analysis for individual identification based on hand-vein biometrics, which combines four feature extraction techniques and three classic machine learning techniques using two main types of images. The results show the reliability of some combinations for hand-vein biometric identification achieving accuracy levels above 98% and an Equal Error Rate under 3.2%.

Keywords: Hand-Veins · Vein feature extraction · Biometric systems · Machine learning

1 Introduction

Biometric characterization systems are generally used in safety-related applications because they allow individual identification or verification based on human body traits. The verification process aims to confirm that an individual is one he/she claims to be. In this case, the system response may be positive or negative. On the other hand, the identification process recognizes an individual within a given dataset [22]. The most common biometric systems extract characteristics from the face [43, 60], fingerprints [16], iris [15, 40], keyboard typing speed [22, 49], among others [17, 21, 32]. Although these techniques have been

extensively used, they present accuracy problems, and because of their nature, it is possible to break their safety.

Hand vein-based biometric systems, both finger and palm veins, have gained attention recently [7, 8, 11, 27, 41, 57]. The hand veins have intricate patterns, reaching up to five million reference points. Vein patterns are unique even in identical twins, proving to be the most critical case for biometric systems because identical twins share the same DNA (Deoxyribonucleic Acid) and comparable cognitive-physical development [32]. In addition, veins are insensitive to environmental changes and become undetectable only in the absence of blood flow [27]. Since these patterns are not visible to the human eye, infrared light-sensitive systems are used for image acquisition, having certain advantages over other biometrics, such as: (1) the biometric patterns remain hidden; (2) biometric traits are unalterable and unchanging over time; (3) it is almost impossible to replicate or copy an identification, leaving no traces to forge the system; and (4) it disappears with the individual's death.

Identifying individuals in biometric systems requires a previous feature extraction stage [8, 27, 48, 53, 55], which can be grouped into four categories: geometry-based techniques, minutiae point techniques, texture-based techniques, and subspace projection techniques. This stage involves computer vision techniques to obtain relevant information from an image. Such information is later used by machine learning algorithms (ML) to determine the identity of a sample. After the feature extraction process, biometric systems implement an identification stage that receives the characteristics of the images to create a model that distinguish each presented sample. However, the feature extraction process does not always lead to the acquisition of useful information for individual identification; therefore, the performance of different techniques for hand-vein patterns should be analyzed to establish a comparative baseline for future studies.

This work presents a comparative analysis of different combinations of representative feature extraction techniques and machine learning (ML) models based on the results for hand-vein identification to evaluate each feature extraction category. Our proposed methodology combines four feature extraction techniques from state-of-the-art and three classical machine learning techniques for hand-vein identification: K-Nearest Neighbors (KNN), Logistic Regression, and Convolutional Neural Networks (CNN). The evaluated techniques were implemented using the ML libraries for Python: ScikitLearn, Keras, and OpenCV. Experimental results show that different combinations of feature extraction and identification processes achieve an accuracy of up to 100% and an Equal Error Rate (EER) of 0.59%. Due to the number of possible combinations of feature extraction and identification techniques, the collected information will be a baseline for further development of biometric systems and other related studies. Thus, the main contributions of our proposal are:

- provides the detailed information required to develop and deploy a computationally lightweight scheme for individual identification using hand-vein patterns, considering the low computational cost and robust representation techniques and recognition algorithms.

- shows the performance of combinations of representative feature extraction methods and machine learning models that allow their implementation in low-cost systems, achieving promising results in terms of accuracy and EER.
- facilitates the selection of an identification framework using hand-vein images based on different combinations of feature extraction techniques and ML models.

The remainder of the paper is presented as follows. Section 2 summarizes the state-of-the-art on feature extraction methods for hand-vein images. Section 3 introduces the proposed methodology, describing the feature extraction techniques and identification models to be compared. Later, Sect. 4 discusses the obtained results for each evaluated method. Finally, Sect. 5 presents the conclusions and future works.

2 Summary of State-of-the-Art

Currently, the feature extraction methods developed for vein pattern images are grouped according to the literature [46, 48, 55] into four categories:

1. **Geometry-Based Techniques:** use geometry information such as vascular pattern shape, direction, length, or location. Generally, the applied techniques involve precise segmentation of the veins, which is achieved with high-resolution images and a controlled acquisition environment. Some of the techniques that correspond to this category include Line Edge Mapping [30], Repeated Line Tracking [9, 31], Wide Line Detector [20], among others [29, 59].
2. **Minutiae Points Techniques:** extract relevant points from the image of the vascular pattern, such as the position of terminations, branches, or forks. These points establish the similarity between two images, regardless of their orientation or location. Although this category is presented as the most promising for non-contact acquisition scenarios because of its invariant nature to scale and rotation, a low number of interest points are often obtained because of the low contrast of NIR images. In addition, lighting changes during image acquisition cause different interest points for the same individual, which may affect the identification process. The most commonly used techniques in this category are Scale Invariant Feature Transform (SIFT) [37, 50], Speeded-Up Robust Features (SURF) [36], and Oriented Fast Rotated Brief (ORB) [43].
3. **Texture-Based Techniques:** use statistical methods to extract characteristics from the image textures. One of the most notable properties of these techniques is their capacity to acquire and combine vascular information with other textures present in images [6]. Generally, its representation is given by histograms of gray-scale images; this feature reduces the size of the data to a predefined dimension vector by decreasing the storage space and processing time. However, since the information is represented as histograms, there are losses of positional information that can sometimes lead to identification errors. Some of the techniques available in the literature include Histogram of

Oriented Gradient (HOG) [5], Discrete Wavelet Transform [58], Local Binary Patterns (LBP) [50], and their modifications [1, 28].

4. **Subspace Projection Techniques:** transform image information into a feature-separable subspace. This projection is made into a space of r dimensions, and the result is a representation with fewer characteristics that yield lower computational costs in the identification process. However, the feature extraction process can require a high computational cost, and sometimes with new data appearing, a new projection of the images may be needed [48]. Available methods include Principal Component Analysis (PCA) [28], Linear Decomposition Analysis (LDA) [54], Independent Component Analysis (ICA) [55], Neural Networks [25], among others.

Considering the above classification, our proposed method evaluates different representative feature extraction techniques based on the performance results of widely used ML models for individual identification.

3 Proposed Methodology

A general workflow of a hand-vein identification system is depicted in Fig. 1. Typically, it comprises four main stages: image acquisition (S0), pre-processing (S1), feature extraction (S2), and recognition (S3). The proposed methodology helps to determine which technique combination is best for developing a computationally lightweight scheme for biometric identification using hand-vein patterns, as shown in Fig. 2.

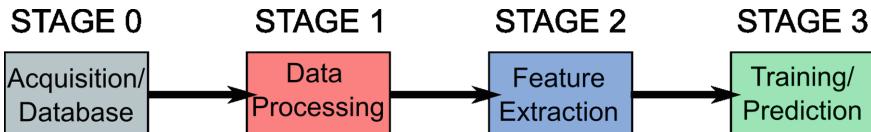


Fig. 1. General workflow of the hand-vein identification system.

Initially, the hand image is captured using infrared devices that reveal the vein patterns. Later, images are processed by the Contrast Limited Adaptive Histogram Equalization (CLAHE) [38] for contrast enhancement to help feature extraction. Considering the given information, we selected a representative technique for the feature extraction process from each approach. For this purpose, the selected techniques are the Freeman Chain Code (FCC), PCA, LBP, and SIFT. In the case of FCC, which describes the geometric information of vascular patterns, it is implemented as an invariant representation of the rotation and position [4, 39, 60]. With the exception of FCC, the other algorithms were selected based on their EER results reported in the literature review [55], with values less than 4%. Finally, the recognition process is carried out by training a classifier and evaluating the performance for each feature extraction and classification technique combination on both RAW and CLAHE images.

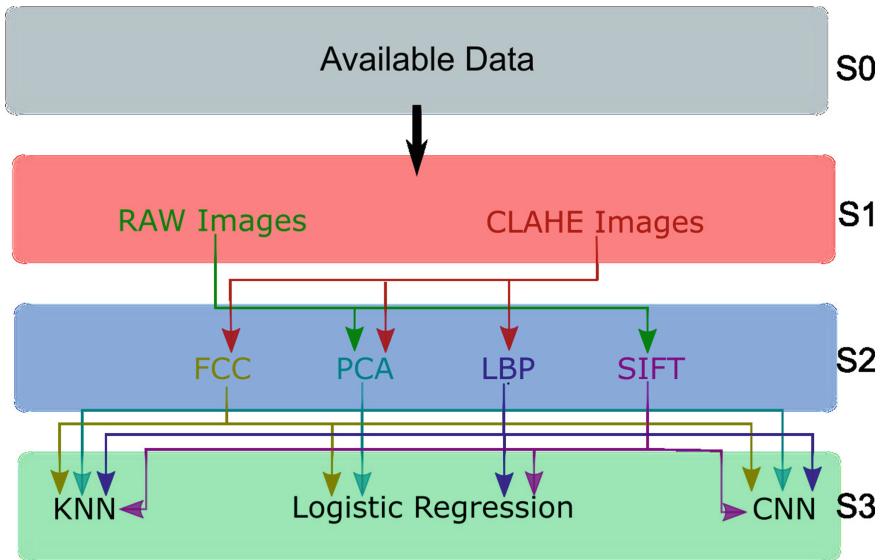


Fig. 2. Evaluation scheme for each combination of the feature extraction approaches using different machine learning algorithms.

Our proposed methodology focuses on the feature extraction and recognition stages because they are decisive processes in the performance of a biometric system [55]. Following, we present the feature extraction techniques and ML-based models used in the evaluation scheme, explaining their respective setup and implementation.

3.1 Feature Extraction

The feature extraction stage obtains a numerical representation of the hand-vein patterns as a descriptor, which will later be the input of the recognition process. Accordingly, we evaluated the following techniques.

- **FCC** is a geometry-based method proposed in [10] and is used to represent a 2D shape using an easy-to-transmit code and eight-digit numeric encoding from zero to seven. Since the FCC requires the position of the points in the geometry of interest, it is necessary to segment the vein pattern geometry and obtain the coordinates of the lines that describe them. In this case, the *findContours* algorithm provided by OpenCV was used to extract geometric shapes. This algorithm finds closed contours within binarized images and produces a list of vectors corresponding to the coordinates of each point within each boundary. However, owing to image quality, it is possible to find small contours that do not represent the geometry of the vascular regions. Therefore, a contour filtering algorithm was developed to discriminate regions with an area variability of lesser than 40 pixels (see Fig. 3).

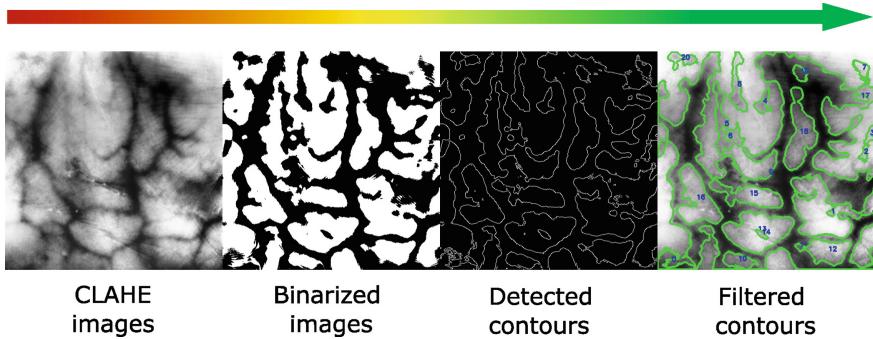


Fig. 3. Method for FCC extraction, starting with a binarization of CLAHE images, followed by contour detection and filtering, and returning contours for FCC representation.

After region segmentation, in order to obtain the chains that describe the geometry of each detected contour, the FCC creates a code that takes an initial pixel and assigns a number based on the location of the next pixel, repeating the assignation until the contour is finished, as shown in Fig. 4. Finally, the obtained strings are sorted using the most extensive series of zeros as the FCC starting flag. Thus, we obtain the same code regardless of the geometric rotation or whether the starting point for generating the string changes during processing (see Fig. 4).

- **SIFT** descriptor [12] is used to identify a set of reference points within the images, which are invariant to rotation and scale [52]. The SIFT method performs a pyramidal decomposition by applying different Gaussian filter sizes. Thus, when the points with the highest Gaussian difference exceed a specific value, their coordinates are stored as points of interest. This algorithm has been used to extract the positions of interest points from vascular images for vein-based biometrics [27, 36, 37, 42, 50]. For this purpose, we used OpenCV the `xfeatures2d.SIFT_create()` function in its default configuration.

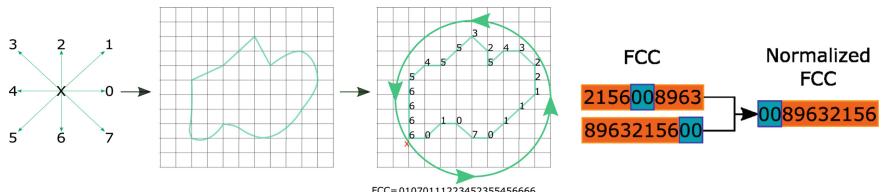


Fig. 4. H. Freeman Chain Coding Process for FCC Representation and Normalization. For the Example Contour, The Codification Numbers are Used for Pixel Location, Acquisition Direction, and Generated FCC Representation. Given Two Codes Corresponding to the Same Contour, the Largest Zeros Chain is the Starting Flag for Normalization.

- **LBP** is a technique for feature extraction in the spatial domain that was proposed in [34] and later improved in [35]. LBP aims to encode the texture information of an image based on the intensity differences between a central pixel and its n neighboring pixels in a circular region of size r by using a sliding window. Although more recent variants have been proposed for LBP, this algorithm is still widely used in biometric systems for feature extraction [1, 28, 50]. For its implementation, we used the function provided by the ScikitLearn Python library (i.e., local_binary_pattern) in its rotation invariant configuration, with a radius $r = 10$ and $n = 8$ neighboring points. Figure 5 depicts different grayscale images with the vascular pattern and other textures present in the processed CLAHE images. A parameter selection was developed using visual inspection followed by a Blind/Referenceless Image Spatial Quality Evaluator for image quality assessment.
- **PCA** is a feature space reduction technique [19] that is widely used to reduce the size and complexity of large-scale arrays and data. In this work, we used the version of the PCA provided in the ScikitLearn Python library, which receives the number of principal components to use and an array of size $s \times t^2$, where s corresponds to the set of available samples and each row in the array is the feature vector of $t \times t$. Initially, the training process for PCA was carried out with 120 components to obtain acceptable results. However, to reduce the computational cost of the model, tests were carried out in each quartile (25, 50, 75, 100), evaluating their effectiveness in feature extraction from RAW and CLAHE images.

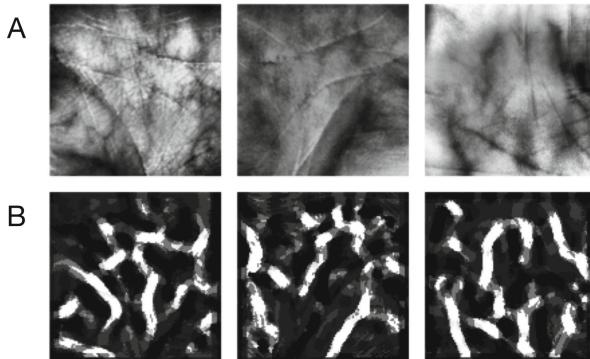


Fig. 5. Texture information extracted from the images by using the lbp with rotation invariant configuration. The first row (A) corresponds to CLAHE images, whereas the second one. (B) belongs to the LBP texture information.

It is noteworthy that SIFT applied to high-contrast CLAHE images detects a high density of interest points, thereby reducing the performance of the identification process. In contrast, the FCC and LBP descriptors cannot capture a

representative number of interest points in the RAW images because of the lack of contrast.

3.2 Identification Process

In this study, we used the Tongji Contactless Palmvein Dataset v.1 [62], which is the largest public dataset from the state-of-the-art. The collected images were captured in two separate sessions with 10 samples per hand from 300 persons, corresponding to 600 hands and 12,000 samples (i.e., $600 \times 2 \times 10$). Since there is no correlation between the vein patterns of both hands of the same person, the two hands are usually considered as different classes for experimental purposes. Later, we divided the database into two subsets, referred to as “Data_real” and “Impostors”, representing 50% of the information (i.e., 6000 images belonging to 300 hands for each partition). The “Data_real” subset was used to train the proposed ML algorithms. Meanwhile, the “Impostors” subset was used to assess the performance of the model security. Each model was trained using a 70-30 cross-validation scheme on the training and validation data for 100 iterations to avoid overfitting. Besides, a data augmentation procedure was performed, which involved creating new samples from some affine transformations of the original image. The proposed procedure randomly includes shifting, rotation, blurring, and scaling of the original samples, as these are usually produced on contactless systems. Its implementation was made with the Keras `ImageDataGeneration` function using the following range of parameters for zoom, shear, shift, and rotation with values of: 0.05, 0.02, 0.2, and 5, respectively. Consequently, a set of 19,215 training images and 8,226 validation images were obtained. Finally, the accuracy and EER metrics were evaluated in a comparative framework for each combination, which served as a baseline for the further development of hand-vein biometric systems.

In combination with the selected feature extraction techniques, we evaluated their performance using the following recognition models.

- **KNN** is an algorithm used to identify individuals based on a voting principle where the class of a sample is established based on the class of the K nearest samples or neighbors [33,63]. KNN, initially proposed by [14], stands out for its ability to classify multiple classes and has been used in different biometric systems [3,24,58]. We used the KNN implementation (`KNeighborsClassifier` function) provided by the SciKitLearn library using a three-neighbor voting density configuration ($K = 3$) owing to the high density of the sampling points. The similarity metric was the standard Euclidean metric. Other configurations with larger K -values were tested; nevertheless, given their low accuracy results, no results were reported.
- **Logistic Regression** is a type of statistical analysis that estimates the probability that a given sample belongs to a class. However, the proposed approach uses a multiclass approximation to compare the class occurrence probabilities for each sample. This method has been used in various biometrics studies [2,13,41,45], showing a high ability to solve classification problems. In

the current work, we used the `LogisticRegression` function (with default parameters) provided by the SciKitLearn library. The learning process configuration was for identifying 302 individuals, with a maximum of 100 iterations.

- **CNN** is a variation of multi-layer perceptron neural networks. In convolutional networks, each neuron processes image information using different filters and convolutional operations. Some CNN-based models, such as AlexNet [26] and VGG16 [47], stand out for their accuracy in vein-based biometric systems [2, 44, 51, 56]. We selected them because of their low computational complexity. Moreover, evaluating shallow architectures before attempting deeper models is commonly recommended. Based on this recommendation, recent works [18, 23, 44] have demonstrated that it is unnecessary to implement deeper models to achieve higher results. Since hand-vein samples are usually small in size, more recent and deeper CNN models are unable to learn richer representations. Therefore, for the objective of our study, both are appropriate in this sense since obtaining good initial results is not required to use more current or complex architectures.

The AlexNet architecture has five convolutional layers connected to MaxPooling layers, each with 3, 64, 256, 384, 384, and 256 filters. In contrast, VGG16 comprises five sets of distributed convolutional layers, as shown in Fig. 6. In this case, transfer learning was applied to the model. This technique selects a set of T trained layers from a pre-trained model and adds N new layers without training. The N layers were trained with their data, whereas the T layers were preserved unchanged. In this study, we modified the last layer of both models to classify 302 classes.

The two architectures were implemented using the Keras library. Besides, we performed a fine-tuning process employing a transfer learning approach. Thus, the CNN models were initially trained on the ImageNet database to recognize 1000 classes, and their weights were optimized to fit the requirements of the hand-vein dataset. Hence, the output layer was modified to identify 302 individuals instead of 1000 classes, as shown in Fig. 6, whereas the remaining pre-trained layers kept their original weights.

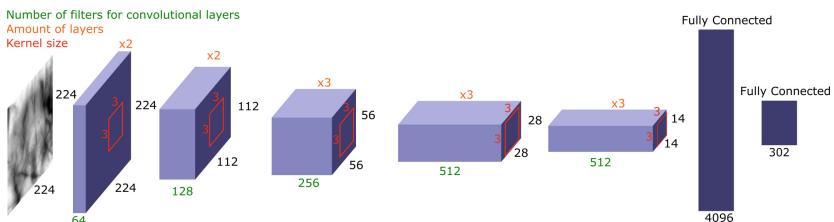


Fig. 6. VGG16 architecture with the last layer modified for transfer learning.

4 Results and Discussion

This section presents the experimental results for each evaluation. We used the accuracy and EER metrics to compare each combination of methods, which are widely used to evaluate biometric systems. For this purpose, the accuracy of the models was verified using a validation process that corresponded to known subjects but unknown samples. Thus, a set of matching scores were obtained for intra-class and inter-class comparisons using the “Data_real” and “Impostors” subsets to analyze the false acceptance rate (FAR), the false rejection rate (FRR), and the EER metric. PCA was implemented for both feature extraction and dimensionality reduction. Since FCC converts images to a larger size representation, PCA was applied to reduce the FCC data dimensionality and decrease the training time. PCA was also used for the raw and CLAHE images with four representations corresponding to 25, 50, 75, and 100 principal components. Finally, LBP and SIFT descriptors were implemented without adjusting or modifying the dataset images. Table 1 presents the accuracy of each evaluated model using the previously selected feature descriptors.

Table 1. Accuracy results of the identification process for each selected combination on the validation set. PCA25, PCA50, PCA75, and PCA100 Correspond to Representations of 25, 50, 75, and 100 Principal Components, Respectively. higher results are identified in bold.

Method variant	KNN	Logistic regression	CNN AlexNet	CNN VGG16
RAW+PCA25	0.8948	0.9777	N/A	N/A
RAW+PCA50	0.9072	0.9830	N/A	N/A
RAW+PCA75	0.9131	0.9911	N/A	N/A
RAW+PCA100	0.9161	0.9892	N/A	N/A
CLAHE+PCA25	0.9945	0.9970	N/A	N/A
CLAHE+PCA50	0.9967	0.9993	N/A	N/A
CLAHE+PCA75	0.9977	1.0000	N/A	N/A
CLAHE+PCA100	0.9984	1.0000	N/A	N/A
FCC+PCA25	0.7941	0.5914	N/A	N/A
FCC+PCA50	0.7977	0.6127	N/A	N/A
FCC+PCA75	0.8038	0.7685	N/A	N/A
FCC+PCA100	0.8048	0.7602	N/A	N/A
CLAHE+LBP	0.0033	0.0033	N/A	N/A
CLAHE+SIFT	0.7162	0.7209	N/A	N/A
Raw images	N/A	N/A	0.8520	0.9860
CLAHE images	N/A	N/A	0.9520	0.9740

The PCA results of the raw images indicate higher accuracy for the representations of 100 and 75 principal components for KNN and logistic regression,

respectively. It suggests that, possibly for logistic regression, the information could be more representative than KNN. The graphs in Fig. 7 show the EER for each evaluated model, where the ratio value is the acceptance rate of the samples that exceed a class probability threshold. Higher threshold values correspond to higher membership chances, thereby ensuring higher security levels for the system. Although the KNN has an EER of 0%, its operation point can be difficult to determine based on plot behavior, which shows low-security levels for impostor detection. Conversely, logistic regression has an EER of 3.2% and a clear tendency to discriminate impostors as the rejection threshold increases.

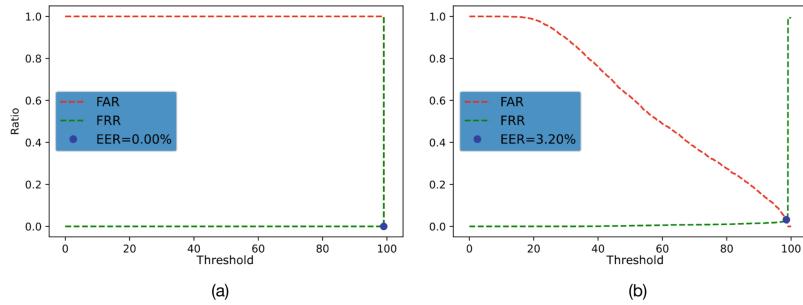


Fig. 7. EER plots for the best combinations of PCA: **a** PCA100 + KNN and **b** PCA75 + Logistic Regression. The green line corresponds to frr and the red one corresponds to FAR.

The PCA applied to CLAHE images among the evaluated models is the model with the best behavior. Although PCA100 has better accuracy for both the KNN and logistic regression, the PCA75 for logistic regression also achieves 100% accuracy, suggesting that no more components are required to identify an individual in a set of 302 persons. The overall behavior of the EER graphs is similar to that described above (i.e., PCA+Logistic regression); in this case, the EER achieved by logistic regression is 0.59%, which outperforms the tested methods and is acceptable for security applications. On the other hand, PCA applied to FCC exhibits behavior similar to PCA on raw images. The most accurate models remain PCA100+KNN and PCA75+Logistic Regression. Although the EER graphs for KNN present the same behavior, as shown in Fig. 8a and 9a, an opposite response is observed in the logistic regression (Fig. 8b and 9b), where an EER of 93.89% with a strong tendency to present false positives (see Fig. 9).

Figure 10 depicts the reported EER for LBP+KNN and the logistic regression of the SIFT descriptor. The LBP presents similar accuracy results for the evaluated techniques. However, given its low accuracy, it is unsuitable for proposed security applications. On the other hand, although SIFT has a higher accuracy than LBP, it does not reach 80%. In addition, since the accuracy of both models for the LBP feature space is low, an EER of 100% is observed, as shown in Fig. 10. However, although the accuracy is not very high in the SIFT

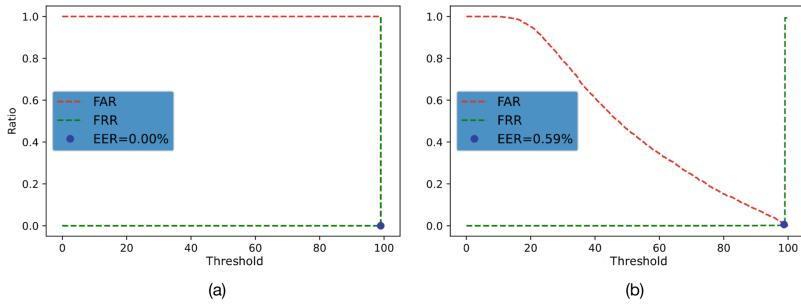


Fig. 8. EER plots for the best combinations of PCA on CLAHE images: **a** CLAHE + PCA100 + KNN and **b** CLAHE + PCA75 + Logistic Regression. The green line corresponds to the FRR and the red one corresponds to FAR.

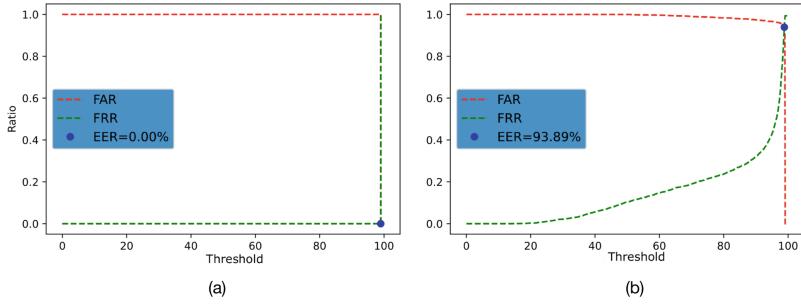


Fig. 9. EER plots for the FCC descriptor: **a** FCC + PCA100 + KNN and **b** FCC + PCA75 + Logistic Regression. The green line corresponds to the FRR, meanwhile the red one to FAR.

plot, it presents an EER of 12.18% and a strong tendency to reject impostors. However, the operating point for this model appears at a very low threshold; therefore, there is a high probability of rejecting non-impostors.

Finally, for the CNN-based models, it was found that the VGG16 architecture was significantly more accurate than AlexNet on RAW images and slightly better on CLAHE images. The obtained results are a consequence of its robustness and of the transfer learning process itself. VGG16 implements small convolutional kernels and increases the number of layers, enhancing the model's expressiveness to obtain a richer deep representation. Hence, there is a reduction in the number of trainable parameters in VGG16, achieving faster and more robust learning. On the other hand, since both networks were trained using transfer learning from the ImageNet database, the VGG network is able to capture more complex structures from hand-vein patterns on the studied database. Besides, it is observed that the EER is high, and it appears for a low acceptance threshold, suggesting a high number of false positives and false negatives, which is disadvantageous for its

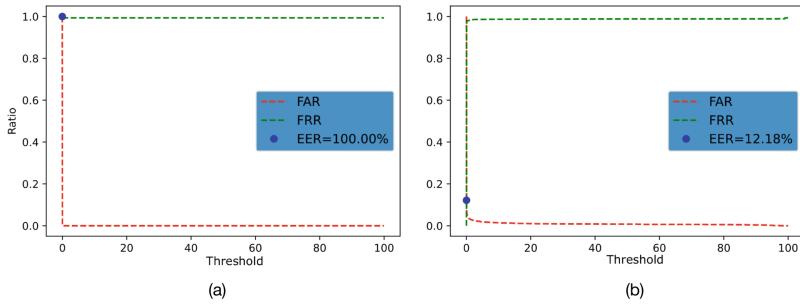


Fig. 10. EER plots for the LBP and SIFT descriptors: **a** LBP + KNN and **b** SIFT + Logistic Regression. The green line corresponds to the FRR, meanwhile the red one to FAR.

execution in safe environments. The reported EER for the CNN models are presented in Figs. 11 and 12. Both models achieve an identification accuracy above 85%.

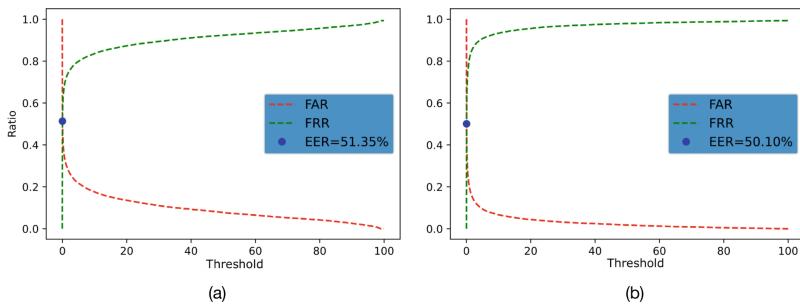


Fig. 11. EER plots for the AlexNet Model on **a** RAW images and **b** CLAHE processed images. The green line corresponds to the FRR, meanwhile the red one to FAR.

4.1 Discussion on Experimental Results

Following, we present the most relevant findings on the results of both stages of the evaluation scheme, feature extraction, and identification process.

Feature Extraction The process of selecting a feature subspace is critical for vein-based biometric systems. Many of these techniques require high-contrast images to obtain relevant information.

- The FCC demonstrated the ability to analyze typical information of vascular regions, such as the area or centroid of the vascular structures. However, given the sensitivity of these techniques to environmental factors, slight variations in lighting can mean considerable changes in the FCC.

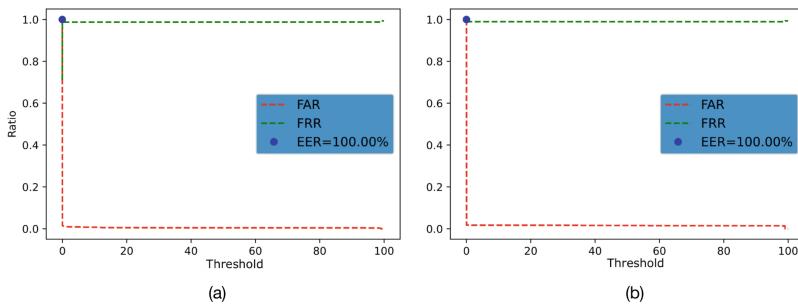


Fig. 12. EER plots for the VGG16 Model on **a** RAW images and **b** CLAHE processed Images. The green line corresponds to the FRR, meanwhile the red one to FAR.

- Although the acquired images do not have much variation in terms of rotation and scale, there is greater accuracy and lower EER for SIFT descriptor than for LBP because the former is invariant to orientation and scale. In contrast, the second is invariant only to rotation.
- PCA for feature space reduction proved to be a key tool for decreasing the data size and selecting relevant information, as seen in the raw images.
- Experiments conducted using CLAHE for PCA showed that image processing is recommended for improving accuracy. Furthermore, this feature space is more suitable for biometric applications than other methods evaluated in this study.

Identification Process Accuracy is an important parameter in biometric systems, but also the EER should be considered as a reinforcing metric for evaluating identification models. The above is mainly due to the behavior of the FAR and FRR curves. Their location provides further insight into the performance of the system. Furthermore, EER alone does not guarantee that the model correctly differentiates between training classes.

- While the KNN involves a low computational cost, it presents acceptable accuracy with all selected feature spaces except for the LBP. It demonstrates the ability of the proposed method to identify sets with multiple classes. However, since the EER is positioned at extreme values and has abrupt cuts in FRR and FAR, it does not appear to be a viable solution for safety applications.
- Although logistic regression requires greater computational capacity for the learning process, it performs well for the different techniques studied. Additionally, the behavior presented in the EER suggests that the tuning process is relatively easy and allows acceptable EER levels for security applications.
- CNN has high accuracy values that are very close to 100%. The VGG16 model achieved the best results in our experiments, which differs from previous works mainly due to the transfer learning process. Thus, we demonstrate that shallow architectures are also suitable for hand-vein identification even

if trained end-to-end. The obtained results also show that it is possible to use this kind of model embedding them in low-resource systems.

After verifying the behavior of the EER graph for each model, it is possible to establish that the most secure identification methodology in the proposed scenario has high levels of accuracy and low EER. This behavior is mainly presented in the graphs corresponding to the logistic regression models. The best combination was logistic regression + PCA75 with images processed using CLAHE, with an accuracy of 100% and an EER of 0.59%.

5 Conclusions and Future Works

In the present work, we proposed a comparative analysis of different representative techniques for individual identification based on hand-vein biometrics. The implemented evaluation scheme combines four feature extraction techniques and three classic machine learning techniques by using two main types of images and the EER and accuracy as comparative metrics. The experimental results show that the obtained accuracy and EER have similar performances to those presented in [5, 61]. Although they used a different number of images, these studies utilized the same type of pictures, similar feature extraction techniques, and ML methods, with an accuracy of 100% and an EER of up to 2.30%. Moreover, our proposal provides a baseline framework for further vein-based biometric studies, considering different evaluated technique combinations.

Based on the obtained results, future studies should compare different combinations using the same metrics (i.e., accuracy and EER) to create benchmark performances that increase the security levels and development of biometric systems. Moreover, it is necessary to implement the best combination of identification and feature extraction techniques found in a fully integrated device that allows hand-vein image acquisition and biometric identification for further time consumption analysis to improve the proposed scheme. Finally, further research is needed to integrate skin thickness or racial profile factors to determine the effects of such variables during the identification stage.

Acknowledgment. The authors thank the Research Project ANID FONDECYT Iniciación en Investigación 2022 No. 11220693 “End-to-end multi-task learning framework for individuals identification through palm vein patterns”, Ministerio de Ciencia, Tecnología, Conocimiento e Innovación, Gobierno de Chile.

References

1. Aglio-Caballero, A., Ríos-Sánchez, B., Sánchez-Ávila, C., De Giles, M.J.M.: Analysis of local binary patterns and uniform local binary patterns for palm vein biometric recognition. In: 2017 International Carnahan Conference on Security Technology (ICCST), pp. 1–6. IEEE (2017)
2. Al-Johania, N.A., Elrefaei, L.A.: Dorsal hand vein recognition by convolutional neural networks: Feature learning and transfer learning approaches. *Int. J. Intell. Eng. Syst.* **12**(3) (2019)
3. Hayat, N.A.M., Noh, Z.M., Yatim, N.M., Radzi, S.A.: Analysis of local binary pattern using uniform bins as palm vein pattern descriptor. *J. Phys.: Conf. Ser.* **1502**(1) (2020)
4. Azmi, A.N., Nasien, D., Omar, F.S.: Biometric signature verification system based on freeman chain code and k-nearest neighbor. *Multimed. Tools Appl.* **76**(14), 15341–15355 (2017)
5. Babalola, F.O., Bitirim, Y., Toygar, O.: Palm vein recognition through fusion of texture-based and CNN-based methods. *Signal, Image Video Process.* (2020)
6. Bashar, K., Murshed, M.: Texture based vein biometrics for human identification: A comparative study. In: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), vol. 2, pp. 571–576. IEEE (2018)
7. Bhilare, S., Jaswal, G., Kanhangad, V., Nigam, A.: Single-sensor hand-vein multimodal biometric recognition using multiscale deep pyramidal approach. *Mach. Vis. Appl.* **29**(8), 1269–1286 (2018)
8. Chen, P., Ding, B., Wang, H., Liang, R., Zhang, Y., Zhu, W., Liu, Y.: Design of low-cost personal identification system that uses combined palm vein and palmprint biometric features. *IEEE Access* **7**, 15922–15931 (2019)
9. Chopra, R., Kaur, S.: M Tech Student. Finger print and finger vein recognition using repeated line tracking and minutiae. *Int. J. Adv. Sci. Res.* **2**(2), 13–22 (2017)
10. Electronic Computers, O.N.: On the encoding of arbitrary geometric configurations. *Ire Trans. Electron. Comput.* 260–268 (1960)
11. Crisan, S., Tebrean, B., Crisan, T.E.: Multimodal liveness detection system for hand vein biometrics. In: 2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA), pp. 1–6. IEEE (2018)
12. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
13. Dreiseitl, S., Ohno-Machado, L.: Logistic regression and artificial neural network classification models: A methodology review (2002)
14. Fix, E., Hodges J.L.: Estimation discriminant analysis nonparametric density. *USAF Sch. Aviat. Med.* **57**(3), 233–238 (1951)
15. Galbally, J., Ortiz-Lopez, J., Fierrez, J., Ortega-Garcia, J.: Iris liveness detection based on quality related features. In: 2012 5th IAPR International Conference on Biometrics (ICB), pp. 271–276. IEEE (2012)
16. Gielczyk, A., Choras, M., Kozik, R.: Lightweight verification schema for image-based palmprint biometric systems. *Mob. Inf. Syst.* (2019)
17. He, C.G., Bao, S.D., Li, Y.: A novel tri-factor mutual authentication with biometrics for wireless body sensor networks in healthcare applications. *Int. J. Smart Sens. Intell. Syst.* **6**(3), 910–931 (2013)
18. Hernández-García, R., Santamaría, J.I., Barrientos, R.J., Salazar Jurado, E.H., Castro, F.M., Ramos-Cózar, J., Guil, N.: Large-scale palm vein recognition on synthetic datasets. In: 2021 40th International Conference of the Chilean Computer Science Society (SCCC), pp. 1–8. IEEE (2021)

19. Hotelling, H.: Analysis of a complex of statistical variables into principal components. *J. Educ. Psych.* **24**, 417–441, 498–520; *J. Educ. Psychol.* **24**, 417–441 (1933)
20. Huang, B., Dai, Y., Li, R., Tang, D., Li, W.: Finger-vein authentication based on wide line detector and pattern normalization. In: 2010 20th International Conference on Pattern Recognition, pp. 1269–1272. IEEE (2010)
21. Imah, E.M., Jatmiko, W., Basaruddin, T.: Electrocardiogram for biometrics by using adaptive multilayer generalized learning vector quantization (AMGLVQ): Integrating feature extraction and classification. *Int. J. Smart Sens. Intell. Syst.* **6**(5), 1891–1917 (2013)
22. Jain, A.K., Ross, A., Prabhakar, S.: An introduction to biometric recognition. *IEEE Trans. Circuits Syst. Video Technol.* **14**(1), 4–20 (2004)
23. Jia, W., Gao, J., Xia, W., Zhao, Y., Min, H., Jing-Ting, L.: A performance evaluation of classic convolutional neural networks for 2d and 3d palmprint and palm vein recognition. *Int. J. Autom. Comput.* **18**(1), 18–44 (2021)
24. Khanam, R., Khan, R., Ranjan, R.: Analysis of finger vein feature extraction and recognition using da and knn methods. In: 2019 Amity International Conference on Artificial Intelligence (AICAI), pp. 477–483. IEEE (2019)
25. Kim, W., Song, J.M., Park, K.R.: Multimodal biometric recognition based on convolutional neural network by the fusion of finger-vein and finger shape using near-infrared (NIR) camera sensor. *Sensors* **18**(7), 2296 (2018)
26. Krizhevsky, B.A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2012)
27. Kumar, R.M., Krishnan, M.: An open source contact-free palm vein recognition system. *Int. J. Adv. Appl. Sci. (IJAAS)* **6**(4), 319–324 (2017)
28. Lee, E.C., Jung, H., Kim, D.: New finger biometric method using near infrared imaging. *Sensors* **11**(3), 2319–2333 (2011)
29. Lee, J.C., Lo, T.M., Chang, C.P.: Dorsal hand vein recognition based on directional filter bank. *Signal, Image Video Process.* **10**(1), 145–152 (2016)
30. Li, X., Zhang, R., Wang, Y.: Hand dorsal vein recognition by matching width skeleton models. In: Image Processing (ICIP), 2016 IEEE International Conference (2016)
31. Liu, T., Xie, J.B., Yan, W., Li, P.Q., Lu, H.Z.: An algorithm for finger-vein segmentation based on modified repeated line tracking. *Imaging Sci. J.* **61**(6), 491–502 (2013)
32. Mohammed, B.O., Shamsuddin, S.M., Hasan, S.: An overview of uni- and multi-biometric identification of identical twins. *IEIE Trans. Smart Process. Comput.* **8**(1), 71–84 (2019)
33. Mucherino, A., Papajorgji, P.J., Pardalos, P.M.: k-nearest neighbor classification. In: Encyclopedia of Database Systems. vol. 1, pp. 83–106. Springer, US, Boston, MA (2009)
34. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. *Pattern Recogn.* **29**(1), 51–59 (1996)
35. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7), 971–987 (2002)
36. Pan, M., Kang, W.: Palm vein recognition based on three local invariant feature extraction algorithms. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7098 LNCS, pp. 116–124 (2011)

37. Parihar, R.S., Jain, S.: A robust method to recognize palm vein using SIFT and SVM classifier. *SSRN Electron. J.* **1**, 1703–1710 (2019)
38. Pizer, S.M., Amburn, E.P., Austin, J.D., Cromartie, R., Geselowitz, A., Greer, T., Romeny, B.H., Zimmerman, J.B., Zuiderveld, K.: Adaptive histogram equalization and its variations. *Comput. Vis., Graph., Image Process.* **39**(3), 355–368 (1987)
39. Prasad, K.K., Aithal P.S.: A study on fingerprint hash code generation based on Md5 algorithm and freeman chain code. *Int. J. Comput. Res. Dev.* **3**(1), 13–22 (2018)
40. Raghavendra, R., Avinash, M., Marcel, S., Busch, C.: Finger vein liveness detection using motion magnification. In: 2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems, BTAS 2015 (2015)
41. Rajalakshmi, M., Rengaraj, R., Bharadwaj, M., Kumar, A., Raju, N.N., Haris, M.: An ensemble based hand vein pattern authentication system. *CMES—Comput. Model. Eng. Sci.* **114**(2), 209–220 (2018)
42. Rojas, C., Hernández-García, R., Barrientos, R.J.: Individuals identification using finger veins under a multi-core platform. In: 2018 37th International Conference of the Chilean Computer Science Society (SCCC), pp. 1–7 (2018)
43. Sajjad, M., Khan, S., Hussain, T., Muhammad, K., Sangaiah, A.K., Castiglione, A., Esposito, C., Baik, S.W.: CNN-based anti-spoofing two-tier multi-factor authentication system. *Pattern Recognit. Lett.* **126**, 123–131 (2019)
44. Santamaría, J.I., Hernández-García, R., Barrientos, R.J., Castro, F.M., Ramos-Cózar, J., Guil, N.: Evaluation of end-to-end CNN models for palm vein recognition. In: 2021 40th International Conference of the Chilean Computer Science Society (SCCC), pp. 1–8. IEEE (2021)
45. Shah, K., Patel, H., Sanghvi, D., Shah, M.: A comparative analysis of logistic regression, random forest and knn models for the text classification. *Augment. Hum. Res.* **5**(1) (2020)
46. Shaheed, K., Liu, H., Yang, G., Qureshi, I., Gou, J., Yin, Y.: A systematic review of finger vein recognition techniques. *Information (Switzerland)* **9**(9) (2018)
47. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings, pp. 1–14 (2015)
48. Thenmozhi, G., Anandha Jothi, R., Palanisamy, V.: Comparative analysis of finger vein pattern feature extraction techniques an overview. *Int. J. Comput. Sci. Eng.* **7**(5), 867–872 (2019)
49. Uludag, U., Pankanti, S., Prabhakar, S., Jain, A.K.: Biometric cryptosystems: issues and challenges. *Proc. IEEE* **92**(6), 948–960 (2004)
50. Van, H.T., Duong, C.M., Van Vu, G., Le, T.H.: Palm vein recognition using enhanced symmetry local binary pattern and sift features. In: 2019 19th International Symposium on Communications and Information Technologies (ISCIT), pp. 311–316. IEEE (2019)
51. Wan, H., Chen, L., Song, H., Yang, J.: Dorsal hand vein recognition based on convolutional neural networks. In: 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1215–1221. IEEE (2017)
52. Wang, G., Wang, J.: SIFT based vein recognition models: analysis and improvement. *Comput. Math. Methods Med.* (2017)
53. Wu, J.D., Liu, C.T.: Finger-vein pattern identification using principal component analysis and the neural network technique. *Expert Syst. Appl.* **38**(5), 5423–5427 (2011)
54. Wu, J.D., Liu, C.T.: Finger-vein pattern identification using SVM and neural network technique. *Expert Syst. Appl.* **38**(11), 14284–14289 (2011)

55. Wu, W., Elliott, S.J., Lin, S., Sun, S., Tang, Y.: Review of palm vein recognition. *IET Biometrics* **9**(1), 1–10 (2020)
56. Wulandari, M., Gunawan, D.: On the performance of pretrained CNN aimed at palm vein recognition application. In: 2019 11th International Conference on Information Technology and Electrical Engineering, ICITEE 2019, vol. 7, pp. 1–6 (2019)
57. Yang, J., Shi, Y.: Finger-vein ROI localization and vein ridge enhancement. *Pattern Recognit. Lett.* **33**(12), 1569–1579 (2012)
58. Yazdani, F., Andani, M.E.: Verification based on palm vein by estimating wavelet coefficient with autoregressive model. In: *2017 2nd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, pp. 118–122. IEEE (2017)
59. Yuksel, A., Akarun, L., Sankur, B.: Hand vein biometry based on geometry and appearance methods. *IET Comput. Vision* **5**(6), 398–406 (2011)
60. Zaaraoui, H., El Kaddouhi, S., Abarkan, M.: A novel approach to face recognition using freeman chain code and nearest neighbor classifier. In: 2019 International Conference on Intelligent Systems and Advanced Computing Sciences (ISACS), pp. 1–5. IEEE (2019)
61. Zhang, L., Cheng, Z., Shen, Y., Wang, D.: Palmprint and palmvein recognition based on DCNN and a new large-scale contactless palmvein dataset. *Symmetry* **10**(4), 1–15 (2018)
62. Zhang, L., Li, L., Yang, A., Shen, Y., Yang, M.: Towards contactless palmprint recognition: A novel device, a new benchmark, and a collaborative representation based identification approach. *Pattern Recogn.* **69**, 199–212 (2017)
63. Zhang, Z.: Introduction to machine learning: K-nearest neighbors. *Ann. Transl. Med.* **4**(11) (2016)



URL Classification with Intrusion Detection System

Veeresh Uppara^(✉), Akif Iqbal, Vishal P, Vinay M V, and Sarasvathi V

PES UNIVERSITY, Department of CSE, Bengaluru, India

uveereshadoni@gmail.com, akif.iqbal28@gmail.com,
vishalupadyaya6786@gmail.com, vinaymds0@gmail.com, sarsvathiv@pes.edu

Abstract. The generation we live in has almost everything processed through the internet. When a user wants a question to be answered they open their web browser and search for a query, many websites show up on the screen which the user can choose from. Dozens of Uniform Resource Locator (URLs) are created everyday, not all of them are safe and some are built for malicious purposes. The attacker who creates these malicious URLs can send multiple responses thereby disrupting the traffic and delaying the process. Our aim is to build and incorporate models which can classify that website and identify the category using URL and check for the presence of any application layer attacks. Our paper deals with capturing live network traffic and packets, extracting relevant values and feed these to URL and Intrusion Detection System (IDS) modules. We use certain heuristics, train and build the Machine Learning (ML) models like Multinomial Naive Bayes and Logistic Regression giving accuracy of 87.51 and 95.89% respectively. Inspection of traffic from the website is analyzed by building a network traffic analyzer using pyshark to capture traffic in real time. For the detection method of IDS, we would be using Anomaly-Based methods and for identifying malicious traffic we would be using Support Vector Machine (SVM) to classify attacks into application layer attacks namely Distributed Denial of Service (DDoS), Cross-site Scripting (XSS), BruteForce, Denial of Service (DoS), Botnets, Structured Query Language (SQL) Injection.

Keywords: Machine learning · URLs · Support vector machine · Anomaly based intrusion detection system · Multinomial Naive Bayes · Logistic regression

1 Introduction

1.1 General Overview

Browsing on the internet has been a common thing today to search for your needs. Websites are built by developers for consumers' demands and as a business for developers. These websites usually help solve particular problems, however,

not all websites are built with good intentions and are often built for malicious purposes, intending to steal users' private information and with the aim to disrupt users' machines or install malicious software. But with the increase in growth for browsing where millions of URLs are created everyday, malicious websites and disruption in traffic with network attacks also have been a concern for many in this current era. People are prone to various different types of attacks, these could include phishing URLs, websites that imitate another legitimate website and upon user entering, credentials stealing their data hence causing concern to people who browse the web. Another type could include websites that upon visiting them causes drive-by downloads and plenty of executable files to get installed along with malicious virus on the users' computer. To rectify this we use URLs and with the Machine Learning models whereby using specific algorithms we can classify the URLs as malicious or non-malicious while also finding the category of the URL like adult, arts, sports, business, references, recreational, movie, computers, games, health, home, kids, news and science for efficient classification. Other methods can also be viable like looking at the elements on the raw Hypertext Markup Language (HTML) page like iframes, reference links, tokenizing the URL or looking at commonly used phishing words in the URL. For network traffic control we use an IDS which checks for malicious activity and then reports the security vulnerabilities to the server. We use pyshark model to record the incoming traffic and SVM to classify the traffic into any of the application layer attacks namely DDoS, DoS, BruteForce, XSS, SQL Injection, Botnets.

1.2 Section Overview

This paper commences with a literature survey section which highlights the previous work and gives a brief overview on each paper's advantages and limitations. Proposed Methodology dives deep into the work that was carried out starting from the datasets that were used, presenting a high level design architecture and explaining the complete implementation of the research in the paper. This section contains algorithms, detailed explanation and overall advantage of the research carried out. A section is dedicated to highlighting results that were obtained from our paper, tabulating them and presenting confusion matrices and output screenshots. Final section concludes the work carried out in the paper and mentions challenges that were faced and scope that hold in future related to our work.

1.3 Problem Statement

This paper focuses to combine functionalities of URL classification and IDS to ensure the websites being visited are non-malicious, free from executable file, and in addition also categorize the websites into the available 12 categories. With the addition of IDS, we bring in the capability to monitor the network traffic and also identify the type of application layer attack present in the traffic. To add

complexity to our paper we have made it more real time with pyshark module for capturing packets being sent and received by the client and server systems.

2 Literature Survey

2.1 URL Classification

A lot of different approaches have been taken up in past and include different methods and different classifiers. Feature selection is incorporated to reduce the number of features required for classification and then rank them on the basis of their importance. Machine learning classification algorithms have been incorporated which include SVM, Naive Bayes, Decision Trees along with Ensemble learning techniques. Detailed studies have been carried out too using various ML classifiers and comparing their accuracy and observing their performances using a confusion matrix. Some of these classifiers that have been compared are K-Nearest Neighbours (KNN), SVM, Logistic Regression, Adaboost, Gradient Boosting, Random Forest and Voting Classifier. Deep Learning approaches is a method wherein deep learning based architectures simple Recurrent Neural Network (RNN), simple Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN-LSTM) are used to classify URLs as benign or malicious. Heuristic based approaches include Page-based features where page rank and quality are checked, Domain-based features, Type-based features and Word-based features.

Ding [4] finds appropriate and typical features in the URLs and put them into a list. To store the features they use a two dimensional array where one dimension they store the URL and other dimension for specific information of the features. ML models they use are Logistic Regression and Naive Bayes. To train the URL with the models they use a tokenizer which converts the URL into tokens and then fit it with the TfidfVectorizer. The results they obtained with different ML models had an accuracy 88.24% on average.

Manjeri et al. [3] uses common ML models for classification but experiments on the results by handling class imbalance, normalization and replacing missing values on the dataset. Additionally it incorporates extra features like Remote App Packets, Source App Packets other miscellaneous values relevant to their work. The results they obtained with different ML models had an accuracy improvement of 1.4% on average without class imbalance. However this paper requires much time for training as it combines lot off decision trees to determine the class which requires very high computational power.

Khan et al. [5] incorporates plenty of ML models namely, KNN, SVM, Logistic Regression, Adaboost, Gradient Boosting, Extra Trees, Random Forest and finally feeds them into a Voting Classifier. Their work included using scoring features Chi-Square and Analysis of Variance (ANOVA). The entire work was carried out on two different datasets from University of New Brunswick (UNB) and Kaggle. Using Voting Classifier they were able to achieve highest accuracy possible of 99.72 and 95.37% respectively. On average all the remaining other

models resulted in an accuracy of 98.41 and 91.64% respectively. Voting classifier takes in all the models that were used in this paper and gives high accuracy however implementing Voting Classifier requires to implement at least 3 good classifiers.

2.2 The Intrusion Detection System

The Network Traffic Analyzer or IDS have been installed at different locations to protect systems or a network from application layer or network layer attacks. They have been designed in various ways to detect any application layer or network layer attack-Signature-based and Anomaly based. Anomaly-based network traffic analyzers make use of ML models like SVM, Artificial Neural Networks (ANN), KNN or several hybrid models like SVM + Principal Component Analysis (PCA) to obtain better results in terms of accuracy and False Positive (FP) ratios. But it has been difficult choosing between the models especially with the SVM + PCA having the highest accuracy but PCA presents a lot of issues when it comes to choosing which attributes have to be merged for making sure the dimensionality is taken care of. Various ML models have been trained using different datasets. KDDCUP is a famous dataset for DDOS attacks. The CSE CIC AWS (Communications Security Establishment and the Canadian Institute for Cybersecurity) dataset is a collaborative dataset that has almost all application layer attacks like DDOS, XSS, and SQL Injection with many attributes that have to be taken into account for better accuracy. The challenge in the making of an ideal network traffic analyzer is the selection of the right features from the datasets.

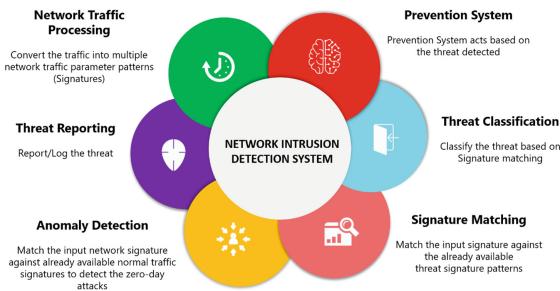


Fig. 1. Intrusion Detection System functionalities.

In Fig. 1, the main use cases of the intrusion detection system are provided. In our paper we are focusing on most of the functionalities from network traffic monitoring to signature matching, from Anomaly detection to Threat Prevention and Classification of the Application Layer attack.

Joshi et al. [1] have discussed the various ML algorithms for coming up with the accurate intrusion detection systems. They have highlighted the pros and

cons of each model-supervised, unsupervised, the accuracies of them on the same dataset. Through their paper it was shown how SVM+PCA (Principal Component Analysis) gave the best accuracy of 99.5%. The importance of selecting the right features amongst the 41 features of KDD cup is also mentioned. But the limitation of the paper is SVM requires more time and memory making it costlier in comparison to other models. The paper focused on a limited number of attacks as it considered the KDD cup dataset.

Netscrapper is a Real-time flow-based Network traffic analyzer developed by Ahmed et al. [2] using three Machine Learning Models Random Forest, KNN and ANN that classifies the 53 most popular online applications (Google, Twitter, Facebook, etc.) on the network and enables network administrators to detect malicious traffic and protect user data. They utilized CIC flowmeter, an open-source toll that reduces network noise and removes incomplete records. Implementation of Netscrapper involves collecting live traffic flow. The traffic stream is fed to a CIC flow meter before being analyzed by an ML pipeline that executes three ML models. Among the three ML models, The ANN model could inspect four application flow streams per second with an accuracy of 99.86% and without requiring any human intervention by automatically extracting features. Hence ANN provides the highest accuracy and shortest prediction time and is best suited for real-time prediction. The paper's limitations are KNN is estimated to be a comparatively slow model than ANN and RF. Since training the data using KNN for each prediction requires a lot of time for large datasets. Hence, unsuitable for real-time inference.

3 Proposed Methodology

3.1 Dataset

The binary dataset obtained from Kaggle [8] consists of 2 labels—good and bad. The dataset has of 420,465 URLs in total. In Fig. 2, the count of label of the binary dataset is shown through a bar graph visual.

Figures 3 and 4 describes the most common recurring word (word cloud) in the URLs string for good and bad URL labels respectively.

The categorical dataset obtained from Kaggle [7] consists of 12 categories. Figure 5 shows a bar plot that summarizes the entire dataset consisting of 1,048,573 URLs in total.

Figure 6 describes the most common recurring word (word cloud) in the URLs string for a category—reference. Similarly to help us understand the dataset clearly, word clouds were generated for the remaining 11 categories.

In Fig. 7, we have shown the correlation of subset of the dataset through a heatmap. The colours corresponding to a value close to one means that the attributes are closely related and hence one amongst them can be chosen for training of the model. The colours corresponding to a value close to zero means that the attributes aren't closely related and both the attributes have to be taken for training of the model.

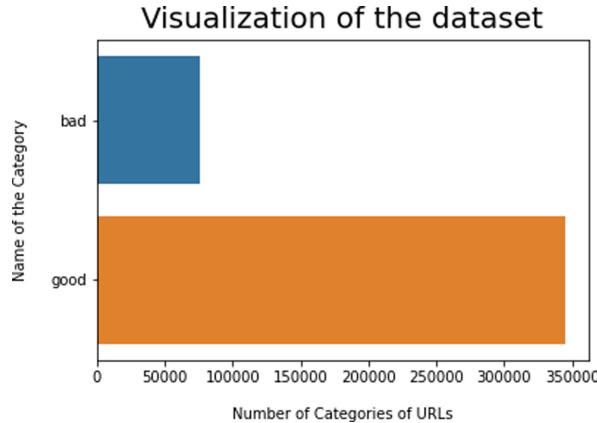


Fig. 2. Binary dataset.

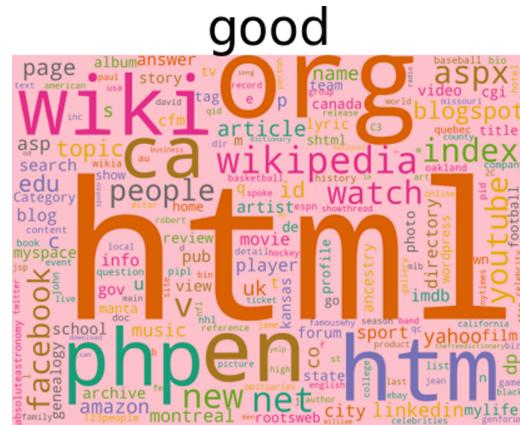


Fig. 3. Word cloud for good URLs.

In Fig. 8, we have given an overview of the seven CSV files of CSE CIC IDS 2018 [6] obtained from Kaggle. The dataset includes the captured network traffic and system logs of each machine, along with 80 features extracted from the captured traffic using CICFlowMeter. Each CSV has its own application layer attacks like BruteForce, DOS, DDOS, XSS, SQL injection and Botnets. The Network Traffic Analyzer has been used to detect suspicious traffic with available signatures. Some have come up with ML models to make the network traffic analyzers and make them detect traffic in real time, commonly called Anomaly based network traffic analyzers. The URL classification has been the topic of discussion as to which features have to be selected for the classification. The number of domains is just increasing each day. To make an ML model which classifies the URLs in real time, we must look at the right heuristic features.

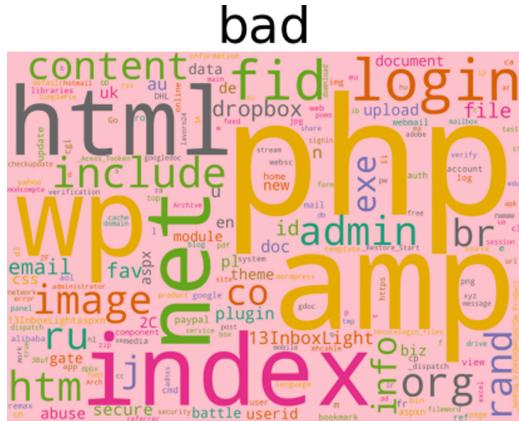


Fig. 4. Word cloud for bad URLs.

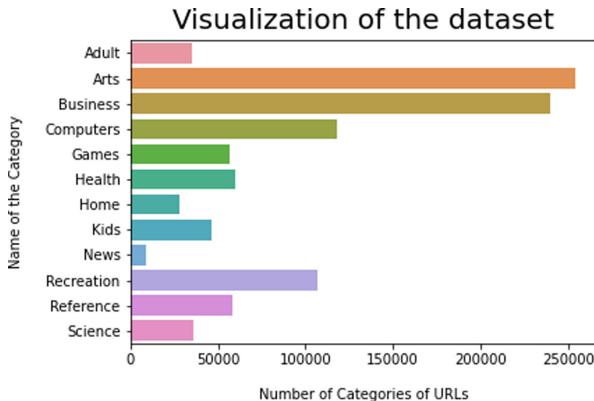


Fig. 5. Categorical dataset.

In addition, we are also looking at categorizing the URLs into the available 12 categories. In our paper, we have identified 3 tasks, Packet Capture, URL Classification, Intrusion Detection System. By doing so, we would be able to provide software that can provide websites classification along with the category of the website using text analysis of URLs and analyze the traffic, both the incoming traffic and the outgoing traffic, and identify the type of application layer attack if traffic is found suspicious.

In Fig. 9, the high level design of our paper is shown which is a simple client server architecture where client requests for services while browsing the internet. Each request is checked by the network traffic analyzer for an anomaly in traffic. The server responds with the services through packets. These packets are again checked by the network traffic analyzer. Only traffic which is not malicious is sent/received. Along with this functionality, we want our work to also identify

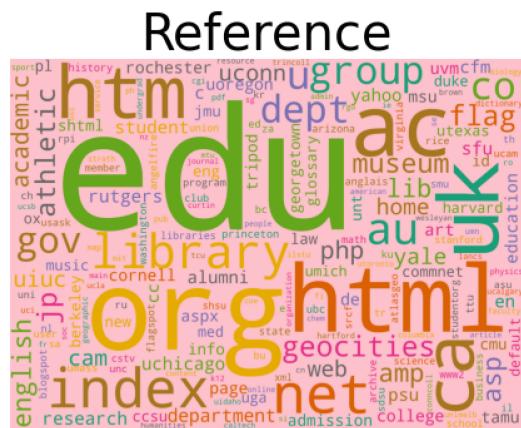


Fig. 6. Word cloud for reference URLs.

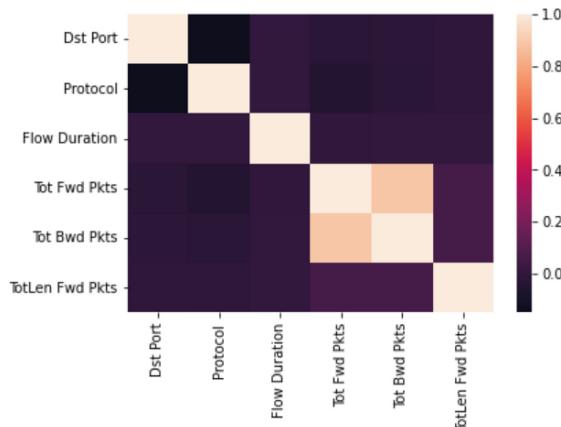


Fig. 7. Correlation of subset of dataset.

the application layer attacks and categorise the websites into the available 12 categories with additional classification of the websites.

Browsing by user is being captured by the packet capture. Packet Capture would then send the arrayofpackets[] and websites[] to the other modules. The URL Classification and Categorization would classify the URLs as malicious or non-malicious and also determine the category of the website. Then the Network Traffic Analyzer would determine the type of application layer attack if present else would classify the traffic as benign traffic.

3.2 The Packet Capture Module

This module captures traffic, both incoming and outgoing. Through this model we would be able to detect the traffic in dynamic way helping the clients and

Benign	4883142
DDOS attack-HOIC	686012
Bot	286191
FTP-BruteForce	193360
SSH-Bruteforce	187589
DoS attacks-GoldenEye	41508
DoS attacks-Slowloris	10990
DDOS attack-LOIC-UDP	1730
Brute Force -Web	611
Brute Force -XSS	230
SQL Injection	87

Fig. 8. Application layer attack types.

servers. This makes use of pyshark module of python which is tsharks CLI. Pyshark module helps in writing python scripts which would help us in accessing the attributes of the captured traffic like IP address, SRC port etc., which would then be sent to the other modules. In Algorithm 1, through pysharks LiveCapture() API we would capture traffic in real time. We have set the Berkeley Packet Filter (bpf filter) to IP and port 53 to capture the browsing activity of the user. The arrofpackets[] and websites[] are returned through this model.

Algorithm 1 Packet Capture

```

1: Capture  $\leftarrow$  LiveCapture(bpf filter=IP, 53
2: arrofpackets  $\leftarrow$  empty list
3: websites  $\leftarrow$  empty list
4: for packet in sniffContinuously(packet = 5) do
5:   print packet
6:   arofofpacket  $\leftarrow$  empty list
7:   if packet.ip or packet.udp or packet.tcp then
8:     arofofpacket  $\leftarrow$  append IP, SRC port
9:     print arofofpacket
10:    arrofpackets  $\leftarrow$  append arofofpacket
11:   end if
12:   if packet.dns then
13:     websites  $\leftarrow$  append dns.queryname
14:   end if
15: end for
16: return arrofpackets, websites

```

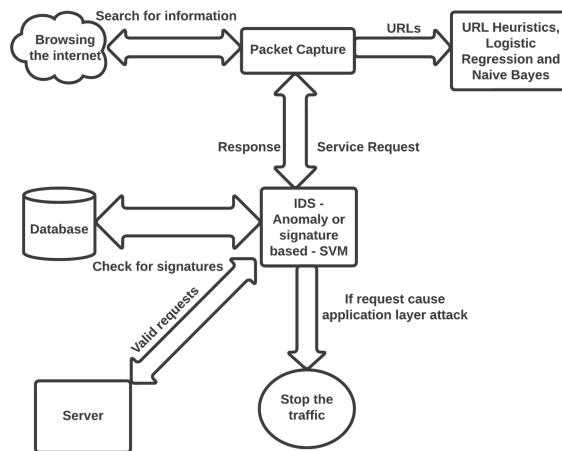


Fig. 9. High level design.

3.3 The URL Classification And Categorization

This task from our work deals with giving results of the website on the basis of URL text analysis. Logistic Regression is used to determine if website is malicious or not and Multinomial Naive Bayes determines the category of the website. Additionally two different heuristics are included which displays if any sort of phishing words were found in the URL string and count of the HTML tags such as anchor tags, script tags, link tags and iframe tags.

Phishing Words Algorithm 2 describes the algorithm to search for commonly used phishing words that could be present in the URL string. A regular expression library is used and the URL string is split on the basis of whitespaces and stores as list. A pre-defined list contains most common words used for phishing purposes and using these lists a search is performed in the URL string.

HTML tags count This task makes use of two main components:

Chrome Webdriver A part of the open source testing framework Selenium, it helps in extracting the page source of the websites page and gives the HTML source code.

HTMLParser A pre-existing class with methods, to implement this, the class needs to be extended and two methods `handle_starttag()` and `handle_startendtag()` methods can be overridden to perform the desired operations whenever any HTML tag is encountered. As shown in Algorithm 3, a count dictionary is initialized which eventually gets returned, in the two methods the count value of the tag is incremented using the tag itself as the dictionary key.

Algorithm 2 Phishing Words

```

1: split URL string on white spaces using regex
2: phishing_words  $\leftarrow$  common phishing words
3: words  $\leftarrow$  empty list
4: count  $\leftarrow$  0
5: for element in phishing_words do
6:   if element in tokenized_words then
7:     count  $\leftarrow$  count + 1
8:     append found word in words list
9:   end if
10:  if count > 0 then
11:    return count, words
12:  end if
13: end for

```

Algorithm 3 HTML tag count

```

1: extend HTMLParser class as MyHTMLParser
2: initialize count as dictionary in MyHTMLParser
3: increment tag count for every tag encounter
4: in handle_starttag(), handle_startendtag()
5: wd  $\leftarrow$  webdriver.Chrome
6: pass URL string in get method of wd
7: html  $\leftarrow$  call page_source method of wd
8: parser  $\leftarrow$  object of class MyHTMLParser
9: return parser.count {refer to section D2 for explanation}

```

Algorithm 4 Logistic Regression

```

1: data  $\leftarrow$  binary dataset
2: y  $\leftarrow$  data[label]
3: urllist  $\leftarrow$  data[url]
4: vectorizer  $\leftarrow$ 
5: TfidfVectorizer(modifiedTokenizer)
6: X  $\leftarrow$  vectorizer.fit_transform(urllist)
7: X1, X2, y1, y2  $\leftarrow$ 
8: train_test_split(X, y, testsize=.3)
9: logit  $\leftarrow$  LogisticRegression(0, sag)
10: logit.fit(X1, y1)
11: logit.predict(X2)

```

Logistic Regression Algorithm 4 for Logistic Regression, we choose the URL dataset [8] having a total of 420465 rows and two columns namely, URL and label where we store in *y* and *urllist* variables respectively. We use a custom tokenizer to split the URL into tokens. The tokenizer splits it by slash, then the slash tokens into hyphen and finally the hyphen tokens to dot. Add the dot and hyphen and store it in an empty list. Also ‘.com’ tokens is removed since it isn’t necessary for this model. These tokens are used in *TfidfVectorizer*, a module from

Python's sklearn which gives numerical values needed to train with the model. We do fit_transform of URL with this vectorizer. The sklearn's train_test_split takes the label and the vectorized tokens for training and the rest for testing. Training set is fitted with logistic regression model where random state is 0 and solver as 'sag' since the dataset is large. Fit the test variables with the logistic model and get the prediction.

M-Naive Bayes Algorithm 5 that uses the categorical dataset [7] first takes 12 different variables and takes out 20,000 rows of URL from each category. Each of these 12 test variables are concatenated together into another test variable, this set contains URL for testing and determining the accuracy and performance of the model in the later stages. The same 20,000 rows of URL from each category is dropped from the main dataframe variable to subject it to model training. After obtaining the train and test set, the variable stemmed_count_vect helps in eliminating commonly used words in the English language from the URL string since they are irrelevant and have no effect in determining the category for the URL. A pipeline method is used which combines the stemmed_count_vectorizer, TfidfTransformer to transform URL string into sparse matrices and the sklearn MultinomialNB function to fit the train set. Upon doing these operations finally the classifier model is generated and the same model is used in our work to predict the category of website using URL.

Algorithm 5 Multinomial Naive Bayes

```

1: dataset ← categorical dataset
2: add 2000 rows of URL for each category into a variable
3: test_data ← concatenate each category
4: drop 2000 rows of URL for each category from dataset
5: xTr ← data['URL']
6: yTr ← data['Category']
7: xTe ← test_data['URL']
8: yTe ← test_data['Category']
9: stemmed_count_vect ←
10: CountVectorizer(stop_words = 'english')
11: nb_clf ← Pipeline(['vect', 'tfidf', 'clf'])
12: 'vect' ← stemmed_count_vect
13: 'tfidf' ← TfidfTransformer()
14: 'clf' ← MultinomialNB(False, 0.0001)
15: nb_clf.fit(xTr, yTr)
16: nb_clf.predict(xTe)

```

3.4 The Intrusion Detection System

IDS is a software or a physical system that protects the client and server applications from any type of attacks at the application or network layer. There are

various types of IDS, Host based, Signature based, Anomaly based. This paper focuses on making an IDS with ML that can detect traffic of known signatures and also learn with different signatures of the traffic. The dataset selected is the CSE CIC dataset [6], a popular dataset for IDS with around 7 application layer attacks.

Algorithm 6 is about the SVM model. We have chosen the CSE CIC dataset [6] and have performed the preprocessing of the dataset. Python's sklearn module has the Support Vector Classifier(SVC()) function which can be fit to the trained dataset. We have chosen our test size as 0.2 for the splitting of the dataset into training and testing datasets with `train_test_split()` before the fit function is applied. There are 5 SVM models that have been trained with 78 attributes of the dataset and 2 more SVM models trained with 5 attributes. These SVM models are deployed with joblib module as files.

Algorithm 6 SVM

```

1: data  $\leftarrow$  CSE CIC dataset
2: drop Timestamp column from data
3: drop FlowDuration column from data
4: drop NA values from data
5: X  $\leftarrow$  data.iloc[:, :-1].values
6: y  $\leftarrow$  data.iloc[:, -1].values
7: xTr,xTe,yTr,yTe  $\leftarrow$ 
8: trainTestSplit(X,y, testsize=.2)
9: xTr  $\leftarrow$  fit.transform(xTr)
10: xTe  $\leftarrow$  transform(xTe)
11: classifier  $\leftarrow$  SVC(linear, 0)
12: classifier.fit(xTr,yTr)
13: classifier.predict(xTe)

```

3.5 Advantages

This paper aims to provide a real time detection of malicious websites and also find the categories of the websites, identify the type of application layer attack the traffic is prone to. By doing so, our paper aims to highlight the use cases of safer browsing and protecting the client and servers from application layer attacks.

4 Result

Through our paper we integrated the functionalities of two models to bring in the use cases of safer browsing and protecting the client and server systems from application layer attacks. Through pyshark we tried making the detection of malicious websites real time by capturing traffic pertaining to the browsing

activity of the user. Since URL classification deals with text analysis of the URL string, Naive Bayes was incorporated after deciding amongst several other ML models such as Random Forest, SVM etc. and it gave the best results in accuracy and time taken for training. Logistic Regression is used to detect whether a website is malicious or not. 5 SVM models were trained for identifying 7 application layer attacks. These SVM model's were fit with 70 attributes of the CSE CIC dataset [6]. Currently these models are restricted to static testing. In Fig. 10, we have shown the accuracy of various models that are mentioned in our paper. Additionally we have 2 SVM models namely, SVM-DDoS and SVM_BruteForce, which are fit with 5 attributes and for real time testing of presence of any application layer attacks on the visited websites. Pickle and joblib files were generated to execute the complete software quicker.

Module	Metrics		
	Train Dataset Accuracy	Test Dataset Accuracy	Dataset Size
Logistic Regression	97.09%	95.89%	420,464
Naïve Bayes	93.47%	87.51%	1,048,596
SVM – DoS	92%	41%	1,048,574
SVM -- DDoS	95%	85%	1,048,576
SVM – Brute Force	99.99%	90%	1,046,298
SVM – Web Application	99%	92%	1,048,574
SVM – Botnets	97%	73%	1,048,573

Fig. 10. Accuracies of all models.

```

[[129411      1  4263]
 [     0  38422      0]
 [  4401      8 33209]]
               precision    recall   f1-score   support
Benign          0.97      0.97      0.97    133675
FTP-BruteForce  1.00      1.00      1.00     38422
SSH-Bruteforce  0.89      0.88      0.88     37618

accuracy           0.96      0.96      0.96    209715
macro avg       0.95      0.95      0.95    209715
weighted avg     0.96      0.96      0.96    209715

svm train accuracy = 0.9580061035214458
svm test accuracy = 0.9586438738287676

```

Fig. 11. SVM bruteforce accuracy and confusion matrix.

Figure 11 shows the confusion matrix of the SVM model for FTP-Bruteforce and SSH-Bruteforce. The accuracy of the model is 95.86% on the testing data. Our model identifies the type of bruteforce Attack. The authors [2] Ahmed, A.A.; Agunsoye, G made use of ANN which gave a better accuracy but it required more computational power and also took time in predicting whether the traffic is malicious or benign, and not specifically type of application layer attacks. Also the paper just focuses on monitoring network traffic but we aim to also identify what kind of traffic is being sent or received.

	precision	recall	f1-score	support
Benign	0.96	0.99	0.98	199076
DoS attacks-GoldenEye	0.52	0.31	0.39	8382
DoS attacks-Slowloris	0.09	0.01	0.03	2257
accuracy			0.95	209715
macro avg	0.53	0.44	0.46	209715
weighted avg	0.94	0.95	0.94	209715

Fig. 12. Confusion matrix SVMDDoS.

Figure 12 shows the confusion matrix of the SVM model for DoS-Slowloris and Hulk. The accuracy of the model is 95% on the testing data. The accuracy of our model is comparable to the accuracy that was obtained in the paper by Joshi [1]. Moreover our model identifies the type of DoS-Slowloris or Hulk, which was not mentioned in their paper. Through the usage of CSE CIC dataset [6], we have expanded our paper's scope to identify the seven application layer attacks which is not available in the KDD cup restricting their IDS to detect only 4 types of malicious traffic.

Figure 13 summarises the TP, FP, TN, FN of the Logistic Regression model. The model obtained an accuracy of 95.89% using the dataset [9] without using hyperparameter tuning. This accuracy was slightly more than the Logistic Regression implementation in the paper by Chiyu Ding that obtained 92.35% without using hyperparameter tuning [4]. The accuracy was also higher than the implementation in the paper by H. M. Junaid Khan et al. that obtained 94.04% with UNB Dataset and 87.21% with Kaggle Dataset [5].

Figure 14 shows the report obtained on the Multinomial Naive Bayes model that was trained using the dataset [7]. An accuracy of 87.51% was obtained on the testing data. This accuracy obtained was higher compared to the MultinomialNB implementation in the paper by Chiyu Ding that obtained an accuracy of 86.9.

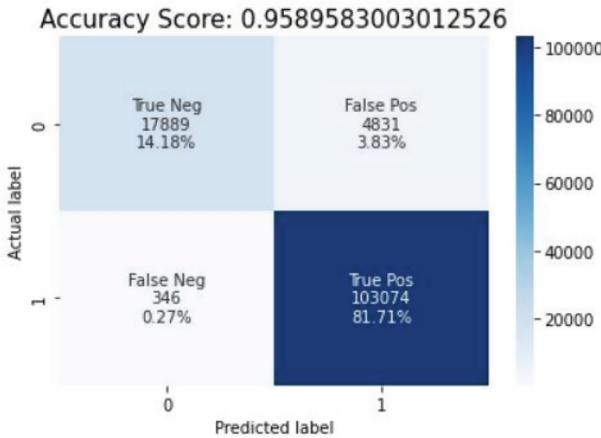


Fig. 13. Logistic Regression Confusion matrix.

	precision	recall	f1-score	support
Adult	0.893	0.229	0.364	2000
Arts	0.705	0.857	0.773	2000
Business	0.807	0.945	0.871	2000
Computers	0.905	0.911	0.908	2000
Games	0.930	0.956	0.943	2000
Health	0.948	0.976	0.962	2000
Home	0.909	0.953	0.930	2000
Kids	0.864	0.849	0.856	2000
News	0.969	0.967	0.968	2000
Recreation	0.835	0.968	0.897	2000
Reference	0.846	0.955	0.897	2000
Science	0.959	0.937	0.948	2000
accuracy			0.875	24000
macro avg	0.881	0.875	0.860	24000
weighted avg	0.881	0.875	0.860	24000

```
Naive Bayes Train Accuracy = 0.9347938114144502
Naive Bayes Test Accuracy = 0.875125
```

Fig. 14. M-Naive Bayes classification report.

5 Conclusion

We have come up with a paper which combines two separate functionalities—The URL classification and categorization and The Intrusion Detection System. It is a paper where functionalities of two different models have been combined to obtain different use cases. In this paper we touched on two use cases namely, protecting the client and server system on the network from application layer attacks and also to provide safe browsing through filtering or classification of

websites. The work carried out by us mainly focuses on seven application attacks available on the dataset—CSE CIS [6]. So there could be further improvement in the IDS by extending it to the other Open System Interconnection (OSI) or Transport Control Protocol (TCP) layers. In addition we have identified 12 categories of the websites, so there is scope of increasing it and also the phishing keyword count can also be increased to get better outputs. Websites are now being hosted on cloud, so this is a challenge for both the modules as the website has the cloud name in its URL making it difficult to process and get the results for the websites. An alternative to this would be to design a browser extension and use full path URLs instead.

References

1. Joshi, M.R.: A Review of Network Traffic Analysis and Prediction Techniques (2016)
2. Ahmed, A.A., Agunsoye, G.: A Real-Time Network Traffic Classifier for Online Applications Using Machine Learning. *Algorithms* **14**, 250 (2021)
3. Manjeri, A.S., Kaushik, R., Ajay, M.N.V., Nair, P.C.: A machine learning approach for detecting malicious websites using URL features. In: 2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA) (2019)
4. Ding, C.: Automatic Detection of Malicious URLs using Fine-Tuned Classification Model (2020)
5. Junaid Khan, H.M., Niyaz, Q., Devabhaktuni, V.K., Guo, S., Shaikh, U.: Identifying generic features for malicious URL detection system. In: 2019 IEEE 10th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON) (2019)
6. <https://www.kaggle.com/datasets/solarmainframe/ids-intrusion-csv>
7. <https://www.kaggle.com/datasets/shaurov/website-classification-using-url>
8. Ding, C.: Automatic Detection of Malicious URLs using Fine-Tuned Classification Model (2020)
9. <https://www.kaggle.com/datasets/antonyj453/urldataset>



NP4G: Network Programming for Generalization

Shoichiro Hara^(✉) and Yuji Watanabe

Nagoya City University, Nagoya, Japan

s.hara@nsc.nagoya-cu.ac.jp, yuji@nsc.nagoya-cu.ac.jp

Abstract. In recent years, the development of Artificial Intelligence systems using neural network has been remarkable. However, this method has low explainability and is illogical. To solve this, there is an automatic programming method based on inductive inference. However, this method has also the problem of low versatility. In this study, we propose NP4G: Network Programming for Generalization, which can automatically generate programs by inductive inference. Because the proposed method can realize “sequence”, “selection”, and “iteration” in programming and can satisfy the conditions of the structured program theorem, it is expected that NP4G is a method that automatically acquires any programs by inductive inference. As an example, we automatically construct a bitwise NOT operation program from several training data by generalization using NP4G. Although NP4G only randomly selects and connects nodes, by adjusting the number of nodes and the number of phase of “Phased Learning”, we show the bitwise NOT operation programs are acquired in a comparatively short time and at a rate of about 7 in 10 running. The source code of NP4G is available on GitHub as a public repository.

Keywords: Inductive Inference · Automatic Programming · Knowledge Acquisition · Genetic Programming · Genetic Network Programming

1 Introduction

Along with the development of technology, artificial intelligence (AI) systems has evolved rapidly and is now widely used in all fields. Recently, the performance of AI has greatly improved due to the development of a technology called deep learning based on neural networks. While this method can acquire advanced recognition ability by learning a large amount of data, the entity of the model is nothing more than a computational model that obtains results by performing calculation processing. So, this method has low explainability. In addition, there is no guarantee that a model that fully matches the data used for training can be created. In order to solve this problem, it is necessary to create a model that generalizes to a concept that encompasses all of individual training data. This can

be rephrased as an automatic programming method based on logical inference. However, Automatic Programming methods based on inductive inference still have problems due to their low versatility.

In order to realize automatic programming based on this logical inference, it is necessary to extract the structure of some kind of solutions to the problem, so that this field is closely related to the knowledge acquisition of artificial intelligence. In the study on the knowledge acquisition of the artificial intelligence, there is still a big problem that knowledge including a lot contradiction and/or an exception is hard to formulate. Especially, the inductive inference generalized by logical inference from one example is an important problem that the artificial intelligence can acquire knowledge by itself.

In this study, we propose NP4G: Network Programming for Generalization, which can automatically generate programs by inductive inference. Because the proposed method can realize “sequence”, “selection”, and “iteration” in programming and can satisfy the conditions of the structured program theorem, it is expected that NP4G is a method which automatically acquires any programs by inductive inference. As an example, we automatically construct a bitwise NOT operation program from several training data by generalization using NP4G. Although NP4G only randomly selects and connects nodes, by adjusting the number of nodes and the number of phase of “Phased Learning”, we show the bitwise NOT operation programs are acquired in a comparatively short time and at a rate of about 7 in 10 running.

Section 2 explains automatic programming, inductive inference, and genetic programming as related research. Section 3 proposes network programming for generalization (NP4G) and explains its basic structure. In Sect. 4, as an example of the proposed method, we describe how to acquire a bit NOT operation program, and in Sect. 5, we show and discuss the verification results. Section 6 describes the significance of this study and future issues. The source code of NP4G is available on GitHub as a public repository.¹

2 Related Research

2.1 Automatic Programming

Automatic programming (AP) is the automation of all or part of the generation of programs, and achieves measurable success as an aid to developers of large systems and small programs [3]. Among automatic programming, a model that automatically generates programs by logical inference has been researched for a long time. Logic Theorist, the world’s first artificial intelligence program published in 1956, was designed to imitate human logical inference using search trees and heuristics [6].

As research on automatic programming has been attracting attention in recent years, there are methods using a large-scale language model based on a neural network, represented by GPT3 [4] of OpenAI. The methods predict

¹ <https://github.com/Amplil/np4g>.

and automatically generate code according to the situation by learning as a language model by deep learning from a huge amount of publicly available code. The methods have been very successful in applications such as predicting the code that the developer is going to write on the editor and suggesting the continuation of the code [1]. However, the methods do not generate code by logical inference because they are language generation models and are inference based on illogical experience with huge amount of learning. In this respect, many problems still remain today.

Recently, some researches have been achieved in the generation of automatic programming by logical inference [16], however there remains a problem in terms of acquiring arbitrary programs since the methods are effective only for domain-specific languages such as SQL.

2.2 Inductive Inference

In a wide sense, logical inference consists of deductive inference, inductive inference, and analogical inference [14]. Representative examples of logical thinking include “inductive thinking”, “analogous thinking”, “thinking of generalization”, and “symbolic thinking” [15]. Of these, “thinking of generalization” corresponds to inductive inference. Inductive inference is an inference that derives general rules to explain given data [2]. Inductive inference by artificial intelligence has been studied for a long time [5, 13], but its scope of application is limited and problems remain in acquiring arbitrary programs. In addition, since inductive inference by artificial intelligence can be said to be a method of automatically acquiring knowledge, it is closely related to research on knowledge acquisition. Much knowledge cannot be formulated, and even if formulated, there are contradictions between rules. Such problem is called knowledge acquisition problem [12, 17], and it is still not resolved.

From the point of view of inference with generalization, neural network methods are also within the scope of the field because they can pick out common features of individual data. However, it is not logical because the correlation is obtained based on a huge amount of data. In addition, the decision process is poorly explained, and this problem is called the black-box problem [7]. And it requires a large amount of data for learning, so it is only an inference that generalizes, but not an inductive inference.

2.3 Genetic Programming

Genetic programming (GP) is a method for automatically generating tree-structured programs by using genetic manipulations [10]. It is used to solve various problems such as the automatic construction of formulas and the generation of agent action sequences. GP handles only tree structures, but genetic network programming (GNP) is an extension of GP to networks [8]. GP/GNP regards each node in the network as the minimum unit for simple processing, and automatically changes the way they are combined to build more optimal programs. Each node can be classified into a decision node, a processing node,

and a start node. GP/GNP can automatically acquire a program represented by a network, but there is no example of its application to inductive inference.



Fig. 1. Generalization from training data.

3 Proposed Method

3.1 Basic Concept of NP4G

NP4G is a method of performing inductive inference by automatically generating a program represented by a network based on training data. For example, in Fig. 1, a bitwise NOT operation program is obtained by generalization from four training data. The training data is one-input, one-output data, and the search is performed until a program is generated that can obtain inputs and outputs that match the training data. The generated program is obtained by connecting multiple nodes with simple functions in a network, similar to the concept of GNP. GNP is network programming using genetic methods, but NP4G is a method that assumes the application of various methods, not limited to the use of genetic methods.

NP4G can be said to be a method of knowledge acquisition by inductive inference in that it acquires one piece of knowledge (program) by generalizing each case (training data). Therefore, unlike methods using neural networks, the number of required training data is very small, only a few that can grasp the characteristics of the target to be generalized. In addition, since the constructed program itself is in the form of a network, it is clear what kind of processing is being done internally.

Attempts to automatically generate programs by combining networks have so far been used only in the genetic method of GNP. However, the proposed method uses network programming as a means of generalization. Network programming, which is not limited to this genetic method, is expected to be widely applied in new fields in the future as a new automatic programming method. In the future, similar to GP/GNP, it is expected to be extended to effective algorithms by combining with other methods such as neural networks and reinforcement learning.

3.2 Structure Theorem

There is a problem that the method that can make logical inference has a limited application. However, NP4G has a structure that satisfies the conditions of

the structure theorem in order to obtain an arbitrary program. Structure theorem states that any program can be constructed by combining three types of basic structures: “sequence”, “selection”, and “iteration” [11]. As explained in Sect. 3.3, NP4G can achieve “sequence” by executing nodes in a predetermined order from the start node, “selection” by providing decision nodes and processing nodes as preliminarily provided functions, and “iteration” by iterable data. So it can be said that the condition of the structure theorem is satisfied.

3.3 Basic Structure of NP4G

NP4G has a directed graph structure, such as Fig. 2, in which nodes such as multiple functions and objects are connected in a network. Nodes are functions (squares in the figure), a start node (a circle marked with “S” in the figure) with input data that is executed at the beginning of the program, an end node (a circle marked with “E” in the figure) that signal the end of the program, and object nodes (circles in the figure) that output pre-stored data without an input link. The number of links connected as inputs to a node varies depending on the function, but any number of links can be connected as outputs regardless of the function.

Execution Order of Nodes Using an example of a network generated by NP4G shown in Fig. 2, we explain the execution order of nodes. By processing the nodes in order from the start node, the “sequence” of the structure theorem can be realized.

First, the nodes connected to the start node with input data are executed in order from the top, starting with ①, ②, ③. When all the nodes are executed, the nodes connected to them are also executed in order from the top, ④, ⑤, ⑥. When the network is regarded as a data series, this operation to be executed in order from the top means the operation to arrange the nodes to be executed next as a list and execute them in order from the front.

Here, like ④ in Fig. 2, if there is no output from other input nodes, the execution result will be “not yet” and no output will occur. This means that if there are nodes that do not derive from the start node, such as ⑨, they will not be the output. On the other hand, in the case of ⑥, ⑤ was executed immediately before, and all the outputs of the input nodes exist, so they are output. Next ⑦, ⑧, and ⑨ are executed, but in the case of ⑧, this node has already been output in ⑥, the execution result will be “already done” and will not be output again. Each node is executed in this way until there are no more nodes to process next.

Finally, the last output node is responsible for the output of the entire network and is connected to the end node. In other words, the end nodes are not connected to the network from the beginning, but are determined after the fact from the relation of the execution order of the nodes.

Iteration In order to satisfy the structure theorem, it is necessary to introduce “iteration”. In NP4G, iterative processing is realized not by providing a feedback

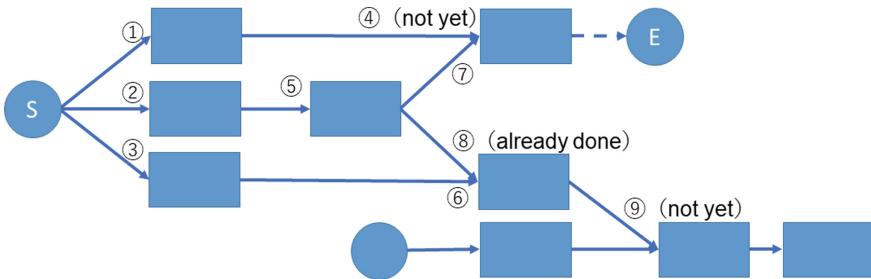


Fig. 2. Basic structure of NP4G and execution order of nodes.

loop on the network, but by inputting iterable data to a function. A function given an iterable data as input does the same process for each element of the data. A function also creates iterable data or converts it back to non-iterable data, and the process is repeated as long as iterable data is used as the function input. By doing so, iteration can be achieved without creating a program that gets stuck in an infinite loop and never terminates. In addition, even when randomly combining nodes and automatically generating a program, the simplicity of the node execution order makes it less likely that errors occur.

Automatically Defined Functions (ADFs) GP uses Automatically Defined Functions (ADFs) because they can be expected to be faster as programs evolve [9]. ADFs make it possible to create more advanced programs in a short period of time by reusing networks that have already been created. NP4G combines ADFs with phased learning described in Sect. 3.3, registering networks as ADFs at each phase and allowing the networks to be reused in the next phase.

Phased Learning Phased learning is a method of learning even a complicated program without difficulty by dividing learning into each phase. Phased learning is a method mainly used in reinforcement learning, and it has the effect of preventing learning time from becoming enormous by lowering the degree of freedom that learning can take.

NP4G starts by reducing the number of training data to be learned and generating a network that performs simple processing. Then, the generated networks are reused as ADFs when constructing the next network. By doing so, we can expect to learn in a shorter time than when obtaining a complicated network from the beginning.

4 Acquisition of a Bitwise NOT Operation Program

Using NP4G, we consider the case of automatically constructing a bitwise NOT operation program, and verify it by actually executing the NP4G program. In this study, we use the programming language Python (Python 3.7.12) to create

a NP4G program. We use Google’s Colaboratory as the execution environment. In addition, list-type objects (hereafter referred to as lists) are used in Python programs as iterable data for realizing iterative processing. In NP4G, all data stored in training data, start nodes, and object nodes are character strings.

4.1 Preliminarily Provided Functions

As a function to construct the network, four nodes are given in advance: split function, sum function, equal function, and control gate function, as shown in Fig. 3. The contents of each function are explained below.

Split Function Split function separates strings with spaces as delimiters and write them as a list. In figures, “split” is enclosed in a square.

Sum Function Sum function smoothes multiple inputs, such as list-type inputs and other character strings, and combines them with a space between them to make one character string. If the input string is “[NULL]”, then “[NULL]” is not concatenated. If the output string is “(empty)”, then “[NULL]” is outputted. In figures, “+” is enclosed in a square.

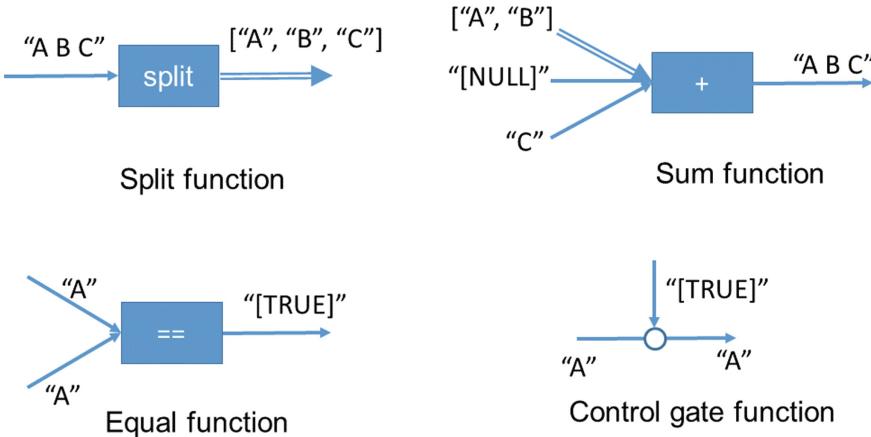
Equal Function Equal function outputs “[TRUE]” when the values of the two inputs match. Otherwise, it outputs “[FALSE]”. It plays the role of decision node in GP/GNP. In figures, “==” is enclosed in a square.

Control Gate Function If one of the two inputs is “[TRUE]”, control gate function passes the value of the other input, otherwise outputs “[NULL]”. It plays the role of a processing node in GP/GNP. In figures, it is indicated by white circles.

4.2 Example of a Bitwise NOT Operation Program

Figure 4 shows an example of a bitwise NOT operation program realized by using these four functions. As the first phase, NP4G constructs the network in the upper left of the figure for 1 bit input. The equal function whose input is an object node containing “0” and a start node containing input “0” outputs “[TRUE]”. The control gate function with “[TRUE]” as input outputs the value of the object node storing “1” as it is. In this way, the “selection” of the structure theorem is realized by the equal function and the control gate function. If the input is “1”, another equal function and a control gate function that output “[FALSE]” and “[NULL]” for input “0” output “[TRUE]” and “0” respectively. In this way, a 1 bit logical NOT is constructed as a network.

Next, as shown in the upper right of the figure, this network is made into a form that can be reused for building the next network as ADFs. In the figure, “not” is indicated by enclosing it in a square. Then, in the next phase where

**Fig. 3.** Function to be given beforehand.

multiple bits are input, NP4G constructs a bitwise NOT operation program by using the iterative processing of iterable data (list) and the ADFs “not”. Multi-bit input strings are made into a list by the split function, the ADFs “not” are repeatedly applied to each element, and the disjoint strings are combined by the sum function to realize multi-bit logical not. By phased learning that builds a network phase by phase in this way, the target program is realized.

4.3 Verification Method

We verify whether a bitwise NOT operation program can be automatically acquired by NP4G simply by randomly selecting and connecting nodes for several training data.

Training Data and Random Search The training data used are only four shown in Table 1, and the training data used at each phase changes depending on the method of phased learning. Each input and output of the training data is written in parentheses such as (input, output).

As a bitwise NOT operation program is automatically generated by NP4G, a network that matches these training data at each phase is searched by randomly combining nodes. The random algorithm first places a start node and an end node, selects nodes at random for the determined number of nodes, and then connects them with other nodes at random for the number of inputs of the function.

Method of the Phased Learning In this study, we prepare the phases used in phased learning from phase 1 to phase 5, and examine the effect of phased learning by changing the combination of each phase. The training data for the five phases used in phased learning are as follows.

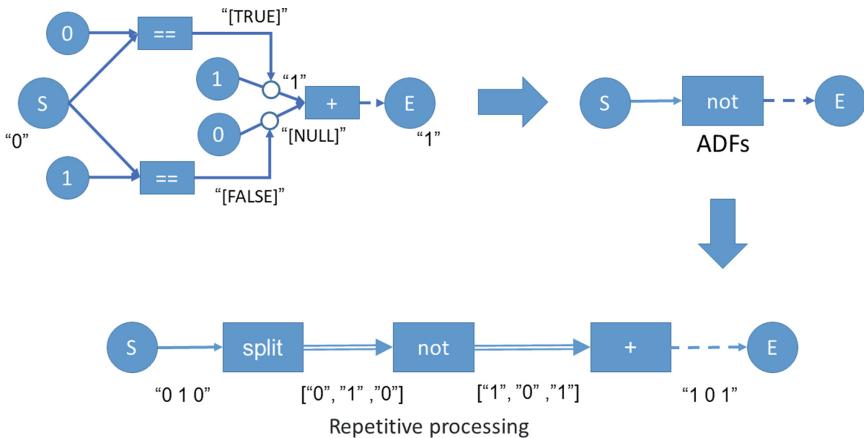


Fig. 4. Implementation of a Bit NOT operation program by NP4G.

Table 1. All training data to use in this study.

(input, output)
("0" , "1")
("1" , "0")
("0 0" , "1 1")
("0 1 0" , "1 0 1")

- Phase 1: ("0", "1")
- Phase 2: ("1", "0")
- Phase 3: ("0", "1"), ("1", "0")
- Phase 4: ("1", "0"), ("0 0", "1 1")
- Phase 5: ("1", "0"), ("0 0", "1 1") ("0 1 0", "1 0 1")

Then each of these phases is compared by changing the number of phases like follows.

- 3 phases: Phase 2, 3, 4
- 4 phases: Phase 1, 2, 3, 4
- 5 phases: Phase 1, 2, 3, 4, 5

The networks obtained at each phase can be reused in subsequent phases by adding them to a list as ADFs using Python lambda expressions. A network that matches the training data at each phase is searched by randomly combining preliminarily provided functions, object nodes, and ADFs. Note that the object node also needs to be prepared in advance in the same way as preliminarily provided functions. In the case of object nodes, both input and output character strings of training data used in learning at each phase are prepared as object nodes.

Verification In the verification, the NP4G program is executed 10 times, and the average values of the generation results and execution time are obtained. The generated result is “success” if the network obtained by searching is a generalized network that can obtain the expected output, and “failure” otherwise. If it is not generated within the time limit for one program execution, it is considered as “exceeded”. To check whether the output is the expected one, enter all binary strings from 1 to 5 digits (“0” to “1 1 1 1 1”) as verification data. Also, the time limit for one execution is 3 h (10,800 s).

5 Results and Discussion

Table 2 shows the execution results of the NP4G program when the number of phases and the number of nodes are changed. The execution results are displayed with the numbers of “success”, “failure”, and “exceeded”. Table 3 shows the average, maximum, and minimum execution times (s) when the number of phases and the number of nodes are changed. From Tables 2 and 3, when there are four phases and 20 nodes, the number of successes is as high as 7 and the average time is relatively short as 1,105 s. In the case of four phases and 15 nodes, the number of successes is six, but the average time is the shortest at 334 s. These are the best results.

As shown in Table 2, there is only one failed network when there are four phases and 10 nodes. Also, from the results when the number of nodes is five, all of them exceed the time except for one successful case of five phases. This is probably because the number of nodes is too small to generate a bitwise NOT operation program with five nodes. On the other hand, when the number of nodes is large, such as 20, the number of time overruns tends to increase, which is thought to be due to the time required to generate the network as the number of nodes increases. Next, looking at the average time for each number of phases in Table 3, we can see that the average time for three phases tends to be long. The reason for this is thought to be that the number of phases to be taken is small in three phases, and it becomes necessary to create a complicated network in one phase. Also, the reason why the average time tends to be longer for five phases than for four phases is that if the number of phases is increased too much, it will take extra time.

Figure 5 shows a five-nodes network with five phases that is the only successful automatic generation. It is shown that the expected network can be obtained even if the number of nodes is small like this network. After automatically generating a network with an output of 1 for an input of 0 in Phase 1, it is used as adf1 in Phase 2, and then the network created in Phase 2 is used as adf2 in Phase 3. In Phase 4, we can see that adf3 is used to create a network similar to the example implementation of a bitwise NOT operation program in Fig. 4. In the case of the network, we confirm that a bitwise NOT operation program is obtained in Phase 4, and the expected network is obtained even in the middle of phased learning.

Next, Fig. 6 shows the network when automatic generation with 10 nodes in four phases fails. In this way, because the network becomes complicated, it

is thought that although there are outputs that match all the training data, there are outputs that do not match the verification data. In the case of this network, an input of “0 1 0” results in “0 1 1” and an output that is not “1 0 1”. Even in this case, it is considered that the expected network can be obtained by increasing the number of phases and learning with training data including (“0 1 0”, “1 0 1”). Also, from Phase 2 and Phase 4 in Fig. 6, we confirm that the ADFs adf_1 and adf_3 generated in the previous phase are not always used because the nodes are randomly selected when constructing the network. As we can see from the networks obtained in the actual verification of Figs. 5 and 6, NP4G has a clear thinking process, unlike methods using neural networks.

Table 4 shows the average, maximum, and minimum generation times (s) at each phase. This table is compiled from all successful results regardless of the number of nodes and phases. From this table, it can be seen that the generation times in Phase 1 and 2 are shorter than the other phases. This is because both Phase 1 and 2 are learning with only one set of training data, so it is easy to find a network that matches the training data even if the network is generated randomly. Next, it can be seen that the average generation time in Phase 3 is the longest at 2585.24 s, followed by Phase 4, and then Phase 5. We consider these factors as follows: First, Phase 3 is the learning of training data (“0” “1”) and (“1”, “0”), and for the first time with one network, it is necessary to search for a network that satisfies two conditions. Therefore, it is considered that it is necessary to generate a complicated network. In Phase 4, a complicated network is not required, but the generation time is considered to be longer because it is necessary to generate a network that required iteration processing for the first time in Phase 4. In the case of Phase 5 after getting the iterative process, even if the number of digits increases, the same network can be used, so it is thought that the generation time can be shortened.

Table 2. Execution result (Success/Failure/Exceeded).

Number of nodes	5	10	15	20
3 phases	0 / 0 / 10	8 / 0 / 2	6 / 0 / 4	2 / 0 / 8
4 phases	0 / 0 / 10	7 / 1 / 2	6 / 0 / 4	7 / 0 / 3
5 phases	1 / 0 / 9	6 / 0 / 4	7 / 0 / 3	5 / 0 / 5

6 Conclusion

6.1 Summary and Significance of this Study

In this study, we proposed Network Programming for Generalization (NP4G). We confirmed that NP4G could acquire a bitwise NOT operation program from several training data just by selecting and connecting nodes at random. NP4G is

Table 3. Execution times (s).

Number of nodes		5	10	15	20
3 phases	Mean	—	3067	5565	3699
	Max	—	10480	8966	9543
	Min	—	258	1	3
4 phases	Mean	—	3024	334	1105
	Max	—	8294	1103	2835
	Min	—	134	7	10
5 phases	Mean	32	4090	1506	1266
	Max	32	9103	7866	2586
	Min	32	1	2	26

Table 4. Generation times (s) in each phase.

Phase	1	2	3	4	5
Mean	1.17	2.24	2585.24	188.31	12.49
Max	6.94	24.20	10425.58	1114.09	347.02
Min	0.00	0.00	0.53	3.95	0.03

an inductive inference because it is a method of finding general properties from several examples. In addition, it has a structure that satisfies the conditions of the structure theorem of “sequence”, “selection”, and “iteration” for programming, and it is a method that can be expected to automatically acquire arbitrary programs by inductive inference. Network programming, not limited to genetic methods, demonstrated by NP4G is expected to be widely applied in new fields as a new automatic programming method. The significance of this study is that it shows the expected realization of versatile and flexible artificial intelligence which can learn by itself from all kinds of knowledge through network programming, and make logical inferences and answer to other problems by making use of that content.

6.2 Future Issues

Turing Completeness We explained that NP4G satisfies the conditions of the structure theorem, but in order to prove that NP4G is a method that can obtain arbitrary programs, it is necessary to show that NP4G is Turing complete. Turing completeness means that a computational model has computational power equivalent to that of a universal Turing machine, that is, it can reproduce arbitrary programs. If it can be shown to be Turing completeness, it means that we have realized a method that can acquire arbitrary programs by inductive inference. In addition to a bitwise NOT operation program, it is necessary to actually try whether other programs can be built automatically using NP4G.

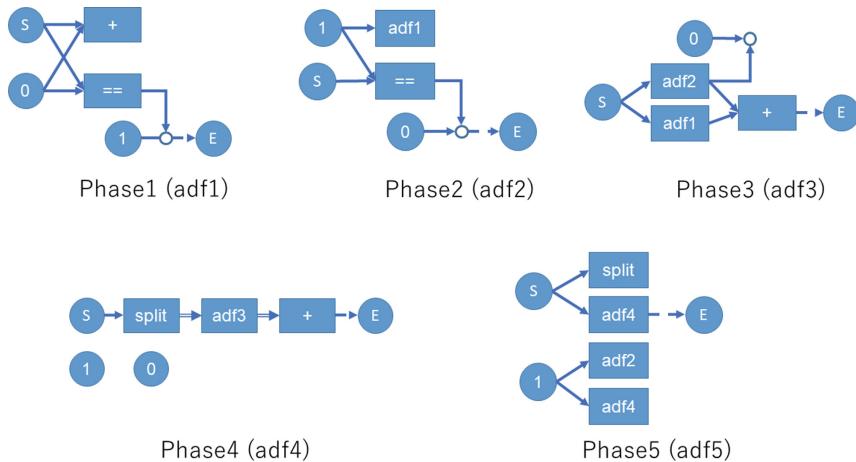


Fig. 5. Network when the expected program is obtained by automatic generation at 5 phases with 5 nodes.

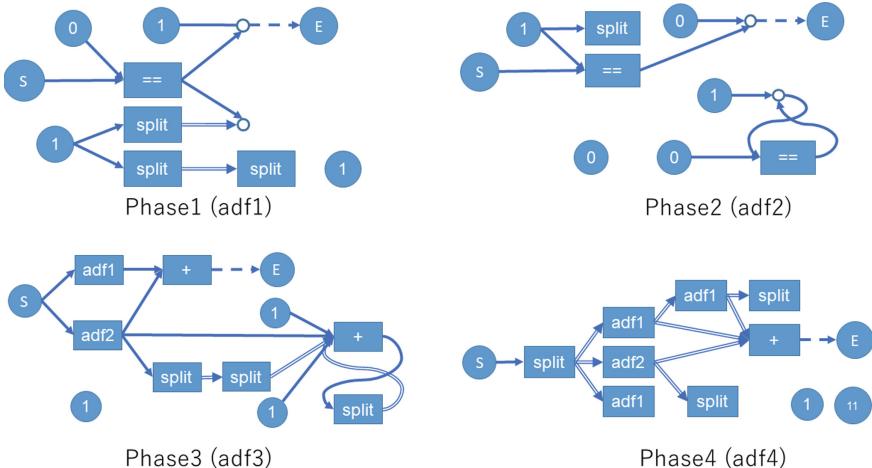


Fig. 6. A network obtained by automatic generation with 10 nodes in 4 phases, where the training data used for learning matches the input and output, but a program different from expectations is obtained.

Exploring Search Methods In this research, we used accidental search by random generation as a network search method. Since this method is primitive, the more types and numbers of nodes used, and the more complex the target program, the more time it takes to search. In the future, by combining with other learning methods such as using reinforcement learning, it is necessary to automatically adjust the types and numbers of nodes suitable for network

components and how to combine them in order to improve the efficiency of network construction.

Evaluation of the Network Other learning methods, such as genetic methods and reinforcement learning, have a mechanism to numerically evaluate generated networks and models using evaluation functions. In this research, it is an evaluation whether or not the target network is generated, and a numerical evaluation by a evaluation function has not yet been performed. In the future, if we can establish a network evaluation method in NP4G, we can change the construction method by evaluating the network. In addition, it becomes easy to combine with other learning methods that use evaluation functions.

Simplification of Generated Networks As explained in Sect. 3.3, in a network generated by NP4G, if a node is not derived from the start node, it is not executed. If there is an algorithm to remove such nodes after generating the network, the generated network can be simplified.

Acknowledgments. This research was supported by JST SPRING, Grant Number JPMJSP2130.

References

1. GitHub Copilot. <https://copilot.github.com/>
2. Arikawa, S.: Program synthesis by inductive inference and analogical reasoning. *Artif. Intell.* **2**(3), 299–306 (1987)
3. Barr, A., Feigenbaum, E.A.: Chapter x—automatic programming. In: *The Handbook of Artificial Intelligence*, pp. 295–379. Butterworth-Heinemann (1982)
4. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. *CoRR* (2020). [ArXiv:abs/2005.14165](https://arxiv.org/abs/2005.14165)
5. Case, J., Smith, C.: Comparison of identification criteria for machine inductive inference. *Theor. Comput. Sci.* **25**(2), 193–220 (1983)
6. Gugerty, L.: Newell and simon’s logic theorist: Historical background and impact on cognitive modeling. *Proc. Hum. Factors Ergon. Soc. Annu. Meet.* **50**(9), 880–884 (2006)
7. Hussain, J.: Deep learning black box problem. Master’s thesis, Uppsala University, Department of Informatics and Media (2019)
8. Katagiri, H., Hirasawa, K., Hu, J.: Genetic network programming—application to intelligent agents. *Proc. IEEE Int. Conf. Syst., Man Cybern.* **5**, 3829–3834 (2000); 2000 IEEE International Conference on Systems, Man and Cybernetics; Conference
9. Koza, J.R.: *Genetic Programming II : Automatic Discovery of Reusable Programs*. MIT Press (1992)
10. Koza, J.R.: Genetic programming as a means for programming computers by natural selection. *Stat. Comput.* **4**(2) (1994)

11. Linger, R.C., Witt, B.I., Mills, H.D.: Structured Programming; Theory and Practice the Systems Programming Series, 1st edn. Addison-Wesley Longman Publishing Co., Inc, USA (1979)
12. Liou, Y.I.: Knowledge acquisition: issues, techniques, and methodology. In: SIGBDP '90, (1990)
13. Michalski, R.S.: Pattern recognition as rule-guided inductive inference. IEEE Trans. Pattern Anal. Mach. Intell. PAMI **2**(4), 349–361 (1980)
14. Nakahara, T.: Basic knowledge of arithmetic course in mathematics important term 300 (in Japanese). Meijitoshoshuppan (2000)
15. Saito, N.: Logical thinking based on mathematical views and ways of thinking (in Japanese). Res. Educ. Pract. Dev., Gunma Univ. **28**, 31–37 (2011)
16. Takenouchi, K., Ishio, T., Okada, J., Sakata, Y.: PATSQL: efficient synthesis of SQL queries from example tables with quick inference of projected columns. CoRR (2020). [ArXiv:abs/2010.05807](https://arxiv.org/abs/2010.05807)
17. Wagner, W.P.: Issues in knowledge acquisition. In: Proceedings of the 1990 ACM SIGBDP Conference on Trends and Directions in Expert Systems, SIGBDP '90, pp. 247–261. Association for Computing Machinery, New York, NY, USA (1990)



Blockchain and AI for Optimizing Bank Data Security

Ibrahima Souare¹(✉) and Khadidiatou Wane Keita²

¹ University Cheikh Anta Diop of Dakar, Dakar, Senegal

ibrahima4.souare@ucad.edu.sn

² Polytechnic Institute, Dakar, Senegal

wane.keita@esp.sn

Abstract. Online banking has become one of the most important applications on the Internet, being provided by most banks around the world. Banking payment systems are based on the progress of technology. Technology is a key driver for banks. Despite this technological progress, bank data remains the real target of cyber attacks. With the use of these two technologies in SouareTech ensures data sharing in a cooperative environment: blockchain guarantees reliable data sharing in a large-scale environment to form real megadata; AI ensures strict access control, prediction of attacks and more intelligent security in the architecture, which help to establish the banking system more reliable. We adopt the combination of these fuzzy logic (FL) and Private Data Center (PDC) mechanisms in SouareTech allows to get these better performances in terms of the impact of attacks on the banking system and the reliability of the data. In this paper, we have proposed SouareTech, an intelligent architecture that shares, computes and stores data securely in different servers, aiming to thwart cybercriminals. In addition, we have presented the SouareTech architecture and the role of its devices and then an analysis of its effectiveness from the point of view of network security has been made.

Keywords: Blockchain · AI · Security · Cyberattacks · Banks · Data

1 Introduction

The banking sector is one of the most essential institutions, ensuring its stability allows the country to succeed. For transactions, banks issue credit cards that allow their customers to make purchases in supermarkets, shops, on the Internet or to withdraw money. The data are stored and transit in the equipment of the banking system. Consequently, these can generate several applications that make data transactions vulnerable. These data are exploited by the attackers by compromising all security properties. The attack is an offensive action that targets the infrastructure and services connected to the Internet using various techniques to steal, modify or make the network inaccessible. A study was made on attacks for data breach, ransomware infection, malware, phishing attacks place the banking sector in the top five for the number of security incidents in 2021. To avoid

fraudulent transactions, it is necessary to modernize this sector. Artificial intelligence (AI) is fully fulfilling this role. The financial industry is likely to be the one to tackle artificial intelligence [1]. It is the next phase of banking as it seems to be using advanced data science to combat unauthorized transactions. Blockchain technology is also at the center of core innovation with promising application possibilities in this banking sector. The key lies in how to make data immutable, traceable and secure. Blockchain technologies are the promising way to achieve this goal, via consensus mechanisms across the network, to guarantee the sharing of data in a reliable way. The complementarity of these two types of technologies will serve the modernization of the banking industry. Indeed, the modernization of the banking system offers advantages also in the protection of data, especially in the transaction. Digital technology and its procession of innovation constitute an opportunity for this sector to reinvent itself and improve customer relations. It offers more flexibility, access to a wider range of services in a modern technological environment and security is available without time constraints. With AI several machine learning and artificial neural network models are used to detect fraudulent transactions with credit cards. These machine learning models are logistic regression (LR), support vector machine (SVM), decision tree (DT), random forest (RF), artificial neural network (ANN). In this paper, we propose a combination of blockchain and AI to secure the data in the banking system by designing a SouareTech architecture to significantly improve the data sharing and then the network security. In the SouareTech architecture we combine the two fuzzy logic type 2 (FL) and PDC mechanisms for sharing, data collection and attack detection at secondary and central nodes. The integration of PDC into SouareTech allows central servers to monitor and reason about what and why data is being used and by whom, which means they can truly control each operation and get accurate results and management of data access behavior. SouareTech provides a more secure and intelligent data storage system via automated server algorithms. The rest of the paper is organized as follows: in Sect. 2 a presentation of the literature review, in Sect. 3 Architecture of the interbank transaction 4 we describe AI and blockchain-based security, in Sect. 5 we present the SouareTech architecture, in Sect. 6 we discuss results and methodology, and in Sect. 7 we draw a conclusion and outlook.

2 Literature Review

Data is the oil of every country and every company; its security is the main concern of all network architectures. It is also the basis for the functioning of AI because of its need for a huge amount of data from many possible locations on the Internet. To enhance the security of data in the banking system a number of works have been carried out. Maharjan and Chatterjee [2] develop a protection system minimizing the risks that exist in the banking sector of Nepal. The framework developed will help to control or avoid significant organizational risks associated with mitigation delays. Single et al. [3, 4] develop an electronic architecture of different banks using blockchain technology to solve the data storage problem and an optimization of their security. To optimize data security, they recommend working in a team spirit of different banks [5]. Edwinraj et al. compare and analyze some techniques to detect credit card fraud [6, 7]. They focus on methods like fusion of Dempster shafer and Bayesian learning. Zhang [8] develops a

real-time monitoring system for bank accounting based on software and an early warning system to optimize security. Btoush et al. [9–11] propose the integration of AI in the banking system and other financial services. They allow for improved strategic decisions and automation of attack recognition. They propose a customer-listening monitoring system for all attempted banking activities at ATMs. Using a graphical modeling tool (Unified Modeling Language), with design incorporating an additional input, via an embedded IP camera that stealthily captures the facial image of the ATM user. They automatically transmit to the mobile device of the bank that owns the account, via a dedicated artificial intelligence. In [12] Kal Wang.al propose models for detecting fraudulent credit card transactions using five classifiers namely logistic regression (LR), support vector machine (SVM), decision tree (DT), random forest (RF) and artificial neural network (ANN). Using the data resampling technique to increase the number of samples, while the second is based on the costs where the error function integrating the weights of each class. In [13] Anuchal Sahu.al present Type-2 fuzzy logic an approach for implementing AI in the banking industry. They show that when optimized with scalable algorithms, Type-2 fuzzy logic offers very good performance, comparable with neural networks in terms of accuracy, but outperforms other models and is recommended as a suitable machine learning approach for any automated decision making in the financial domain. In [14] Janet Adams.al combine blockchain and AI using Secnet architecture to improve data security and sharing, network security. They adopt in the architecture the PDC Private Data Center which is more able to deploy. They continue that blockchain allows AI with its rise to power a hopeful, efficient bring to enable reliable data sharing in a trustless environment which can help AI to make its decisions more accurate due to real megadata collected from more several places on the Internet. This is the reason why blockchain can enable trust mechanisms is that it can provide transparent and unforgeable metadata to an infrastructure. All the ideas and solutions proposed above to secure data by designing a new application paradigm or by integrating blockchain and AI to analyze data security and storage. However, none of them address the security of the banking system point of view architecture decentralize to solve the storage problem and data security. To fill this gap, SouareTech is a common and general network architecture of an interbank system that combines the power of AI and blockchain together at large scale. This can support a dynamic update of all these functional components i.e., both types of nodes (secondary nodes and core nodes) at any time if needed, to improve data security. Firstly, the AI in SouareTech takes care of generating artificial data to form a more robust security, predicts and recognizes attacks. In Secnet, the AI is limited to defense only. Second, SouareTech maximizes data sharing through cooperative communication between nodes, followed by control of the central nodes.

3 Architecture of the Interbank Transaction

3.1 Architecture

Figure 1 is a banking transaction architecture that is used around the world. The internet is the communication channel using Visa, Mastercard, or American Express payment cards. For this network, switches ensure the payment and the approval of the transaction. The computers are connected to the switches, ATMs and database servers. The information

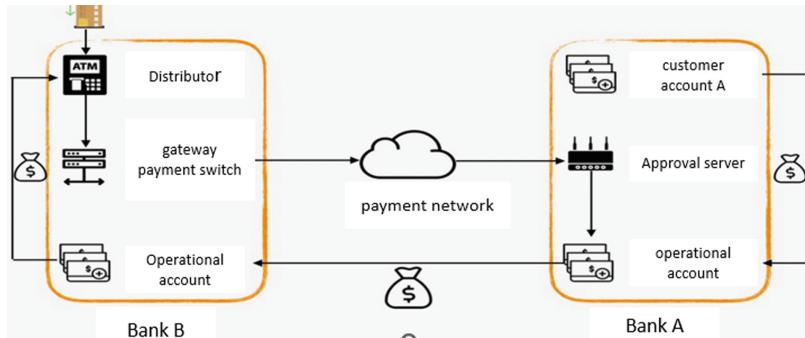


Fig. 1. Interbank transaction architecture.

is transferred to the cardholder's bank, via this payment network, which will approve or not the transaction. After the approval of the transaction the terminals take care of the distribution of the tickets [15].

3.2 Analysis of the Vulnerabilities of the Existing Architecture

With the progress of technology, attacks have become more and more intelligent and the banking system is not spared. The effectiveness of a computer system relies on its ability to ensure the integrity, authenticity, confidentiality and availability of the data in its network. Data is subject to numerous operations such as capture, storage, retrieval, update and transfer. By using the Internet as a communication channel, this system remains ineffective against these new threats. This system does not allow the traceability of the different possible attacks. Attacks by phishing to this system are undetectable. Indeed, phishing allows the installation of malicious software and gives the possibility to compromise the machines having access to the payment network and data. Advanced persistent threat (APT) attacks are very sophisticated and stealthy. These types of attacks combine advanced intrusion and spoofing techniques that can recover sensitive data. Thus, Trojan horse attacks, denial of service (DDoS), malware attacks, zero-day attacks, data breaches, SQL code injection remain very attractive to cybercriminals. This model does not address these types of attacks. The implementation of an automatic detection system and the use of powerful cryptographic algorithms are necessary to predict known or unknown types of attacks and a tamper-proof distributed database.

4 IA and Blockchain Security

4.1 Blockchain

In general, there are two types of blockchain: public and private. A public blockchain is open to the public and anyone can participate as a node in the decision-making process. Due to a large number of nodes participating in the network, the transaction speed is low but the security is high. A private blockchain is open only to a group of individuals or organizations that have decided to share the ledger only among themselves. When

a block is added to the blockchain network, a digital signature is required based on asymmetric cryptography and the chosen protocol allows to validate the recording or not of a transaction. Regarding the digital signature, each user has a pair of cryptographic keys—one public and one private. The private key is used to sign transactions. Signed digital transactions are distributed throughout the network and are then accessed by public keys, which are visible to all members in the network. Figure 2 shows an example of a signature used in a blockchain. There are two phases in a typical digital signature process: the signing phase and the verification phase. When a client wants to sign a transaction, they first generate a hash value derived from the transaction. Then they encrypt this hash value using their private key and send the encrypted hash with the original data to the agency. The agency verifies the received transaction by comparing the hash decrypted via the client's public key, and the hash value derived from the receipt given by the same hash function as the clients. The digital signature algorithms used in blockchains include the elliptic curve digital signature algorithm to ensure the security of communications between different participants in the blockchain network. These techniques allow our model to solve the storage of data, the traceability of transactions, the distribution of information and its validity across all nodes of the network. They also allow the creation of identifiers for each client. In addition, they allow our model to validate the transaction after a consensus of all the nodes.

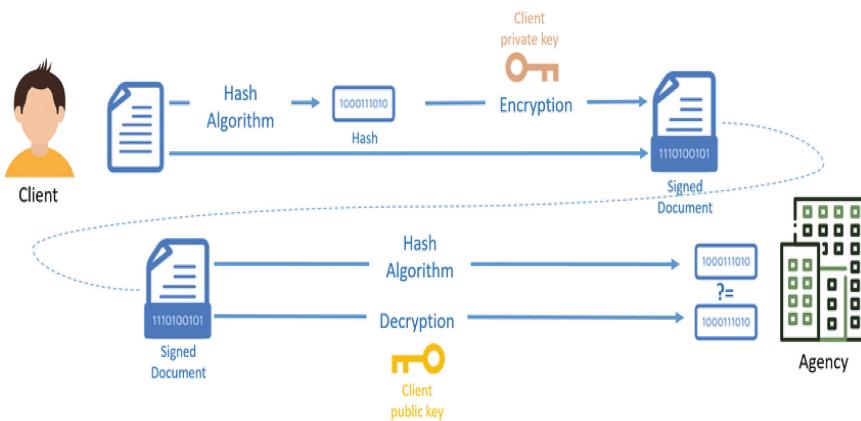


Fig. 2. Digital signature used in the blockchain.

Despite its unbreakable and distributed register, blockchain technologies are not immune to cyberattacks. Faced with the use of these quantum computers in cyber attacks, the cryptographic techniques of the blockchain remain ineffective. The lack of update of encryption algorithms, irreversible transactions and low processing speed make blockchain technology vulnerable. The absence of a warning system is also a weakness. The complexity of implementation, the compatibility of the various techniques of blockchain constitute a challenge for the banking system. To face certain types of attacks and complexities, it is necessary to integrate artificial intelligence in order to optimize the security of the banking system data.

4.2 Artificial Intelligence

It is important to note that AI works consistently with intelligent devices that ultimately benefit banks on the effectiveness and efficiency of these algorithms. Advances in such intelligence play a crucial role in the future of banks because they possess the potential to assist and improve the automation of activities and especially the defense of cyber attacks. With the complexity of attacks, AI detects and predicts attacks like malware and programs. Today mobile banking also uses biometrics to authenticate customers. AI biometric security techniques not only improve the security of banking data, but also offers more convenience to customers because they no longer have to remember or enter their login credentials [16]. With AI optimizes fraud detection. Neural networks learn each customer's password entry styles when combining keystrokes on the keyboard and compare to make a decision to validate or not.

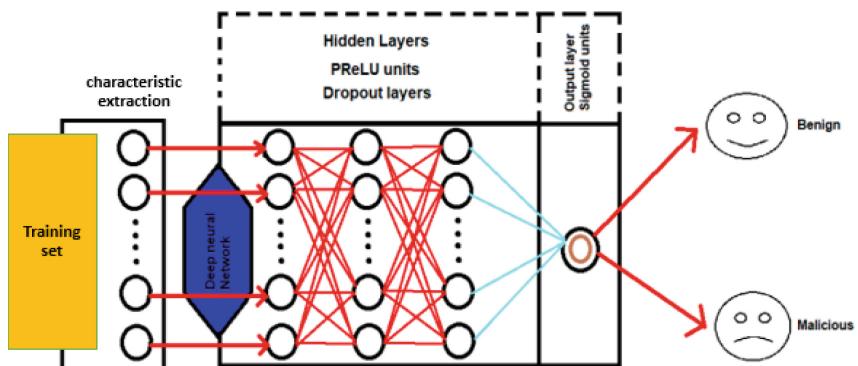


Fig. 3. Neural network.

AI allows us to optimize data security. The integration of these two techniques in the banking system allows us to propose this intelligent security model. It better registers the attacks that the interbank system encounters.

5 Architecture Proposed by SouareTech

This model is an architecture for optimizing the security of the interbank system data against cyber attacks. Blockchain and AI provide security for the model's computing platform. The blockchain enables data sharing, storage, encryption and hashing. The AI-based security produces intelligent and dynamic security rules based on the distributed ledger of the blockchain. Figure 3 illustrates the overall architecture of the model. In this network the nodes are connected and communicate with each other based on its consensus algorithm. For this model we have three servers: centralized automatic servers, local servers and client servers. The local servers manage the bank's operational accounts and are connected to the payment terminals. The data servers manage the customers' accounts. The switches and the computer share the data to both servers. The automatic server is connected to these different components to have access to the different data of the

system. They cooperate to share data and facilitate threat detection; each node contains a blockchain registry and synchronizes its status with other nodes. These two central servers monitor the network activity, updating the data. They identify vulnerabilities, malware, memorize attacks, analyze user behavior and control all transactions in this SouareTech model (Fig. 4).

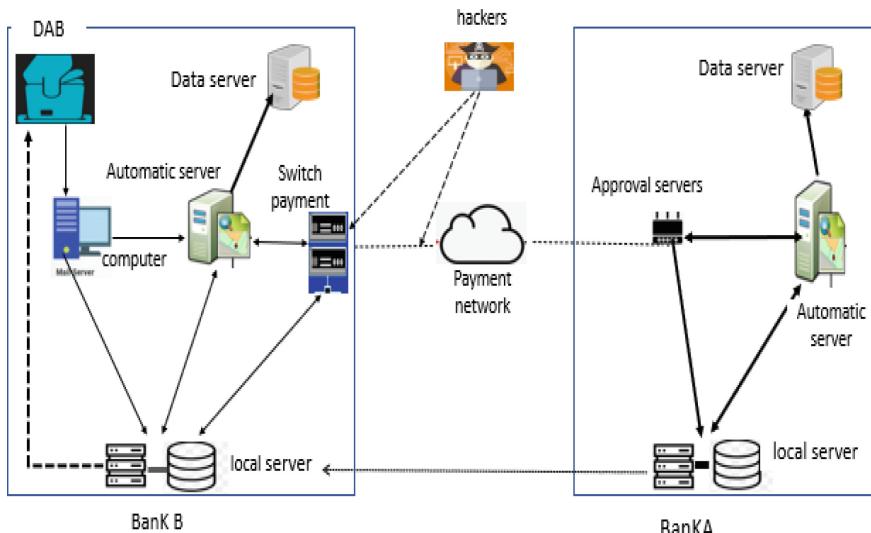


Fig. 4. Architecture SouareTech.

In the face of these intelligent attacks, our model provides a response to cyber threats. This SouareTech model combines blockchain security techniques and artificial intelligence to optimize detection automation, vulnerability prediction and data non-breach. These two technologies are different but complementary. Blockchains are deterministic, permanent and immutable, its security is based on consensus algorithms and cryptography. This model has a decentralized database register protected by several digital signatures. As a result of the distribution of its database, attackers will find it more difficult to find a single point of entry, and thanks to its platform, cyber-defense is improved by preventing fraudulent activities via consensus mechanisms of the various nodes. In line with this trend of new, more intelligent attacks, the system also plans to respond effectively to malware as a result of the application of artificial intelligence to the detection of varied and sophisticated malicious code. It has a predictive capability based on an analysis of normal or abnormal behavior of all network traffic and guarantees the non-alteration, provides authenticity, integrity and traceability of all data transactions in the network.

6 Methodology and Results

6.1 Methodology

The research was conducted in the context of detection, data reliability and prediction of attacks of the banking system transaction. The fuzzy logic training driven by the PDC gold of a predictive analysis of attacks using neural networks that are based on public datasets available in the kaggle provides solid results to achieve very high levels of accuracy. Fuzzy logic is a model that relies on fuzzy rules for built-in explainability, combined with scalable optimization used to maximize the performance of the architecture. Fuzzy logic also allows for a full understanding of the factors influencing the predictive outcome of attacks in this architecture, as a result of its full global as well as local explicability respectively at each data output. PDCs provide a more secure and intelligent data storage system at the node level. The integration of PDC in SouareTech allows to monitor and reason about what and why the data is used in order to detect attacks as well, which means that users can track at each operation its data. These allow us to get both charts using publicly available data namely the credit card fraud detection we pull in kaggle.

6.2 Results

Following the use of our proposed architecture SouareTech and the fuzzy logic and PDC mechanisms gives very satisfactory results on the reliability of the data, which we will compare with an existing architecture in terms of performance on the security of the data and the impact of attacks on this system. These tables show the performance of the SouareTech model to attacks on the banking system (Tables 1 and 2).

Table 1. Security performance of the two models.

Security	Architecture existing	Architecture SouareTech
Confidentiality	+	++
Integrity	+	++
Availability	+	++
Non-repudiation	+	++
Access control	+	++

+: performance

The ++ shows the performance of the SouareTech model on its ability to ensure data integrity, confidentiality and availability. These are explained by the fact that the model has a distributed database registry, the cryptographic techniques each node has the two types of keys. Any transaction or modification is only possible through its protocol consensus of all the nodes of the network. Blockchain technologies facilitate the maintenance of data provenance and immutability. In addition, the implementation of

Table 2. The impact of cyber attacks on the two models.

Types of attacks	Architecture existing	Architecture SouareTech
Phishing	+	--
Advanced Persistent Attacks (APT)	+	--
Denial of Service attack (DDOS)	+	--
Zero-day attacks	+	--
Trojan horses	+	--
Malware and ransomware	+	--

+: possibility to attack

--: no possibility to attack

intrusion and anomaly detection systems optimizes data access to these different attacks. This model has a better optimization on the data security of the banking system.

The -- shows the performance of our model to these different attacks on the banking system. The + shows that the existing model remains more exposed to these different attacks. The more complex the threats are, the more inefficient it remains, because the deployment is specific and requires human intervention. The effectiveness of the proposed model is based on its ability to learn the normal operation of the system. The combination of these two techniques based on digital signature and anomaly detection allows this model to identify the different types of intrusions and unknown attacks in the current banking system. This table shows the performance of the SouareTech model and the vulnerability of the existing model to cyber attacks.

7 Conclusion

In this paper the use of blockchain and AI allowed us to propose the SouareTech architecture to optimize the detection of cyber attacks in the banking system. AI with the help of blockchain combining these fuzzy logic type 2 and PDC models have provided strong performance on data security and network in terms of recall, precision and access, in that they are able to derive understandable human interpretations of the underlying factors when making decisions at both global and local levels. Reliable data management in a trustless environment the SouareTech architecture allows, with a paradigm focused on decentralized, secure storage. Thus, these attacks on the banking system have been presented. This architecture optimizes the security of data by ensuring its confidentiality, integrity, authenticity and availability. The goal of the combination of AI and blockchain is to obtain an intelligent architecture to overcome the flaws of the banking system. Today many countries are contributing to the research of these new technologies and their applicability, but China, Europe and the United States are still leading the contribution. In future work, we will explore how to leverage blockchain to reduce data access authorization in the communication-rich internet of things. In addition, we will model the SouareTech architecture and analyze its performance through extensive experiments based on quantum cryptography.

References

1. Swain, S., Gochhait, S.: ABCD technology—AI, Blockchain, Cloud computing and Data security in Islamic banking sector. IEEE, (2022)
2. Maharjan, R., Chatterjee, J.M.: Framework for minimizing cyber security issues in banking sector of Nepal, vol 1, Issue 1. pp. 82–98 (2019)
3. Singh, K., Singh, N.: Dharmender Singh Kushwaha, An Interoperable and Secure E-Wallet Architecture based on Digital Ledger Technology using Blockchain". Galgotias University, Greater Noida, UP, India (2018)
4. Roy, P., Rao, P., Gajre, J., Katake, K., Jagtap, A. and Gajmal, Y.: Comprehensive analysis for fraud detection of credit card through machine learning. In: 2021 international conference on emerging smart computing and informatics (ESCI), vol 10. IEEE, pp. 765–769 (2021)
5. Acharya, S., Joshi, S.: impact of cyber-attacks on banking institutions in India: a study of safety mechanisms and preventive measures. Parach's journal **17**(6), (2020)
6. Benson Edwin Raj, S., Annie Portia, A.: Analysis on credit card fraud detection methods. Tirunelveli, India (2011). Editor: IEEE
7. Thisarani, A. M., Fernando, S.: Artificial intelligence for futuristic banking. IEEE, (2021)
8. Zhang, J.:Design and Implementation of real time warning module for bank financial supervision system. Date of Conference: December 2017, Editor: IEEE
9. Btoush, E., Zhou, X., Gururaiyan, R., Chan, K.C. and Tao, X.: A survey on credit card fraud detection techniques in banking industry for cyber security. IEEE, Doha Qatar, p 7 (2021)
10. Kouraogo, Y., Zkik, K., Orhanou, G.: Attacks on android banking applications. IEEE, Agadir Morocco (2016)
11. Musthyala, H., Reddy, P.N.: Hacking wireless network credentials by performing phishing attack using Python scripting. IEEE (2021)
12. Wang, K., Dong, J., Wang, Y., Yin, H.: Securing data with blockchain and AI", date of publication June 7, 2019, date of current version June 27, 2019. IEEE
13. Sahu, A., Harshvardhan, G.M., Gourisaria, M.K.: A dual approach for credit card fraud detection using neural network and data mining techniques. New Delhi, India (2021)
14. Adams, J., Hagras, H.: A type-2 fuzzy logic approach to explainable AI for regulatory compliance, fair customer outcomes and market stability in the global financial sector. IEEE, Glasgow, UK (2020)
15.
16. Tran, V.B.H.: Converging artificial intelligence and blockchain technologies for security and risk management in banking. University of Northern Iowa, (2021)



Streamlining Conceptual Modeling

Hermann Bense^(✉)

bense.com GmbH, Schwarze-Brüder-Str. 1, 44137 Dortmund, Germany
hb@bense.com

Abstract. Research into knowledge and expert system engineering has seen numerous attempts to streamline conceptual modeling, especially in the field of so-called *Multi-Level Modeling* (MLM). In this paper, we will refine and augment the approaches made there, so that these can also be used for conceptual modeling in general. To this end, we introduce a formal methodology for the definition of attributes (data properties) and relational relationship types (object properties). Rules are then defined as instantiation regulations in this connection. As a new methodology, we introduce the so-called *Value Assignment Propagation* (VAP), and then show how this streamlines conceptual modeling, how layer-mistakes can be avoided and how previously unfulfilled requirements for MLM can be satisfied.

Keywords: Knowledge engineering · Conceptual modeling (CM) · Multi-level modeling (MLM) · Data properties (DP) · Value assignment propagation (VAP) · Filters · Inheritance · (Deep)instantiation · Pow-erotypes

1 Introduction

The past two decades have seen a number of seminal publications and books about ontology engineering, yet, we are unable to arrive at a mathematical formalism for instantiation. Intuition is the predominant aspect in identifying what data properties can be invoked via inheritance, but a precise definition of which sets of universals and particulars are affected, is nowhere to be found; in particular, the question of how inheritance affects object properties (relationship types) with regard to instantiation is left largely unanswered.

The way in which the terms *inheritance* and *instantiation* are applied by ontologists is often features an intuitive type of understanding, which is based on the methods of object-oriented programming (OOP), object-oriented analysis (OOA) and object-oriented design (OOD). *The Unified Modeling Language* (UML) and UML diagrams are utilized to visualize conceptual models and to specify the logic of applications. Ontologists also rely on modeling languages such as the Web Ontology Language (OWL), which is based on *Description Logic* (DL). The concepts of inheritance and instantiation also play a key role in relational and NO-SQL database systems. When attempting to actually break the

modeling down into its implementation levels, a mismatch often occurs because the methodology and the power of the various representations vary from each other. By applying mathematical set theory, we will extend and generalize the formalisms for describing inheritance and instantiation and we will have to distance ourselves from known methods, where necessary.

The rest of the paper is organized as follows: In the Related Work Sect. 2, we discuss what approaches already exist for streamlining conceptual modeling and what disadvantages they entail. The Methodology Sect. 3 describes the naming conventions on which our methodology are based and the mathematical foundations for the formalizing inheritance and instantiation. Using concrete modeling examples, the application of the methodology is discussed in the Application Sect. 4. In the Evaluation Sect. 5, we evaluate the extent to which our methodology contributes to fulfilling the criteria for Multi-Level Modeling (MLM), and to what extent our approach contributes to streamlining conceptual modeling in general, and how metadata can be managed more systematically; we also document the savings potential of our modeling. Finally, in the Conclusion Sect. 6, we summarize the results and describe the outlook fore future research activities.

2 Related Work

The article by Humm et al. [9] differentiates between lightweight and heavyweight ontologies and associates developer-oriented communities with the first, and academic communities with the latter. Our primary motivation is to introduce a methodology for conceptual modeling that has no limitations to the power of modeling, but avoids the ballast and complexity of other approaches. That is, our approach aims to support the development of both lightweight and heavyweight ontologies. As a typical layer mistake in *Meta-Object Facility* (MOF)¹ we find that `Class` appears on both levels 2 and 3, which we believe to be a flaw. The article by Carvalho and Almeida [5] describes how a powertype `MobilePhoneModel` can be modelled using *regularity attributes*. But the classes `MobilePhone` and `MobilePhoneModel` cannot be merged with regularity attributes, since each particular phone would then also have a `launchDate`. Another current subject of discussion in ontology modeling and in the Semantic Web is the question of how we can model *Labeled Property Graphs* (LPG). Suggested solutions come from the NO-SQL community, particularly from neo4j.² So far, however, these have not been systematically incorporated into modeling languages such as RDF and OWL. Within the Dublin Core Metadata Initiative,³ metadata is defined in the *Dublin Core Metadata Element Set*. Are data properties like `Title`, `Publisher` and `Description` really always metadata, or do they belong more to the application domain? There needs to be a clarification on how to achieve flexibility in modeling so that such attributes can be used either as meta attributes or attributes of the

¹ https://en.wikipedia.org/wiki/Meta-Object_Facility.

² <https://neo4j.com/>.

³ <https://www.w3.org/2001/09/rdfprimer/rdf-primer-20021108.html#dublincore>.

application domain, or in both ways, if necessary. An article [1] by Atkinson and Kühne argues that the shallow instantiation approach cannot solve the issues surrounding *multiple classification* and *replication of concepts* problems. The authors add the concept of *potency* to each model element. Potency defines the depth to which a model element can be instantiated. In each instantiation step, the potency is reduced by 1. However, this requires the concepts to be modeled such that they can be divided into logical levels, each of which must be assigned a meaning. This harbors the potential for layer mistakes, especially in ontologies that are undergoing frequent changes during development. The subdivisions of base classes are also often performed using terms such as `category`, `family`, `group`, `order`, `subordinate`, etc. For example, in a paper by Neumayr et al. [11] on page 9, in Figure 3, the basic classes `Book` and `Car` are modeled as subclasses of the `ProductCategory` and assigned a classification level of 3. To model that `marketLauch` has to be instantiated at brand level and `mileage` has to be instantiated by objects at level physical entity, the authors claim it is necessary to use explicit constraints. Since the authors do not specify the constraints, the question of how this can be achieved is left open. We also see the potential for layer mistakes in taking this approach. A paper by Guizzardi et al. [7] also discusses *deep instantiation*. The classes `BirdSpecies` and the class `Bird` are modelled in two parallel class hierarchies. In this we identify a modeling deficiency whereby the characteristics of all other species would have to be modeled redundantly as in `BirdSpecies`, or would need to be shifted to a higher class `Species`. The authors' modeling of roles for `Gender` and `ThreatPhase` does not answer how one of the values for the enumeration types should be associated with a specific particular. We make the assumption that it should be a 1:1 connection via an object property. Nevertheless, we consider the graph rewriting rules on page 6 of [8] to represent an important contribution in reducing the complexity of ontology models. The approach of Bense and Humm [3] proposes satisfying the instance and type character via two different inheritance mechanisms of properties. These are `BroderInheritedProperty` based on `skos:broader` and `TypeInheritedProperty` based on `rdfs:class`. The case that is not covered by this is if we want to model that all `polar bears` have a `white fur`.

Other proposals made in the area of *Multi-Level Modeling* (MLM) [4,6,13, 14] describe approaches with which conceptual modeling is not only enriched but also streamlined. What the cited approaches have in common, however, is that they require additional complex or even overly-complex annotations and/or the extension of query languages, meaning they are no longer compatible with RDF/RDFS/OWL and query languages such as SQL/SPARQL. In addition, not all of the requirements previously placed on MLM in [6, 7, 10] could be met. So, major problems in conceptual modelling result from the non-availability of suited uniform methods or the inadequate application of modeling methods. In addition to other problems, this causes *layer mistakes*, which penetrate even into the basic methods of software engineering.

The key research question we answer is: What are the limitations in ontology engineering when it comes to creating leaner, less redundant, better comprehensive and more accurate models, and what methods can be used in overcoming the limitations discussed above. This raises other additional questions: What naming conventions can be used to reduce ambiguities? What patterns can be specified to define properties and what are the permissible instantiations that result from this? What influence does this have on the modeling style of the ontologists? How can existing ontologies benefit from this, and what advantages emerge here for maintaining accuracy and consistency in the evolution of ontologies.

3 Methodology

Before answering questions, the methodology used here has to be introduced. Basic methodologies for naming conventions in ontological engineering were described in [2]. Figure 1 is a so-called *ontograph* visualization of an example of a vehicle ontology that makes use of these naming conventions. The open source library graphviz⁴ is used to create ontographs. The quick switching between ontology modeling and visualization supports rapid prototyping, thus supporting error detection and correction. The example in Fig. 1 also illustrates the principle of (multiple) inheritance, as well as depicting the instantiation of object properties. The following sections will explain in greater detail the naming conventions used, the different types of property definitions, and their instantiations.

But first of all, we need to introduce a set of basic definitions, naming conventions, axioms and rules for conceptual modeling. Our aim is to manage with as few concepts as possible. These should be understood as recommendations and they may be adapted or implemented in alternative ways. For example, the prefixes used in the naming conventions encode the type of the concepts within their names. Of course, by explicitly specifying further meta information, each represented concept can alternatively be classified without using prefixes. The core concepts we define here include *Knowledge Graph* (KG), *Knowledge Subject* (KS), *Atomic Data Type* (ADT), *Value Set* (VS), *Value Set Range* (VSR) and *Particulars Layers* (PL) as well as *Universals Layer* (UL). The mathematical concepts *Principal Filter* and *Principal Ideal* allow for a compact definition of hierarchies of universals. The aforementioned concepts consequently form the basis for formally defining *Data Property Definitions* (DPD) and *Object Property Definitions* (OPD). The instantiation rules then determine how DPDs and OPDs may be applied. The instantiation of data properties is referred to as *Data Property Instantiation* (DPI), and that of object properties is referred to as *Object Property Instantiation* (OPI). *Relator Instantiation* is a special type of OPI for modeling n-ary relationships. The central approach taken by this paper is called *Value Assignment Propagation* (VAP) and consists of introducing a new methodology for the different application of the data property instantiation (DPI) based on two different types of the data properties, namely the *Propagation Data Property* (PDP) and the *Meta Data Property* (MDP).

⁴ <http://www.graphviz.org>.

3.1 Definitions

The key concepts in ontology engineering are classes, attributes, relationship types, particulars, relators, processes and functions. In line with language usage in the Semantic Web, we also refer to attributes as *Data Properties* (DP) and relationship types as *Object Properties* (OP). It is therefore consistently true that a property p is either a DP or an OP. Classes, object properties and processes are also gathered under the general term *universal*. The most important concepts in ontology engineering are classes, attributes, relationship types, particulars, relators, processes and functions. According to the language usage in the Semantic Web, we also refer to attributes as *Data Properties* (DP) and relationship types as *Object Properties* (OP). It applies consistently that a property p is either a DP or an OP. Classes, object properties and processes are also collected under the general term *universal*. Particulars are instances of universals which are connect by the object property »iof (is instance of) with a universal, e.g. ($>\text{Toms_Lambo}$, »iof, $\wedge\text{Lamborghini_Miura}$). Besides through duplication particulars cannot be instantiated (s. Axiom A1 in [4], Page 8). Thus, the assertion (X , »iof, $>\text{Toms_Lambo}$) is not allowed.

3.2 Name Sets

The name of an ontological concept always has the index of the designating set as its prefix: $N_x = \{xn \mid n \text{ is a string}\}$. To enable the names of ontological concepts to be unambiguously designated, we introduce the following sets: $\mathbf{N}=\text{Strings}$ with the subsets $\mathbf{N}^{\sim}=\text{Class Names}$, $\mathbf{N}_{\cdot}=\text{Data Property Names}$, $\mathbf{N}_{\diamond}=\text{Object Property Names}$, $\mathbf{N}_{>}=\text{Particular Names}$, $\mathbf{N}^{-}=\text{Relator Names}$, $\mathbf{N}_{\circ}=\text{Process Names}$, $\mathbf{N}^{\sim}=\text{Function Names}$. The names of universals is the set $N_U = N^{\sim} \cup N_{\diamond} \cup N_{\circ}$. Later, we will show how the prefixes of data properties, in particular, are used to define instantiation rules. Similar naming conventions for so-called regularity attributes are introduced in [5], e.g. the prefix `instances` in `instancesScreenSize` indicates that an attribute of the type class `MobilePhoneModel` correlates with the corresponding attribute of the base class `MobilePhone`.

3.3 Knowledge Graphs

An ontology is represented by a *Knowledge Graph* (KG). The smallest syntactic storage unit is a (RDF) triple⁵ in the form (s, p, o) . A *Knowledge Graph* (KG) is defined as $\text{KG} \subseteq N \times N \times N$. With $s, p, o \in N$, a *Knowledge Atom* (KA) is a triple $(s, p, o) \in \text{KG}$. A *Knowledge Subject* (KS) is the set of triples with the same subject x is denoted by Σ_x :

Df 31. (Knowledge Subject) $\Sigma_x = \{(s, p, o) \in \text{KG} \mid s = x\}$.

⁵ https://en.wikipedia.org/wiki/Semantic_triple.

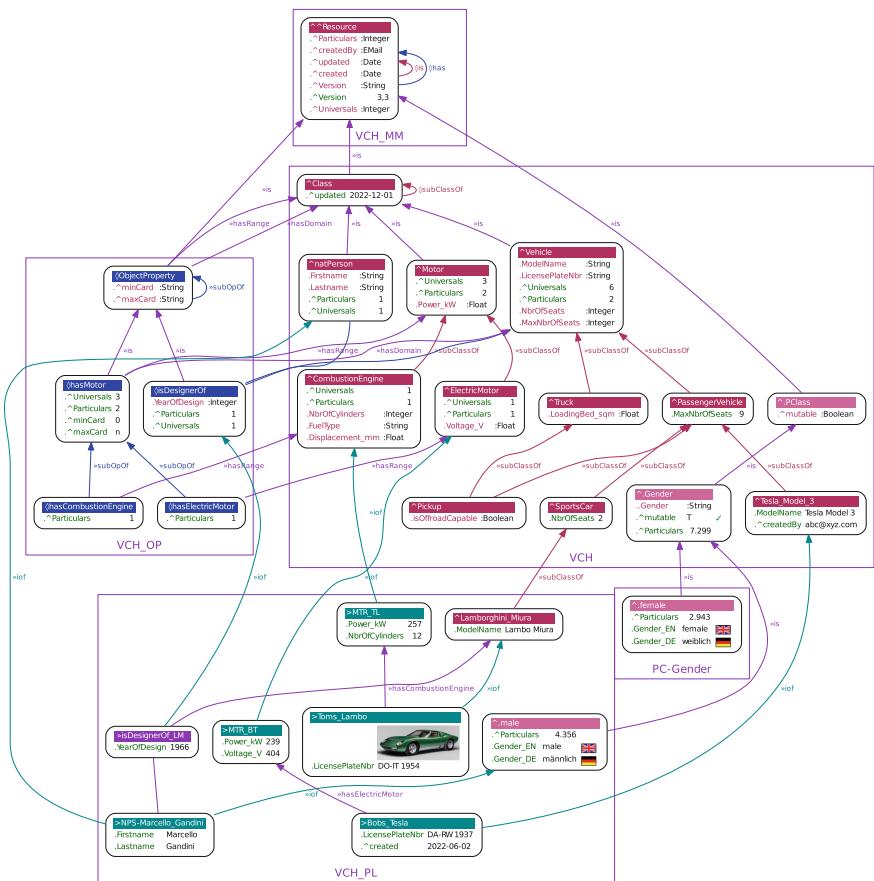


Fig. 1. Vehicle ontology.

A knowledge subject Σ_x is represented as a node in the graphical representation of a knowledge graph (OntoGraph) and has the node name x. According to our understanding, a knowledge subject also corresponds to an instance/singleton that is unique, regardless of whether it is a particular or a universal. Even if all pairs (p,o) of values of two knowledge subjects Σ_x and Σ_y with $x \neq y$ are identical, they are different objects simply because their names x and y are different. The physical representation of a knowledge graph is called a *Knowledge Repository* (KR). A KR can be a file, a table, or a database.

3.4 Atomic Data Types

Data properties are defined by exactly one so-called *Atomic Data Type* (ADT). An ADT name is an element of the set $ADT = \{ :String, :Integer, :Decimal, :Float, :Double, :Boolean, :Duration, :DateTime \}$.

:Date, :Time, :anyURI, ...}. To simplify matters here, we only use a subset of the datatypes defined in OWL⁶ and XML⁷ and we also use a different notation. Instead of `DataPropertyAssertion(a:NbrOfCylinders a:>MTR_TL "12"^^ xsd:integer)`, we would make assignments in the more readily editable and readable form (`>MTR_TL, .NbrOfCylinders, 12`). Nor do we encode the type of the attribute value at the attribute value itself, but rather within the definition of the data property with (`^CombustionEngine, .NbrOfCylinders, :Integer`) instead. We consider it a major disadvantage that, according to OWL conventions, the atomic data type always must be specified with the value itself, as this leads to unnecessarily high additional storage overheads and needlessly complicates queries and makes comparisons difficult.

3.5 Value Sets

The set of all possible values of a data property p is defined by the *Value Set Range* (VSR). The set of all values actually applied in the ontology for a data property form the $Value\ Set\ VS(p) \subseteq VSR(p)$.

Df 32 (Value Set). $VS(p) = \{ v \mid (s, p, v) \in KG \}$. Examples:

- $VSR(:Natural) = \mathbb{N}$, $VSR(:Integer) = \mathbb{Z}$, $VSR(:Float) = \mathbb{R}$
- $VSR(:Boolean) = \{0, 1\} \cup \{T, F\}$; $VS(.warmblooded) = \{T, F\}$
- $VSR(.Gender) = :String$; $VS(.Gender) = \{\text{male}, \text{female}\}$
- $VSR(.WaveLengthOfVisibleLight_nm) = \{wl \mid wl \in [400, 780]\}$

3.6 Layers

We differentiate between two ontology layers, the *Universals Layer* (UL) and the *Particulars Layer* (PL). The PL contains all particulars which are connected via the »iof object property to an universal and all object property instantiations (OPI) between any two particulars. Universals are organized into hierarchies by the following sets of *semantic relations* (SR).

- $SRU = \{\Diamond\text{is}\}$; $SRM = \{\Diamond\text{has}\}$
- $SRR = \{\Diamond\text{subClassOf}, \Diamond\text{subOpOf}, \Diamond\text{subDpOf}, \Diamond\text{subProcessOf}\}$
- $SRO = \{\Diamond\text{hasSubClass}, \Diamond\text{hasSubOp}, \Diamond\text{hasSubDp}, \Diamond\text{hasSubProcess}\}$

The object properties of the SSR set express hyperonymy, those of the SRO set hyponymy and the respective inverse relationship results from the naming convention: $\Diamond\text{subXOf} = \Diamond\text{hasSubX}^{-1}$. The UL contains all universals including all data property definitions (DPD), all object property definitions (OPD) and all hierarchy relationships defined through an object property $op \in SR = SRR \cup SRO$. The number of levels in a hierarchy is not restricted. Cycles are not permitted in hierarchies.

⁶ https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/#Datatype_Maps.

⁷ <https://www.w3.org/TR/2012/REC-xmlschema11-2-20120405/#anyAtomicType>.

3.7 Filters

Let \leq be a non-strict order relation, while the pair (M, \leq) is called a non-strictly ordered set. A filter or *order filter* (OF) is a special subset of a partially ordered set (poset). We use filters to construct parent and child hierarchies of universals. With $x, y \in N_U$ and $\diamond_{SRR} \in SRR$ or $\diamond_{SRO} \in SRO$ we define:

- $x \leq y \Leftrightarrow ((x, \diamond_{SSR}, y) \in KG \text{ or } (y, \diamond_{SSR^{-1}}, x) \in KG)$
- $x \geq y \Leftrightarrow ((x, \diamond_{SRO}, y) \in KG \text{ or } (y, \diamond_{SRO^{-1}}, x) \in KG)$

Df 33 (principal filter).

$\therefore a = \{x \in M \mid a \leq x\}$ is the principal filter of a.

Df 34 (principal ideal).

$\therefore a = \{z \in M \mid z \leq a\}$ is the principal ideal of a.

3.8 Inheritance

We use the term inheritance essentially as it is applied in programming. Properties are inherited across a hierarchy of universals. As the root element of all such inheritance hierarchies, we introduce the `^^Resource` element. The elements of a class hierarchy are connected by the transitive object property `◊subClassOf`. A class is by definition a subclass of itself (as the subset may be the complete set).⁸ For example, the triples `(^ElectricMotor, »subClassOf, ^Motor)`, `(^CombustionEngine, »subClassOf, ^Motor)` and `(^Motor, »subClassOf, ^Class)` define a so-called class hierarchy (Fig. 1). The data property `.Power_kW` is therefore inherited from the `^Motor` class to all of its subclasses. This means that any particular motor can be assigned a value for the `.Power_kW` data property.

3.9 Inheritance Hierarchies

Be $u \in N_U$ the name of a universal. The principal filter $\therefore u$ is the set of all super universals of u including u, the principal ideal $\therefore u$ is the set of all sub universals of u including u, e.g.:

- $\therefore ^{^P}ickup = \{^Pickup, ^Truck, ^PassengerVehicle, ^Vehicle\}$
- $\therefore ^{^M}otor = \{^Motor, ^CombustionEngine, ^ElectricMotor\}$
- $\therefore \diamond_{hasCombustionEngine} = \{\diamond_{hasCombustionEngine}, \diamond_{hasMotor}\}$
- $\therefore \diamond_{hasMotor} = \{\diamond_{hasMotor}, \diamond_{hasElectricMotor}, \diamond_{hasCombustionEngine}\}$

3.10 Property Definitions

We distinguish between data and object property definitions. For both, we provide a compact and a traditional definition pattern. The two triple definition pattern, which uses the object properties `»hasDomain` and `»hasRange`, can be always transformed into the compact one triple definition pattern.

⁸ <https://www.w3.org/TR/owl-ref/#subClassOf-def>.

3.11 Data Property Definitions

Internal properties of instances are referred to as data properties (DP) according to the convention in the Semantic Web/OWL/RDF. These include characteristics such as `.Color`, `.Length`, `.Weight`, `.Age`, etc., which establish the corresponding quality dimensions. Traditionally a *Data Property Definition* (DPD) is built using two triples e.g. `(.LicensePlateNbr, »hasDomain, ^Vehicle)` and `(.LicensePlateNbr, »hasRange, :String)`. The short form of a DPD determines the associated type for the class, e.g. `(^Vehicle, .LicensePlateNbr, :String)` and an instantiation like `(>Toms_Lambo, .LicensePlateNbr, DO-IT 1954)`. Be $a \in N_\alpha$ the name of a data property, $:adt \in \text{ADT}$ an Atomic Data Type, and $u \in N_U$ the name of a universal. Both versions of the DP definitions are equivalent according to the following rule: $(u, a, :adt) \Leftrightarrow (a, »hasDomain, u) \wedge (a, »hasRange, :adt)$. So, data properties (DP) connect instances to values from the value set VS(DP). *Data Properties* (DP) are defined using *Atomic Data Types* (ADT) with an associated Value Set (VS). We distinguish between two fundamentally different types of data properties with their *Data Property Definition Patterns* (DPDP) defined below. DPDPs are inherited down the universal hierarchy, in which they are defined. Be `.dp` the name of a *Propagation Data Property* (PDP) and `.^dp` the name of a *Meta Data Property* (MDP).

- A *Propagation Data Property Definition* (PDPD) has the form $(u, .dp, :adt)$, for example, `(^Motor, .Power_kW, :Float)`.
- A *Meta Data Property Definition* (MDPD) has the form $(u, .^dp, :adt)$, for example, `(^Car, .^YearOfDesign, :String)`.

3.12 Object Property Definitions

External properties of instances are referred to as *Object Properties* (OP) according to the convention in the Semantic Web/OWL/RDF. Object properties connect instances with another instance or with itself. The traditional definition for an object property e.g. `◊hasMotor` is defined using the two triples `(◊hasMotor, »hasDomain, ^Car)` and `(◊hasMotor, »hasRange, ^Motor)`. The short form of an *Object Property Definition* is `(^Car, ◊hasMotor, ^Motor)`. As an example from the vehicle ontology we have `(>Toms_Lambo, ◊hasCombustionEngine, >MTR_TL)` where `◊hasCombustionEngine` is a subobject property (`◊subOpOf`) of `◊hasMotor`. This modeling is an example of how it is possible to define relationship types between instances and their specialization on different abstraction levels according to the requirement set out in [10], page 3. Be $\diamondop \in N_\diamond$ the name of an object property, and $c, d \in N_\circ$ the name of classes. Both versions of the OP definitions are equivalent according to the following rule: $(c, \diamondop, d) \Leftrightarrow (\diamondop, »hasDomain, c) \wedge (\diamondop, »hasRange, d)$.

3.13 Instantiation

How can we formally define *instantiation*? We essentially understand instantiation to be the creation of a new knowledge subject corresponding to a node in the knowledge graph, whose identifier $x \neq y$ for all previously existing instances of y . During the creation process, all possible value assignments should be preceded by the typing of the instance either as a particular of a class with $(x, \gg\text{iof}, u)$ or as a subclass of another class with $(x, \gg\text{subClassOf}, y)$. Any further value assignments can then be made to the instance, regardless of whether they are data properties or object properties. Be $\wedge C, \wedge D \in N$ the names of classes and $\cdot \cdot \wedge C$ and $\cdot \cdot \wedge D$ their super-class hierarchies and $\cdot \cdot \wedge C$ and $\cdot \cdot \wedge D$ their subclass hierarchies and $SOP(\cdot \cdot u) = \cup \{sop(c) \mid c \in \cdot \cdot u\}$. Be $>P, >P1, >P2 \in N$ the name of particulars and $>P1 \in SOP(\cdot \cdot \wedge C), >P, >P2 \in SOP(\cdot \cdot \wedge D)$. Be $r \in N_{\diamond}$ the name of an object property and $\cdot \cdot r$ the set of names of the super object properties of r . Be $DP \in N$ a data property name and $v \in VS(DP)$ a value from the value set of DP .

3.14 Data Property Instantiation

A short form *Data Property Definitions* (DPD) follows the pattern $(u, A, :adt)$ where u can be the name of a class or an object property, and $A \in \{\cdot A, \cdot \wedge A\}$ is the name of an principal or meta data property and $:adt \in ADT$ denotes an *Atomic Data Type* (ADT). The *Value Set Ranges* (VSR) of $\cdot A$ and $\cdot \wedge A$ are identical: $VSR(\cdot A) = VSR(\cdot \wedge A)$. Thus with $v \in VS(\cdot A) \subseteq VSR(\cdot A)$ every *Data Property Instantiation* (DPI) must map to one of the following rules where the symbol \models relates the property definition to its allowable property instantiation.

- $(u, A, :adt) \models (x, A, v)$ where $(x, \gg\text{iof}, u)$ or $x \in \cdot \cdot u$
- $(\diamond OP, \cdot A, :adt) \models (\gg OPI, \cdot A, v)$ for relators with $(\gg OPI, \gg\text{iof}, \diamond OP)$.

3.15 Value Assignment Propagation (VAP)

In a hierarchy of universals $\cdot \cdot u$ a value v can be assigned to a data property starting from the class where it firstly has been declared in the form $(u, a, :adt)$, for example, $(\wedge Car, \cdot \text{ManufDate}, :Date)$ or $(\wedge Car, \cdot \wedge \text{LaunchDate}, :Date)$. If the name of a data property contains the prefix dot(.) it is a *Propagational Data Property* (PDP) and if it contains the prefix dot-circumflex (.)⁵ it is a *Meta Data Property* (MDP). Value assignments of PDPs are propagated along the hierarchy of universals from the universal u , where the assignment was first made, to all subclasses that lie below u in the hierarchy $\cdot \cdot u$ and also to particulars that are connected to the universal u via $\gg\text{iof}$. According to Carvalho and Almeida [5], page 37 this is characterized by $\text{durability}=\infty$. For MDPs there is no propagation of value assignments ($\text{durability}=0$), i.e., they can only be assigned

once and are not inherited downwards. In addition, we provide the characteristic mutability for a data property for the case, that the value of an attribute needs to be forbidden to change, once it has been assigned, for example (.warmblooded, .mutable, F) or (.Gender, .mutable, T). This corresponds to setting `mutability=0` in [5]. For MDPs \wedge^a we have in general that they must not instantiate into particulars with the only exception of Adminstrative Meta Data Properties as described in the Evaluation Sect. 5: If p is a particular then there must not exist a $(p, \wedge^a, v) \in KG$. Another desirable characteristic to be modeled for data properties is that of *disjointness*. If a data property a is declared to be ‘disjoint’ then in a hierarchy of universals it is not allowed that a subclass c_1 of a class c_2 has a different value for a : if $\exists (c_1, a, v_1) \in KG$ then there may not exist a $(c_2, a, v_2) \in KG$ with $v_1 \neq v_2$. As a consequence, the classes c_1 and c_2 have to be remodeled in a way that they become siblings with respect to another superclass which they have in common. Examples:

- Be \wedge^{SportCar} a direct or indirect subclass of \wedge^{Vehicle} . The data property definition $(\wedge^{\text{Vehicle}}, \text{.MaxNbrOfSeats}, : \text{Integer})$ and the assignment in $\wedge^{\text{SportsCar}}$ with $(\wedge^{\text{PassengerVehicle}}, \text{.MaxNbrOfSeats}, 7)$ effect that every subclass of $\wedge^{\text{PassengerVehicle}}$ also has a maximum of 7 seats and that every particular of $\wedge^{\text{PassengerVehicle}}$ or any particular of a subclass of $\wedge^{\text{PassengerVehicle}}$ has also 7 seats at maximum by default.
- Value assignments for disjoint PDPs must not be overwritten. If the assignment $(\wedge^{\text{SportsCar}}, \text{.NbrOfSeats}, 2)$ and $(\wedge^{\text{Formula1_Car}}, \text{.NbrOfSeats}, 1)$ are applied then the class $\wedge^{\text{Formula1_Car}}$ cannot be a subclass of $\wedge^{\text{SportsCar}}$, because the set of single-seaters cannot be a subset of two-seaters. I.e., the two classes have to be aligned as sibling classes below a common superclass.
- Value assignments for non-disjoint PDPs may be overwritten i.e., a particular instantiates all values of the PDP of all superclasses.
- Be $(\wedge^{\text{Resource}}, \wedge^{\text{produced}}, : \text{Integer})$ a definition of an MDP. Then the attribute \wedge^{produced} can be assigned in any universal or any particular of a universal in the hierarchy below \wedge^{Resource} . E.g. $(\wedge^{\text{Lamborghini_Miura}}, \wedge^{\text{produced}}, 763)$ models the fact that 763 cars of type $\wedge^{\text{Lamborghini_Miura}}$ have been produced. Since \wedge^{produced} is an MDP its value is **not** propagated to subclass of $\wedge^{\text{Lamborghini_Miura_P400}}$ because this never makes sense, and would actually be an error source. This also implies that it is not mutable $(\wedge^{\text{produced}}, \text{.mutable}, F)$.

3.16 Object Property Instantiation

An *Object Property Definition* (OPD) has the short form $(\wedge^C, \diamondsuit^{\text{OP}}, \wedge^D)$, where \wedge^C is the domain class and \wedge^D is the range class. Then there are the following possible *Object Property Instances* (OPI): (1) $(>P1, \gg^{\text{OP}}, >P2)$, where $>P1$ must be a particular instance of a class

from $\cdot : ^C$ and $>P2$ must be a particular instance of a class from $\cdot : ^D$ (C and D can be identical), i.e. $(>P1, \gg OP, >P2)$, is appended as particular object property instance under $(^C, \diamond OP, ^D)$. (2) $(>P, \gg OP, ^C)$, or also $(^C, \gg OP, >P)$, which means that connections between the particulars layer and the universals layer are possible. In the vehicle ontology in Fig. 1, we find the following Object Property Definition (OPD): $opd = (^Vehicle, \diamond hasCombustionEngine, ^CombustionEngine)$. To find all instances of the OPD, the subclass hierarchies of $c = ^Vehicle$ and $d = ^CombustionEngine$ including the particulars of the subclass hierarchies must be examined for OPIs of the form $(x, \gg hasCombustionEngine, y)$, where it must not simultaneously hold that $x = c$ and $y = d$, otherwise the OPD itself would also be included. The superclass hierarchies of s and o are $\cdot : s$ and $\cdot : o$. The sets of particulars of $\cdot : s$ and $\cdot : o$ are $SOP(\cdot : s)$ and $SOP(\cdot : o)$. Be $>P \in SOP(\cdot : ^D)$.

3.17 Object Property Instantiation Rules

The *Object Property Instantiation Rules* (OPIR) can consequently be derived as follows, where $\diamond OPOF = \diamond OP^{-1}$. For *Particular-Particular Relationships* (PPR) and for the *Class-Particular Relationships* (CPR) and their inverses, we get:

- PPR: $(^C, \diamond OP, ^D) \models (>P1, \gg OP, >P2);$
 $(^D, \diamond OPOF, ^C) \models (>P2, \gg OPOF, >P1)$
- CPR: $(^C, \diamond OP, ^D) \models (^C, \gg OP, >P);$ $(^D, \diamond OPOF, ^C) \models (>P, \gg OPOF, ^C).$

4 Application

In this chapter we show how our methodology can be applied in such a way as to reduce the complexity in ontology modeling, while fulfilling the requirements of Multi-Level Modeling (MLM).

4.1 Class-Particular Relations

The class Lamborghini_Miura in Fig. 1 is a subclass of the class SportsCar , which itself is a subclass and instance of the class PassengerVehicle . The short Object Property Definition (OPD) for $\diamond isDesignerOf$ is $(^natPerson, \diamond isDesignerOf, ^Car)$. The relationship

$(>NPS-Marcello_Gandini, \gg isDesignerOf_LM, ^Lamborghini_Miura)$ indicates that $>NPS-Marcello_Gandini$ is the designer of the car model Lamborghini_Miura rather than each particular Lamborghini_Miura . This fulfills one of the essential requirements ([10], page 2 and 3) of multi-level modeling. Generally speaking, relationships between particulars and classes are permitted. The general instantiation principle here is illustrated by the following examples:

```

- (>Yo-Yo_Ma, »ExpertOf, ^Violin);
  (^Violin, »Expert, >Yo-Yo_Ma)
- (>NPS-Albert_Einstein, »isCreatorOf, ^Special_Theory_
  of_Relativity).

```

4.2 Powertype Subsumption

As one of the obstacles to having simpler conceptual models, we have identified the fact, that without a further differentiation of data properties instantiation it is not possible to merge pairs of classes such as (*Phone*, *Phone_Model*), (*Bird*, *Bird_Species*), ([^]*Car*, [^]*Car_Model*), etc., where the second class is the powertype of the first class, e.g., [^]*Car_Model* = Powertype ([^]*Car*). The reason for having class pairs such as ([^]*Car*, [^]*Car_model*) is that both classes in Fig. 2 have different kind of properties. The class [^]*Car* has data properties such as .*LicensePlateNbr*, .*Seats* etc. while [^]*Car_Model* has properties such as .*YearOfDesign*, .*produced* (number of produced cars) etc. Particulars pertaining to both classes [^]*Car* and [^]*Car_Model* are instantiated on the Particulars Layer (PL).

With our methodology it is now possible to subsume powertype classes below their base classes. Compared to Fig. 2, the OntoGraph in Fig. 3 shows that the [^]*Car_Model* class and the [^]*Car* class have been merged into the [^]*Car* class. At the same time the particular >*Lamborghini_Miura* has been transformed into the class [^]*Lamborghini_Miura* and the connecting object property has been changed from »iof to »subModelOf where »subModelOf ⊂ »subClassOf. This transformation was easily achieved by changing the prefix > into [^] for all triples (s, p, o) ∈ KG where s = >*Lamborghini_Miura* or o = >*Lamborghini_Miura*. Consequently, the instance >*Lamborghini_Miura* moves from the particulars layer to the universals layer. In addition, the two object property definition ([^]*Car*, ◊*Car_Model*, [^]*Car_Model*) and the object property instance (>*Toms_Lambo*, »*Car_Model*, >*Lamborghini_Miura_P400*) for ◊*Car_Model* are no longer needed.

The data property .*Model_Name* was modeled as a Propagation Data Property (PDP). Due to the Value Assignment Propagation (VAP), the triple ([^]*Lamborghini_Miura*, .*Model_Name*, *Lamborghini Miura*) it is propagated to all particulars of that car model, in this case to [^]*Lamborghini_Miura*. In the subModel [^]*Lamborghini_Miura_P400* the value for .*Model_Name* is overwritten with *Lamborghini Miura P400*. Please note that this transformation from the model in Fig. 3 to the model in Fig. 2 can be reverted. Mixing the two class hierarchies results in the original powertype classes shifting downwards and the car's base classes moving upwards in the car hierarchy. Please also note that the data property *YearOfDesign* is modeled as an MDP in Fig. 3 and as a PDP in Fig. 1. The latter is an attribute of the object property instantiation »*isDesignerOf_LM* and therefore contains more information, namely who designed the car and not only when it was

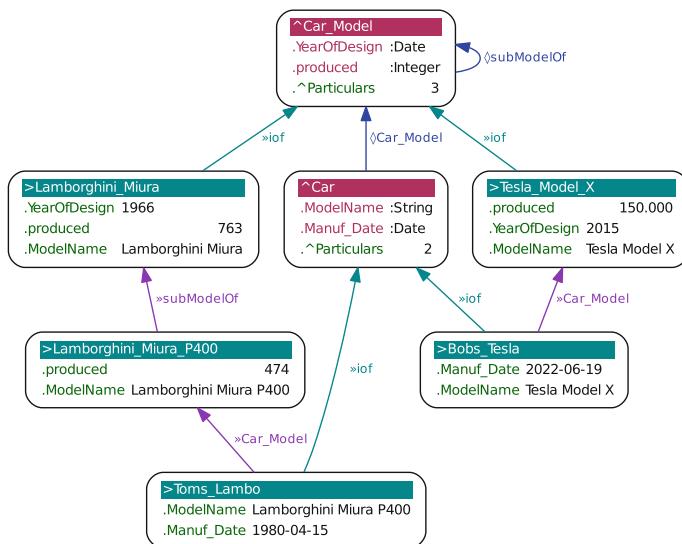


Fig. 2. Car model with materialization pattern.

designed. The example in Fig. 3 is also proof that the possibility exists of an extension with regard to non-uniform sub-hierarchies according to Neumayr et al. [10], page 2. This modeling of $\wedge\text{Car}$ avoids fragmentation and redundancy since every piece of information concerning domain objects are now described locally to $\wedge\text{Car}$ objects.

Ultimately, as called for in [3] the modeler is released as far as possible from deciding when to model instances as classes or as particulars. In comparison, by taking our approach, there is not necessity to explicitly declare a subclass as a class. For example, the instance character of $\wedge\text{Tesla_Model_X}$ is revealed by assigning values to the data properties. The class character as the second facet arises because $\wedge\text{Tesla_Model_X}$ was modeled as $\gg\text{subModelOf } \wedge\text{Car}$.

If $\wedge\text{Car_Model}$ is the *PowerType* (PT) of $\wedge\text{Car}$ then we can write $\text{PT}(\wedge\text{Car}) = \wedge\text{Car_Model}$. The question then arises, is there also a $\wedge\text{Car_Model}$ powertype, for example, $\wedge\text{Car_Model_Model} = \text{PT}(\wedge\text{Car_Model}) = \text{PT}(\text{PT}(\wedge\text{Car}))$? We think not, because what attributes can $\text{PT}(\wedge\text{Car_Model})$ have? Of course, subsets could be formed from models of cars, but that could then be done using object properties such as $\diamond\text{CategoryOf}$, $\diamond\text{SubordinateOf}$, $\diamond\text{classifiedBy}$, $\diamond\text{GroupOf}$, $\diamond\text{OrderOf}$, $\diamond\text{FamilyOf}$, $\diamond\text{classifiedBy}$ or something similar at choice being subsets of $\diamond\text{subClassOf}$. The example shows that, compared to other instantiation methods such as deep or dual deep instantiation, we do not need to annotate additional properties of the data properties such as potency, mutability or durability. By using the method of Value Assignment Propagation (VAP) for Propagation Data Properties (PDP) and by banning the application of VAP for

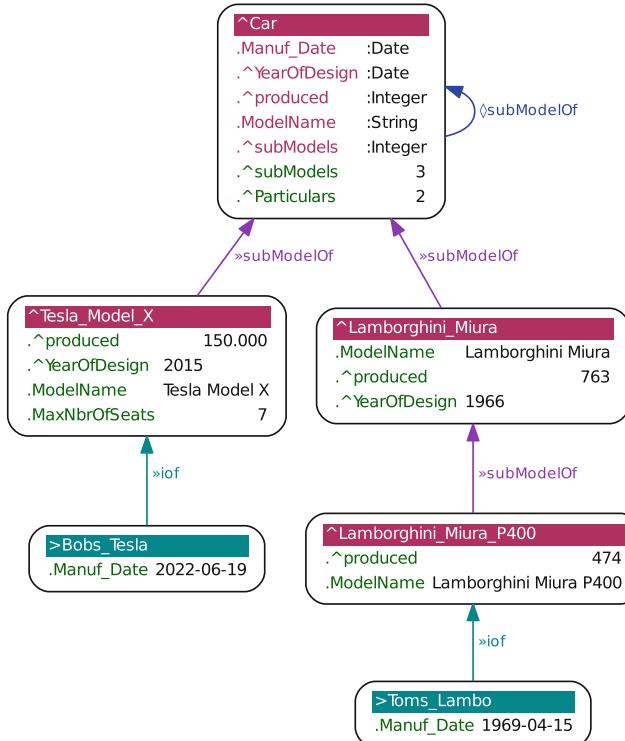


Fig. 3. Car model with powertype subsumption.

Meta Data Properties (MDP) in all cases, we arrive at the targeted instantiation behavior.

What influence can the new methodologies have on the modeling style? First, it is noticeable that in conventional modeling there are fewer classes and more particulars. When subsuming the powertypes, the powertypes then become regular classes and become subsumed under their corresponding base types. It is also noticeable that the powertypes are universals by nature. So, the question is legitimate, why are they not modeled as classes from the outset. We have already argued that this is due to the lack of differentiation for the data properties found in traditional modeling. Since the VAP method now provides this, one modeling recommendation could be to model powertypes as classes and to subsume them under their base types from the start. This automatically creates the advantage of savings possibilities for object property instances.

5 Evaluation

We at first discuss the requirements for Multi-Level Modeling (MLM) that can be satisfied or which are no longer relevant. The reduction in the complexity

of the modeling largely results from the applied naming conventions and the possibility of merging classes and saving object property instances by means of *Powertype Subsumption* (PS). Applying *Value Assignment Propagation* (VAP) shows that it is possible to manage with just two abstraction layers, namely $M_0 = \text{Particulars Layer (PL)}$ and $M_1 = \text{Universals Layer (UL)}$. The consequence of the instantiation method based on PS and VAP is that we can conclude that no *powertypes of types* or *types of types of types* exist.

5.1 Satisfying MLM Requirements

Many of the requirements posited for MLM in [4, 6, 10] then turn out to be irrelevant or can be satisfied by our approach as follows: In the universals layer (UL) with an object property $\diamond_{\text{ssr}} \in \text{SRR}$, it is possible to form arbitrary instantiation chains, since each relationship $(u_1, \diamond_{\text{ssr}}, u_2)$ between universals corresponds to an »iof relationship $(u_1, \text{»iof}, u_2)$. The principles of the organization of universals and particulars within the two layers UL and particulars layer (PL) are clearly defined by the property instantiation patterns. This also applies to the relationship between instances of the UL and PL. Fragmentation and redundancy are avoided by allowing base classes and their powertype classes to be merged into the base class. Since the classification and assignment of objects to one of the levels is only controlled by means of naming conventions, the query methodology does not change. Therefore, the requirement stipulated in [10], page 3, i.e. the capacity to navigate between different levels of abstraction with queries, is fulfilled.

5.2 Parsimony

Compared to conventional modeling, the use of naming conventions and special instantiation methods in some cases actually saves a considerable number of triples. Taking the version containing the saved triples, a modeler can revert to traditional modeling at any time. The use of prefixes in the names of ontological concepts enables direct conclusions to be drawn about the nature of the concept, while facilitating parsimony through naming conventions. When defining property definitions, rather than have two triples, each applying the object properties »hasDomain and »hasRange, we only need one triple in each case. Using the powertype subsumption method, powertype classes can be subsumed under their base classes. As a result, at least two object property instances can be saved for each pair of connected particulars. Due to the instantiation mechanisms we introduced, it is only necessary to differentiate between the universal layer (UL) and the particular layer (PL). Unlike numerous other MLM approaches, this also removes most of the restrictions on how elements can be connected between layers. This reduces or even eliminates the risk of making layer mistakes during modeling. Compared to the modeling in [5], page 26, we regard our approach to be more flexible as it allows the merger of the base class `MobilePhone` and its powertype `MobilePhoneModel` into one class, while preserving the intended semantics at the same time. We solve the problem by defining `launchDate` as

a Meta Data Property (MDP) of the class `MobilePhone`. Another argument for not distinguishing more than one level of abstraction for classes is provided by the definition of metaclasses: *In object-oriented programming, a metaclass is a class whose instances are classes.* As a result, every class would be a metaclass in any case and every class including `Class` would also be an instance of a class. In contrast to MOF, where `Class` appears on both levels 2 and 3, we only have the two layers $M_0 = \text{Particulars Layer (PL)}$, and $M_1 = \text{Universals Layers (UL)}$, while `Class` only appears on level M_1 .

5.3 Singletons

Classes and particulars can also be connected by relators and thus connect instances between the universals layer and the particulars layers. The following example corresponds to modeling *Singleton Properties* (SP) as described in [12]. The relator »`isDesignerOf_LM` in Fig. 1 is an instantiation of the object property $\Diamond\text{isDesignerOf}$ and attributes the knowledge graph with the characteristics of a *Labeled Property Graph*:

- (`>isDesignerOf_LM`, `.YearOfDesign`, 1966) and
- (`>NPS-Marcello_Gandini`, »`isDesignerOf_LM`, `^Lamborghini_Miura`).

5.4 Metadata

We distinguish between domain-specific metadata (DMD) and administrative metadata (AMD). DMD form part of the application domain modeling, whereas AMD encompass information relating to the modelers, versions of the model, as well as technical and statistical information. Both types of meta data are modelled using Meta Data Properties (MDP). As a convention, we agree that AMD attributes are defined in `^^Resource`, while DMD attributes are defined in other universals. This is similar to the Dublin Core Metadata Initiative. In our approach, the mapping to the appropriate area can be easily altered by shifting the data property definitions (DPD) from `^^Resource` to a universal or vice versa. In the example `AnnotationAssertion(a:addedBy a:Dog "Seth MacFarlane")` from OWL2⁹ the attribute `addedBy` can be regarded as a typical AMD attribute and we would place its definition in `^^Resource`.

5.5 Deep Instantiation

Compared to [1] we do not have to add and maintain a property such as *potency* to each data property. Our methodology dictates which instantiation rule is to be applied by using the `^` symbol in Meta Data Property Names or withholding it from Propagation Data Property Names. In our approach, as with deep

⁹ <https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/#Metamodeling>.

instantiation, universals can exhibit both instance and type character simultaneously. In [11], page 9, the base classes `Book` and `Car` are modeled as subclasses of `ProductCategory`. We would merge the `ProductCategory` with the `Product` class and avoid a forced division into classification levels. The modelling in [7], page 2, could be simplified as follows: The powertype `BirdSpecies` can be subsumed under its base type `Bird` by converting the data properties in `BirdSpecies` from PDPs to MDPs and moving them into the base class `Bird`. `GoldenEagle` and `EmperorPinguin` become direct subclasses of `Bird`. So, instead of having two parallel class hierarchies for `Bird` and `BirdSpecies`, we have just one for `Bird`. This also resolves the deficiency in their modeling whereby the characteristics of all other species would have to be modeled redundantly as in `BirdSpecies`. We believe that our modeling methodology is in line with this and not only complements the authors' approach, but also extends it to streamline Multi-Level Modeling (MLM). Our approach makes it possible to solve the unresolved problem from [3], i.e. the inability to state that all polar bears have white fur. Using the VAP method we simply model `(^Polar_Bear, .ColorNameFur, white)`. We therefore do not require two different inheritance mechanisms for properties such as `TypeInheritedProperty` and `BroaderInheritedProperty`, rather only the distinction of whether or not the value assignment is propagated.

6 Conclusion

Ontology engineering follows strict rules. We regard the triple representation of a knowledge graph as the form of representing knowledge which is common to at least all the approaches discussed in this paper. Triples can be transformed into the syntax of any modeling paradigm such as RDF/RDFS/OWL etc. The key outcome is that we have shown that, by applying a differentiated instantiation methodology, it is possible to remodel ontologies in a way that at the same time complexity can be reduced and transparency enhanced. The introduced naming conventions combined with instantiation rules are a minor necessary prerequisite to enable the advantages of our modeling approach to be utilized. To achieve the same goal, other approaches to multi-level modeling (MLM) require extra added information that cannot be stored in the same compact manner. Our approach recognizes classes as instance of other classes in general. Compared to other approaches our meta modeling approach only needs particulars and universals. For the modeler, the key decision regarding the right ontology design is narrowed to choosing whether a data property should be defined as a propagation data property (PDP) or as a meta data property (MDP).

6.1 Future Work

We will continue to work out how our presented instantiation methods can be used to systematize the partitioning of classes in such a way that considerable

savings potential can be applied for storing ontologies. In addition, we will examine how the application of our methodology affects the simplification of ontology design patterns.

References

1. Atkinson, C., Kühne, T.: The Essence of multilevel metamodeling. In: International Conference on the Unified Modeling Language, pp. 19–33 (2001)
2. Bense, H.: The unique predication of knowledge elements and their visualization and factorization in ontology engineering. In: Kutz, O., Garbacz, P., (eds.) Proceedings of the Eighth International Conference (FOIS 2014), pp. 241–250, Rio de Janeiro, Brazil (2014). <https://doi.org/10.3233/978-1-61499-438-1-251>
3. Bense, H., Humm, B.: An extensible approach to multi-level ontology modelling. In: KMIS 2021, 13th International Conference on Knowledge Management and Information Systems (2021)
4. Brasileiro, F., Almeida, J.P.A., Carvalho, V.A., Guizzardi, G.: Applying a multi-level modeling theory to assess taxonomic hierarchies in wikidata. In: Wiki Workshop at 25th International Conference on Companion World Wide Web, pp. 975–980 (2016)
5. Carvalho, V.A., Almeida, J.P.A.: Toward a well-founded theory for multi-level conceptual modeling (2016)
6. Frank, U.: Toward a new paradigm of conceptual modeling and information systems design. *Bus. Inf. Syst. Eng.* **6**(6), 319–337 (2014)
7. Guizzardi, G., Almeida, J.P.A., Guarino, N., Carvalho, V.A.: Towards an ontological analysis of powertypes. In: International Workshop on Formal Ontologies for Artificial Intelligence (FOFAI) (2015)
8. Guizzardi, G., Figueiredo, G., Hedblom, M.M., Poels, G.: Ontology-based model abstraction. In: IEEE Thirteen International Conference on Research Challenges in Information Science, Brussels (2019)
9. Humm, B., Archer, P., Bense, H., Bernier, C., Goetz, C., Hoppe, T., Schumann, F., Siegel, M., Wenning, R., Zender, A.: New Directions for Applied Knowledge-Based AI and Machine Learning. Springer, Informatik Spektrum (2022)
10. Neumayr, B., Grün, K., Schrefl, M.: Multi-level domain modeling with M-objects and M-relationships. In: 6th Asia-Pacific Conference on Conceptual Modeling (2009)
11. Neumayr, B., Schrefl, M., Thalheim, B.: Modeling techniques for multi-level abstraction (2008)
12. Nguyen, V., Bodenreider, O., Thirunarayan, K., Fu, G., Bolton, E., Rosinach, N.Q., Furlong, L.I., Dumontier, M., Sheth, A.: On reasoning with RDF statements about statements using singleton property triples (2015)
13. Odell, J.J.: Power types. *J. Object-Orient.Programm.* **7**(2), 8–12 (1994)
14. Pirotte, A., Zimanyi, E., Assart, D., Yakusheva, T.: Materialization: a Powerful and Ubiquitous Abstraction Pattern, pp. 630–641. VLDB, Morgan Kaufmann (1994)



E-Step Control: Solution for Processing and Analysis of IS Users Activities in the Context of Insider Threat Identification Based on Markov Chain

Oksana Nikiforova¹(✉), Vitaly Zabiniako², and Jurijs Kornienko²

¹ Riga Technical University, Riga, Latvia

oksana.nikiforova@rtu.lv

² ABC Software Ltd., Riga, Latvia

Abstract. As the digitalization of everyday life develops and the use of information technology in various business domains increases with it, the demand for the existence of automatic tools for information protection and security also increases. In recent decades, information technology specialists have learned to sufficiently protect information systems against the evil actions of external attackers and to identify potential points, where systems can be hacked, but information systems insiders' threats for unauthorized use of information are increasing. One of the approaches to reduce the risk of unauthorized data use by internal users of information systems is to base the monitoring of information systems usage on the analysis of users behaviour. The authors of the paper have implemented such an approach in the product "e-StepControl", in which the work of each user in the information system can be analysed according to the typical behaviour model of this individual, and in cases where the user acts differently from the expected behaviour, a security incident can be identified due to the unexpected (therefore, suspicious) activity of the user. Also, such security incidents can be identified by comparing the behaviour of an individual user with other users with the same or similar behaviour within the information system usage, in other word—with the expected behaviour of the representatives of this user's group or class. This grouping of users is essentially the machine learning-based performance of the task of clustering users of information systems according to such parameters as the activities performed by the user and their regularity, the sequence of activities to be performed, attributes of users and user sessions. Both individual and group user behaviour models can serve to identify security incidents, which are further confirmed or rejected by a security specialist by drawing relevant conclusions.

Keywords: Insider threats · Machine learning · User behaviour modelling · e-StepControl

1 Introduction

With the development of IT capabilities and the expansion of the spectrum of usage of information systems (IS) in solving multiple business problems in various fields, the problem when security threats and unauthorized use of data comes from internal users of

IS is becoming more and more relevant (to a point of being equal to the potential damage caused by external hackers). The traditional ways to control the access of registered and authenticated IS users to the electronic information is to grant them IS access rights in scope of the authorization process. As a result, users are granted access to data only to the extent and in such manner, which is necessary for the performance of their work duties. However, the user authentication and authorization mechanism only partially solves the problems of proper control of information availability and its usage. Another approach to reduce this security risk is to base the monitoring of the usage of IS on the analysis of the typical users behaviour models (also known as *UBA/UEBA – user (and entity) behaviour analytics*). This paper offers an approach to insider threats identification by the usage of the supporting tool called “e-StepControl” [1], which is based on the following two main sub-processes:

1. *Self-learning* of the e-StepControl solution by creating typical behaviour models of individual IS users and groups of such users. AI/ML self-learning algorithms are used for creation of these behaviour models—both for each user individually and for a corresponding users (which are based on the same or similar access rights) group.
2. *Monitoring of users activities*. After individual and group behaviour models were created, each new user session is evaluated regarding its conformance with typical user behaviour—by comparing new IS usage sessions content with both individual and group behaviour models [2]. Comparing the content of new sessions not only with an individual, but also with a group behavioural model is of great importance, otherwise (if only the individual user model would be used), some anomalies would be invisible (in case if this user already performed suspicious actions during the initial process of e-StepControl self-training) and the algorithm would consider such suspicious actions as a typical behaviour of a user. In contrast, comparing the behaviour of an individual user with a behaviour model of corresponding users group will give a more accurate evaluation result.

The paper is structured as follows: Sect. 2 outlines the background and related work in the field of insider threat identification, Sect. 3 describes the formalized mathematical model for evaluation of IS users activities audit data, Sect. 4 provides an insight onto application cases of the e-StepControl solution. Lastly, conclusions are being made and future research work is outlined.

2 Background and Related Work

There are IS where insiders are mostly company workers (as internal users of IS with appropriate rights and privileges), or the IS might be offered for usage for a wider range of external users (but they are authorized to use the system with login data). IS security studies show that unauthorized use and even theft of information is not only carried out by external hackers, but also by internal business environment IS users around the world. E.g., according to the source [3], the percentage of insider threat incidents in 2020 was divided between negligent insiders (~61.4%), stolen credentials (~24.8%) and malicious insiders (~13.9%). According to the source [4], the most dangerous types of insiders who do violate data access are:

- Privileged users and administrators—these users pose a unique concern because they have complete access to the organization's critical data and infrastructure. Due to their high level of access, privileged users malicious behaviour is hard to spot because they don't explicitly violate any cybersecurity laws when they access sensitive resources.
- Regular employees—compared to privileged users, normal users are not as dangerous, yet they can still cause harm to an organization. For instance, they might misuse company data, set up illegal software, send private emails to the wrong recipient, fall prey to a phishing scam, etc.
- Third parties and temporary workers—vendors, business partners, and temporary workers might not comply with the cybersecurity policies and procedures adopted by the company or might unintentionally break these. In order to infiltrate a secured perimeter, hackers may use a third-party vendor with low security level.
- Privileged business users and executives—c-level executives get access to an organization's most sensitive and private data. Users in this group may misuse their information for insider trading, personal gain, or business or governmental espionage.

According to the source [5], the top insider threat actors are: privileged IT users (63%), managers with access to sensitive information (60%), contractors and consultants (57%) and employees (51%). Also, regarding the insider threat frequency of attacks, 68% of organization say that insider attacks have become more frequent over last 12 months, 61% of companies have had an insider attack in the past year, in 60% of cases data breaches are caused by insider threats, 60% of organizations had more than 20 incidents in the past year, insider-caused cybersecurity incidents number is increased by 47% since 2018. The same source also states that the top motivations for insider's attacks are: fraud (55%), monetary gain (49%) and IP theft (44%).

All owners of electronic information, who may potentially suffer losses in case of its unauthorized use, are interested in the solution of the problem. International studies and survey data accurately focus both the market niche where appropriate cyber security solutions are needed, as well as the risk audience that poses threats. E.g. [6] states that among the users groups who do pose the largest security risk to organizations are: privileged IT users/admins (60%), contractors/consultants and temporary workers (57%) and regular employees (51%).

The main groups of industries (business segments) that are at risk and can experience various types of losses for data owners are:

- IS containing business critical or sensitive data;
- healthcare (patient's data, medical history);
- financial sector (customer's data, account balances and credit history);
- various state registers which contain personal/restricted access data;
- military domain;
- etc.

All this indicates the need to rapidly develop the field of solutions and tools which would give an opportunity to prevent and fight such misbehaviour cases of internal users, which can bring heavy financial and reputation loses to many companies around the World.

UBA/UEBA solutions are very relevant, both in Europe, the US and elsewhere in the World. The information available in the source [4] shows that the detection and

prevention of violations of internal business laws in organizations is becoming a serious problem and organizations feel increasingly vulnerable, because 98% of companies believe that insider threats are becoming more frequent, and 53% of companies state that detection of insiders attacks has become harder since shifting towards the modern cloud infrastructure.

A number of competing products have been identified that are positioned as operating in the UEBA segment, the most popular examples of these would be: *Splunk* [7], *Fortinet* [8], *Forcepoint* [9] and *Teramind* [10]. The UEBA segment is relatively broad and this category includes products that analyse atypical behaviour of IS users by monitoring the following data sources:

- authentication events;
- devices which are used for connections;
- DNS log files;
- business applications logs;
- audit records of users activities;
- data usage log files from various sources and databases;
- TCP/IP/UDP network traffic;
- etc.

Each of the competing UEBA products is specialized for IS users behaviour anomalies analysis using one or more of the mentioned data sources, but there is no such a single universal UEBA product that would be equally well specialized for the integrated analysis of information obtained from all mentioned data sources.

Moreover, each of these solutions has its own problems and disadvantages (which can be found, e.g., on widely available reviews aggregators):

- *Splunk*—expensive for very large data volumes, difficult to implement optimizing searches for speed, less reliability, account monitoring requires advanced knowledge and also prior configuration, dashboard customization might be needed, training is required to be able to use this tool properly, etc.;
- *Fortinet*—the unnecessary complication of the network, time-consuming configurations, can seem pricey for small companies on limited budgets, there could be some improvement to the Logging Services to include more details, some features are only available in command-line interface, etc.;
- *Forcepoint*—getting unwanted incidents using pre-defined policies, deployment on multiple servers is difficult, connectivity is not secure, additional cost to professional support, etc.;
- *Teramind*—with antivirus in place it may be hard to install because it often flags it as a virus, for the new users the experience might be overwhelming because of quantity of features, it is somewhat confusing to set up triggers that gauge productivity, etc.

Market research indicates a significant increase in market share specifically in the UEBA segment, where we conduct research and develop the e-StepControl solution. By using such new, innovative UEBA-type data security solution as the e-StepControl, managers of the mentioned business segments and according data security officers will be able to efficiently manage trending security risks of insider threats. Within companies, the main beneficiaries are:

- High-level managers and business owners who will gain automated control over employee's activities in the IS.
- Security officers and administrators who will receive notifications regarding users performing suspicious or unusual actions. This allows investigating these cases in detail in order to prevent this type of problem in the future (or to carry out additional user training, if such cases will be discovered and needed).
- IS developers and/or support staff, who will have access to: (1) information regarding problem areas in the IS infrastructure; (2) information regarding the correspondence of users roles implemented in the IS to the real usage of these roles by IS users. Timely troubleshooting of IS infrastructure increases system efficiency and is especially useful in cases when this infrastructure is frequently updated with new patches.

We confirm according market trends with statistical data from the sources regarding a significant increase in the market share of UEBA products. The source [11] states that, based on the global UEBA market report which have been published by value market research, the value is expected to reach USD 2453.4 Million by the year of 2024. The source [12] states that 59% of IT security leaders say that insider risk will increase in the next two years.

Actual negative experiences and losses that prompt companies to think about mitigating the risks of insider threats, according to the source [4] are as follows: Companies from North America suffer the most from insider attacks and their consequences—the average cost in this region increased from \$11.1 million to \$13.3 million; the average total spending on a single insider threat incident (including monitoring, investigation, escalation, incident response, containment, ex-post analysis, and remediation) also went up from \$513,000 to \$756,760.

Therefore, the analysis of security incidents of IS shows that threats created by users of internal IS are on the rise. Recent studies show that on average, 60% of all detected breaches are committed by internal users of the organization. This is a huge risk for which the data controller must take responsibility.

The average cost of preventing the consequences of one incident is generally high, so it is more profitable to invest in cyber security solutions than to struggle with the elimination of the consequences of violations, which will clearly cost very expensive in financial terms (as well as other losses, e.g., in relation to reputation risk, loss of potential customers, etc.).

To address this problem, user behaviour can be tracked in audit data [13] and user behavioural models of individual's users and users groups can be created automatically. These behavioural models can be used to identify anomalous deviations in subsequent users' actions to immediately alert the data security officers of such suspicious cases.

Existing research (e.g., [5]) provide recommendations and indicate the main action strategy and cyber security tools that help to minimize the risks caused by users of internal IS, where the use of UBA-type tools is very high—about 55% of surveyed companies use tools and activities that reduce insiders threats.

The need for these types of methods and products is on the rise because:

- The demand for cyber security solutions in the world as a whole is growing.

- Along with the development of IS, the amount of electronically processed data increases, which in turn contributes to the increase in violations of the use of this data, which is also confirmed by appropriate market researches.
- The risk of negative reputation, which can potentially arise for the merchant, if, for example, unauthorized use of sensitive information is discovered—there are unpredictable losses.
- Elimination of the consequences of late detected violations may require huge time and financial resources.
- Compliance with the requirements of the new GDPR regulation regarding the processing of personal data is strict, but non-compliance with the regulation threatens with significant financial penalties.

The conclusion is clear that the topic of this research is and will be relevant, the UEBA market share is growing and products that can directly measure the behaviour of IS users at the business level of the system (rather than based on some traffic statistics), are relevant and this market niche is not saturated yet. Therefore, our product e-StepControl has a good growth potential.

The approach offered in the paper and integrated within its supporting tool e-StepControl is based on the idea of analysing the typical activity of IS users, profiling it for both individual users and users' groups (or classes), which is described in the next section of this paper. Authors of this paper propose the approach for insiders' threats identification, which replies to all challenges stated above, namely:

- it proposes a set of collectable initial data of user's behaviour, which are registered in system usage logs and therefore does not require complicated integration activities to access the data needed to run the method,
- it defines the algorithm to automatically identify cases of insiders' threats thus minimising the amount of data, which the system security administrator needs to process every day, and
- it provides the tool to support initial data collection and expected output data visualisation in the form suitable for security incidents analysis with an ability to accept or reject each particular identified case.

3 Approach to Obtain and Visualise Suspicious Insiders Activities from Audit Log Data

Any problem could be seen as a function which processes particular input data and produces predicted output outcome, according to System Theory [14]. General schema of the solution described in the paper is shown in Fig. 1, where the function of source data, corresponding target data and methods used to obtain ones from the others are presented. The information expected on the security administrator's side as a list of sessions, where the suspicious activities are identified, are the output data of the approach offered in the paper and supported by e-StepControl. To produce this list the data of IS user behaviour is collected and passed as an activity log [13] to the algorithm, which updates this data, visualises user sessions in the form of timelines diagram and outputs the list of sessions activities.

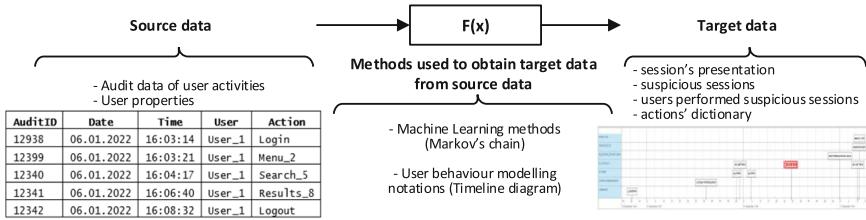


Fig. 1. Function of source data, corresponding target data and methods used to obtain ones from the others.

To make the application of the approach automated it is required to search for some type of data which could be generated from the IS usage audit in real time and won't require any efforts from IS support staff. IT system audit data of user's actions may hold all the necessary information. This process is shown at a high level of abstraction in Fig. 2.

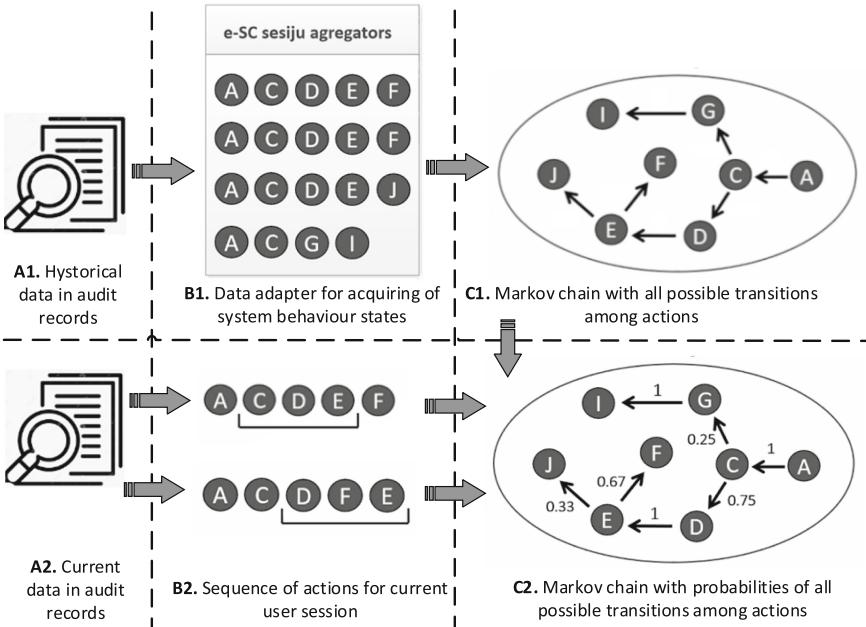


Fig. 2. The functioning of the e-stepcontrol solution for determining suspiciousness of users behaviour (at a high level of abstraction) [15].

Thus, the first step of integration the e-StepControl solutions with a target IS under monitoring is related to obtaining the log data regarding IS users business activities. For this purpose, a data retrieval adapter is usually required in order to export such log data from different sources (e.g. relational/object databases, event logs, text files with IS logs data, etc.) into the e-StepControl analysis environment (data aggregator).

The functioning of e-StepControl for determining suspiciousness is based on the usage of the Markov chain model and relevant algorithms. After the deployment of the e-StepControl solution, data regarding the target IS usage is accumulated for a certain period of time (e.g., 1–3 months), the creation and training of e-StepControl users behaviour models (in the context of the accumulated users behaviour data) takes place, and as a final result—these behaviour models are used for the analysis of new users sessions and determining suspiciousness of these [15].

The formalization of the corresponding sub-steps at the level of mathematical models and algorithms is detailed below. The set of actions performed by the user in his/her IS work sessions serves as the basis for constructing of the graph of according user behaviour model. Initially, each transition between performed actions is recorded in the so-called “sessions graph”. In case if the specific transition between actions is repeated—the numerical value of the respective graph edge is auto-incremented—see Fig. 3.

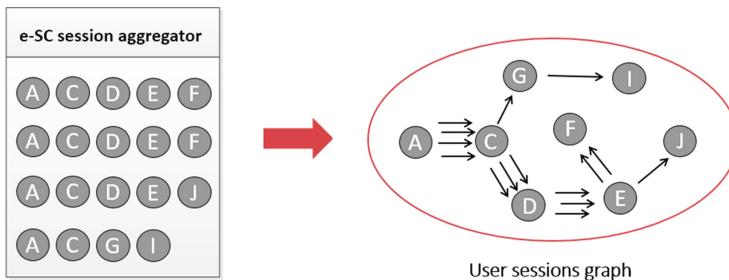


Fig. 3. Construction of the user behaviour graph.

During the training process the initial user behaviour graph is refined with probabilities in the correspondence with Markov chain model [16].

When the number of sessions of user's actions used for training becomes high enough (e.g., this can be configured by the length of the training period, or the minimum number of performed sessions that the model needs to learn before switching to the analysis mode)—this user behaviour model is considered to be trained sufficiently enough, and can be used for analysis of new actions sessions of this particular user.

As the result, the refined user behaviour graph is highlighted with probability values of the actions to be performed, which gives an ability to analyse new incoming sessions in accordance with this refined user behaviour graph. Two types of behaviour models are maintained by the e-StepControl solution—see Fig. 4:

- *The individual user behaviour model*—reflects the typical activities of the particular user in the target IS being monitored (it is built only from the actions of this particular user within sessions that he/she performed);
- *The behaviour model of a group of users*—it is created for several users with similar attributes in the target IS being monitored, according to clustering principles [17–19]. Building this model is based on the assumption that users with a similar role(s) in the IS do behave statistically similarly [2]. Such model is useful in case when the suspiciousness of the previously unknown user's actions needs to be evaluated, as

well as in the case when the specific user performed suspicious actions from the very beginning (during the training period of the e-StepControl solution) and this affects the content of his individual behaviour model.

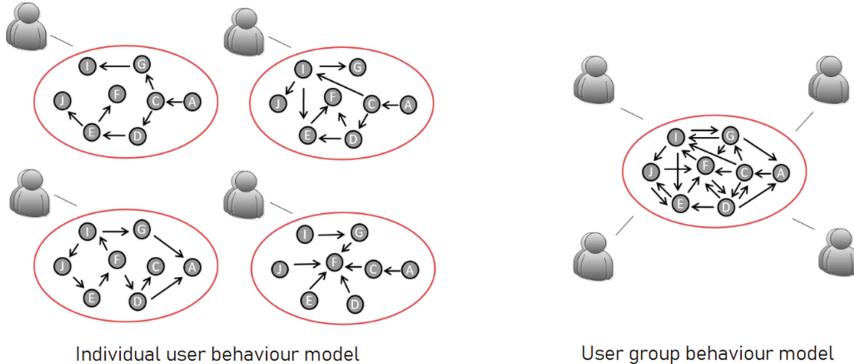


Fig. 4. Types of behaviour models in the e-stepcontrol solution.

Despite the semantic differences in the content of these two models types, the procedures for creating and updating these are identical.

In case if the user's actions, which were not common to him during the previous system usage (training) period, are detected—the relevant session should be marked as “suspicious”, allowing the security specialist of the monitored target system to carry out its further detailed manual evaluation [20].

To ensure this—each new session of this user is verified against his behaviour model, evaluating to what extent this session corresponds to usual user behaviour. The evaluation principle is shown in the Fig. 5.

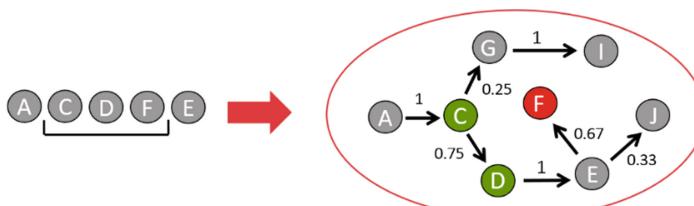


Fig. 5. Evaluation of suspiciousness level of new user session.

A fragment of the analysed session of actions “C—D—F” is shown, which is compared against the content of the behaviour model. The transition “C—D” is known and has an allowed probability of 75%, while the transition “D—F” is considered with the probability 0%. As a result, the session in whole produces a number of expected/not expected or allowed/not allowed transitions. In case of this number is higher than defined for the analysed problem domain [20], the session is identified as suspicious, otherwise

the session is acknowledged as typical and is added to aggregator. In case if a behaviour anomaly is not identified for a particular session—this session is used to update the user's individual and corresponding group models. In this way the so called “dynamic training” is ensured, when the daily usage of the monitored target IS increases the reliability of the e-StepControl solution analysis results.

4 Application Case

The e-StepControl solution implements the method described in the paper. The screenshot with session presentation, where suspicious actions are highlighted, is shown in Fig. 6.

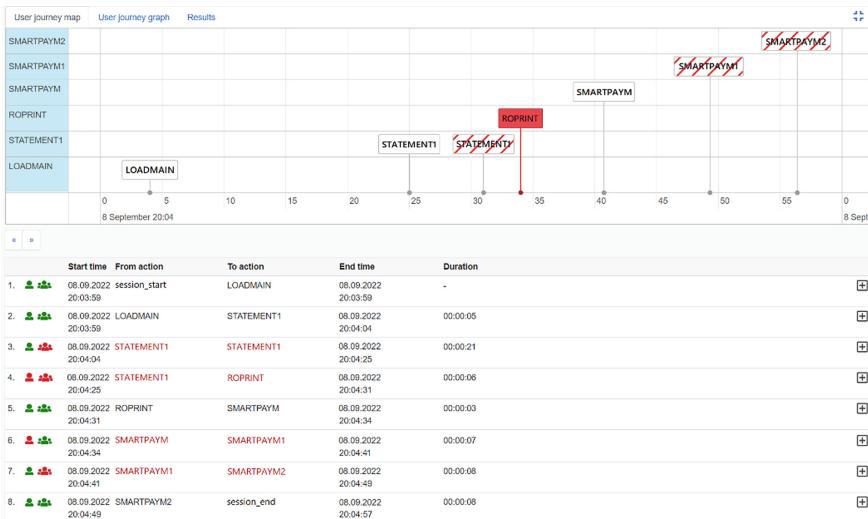


Fig. 6. Analysis of the particular IS user's session content.

On the upper part of the figure, the session content is presented in a form of timelines diagram, where each subsequent user's action is depicted as a separate element with the label of according action identifier. The colour and background of each such element depicts how suspicious the particular action in a sequence is (according to the analysis procedure explained in the previous section).

The lower part of the image contains a detailed table of performed actions and transitions between these in scope of user's session. Additional attributes are shown for each transition between actions (e.g., transition's start/end time, according actions, transition duration time and suspiciousness type). Two coloured icons at the left part of each row in the table show how suspicious a particular transaction is in relation to individual behaviour model (marked by a single user icon) and group behaviour model (marked by a multiple users icon). Non-suspicious transactions are represented with green icon colour, while suspicious transactions are represented with red colour.

In case if a particular transition is evaluated as suspicious according to both individual and group behaviour models—the corresponding action on the timeline is highlighted as fully red, if the suspicion is evaluated up to one indicator—the corresponding action is highlighted as partly (striped) red. Otherwise, if the transaction was not recognized as suspicious, the action is nonhighlighted on the timeline diagram.

Considering that e-StepControl is an insider's threats identification tool, and taking into the account appropriate reasons of security and undisclosed information—it is not permitted to publish real data regarding security incidents identified by e-StepControl in real problem domains (such as finances, insurance, healthcare, law enforcement cases, etc.). Still, in order to demonstrate potential usage and abilities of this solution—authors do outline several typical cases of e-StepControl usage as short obfuscated scenarios.

1st scenario (banking domain)—while analysing the set of e-StepControl alerts regarding suspicious activities, the security administrator finds that one of the IS users had started behaving atypically and this was evidenced in multiple work sessions. By using the ID of one such suspicious session, the security administrator determines that it was created with credentials of the user John. Many atypical transitions between IS actions are found, which do not correspond to John's behaviour model which is constructed during e-StepControl training time period. It was also revealed that the web browser version is used, which John used very rarely, and only when working from his home computer. Additionally, such work sessions are performed during the morning hours, when John was usually on his way to clients and did not use the IS. The e-StepControl analysis algorithm took into consideration all these factors and calculated the suspiciousness level which significantly exceeded the allowed threshold set by security staff, which, in turn, triggered the e-StepControl alert.

A face-to-face meeting with John was immediately arranged. John could not recognize that he had performed the actions recorded in the audit trail, these do not correspond neither to his typical system usage practice, nor to the typical usage time. During the conversation, the parties found out that John had recently received an e-mail with a request to connect to one of the customer's IS and to enter additional data regarding monthly report. For convenience, a URL link was added directly to the e-mail, for quick authentication and filling in the required data. John clicked on this link and entered the data in a separate webform as it was asked in the e-mail.

As a result of an in-depth data analysis and this conversation, the security administrator finds out that in such way hackers had stolen John's credentials and from that moment were able to freely work in the IS under his username and had a full access to a significant set of company's confidential information. Luckily, the anomaly warning was triggered in time, and it was quickly revealed that hackers were hiding under John's user account.

2nd scenario (insurances domain)—while analysing the set of suspicious activities, the security administrator determines that the atypical sessions were created with Mike's IT system credentials. The sequences or transitions of the IS actions did not correspond to Mike's typical behaviour model—frequent searches of customer's data outside the region under Mike's responsibility was revealed. Additionally, such IS work sessions took place both during and after working hours. As with the previous scenario, the

threshold of suspiciousness was significantly exceeded, and several work sessions were marked with security alerts.

After an in-depth study of these work sessions, it was concluded that the identified anomaly was justified and the reason for it was the atypical behaviour in an atypical volume and time (which could not have been detected manually without the assistance of e-StepControl). A face-to-face meeting with Mike and his manager was arranged. It was also found out that Mike was going to submit his resignation soon. By mapping all this data together during the conversation, a conclusion was made about Mike's disloyal attitude towards his current employer. It turns out that he had a job offer from a competitor with the promise of a higher salary if he would obtain the data regarding profitable customers, which would allow him to lure those customers with favourable contract terms to the competitor, so Mike did a massive information search and retrieval, hoping that by working remotely he is protected from an unwanted attention of the security staff.

3rd scenario (healthcare domain)—while analysing the set of e-StepControl alerts, the security administrator found out that one of the doctors was frequently accessing information that he should not have access to, i.e., sensitive medical data of patients not under his supervision. During the investigation of the situation, it was found that Adam (although the duties of his job position did not allow this), had an access to such very extensive and confidential information, because the administrator of the healthcare IT sub-system granted this user rights unjustifiably. This mistake was made because he mixed up usernames of several doctors.

Immediately after the cause of the incident was discovered, Adam's user rights were reduced in accordance with his position and duties. Incident detection in this case was provided by the e-StepControl ability to create individual profiles of each user and group profiles of employees with the same positions. A profile accumulates the typical activities of an employee and the corresponding group of users. By comparing Adam's profile with the behaviour model of other doctors of this job group, e-StepControl was able to recognize differences that occurred when performing atypical activities in this particular case.

4th scenario (law enforcement domain)—while analysing the set of suspicious activities, the security administrator found out that David had started to use the functions available in the criminal cases IS in an atypical manner for him. All transitions between IS function calls happened at a very high speed, with a frequency uncommon for a human being's abilities. Additionally, David's usual browser type was not used.

It was found out that David's atypical actions were caused by requests from scanning software that performed SQL injection attempts. Immediately, the access for third-party software that David had tried to use was disabled, and David had to explain to his administration exact reasons why he was using such potentially dangerous software.

In all these cases the usage of the e-StepControl tool was highly beneficial to company's security specialists in revealing potential sources of malicious insider threats.

5 Conclusions and Future Research

Insiders' threats are a trending topic in modern IT systems security domain. A number of tools already exist, which do help to identify such cases in scope of UBA/UEBA procedures, although no universal solution has yet been developed, which could reveal all according potential suspicious cases (including the analysis of business audit trails of IS usage). e-StepControl is an original solution in this domain, which (in comparison to such tools as previously mentioned *Splunk*, *Fortinet*, *Forcepoint*, *Teramind*, etc.), allows to enable such an analysis of business audit trails, based on the evaluation of IS users activities against according individual and group behaviour models.

The idea of the method used in the e-StepControl solution is based on the analysis of the user's behaviour and identification of cases when it began to suddenly differ from his daily routine duties and activities patterns.

The work of each individual user of the IT system can be analysed based on the audit records of the user's actions performed in the system itself, creating a typical behaviour model of this particular user according to its business functionality. For this, the idea of analysis of individual user behaviour, based on his own behaviour model is expanded with user clustering, which is essentially the automatic grouping of IT users with the same or similar behaviour models into separate classes using machine learning algorithms. The user class behaviour model can further serve to identify anomalies by comparing the behaviour of the individual IT system user against the user class model.

The approach offered in the paper is used in the existing e-StepControl solution, which provides the following practical benefits:

1. Analysis of individual behaviour models allows to identify suspicious cases, when a particular user suddenly starts to act differently (in comparison with his previous actions patterns), while the additional analysis of groups behaviour models enables great further improvement of overall quality and reliability of automated security evaluations and deductions.
2. Automatic users clustering improves the quality of user grouping by creating optimal groups of users not only with formally matching rights, but also including users with a similar range of rights in one group, where the behaviour is not significantly different. This allows solving the problem case when grouping users in IS with a large number of users, a very large number of groups with an insufficient number of users for analysis is formed by formal and manual methods. For such groups, it is impossible to create a model of user group behaviour of higher quality, which significantly reduces the accuracy of detecting atypical user behaviour.
3. e-StepControl solution is able to recognize users who have combined roles, or users who work in the IT system with privileges combined from several jobs.
4. The high degree of automation of user behaviour model analysis enables to regularly look for insiders threats, which primarily provides opportunities to monitor user behaviour in real time and to quickly respond to reports of incidents of suspicious behaviour, and in addition:
 - To recognize users who have been granted too broad rights, which the user actually does not use in practice.

- Recognize users whose accounts are remaining in the IT system but are not being used.

Thus, the method will be able to regularly provide recommendations both for unreasonably large reductions in rights allocation and for closing unused accounts, if such are found.

Both the analysis of individual user behaviour and the automatic clustering of users and the analysis of user behaviour against the typical analysis of a group allows solving advanced suspicious identification problems that have not yet been solved in other UEBA solutions. And by using both individual and group behaviour models together in the analysis of user behaviour, it will be possible to create a cyber security solution with decision-making support functions and real-time reporting of security incidents regarding the non-compliance of a user's activity with the permitted behaviour of the group. Currently such an approach cannot be achieved with existing knowledge. So, the method offered in the paper and applied in e-StepControl significantly improves the existing opportunities to detect violations of data usage in the electronic environment caused by internal users of the organization's information systems.

Acknowledgment. The research leading to these results has received funding from the research project “Competence Centre of Information and Communication Technologies” of EU Structural funds, contract No. 5.1.1.2.i.0/1/22/A/CFLA/008 signed between IT Competence Centre and Central Finance and Contracting Agency. The research title is “Development of a method for analysis and automatic grouping of information system users with similar behaviour, using an AI/ML approach”. The project is co-financed by the Recovery Fund of the Action Program “Latvian Recovery and Resilience Mechanism Plan 5.1.r. 5.1.1.r. of the reform and investment direction “Increasing productivity through increasing the amount of investment in R&D” reforms “Management of innovations and motivation of private R&D investments” 5.1.1.2.i. investment “Support instrument for the development of innovation clusters” implementation rules within the competence centres” framework.

The intellectual property “System and Method for Detecting Atypical Behavior of Users in an Information System by Analyzing their Actions Using a Markov Chain and an Artificial Neural Network” is submitted to World Intellectual Property Organization on 2021/02/26.

References

1. e-StepControl Homepage. <https://estepcontrol.com>. Last accessed 14 Jan 2023
2. Osipov, P., Cizovs, J., Zabinako, V.: Distributed profile of typical user behavior in a multi-system environment. In: Proceedings of 9th international scientific conference of economic and business development. Latvia, pp. 377–386 (2017)
3. Insider Threat. Definition, Types, Examples and Prevention Strategies. <https://heimdalsecurity.com/blog/insider-threat>. Last accessed 11 Dec 2022
4. Insider Threat Statistics for 2022: Facts and Figures. <https://www.ekransystem.com/en/blog/insider-threat-statistics-facts-and-figures>. Last accessed 14 Oct 2022
5. Crucial Insider Threat Statistics: 2023 Latest Trends & Challenges. <https://financesonline.com/insider-threat-statistics>. Last accessed 14 Jan 2023
6. Haystax Technology—Insider Attacks, Industry Survey. http://haystax.com/wp-content/uploads/2017/03/Insider_Threat_Report_2017_Haystax_FINAL.pdf. Last accessed 28 June 2022

7. Splunk Homepage.: https://www.splunk.com/en_us/data-insider/user-behavior-analytics-ueba.html. Last accessed 29 Dec 2022
8. Fortinet Homepage.: <https://www.fortinet.com/products/ueba>. Last accessed 14 Jan 2023
9. Forcepoint Homepage.: <https://www.forcepoint.com/product/ueba-user-entity-behavior-analytics>. Last accessed 02 Dec 2022
10. Teramind Homepage.: <https://www.teramind.co/solutions/user-entity-behavior-analytics>. Last accessed 14 Nov 2022
11. Yousef, R., Jazza, M.: Measuring the effectiveness of user and entity behavior analytics for the prevention of insider threats. *J. Xi'an Univ. Arch. & Technol.*, **8**, 175–181 (2021)
12. Code42, 2021 Data Exposure Report. <https://www.code42.com/resources/reports/2021-data-exposure>. Last accessed 8 Jan 2023
13. Nikiforova, O., Zabiniako, V., Kornienko, J., Gasparoviča-Asīte, M., Siliņa, A.: Solution for analysis of IT systems users behaviour by using AI/ML algorithms. *Appl. Comput. Syst.* **26**(2), 107–115 (2021). <https://doi.org/10.2478/acss-2021-0013>
14. Grundspenkis, J.: Intelligent knowledge assessment systems: Myth or reality, *Frontiers in Artificial Intelligence and Applications, Databases and Information Systems X*, pp. 31–46 (2019). <https://doi.org/10.3233/978-1-61499-941-6-31>
15. Nikiforova, O., Zabiniako, V., Kornienko, J., Gasparoviča-Asīte, M., Siliņa, A.: Mapping of source and target data for application to machine learning driven discovery of IS usability problems. *Appl. Comput. Syst.* **26**(1), 22–30 (2021). <https://doi.org/10.2478/acss-2021-0003>
16. Johnson, A.: Markov chain and its applications an introduction, applied data analytics principles and applications, pp.1–16 (2020)
17. Everitt, B., Landau, S., Leese, M., Stahl, D.: Cluster analysis, 5th edn. Wiley Series in Probability and Statistics, (2011)
18. Mirkin, B.: Clustering: A data recovery approach, 2nd edn. Chapman and Hall, London (2012)
19. Landau, S., Ster, I.: Cluster analysis: Overview. *Int. Encycl. Educ.*, 72–83 (2010). <https://doi.org/10.1016/B978-0-08-044894-7.01315-4>
20. Osipov, P., Aleksejeva, L., Borisov, A., Chizhov, Y., Zmanovska, T., Zabiniako, V.: Implementation and operation aspects of a system for detecting abnormally level of user activity. *Autom. Control. Comput. Sci.* **51**(6), 417–425 (2017). <https://doi.org/10.3103/S0146411617060050>



Machine Learning Based Intelligent Irrigation System Using WSN

Benhamada Abdelhak^{1(✉)} and Kherarba Mohammed^{1,2,3}

¹ Hassiba Benbouali University of Chlef, B.P 78C, 02180 Ouled Fares Chlef, Algeria
a.benhamada@univ-chlef.dz

² Research Center for Scientific and Technical Information,
Q253+JH8, Rue Frères Aissou, 16028 Ben Aknoun, Algeria

³ Embedded systems research unit, Hassiba Benbouali University of Chlef, Bloc 6,
Ouled Fares Chlef, Algeria

Abstract. Water is more than just a necessity to sustain life on the planet by quenching the thirst of humans, animals and plants. There are many reasons why we may face a worse global water crisis in the future than we are currently experiencing. Among the most important of these reasons is the loss of large quantities of fresh water during the irrigation process. In this paper, we present a new irrigation technique that focuses on studying the stages of plant development and estimating the actual amount of water needed at each stage, in order to minimize Over-watering and Under-watering of the plant during its life stages. We use a high amount of data previously gathered through a Wireless Sensor Network (WSN) spread in different places in the agricultural field, then we use k-Nearest Neighbors (KNN) and Weighted-k Nearest Neighbors (W-KNN) to train the Machine Learning model. However, in most existing methods of irrigation the estimated amount of water directed to the plant is constant during all stages. Our proposed solution is able to overcome this disadvantage by introducing the development stages of the plant to the learning model. The results obtained through W-KNN algorithm outperform manual irrigation and automated irrigation without stages.

Keywords: Machine learning · Smart irrigation · Irrigation system · Wireless sensors network

1 Introduction

Achieving water security in the world requires investigating the sources of water loss. More than 75% of fresh water in the world is directed to irrigate agricultural crops to achieve food security [1]; this percentage is likely to rise because the population of the earth is constantly increasing, as the United Nations (UN) [2] confirms that humans are expected to reach 8.6 billion in 2030.

<https://www.univ-chlef.dz/>.

Irrigation system is an essential element for the development of plants in agriculture. Over 74% of the daily food consumption comes from agriculture [3].

Irrigation is the operation consisting of artificially bringing water to cultivated plants to increase their production and allow their normal development, in the event of water deficit induced by a rainfall deficit, excessive drainage especially in arid areas. The farmers keep using traditional agricultural techniques, which results an over-irrigation or under-irrigation, so that it often leads to higher consumption of water and thus, waste of this natural resource. There is a requirement to implement modern science and technology in the irrigation process.

Thus, the main objective of this work is the possibility to predict automatically using machine learning solutions the amount of water required to grow a plant of each stage of development, based on local sensors data like temperature, humidity and soil moisture.

In the development phase of our proposed approach, we have evaluated a supervised machine learning algorithms named W-KNN and KNN on the gathered irrigation data. In our system we used a wide range of sensors that are strategically distributed over the agricultural field, in order to collect data required for correct monitoring in different places in the field, we used WSN to send this data to a central server where a database is located. As a result, the system is actuated on the decision for irrigation.

WSN can be classed as a rapid developing technology that can be used to manage and control tasks with various nodes owned [4]. The WSN architecture installed in our proposed smart irrigation system comprises of ESP8266 NodeMCU-based sensor nodes for measuring soil moisture, temperature and humidity equipped with Wi-Fi communication module. These sensor nodes transmit the measured data to the network coordinator installed on the central control unit of the system, which is based on raspberry pi.

The correlation of irrigation with WSN possible to get better results from the irrigation process using several innovative techniques. It has the possibility to estimate the robust components of smart irrigation like sensing the climatic conditions, determine the plant stages, transmit the data gathered to the server in real time, estimate the best time for watering during the day, and finally the decision obtained via the system is sent through a message alert to the farmer.

In this work we applied W-KNN and KNN to compare the sensor data collected from the field in to estimate results efficiently. Using this algorithm allows to solve both categories' classification and regression problem so that we can treat the Overfitting problem. The rest of the paper is organized as follows: Sect. 2 deals with proposed system and describe the different components used. Data set preparation steps is presented in Sect. 3. Section 4 explains the planned irrigation system's flow diagram. In Sect. 5 we describe the details of the implemented algorithm. Experimental results and discussion are illustrated in Sect. 6. Finally, Sect. 7 presents conclusions followed with future scope.

1.1 Related Work

In the research made in [5], the authors developed a soil moisture estimation system for a lychee orchard applying the Deep Long Short-Term Memory, which

use a linear time series approach that utilize over time the sequential processing. However without any information about the plant development, the system tries to provide an equal environment of soil moisture during all plant life cycle, which leads to over or under watering.

The use of machine learning in agriculture is provided by the authors in [6]. Organic carbon and moisture content in the soil, agricultural yield prediction, disease and weed identifications in crops, and species identification are all areas of study. Classification of crop photos based on machine learning (ML) and computer vision is evaluated to the yield evaluation and monitor crop quality. The effect of internet of things (IoT) and artificial intelligence (AI) in smart farm management is examined with a brief introduction to machine learning algorithms which are most widely used in precision agriculture.

In [7], the authors have developed an irrigation system which applies the techniques of artificial neural network in order to predict the level of water, estimate the period of sluice gate setting using a fuzzy logic control algorithm, and adapting the hydraulics equations for sluice gate level. In the prediction step the author applies the algorithm only to a dataset of last three days of water level, so that considered insufficient in order to reach most exact prediction possible.

In [8] the authors developed a support system for the automatic prediction of intelligent irrigation called (SIDSS) this system helps to estimate the amount of water needed for one week rely on two models, linear model named a Partial Least Squares Regression and a layered model named Adaptive Neuro-Fuzzy Inference System.our system uses live prediction relying on data gathered from the field this provide more stable environment for the plant.

In the study presented by the author of [9], this article presents a research of machine learning applications in supply chains agriculture. However the author have not evolved any system or script, had deduce, through different researches that the neural Networks algorithms are the most explored for agricultural solutions, we used KNN and W-KNN and they gives excellent results.

2 Proposed System

2.1 Working

In this paper we propose an intelligent irrigation system, that helps to minimize the amount of water to be wasted during the development growth stages since, the amount of water required varies according to the specie of plant and the growth stage. Overwatering heavily damages plant and could lead to their decline; a dry surface is not always indicative of the need for water. Usually in many previous studies there is water given to the crop when the conditions perceived by the system attains the targeted conditions, which often leads to Overwatering problem. Thus, the main objective of our proposed system is to predict the best time for irrigation, based on:

- Stages of plant growth (sapling, plant and tree)
- Weather data Temperature and Humidity
- Soil Moisture

We choose Solanum tuberosum (potato plant) because it is widely cultivated and consumed in our region and in the world, Fig. 2 illustrate its growth stages and our dividing into three stages based on water consumption. As shown in Fig. 1, the realization of our smart irrigation system followed different steps. First the system perceives the weather conditions (temp and hum) using DHT11 sensor and soil moisture using a soil moisture sensor, these sensors connected to ESP8266 NodeMCU, which sends in real time through WSN the data collected data from sensors to the server (raspberry pi).

The Raspberry Pi plays an essential role in our system by providing the training model for decision, hosting a web server and storage to the datasets. In addition to the data collected from the sensors the system determines automatically the current stage of the plant by calculating the number of days starting from the first day of planting.

The system trained from the given dataset using the data collected from the sensors and the current stage of the plant. The data is processed in real time and generates a decision (needed/not needed), the raspberry pi is connected with a Global System for Mobile (GSM) module to send the decision to the farmer through an SMS.

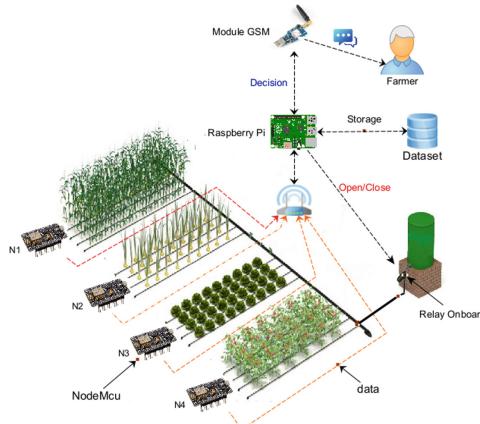


Fig. 1. System architecture.

Divide the stages of plant growth according to the amount of water it needs.

2.2 Hardware Components

The different components used in this system are:

- Raspberry pi: used as calculation unit for training model, hosting a web server and storage to the database.

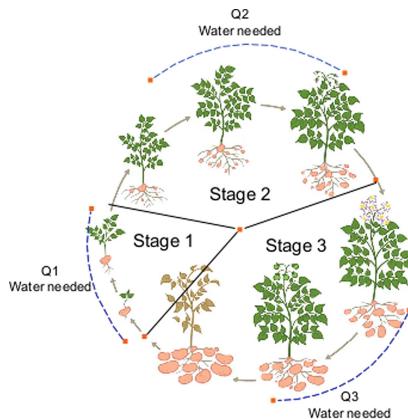


Fig. 2. Stages of plant growth according to the amount of water needed.

- NodeMCU: it enables the integration of various sensors using GPIO pins [10]. The Soil moisture and DHT11 sensor are installed in the field and connected to the NodeMCU as shown in Fig. 6.
- Soil Moisture Sensor: Used to measures the quantity of water in the soil, act as a variable resistor. Thus, when the water content in the soil changes the resistance varies.
- Humidity and Temperature Sensor: The DHT11 sensor is used to measure the temperature and humidity near the area surrounding the plant.
- Module Xbee: Wireless data transmission module, which plays a central role in the implementation of WSN routing protocols.
- GSM module: Is an electronic module equipped with a SIM card, which connects to the GSM network like a mobile phone. Works such as receiving and sending messages [11]. This module used to send SMS notification to the farmer.
- Relay Onboard ESP8266: The relay couples the microcontroller, electrovalves and water motor. It controls the water provided to plant. Depends on the decision issued by the system, relay can OFF or ON the water supply in the target area.

2.3 Software

- IDE Python: Python 3 installed on the raspberry pi4. This is used to program a code that connects the raspberry pi with the other components (NodeMCU, GSM Module) train and run the machine learning model. In this system, we predict the right time to water the crop and this is only feasible using ML.
- NodeMCU IDE: The NodeMCU will be connected with the raspberry, the connection will be done via a built in WIFI module, using NodeMCU IDE.

2.4 Proposed WSN Architecture

Figure 3, illustrates the proposed WSN system. We choose DSDV (Distance-Vector Routing Protocol RFC8965) as routing protocol, at first time the system is up, DSDV create routing tables and maintains them periodically, the sensor nodes gather data and send them to sink using the routing protocol. The DSDV is running in Zigbee network using XBee module.

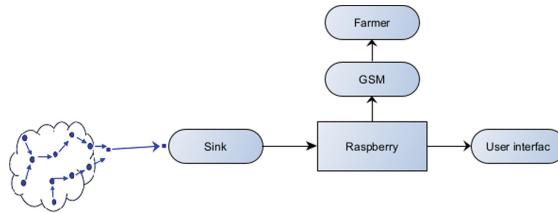


Fig. 3. Illustration of the proposed irrigation system.

3 Data Set Preparation

Dataset containing values of humidity, soil moisture, temperature and plant development stage are loaded into the W-KNN algorithm. This dataset contains values of different scenarios in the field relying on manual watering in order to increase the model accuracy. The values of soil moisture and humidity are represented in percentages and the temperature is represented in Celsius. In Fig. 4 we display some sample dataset.

The collection of dataset values is followed by the following process:

- Collect labeled data from the field.
- Outliers and missing values removal.
- Calculate the mean of every three minutes reading.
- Convert Soil Moisture to percentage.
- Split data to training and test.

4 System Diagram Flow

Figure 5 illustrates the dataflow diagram of our smart irrigation system:

- The sensors gather the data in different scenarios from the soil and surrounding environment of the crop. This data transmits as input data to the NodeMCU.
- The NodeMCU module sends the data in real time to the raspberry (ML module) using DSDV routing Protocol.

- The ML model calculates the stage of plant depending on the Algorithm 1 and run the model in order to determine if the plant needs watering or not (need/not need).
- When the result of the algorithm is “need”. The system provides water to the plant and sends notification to the farmer.
- After providing water to the plant the system checked the required parameters after a fixed period of time.

Sample.No	Stage.No	Temperature	Humidity	Soil Moisture	Irrigation Designion
1	1	32	53	68	1
2	1	38	60	56	0
3	1	38	64	58	0
4	1	37	58	66	1
5	1	43	47	60	0
6	1	40	46	66	0
7	1	34	61	58	1
8	1	35	42	35	0
9	1	33	45	62	1
10	1	48	43	47	0
11	2	32	61	69	0
12	2	37	67	58	0
13	2	39	45	64	0
14	2	40	65	56	0
15	2	44	61	53	1
16	2	41	59	68	1
17	2	31	51	56	1
18	2	33	42	34	1
19	2	37	44	69	0
20	2	47	45	49	0
21	3	31	47	73	1
22	3	35	73	64	1
23	3	34	67	58	0
24	3	36	71	58	0
25	3	44	46	60	1
26	3	41	59	69	0
27	3	32	52	56	1
28	3	38	40	43	1
29	3	35	57	61	1
30	3	46	57	52	1

Fig. 4. Dataset sample.

5 Machine Learning Algorithm

Weighted KNN (W-KNN) algorithm is a modified version of KNN developed to minimize the error rate.

The KNN misclasses the objects because the nearest neighbours are widely separated in distance and it can be classed to other far classes because it has

Algorithm 1 Get irrigation stage.

```

1: DayNo ← DayCounter()
2:
3: if DayNo ≤ endofstage1 then
4:     return stage1
5: else if DayNo ≤ endofstage2 then
6:     return stage2
7: else
8:     return stage3
9: end if

```

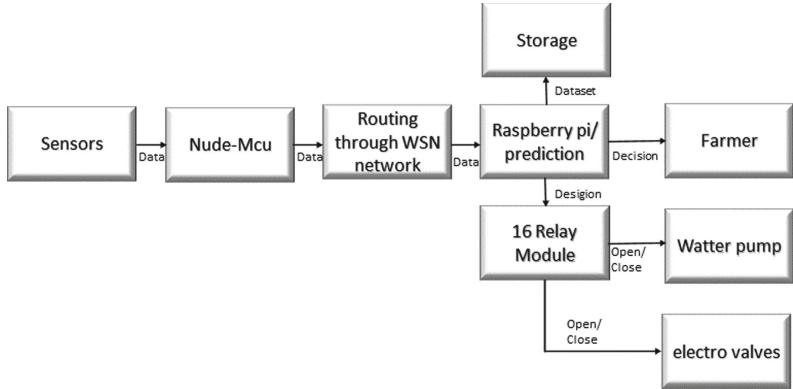


Fig. 5. Data flow diagram.

more condensed points. The W-KNN algorithm is used to overcome this problem by giving a weights to the k nearest point.

W-KNN Algorithme steps [12]:

- K parameter is chosen and initialized.
- Calculate and order the distances between the new sample and the other samples one by one.
- Chose the smallest k distances among the distances calculated in the previous step.
- The weights of k selected samples are calculated using Eq. 1.

$$E = 1/d^2 \quad (1)$$

- The weights of the same classes are grouped together and the class of the new sample is estimated by observing the total weights of the classes of the nearest neighbors.

Advantages to choose W-KNN:

In a classification with W-KNN and KNN algorithms, the W-KNN it is more successful, considering the role of the weighting to increase the accuracy. In the classification with W-KNN and KNN algorithms, the W-KNN algorithm is more successful, considering the role of the weighting to increase the accuracy [13].

6 Experimental Results and Discussion

In the experimental setup, we built a prototype compatible with the system scheme in Fig. 6 we implement the sensors section of the system described in part 2.1, ESP8266 NodeMCU module this module communicates with the server part (raspberry) of the system through WSN network.

The procedure of the server part explained in Sect. 2.1 is implemented in a raspberry pi4. The data received from the sensors is stored in local database

server. The system interacts with the farmer using GSM module, by sending a notification message.

In our experiments, we recorded humidity, temperature and soil moisture data for several weeks and save them in dataset. In order to evaluate the performance of our proposed model in the case of automatic irrigation, we apply both KNN and W-KNN on the data collected. As show in Table 1, the W-KNN algorithm has attained more accuracy classification than the KNN algorithm.

Table 1. Accuracy comparison with W-KNN and KNN.

Algorithm	Knn	W-KNN
Accuracy	96.3%	98.2%

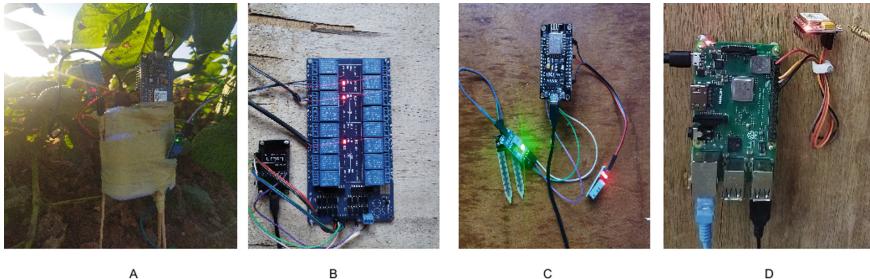


Fig. 6. Hardware components.

We compared our model results with two other datasets, automated irrigation and manual irrigation without considering the stages of the plant, based on soil moisture readings during the development stages of the plant. Figure 7 illustrates the difference between the three strategies, we calculate the average of one day readings.

In the manual strategy, we notice a significant increase in soil moisture in periods when the plant does not need it and decreases in periods when the plant needs it, this behaviour leads to decrease in yield and a large waste of water. In the automatic strategy, the system tries to stabilize the soil moisture percentage without taking into account the plant development stages. This causes overwatering in some stages and under watering in other stages, which also leads to a decrease in yield and waste of water. In our strategy, the system changes the soil moisture along with the changes of plant development stages, as it provides an ideal environment for the plant and decreases water wastage.

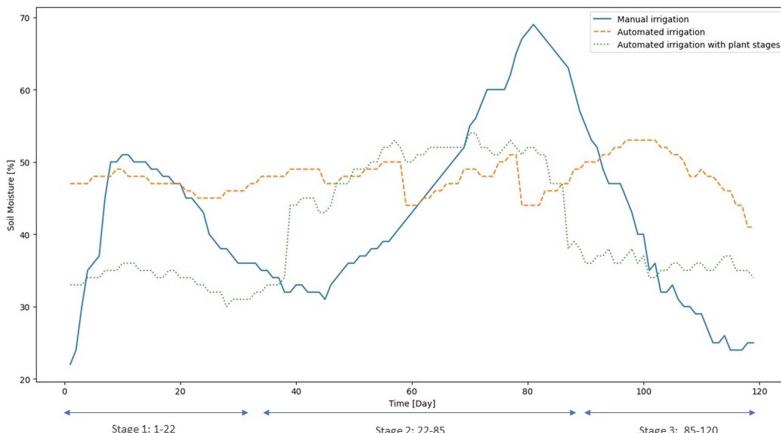


Fig. 7. Comparison of irrigation strategies.

7 Conclusion

In this study, we proposed and implemented an intelligent irrigation system, using machine learning and WSN. The system was trained on given dataset using gathered data from the sensors and taking into account the stages of development of the plant.

On the real time the system process the data and sends the decision to the farmer through an SMS. The results shows that our model provides the perfect conditions for the plant in each stage of development; in addition the system reduces water wastage in significant manner. The results show that our proposed system helps in making irrigation decisions with high accuracy.

As future works we aim to introduce more plants to the system, that are extensively cultivated and measures the exact amount of water used.

References

1. Akshay, S., Ramesh, T.K.: Efficient machine learning algorithm for smart irrigation. In: 2020 International Conference on Communication and Signal Processing (ICCS), pp. 867–870. IEEE (2020)
2. United Nations page. <https://www.un.org/en/desa/world-population-projected-reach-98-billion-2050-and-112-billion-2100>. Accessed Nov 2023
3. Cardoso, J., Glória, A., Sebastiao, P.: Improve irrigation timing decision for agriculture using real time data and machine learning. In: 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), pp. 1–5. IEEE (2020)
4. Permana, I.K.Y.T., Mantoro, T., Irawan, E., Handayani, D.O.D., Safitri, C.: increased efficiency of smart water systems for vegetable plants using the deep learning classification approach. In: 2019 5th International Conference on Computing Engineering and Design (ICCED), pp. 1–5. IEEE (2019)

5. Gao, P., Qiu, H., Lan, Y., Wang, W., Chen, W., Han, X., Lu, J.: Modeling for the prediction of soil moisture in litchi orchard with deep long short-term memory. *Agriculture* **12**(1), 25 (2022)
6. Sharma, A., Jain, A., Gupta, P., Chowdary, V.: Machine learning applications for precision agriculture: A comprehensive review. *IEEE Access* **9**, 4843–4873 (2021)
7. Suntaranont, B., Aramkul, S., Kaewmoracharoen, M., Champrasert, P.: Water irrigation decision support system for practical weir adjustment using artificial intelligence and machine learning techniques. *Sustainability* **12**(5), 1763 (2020)
8. Pattnaik, P.K., Kumar, R., Pal, S. (eds.): *Internet of Things and Analytics for Agriculture*, vol. 2. Springer, Singapore (2020)
9. Sharma, R., Kamble, S.S., Gunasekaran, A., Kumar, V., Kumar, A.: A systematic literature review on machine learning applications for sustainable agriculture supply chain performance. *Comput. Oper. Res.* **119**, 104926 (2020)
10. Pernapati, K.: IoT based low cost smart irrigation system. In: 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 1312–1315. IEEE (2018)
11. Gupta, M.S.D., Patchava, V., Menezes, V.: Healthcare based on iot using raspberry pi. In: 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), pp. 796–799. IEEE (2015)
12. Jung, W.H., Lee, S.G.: An arrhythmia classification method in utilizing the weighted KNN and the fitness rule. *IRBM* **38**(3), 138–148 (2017)
13. Tarakci, F., Ozkan, I.A.: Comparison of classification performance of kNN and WKNN algorithms. *Selcuk. Univ. J. Eng. Sci.* **20**(2), 32–37 (2021)



Comparison of Artificial Neural Networks Algorithms on Datasets with Different Characteristics

Bruno Pilosta, Dijana Oreski^(✉), and Nikola Kadoic

Faculty of Organization and Informatics, University of Zagreb, Varazdin, Croatia
dijana.oreski@foi.hr

Abstract. This paper deals with the research of different artificial neural network algorithms and their application on data sets with different characteristics. In the first part of the paper, a description of six neural network algorithms is given, on the one hand, and the characteristics of data sets measured through meta-features, on the other hand. The empirical part of the paper describes the development of the predictive models through the process of data preparation for modeling, hyperparameters optimization, and analysis and empirical comparison of the algorithms' performance on different data sets. The research results show differences in the performance of the algorithms: Adam algorithm and its modifications have better performance than the AdaGrad algorithm and the basic gradient descent algorithm.

Keywords: Artificial neural network · Meta-feature · AdaGrad algorithm

1 Introduction

Numerous artificial neural network algorithms are developed and applied for predictive and descriptive models' development. This research focuses on artificial neural network algorithms for predictive model development. Previous research papers analyzed the performance of machine learning algorithms of data sets with different characteristics. However, as far as we know, there is no paper dealing with analyzing the performance of different artificial neural networks on data sets with different characteristics.

Kwon and Sim [1] researched data set features' effects on the performances of classification algorithms. The authors study the influence of different data set characteristics on the performance of several classification algorithms. The main aim of the research was to investigate how various characteristics of data sets affect the overall accuracy as well as the time required to train predictive models. Examined characteristics of data sets were: type variables (categorical type and numerical type), size of data set (number of instances), data dimensionality, frequency of missing data in the data set, dataset imbalance, and dimension of classes. Eight classification algorithms were used in the research, including Bayesian networks, logistics regression, RBF networks, Sequential Minimal Optimization (SMO), decision trees, and KNN. Performance evaluation was done by using Cohen's Kappa statistic, RMSE, MAE, and elapsed time. It was concluded

that Bayesian networks and decision trees had the best overall performance. As for the characteristics that affect the accuracy of prediction, it was observed that if the input variables are categorical type, the accuracy drops with logistic regression algorithms, RBF network, and KNN algorithms. The type of input variable does not affect the accuracy of the model created by the Bayesian network algorithm. In addition, the accuracy improves by increasing the number of instances in the data set. When modeling with Bayesian networks the characteristics of the datasets have no influence [1].

The second study [2] refers to the examination of the data set characteristics on the performance of algorithms in the classification domain. In this research, the performance of several different algorithms on datasets from the medical domain is empirically tested. The researchers used 20 data sets based on which they examined the performance of the SVM (Support Vector Method), neural networks, Naive Bayes algorithm, decision trees, random forests, and AdaBoost algorithms. Performance evaluation was done using several different metrics including accuracy, precision, response, F measure, and ROC AUC. Out of 20 data sets, 18 of them are small and two are large. To examine the performance of smaller data sets compared to large ones, the segmentation of large data sets was made into sets of smaller sizes of 980, 490, and 98 instances, which represented the data size categories “large”, “medium” and “small”. The performance of the algorithms was tested on segments concerning the remaining smaller data sets, which were 18–1030 instances in size.

The most robust algorithms proved to be AdaBoost and Naive Bayes. In addition, it was concluded that the size of the data set does not affect the algorithms’ performance [2].

These two studies examined performance comparison of various classification algorithms, but only the second study included neural networks in performance testing. In addition, only prediction accuracy was examined, without including the time required for training/testing predictive models, especially with datasets of different sizes where it can be assumed that the time required for training the model with large datasets will be higher than with smaller datasets.

As far as we know, there is no research dealing with the comparison of different neural network algorithms on data of different characteristics. This is the motivation for our work.

This paper is organized as follows. The second section describes the research methodology focusing on data description and artificial neural network algorithms description. The third section presents research results, whereas section four discusses the results. Finally, section five concludes the paper.

2 Methodology

This chapter describes two datasets used in the empirical research and provides insights into six artificial neural network algorithms applied for predictive model development.

2.1 Data Description

This research examines two datasets. The first data set [3] to be analyzed relates to the prediction of fraud in credit card transactions. The data contained in the dataset refers to

credit card transactions owned by Europeans that took place over two days in September 2013. By using this data set, predictive models will be developed to predict whether there is fraud in the transaction or not employing artificial neural network algorithms.

The second data set [4] refers to the regression problem since the prediction is for the value of a continuous numerical variable. Neural networks are used for predicting the price of real estate. The average price of real estate in the suburbs of Boston is predicted.

2.2 Neural Networks Algorithms

The neural network optimization algorithms implemented in Tensorflow which will be tested in this research are Adam, Adamax, Adagrad, NAdam, SGD, and RMSprop. Each of the algorithms is briefly described in this section.

Gradient Descent (GD) is an optimization algorithm for solving a wide range of different optimization problems, and its use in neural networks is just one of them. In artificial neural networks, this algorithm is used to optimize the weights of their input connections, based on the errors obtained by individual neurons, which ultimately results in a smaller overall error of the neural network model, and thus the accuracy of the model [5].

One of the fundamental variants of gradient descent is Stochastic Gradient Descent (SGD). The SGD algorithm refers to a certain stochastic process, that is, the randomness that appears in its operation. With the SGD algorithm, an instance of the data set is taken, and propagated through the network forward, calculates the gradients, and updates the weights of the links in the network in the same step. Stochasticity is manifested in the randomly selected instance of the data set for propagation through the network, that is, the instances entering the training are not selected in order from first to last, but are randomly selected. Likewise, it does not remember which instances were selected, so certain instances can be propagated through the network several times, and some never [5].

This approach solves the problem of computing gradients over a large training data set, which greatly reduces the time complexity of training, but with a less negative impact on training accuracy.

The AdaGrad algorithm is one of the modifications of the GD algorithm. The AdaGrad algorithm is the simplest modification of the GD algorithm. The name AdaGrad comes from “Adaptive Gradient”, from which we can conclude that it is an adaptation. Adaptation manifests in the adaptation of the learning rate during the optimization step, and how this is done is very simple [6].

Adam refers to *Adaptive Moment Estimation* and it is considered to be the best algorithm, not only because of the speed with which it converges towards the minimum of the cost function but also because of its accuracy, i.e. convergence towards the global, not the local, minimum. The Adam algorithm is a combination of two types of algorithms, RMSPro and gradient descent optimization with momentum (“Gradient Descent with Momentum”).

The **AdaMax** algorithm is one of the variants of the Adam algorithm and it is almost identical to the “Adam” algorithm with the only difference being the calculation of the variable st .

Another variant of the Adam algorithm is called **NAdam** as a derivative of “Nestor’s Adaptive Moment Estimation”. This algorithm uses the so-“NAG” or “Nestor’s Accelerated Gradient” method of calculating the first moment (variables m before). NAG proved to be a better moment calculation method because the gradient of the cost function is not used on the current value of the link weight, but for the weight increased by $\beta * m$. Calculating the moment in this way turned out to be better, that is, it turned out that the algorithm converges to the minimum of the cost function much faster by calculating the moment using the “NAG” method [5].

This modification of the Adam algorithm was performed by Timothy Dozat, who described how “Adam” works using the “NAG” method of moment calculation [7].

3 Research Results

3.1 Comparative Analysis of Algorithms on the First Dataset

Several different neural network architectures were tested, and the structure with three hidden layers of 8, 16, and 32 neurons in the hidden layers gave the best results. The activation functions in the hidden layers are ReLU and in the output “Sigmoid”. The loss function is binary cross-entropy, and the used metric is ROC AUC. The sample size for one iteration of backpropagation is larger due to a much large data set and amounts to 32 instances.

The first algorithm that was tested was Adam. After 5 epochs, Adam reached the minimum of the loss function, and after 22 s the training was completed.

Table 1. Results of testing for ADAM and ADAMAX.

Algorithm	Adam	Adamax		
Accuracy	0.99950	0.99948		
ROC AUC	0.91165	0.91532		
Recall	0.82352	0.83088		
Precision	0.86153	0.84328		
F1 score	0.84210	0.83703		
Training time	21.59661	115.82010		
Confusion matrix				
	P0	P1	P0	P1
S0	85289	18	85286	21
S1	24 112	112	23	113

The loss function is very close to the minimum after the first epoch of training (Fig. 1).

The second algorithm applied is Adamax. Adamax is slower than the Adam algorithm (Table 1), and the number of epochs after which the loss function stabilized at the

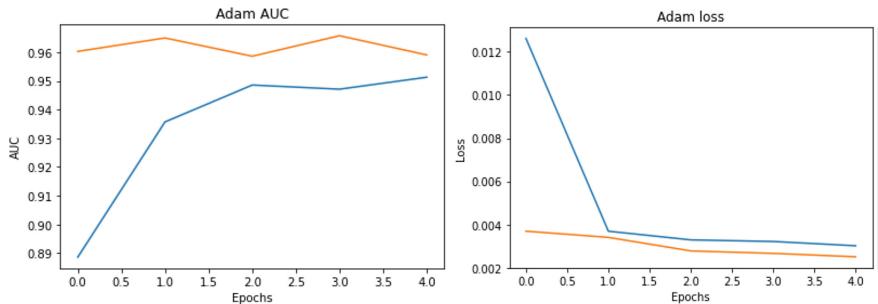


Fig. 1. Accuracy and value of the loss function by epochs adam.

minimum was 28 (Fig. 2). This also means that the total training time is slightly longer, amounting to 116 s. We can see that the loss function has stabilized even before epoch 30.

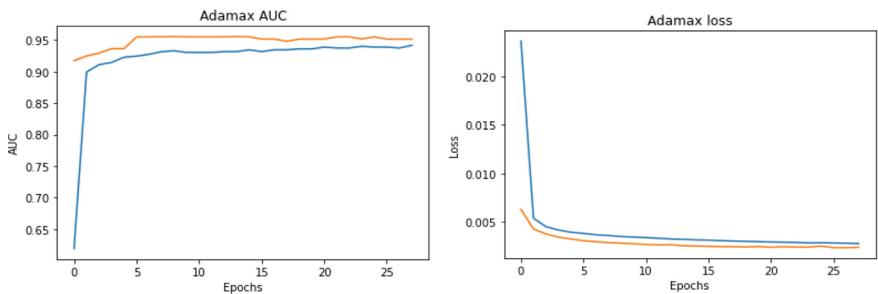


Fig. 2. Accuracy and value of the loss function by epochs Adamax.

Adagrad algorithm has the slowest training time and the lowest accuracy (Table 2). The training lasted for 100 epochs, but around the 60th epoch, the value of the loss function became minimal and stopped decreasing (Fig. 3). Compared to other algorithms, the algorithm has significantly lower accuracy, especially the response value, where it made 42 incorrect negative classifications.

NAdam, with the Adam algorithm, had the fastest training time, reaching the minimum of the loss function occurred after 7 epochs (Fig. 4), but also in this case, after the first epoch, it already approached the minimum. Unlike the “Adam” algorithm, it made a few more mistakes, which makes the algorithm inferior according to all metrics.

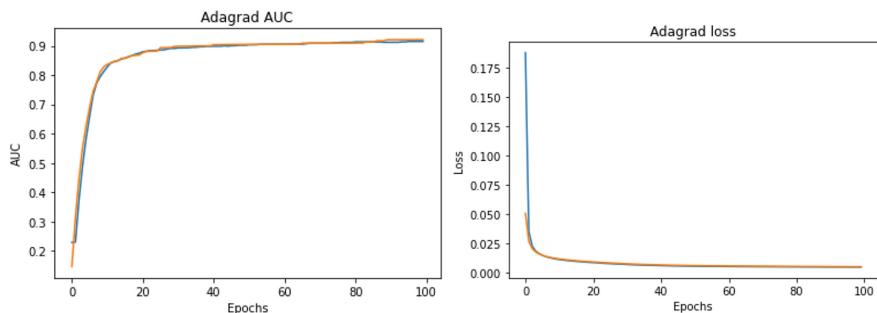
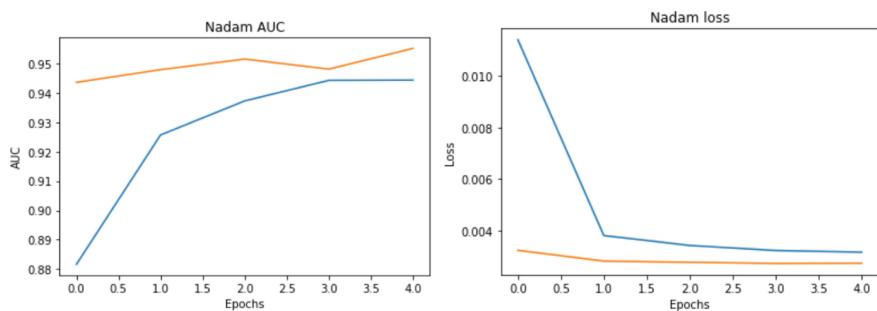
SGD algorithm required longer training time (Table 3), but the algorithm showed very good performance results. The training lasted for 80 epochs where the value of the loss function stabilized (Fig. 5).

RMSProp algorithm had a much shorter training time and the loss function reached a minimum very quickly (Fig. 6). The training lasted 8 epochs, i.e. 35 s. Compared to other algorithms, the loss function had large oscillations, especially when training through multiple epochs, but the accuracies are very similar to other algorithms.

The list of metric values of all algorithms can be seen in the Table 4.

Table 2. Results of testing for ADAGRAD and NADAM.

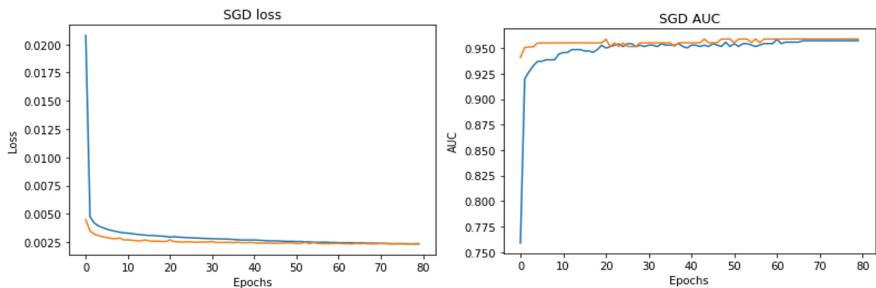
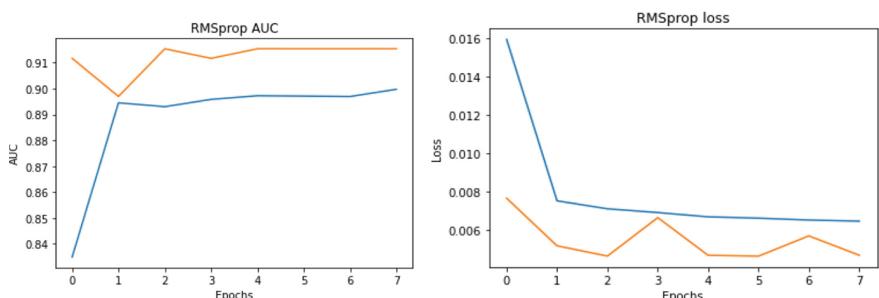
Algorithm	Adagrad	NAdam		
Accuracy	0.99932	0.99941		
ROC AUC	0.84549	0.91528		
Recall	0.69117	0.83088		
Precision	0.85454	0.80714		
F1 score	0.76422	0.81884		
Training time	426.98640	23.42632		
Confusion matrix				
	P0	P1	P0	P1
S0	85291	16	85280	27
S1	42	94	23	113

**Fig. 3.** Accuracy and value of the loss function by epochs Adagrad.**Fig. 4.** Accuracy and value of loss function by epoch NAdam.

The fastest algorithms were Adam and NAdam although according to the overall performance, they were not the best. However, “Adam” had a very similar performance to the best one, SGD. According to the ROC AUC and F1 measures and examining

Table 3. Results of testing for SGD and RMSPROP.

Algorithm	SGD	RMSProp		
Accuracy	0.99955	0.99937		
ROC AUC	0.91186	0.90425		
Recall	0.82352	0.80882		
Precision	0.88888	0.80291		
F1 score	0.85496	0.80586		
Training time	323.09846	35.02104		
Confusion matrix				
	P0	P1	P0	P1
S0	85293	14	85280	27
S1	24	112	26	110

**Fig. 5.** Accuracy and value of loss function by epoch SGD.**Fig. 6.** Accuracy and value of loss function by epoch RMSProp.

overall performance, SGD was slightly better than the fast “Adam” algorithm. This algorithm also had very stable loss function values across epochs, which is expected since there is no variability in the learning rate. The weakest algorithm according to the ROC AUC and F1 measure turned out to be Adagrad, which seems to have prematurely

Table 4. Comparison of algorithms.

Algorithm	Accuracy	ROC AUC	Recall	Precision	F1 score	Training time (sec)	Test time (sec)	Epochs
Adam	0.999508	0.911659	0.823529	0.861538	0.842105	21.596610	0.637579	5
Adamax	0.999485	0.915318	0.830882	0.843284	0.837037	115.820105	0.605538	28
Adagrad	0.999321	0.845494	0.691176	0.854545	0.764228	426.986407	0.603548	100
Nadam	0.999415	0.915283	0.830882	0.807143	0.818841	23.426325	0.609556	5
SGD	0.999555	0.911683	0.823529	0.888889	0.854962	323.098467	0.630573	80
RMSprop	0.999380	0.904254	0.808824	0.802920	0.805861	35.021044	0.632574	8

stopped additional learning due to too low a learning rate at higher iterations. Adamax and RMSProp had an almost equal number of false positive and negative predictions, so the response and precision values are similar, but RMSprop had slightly more false predictions, so overall it is slightly weaker. Due to the very short training time, but also very high accuracy, the recommendation would be to use the Adam and NAdam algorithms for large data sets.

3.2 Comparative Analysis of Algorithms on the Second Dataset

The second dataset refers to the regression problem since it predicts the value of a continuous numerical variable: the average price of real estate in the suburbs of Boston is predicted. According to the different type of problem, different metrics were used. Neural network architecture shown to be optimal for this data set consists of two hidden layers with 64 neurons each, again with the ReLU activation function. Since we are dealing with a regression problem, there is one neuron with a linear activation function on the output layer. The loss function is the average squared error.

The fastest algorithm was Adam. After 45 epochs and 1.15 s of training, the model trained with this algorithm reached a coefficient of determination value of 0.857 with an average absolute error (MAE) of 2,377 or 2,377 dollars. Looking at the average property value of around \$23,000, we can conclude a low error rate.

The decline of the loss function was examined and it is seen that increasing the number of training epochs from 45 leads to overfitting (Fig. 7).

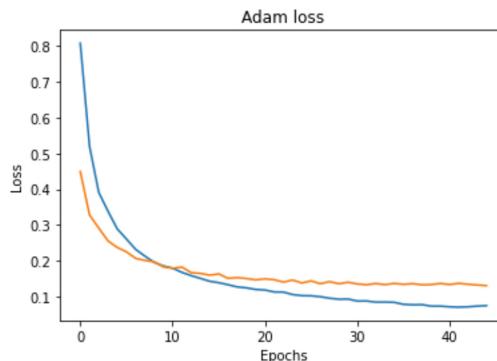


Fig. 7. Value of loss function by epoch ADAM.

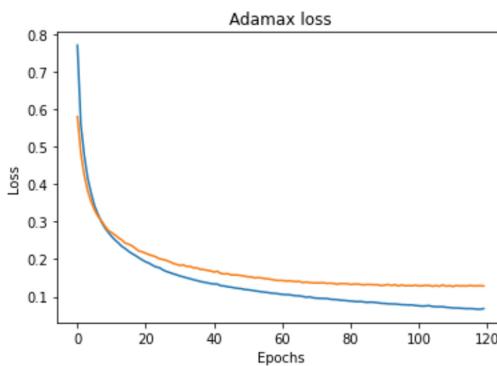
Slightly better results than the Adam algorithm were shown by Adamax (Table 5) with a much longer training time of 120 epochs, which resulted in a training time of 2.6 s. Compared to Adam's model, Adamax's model had a slightly higher average absolute error of 2.382, but a slightly smaller rooted average squared error.

The loss function was decreasing until the 120th epoch, after which an increase in the value of the validation loss function was observed (Fig. 8).

The weakest was the Adagrad algorithm (Table 6). Although similar accuracies were achieved as with other algorithms, the training time was much longer. Training the model

Table 5. Results of testing for ADAM and ADAMAX on second dataset.

Algorithm	Adam	Adamax
R2 score	0.856973	0.85956
Mean absolute error	2.37666	2.3816
Mean squared error	12.012188	11.7947
Root mean squared error	3.46586	3.43435
Training time (sec)	1.15405	2.55184
Test time (sec)	0.025022	0.02102
Epochs	45	120

**Fig. 8.** Value of loss function by epoch Adamax.

with the Adagrad algorithm took as many as 3000 epochs, and the training time was over a minute, (66 s). The coefficient of determination was the lowest 0.851, but the MAE value was slightly lower than the described algorithms at 2.367. On the other hand, the model with this algorithm had the highest RMSE of 3.53.

Table 6. Results of testing for ADAGRAD and NADAM on second dataset.

Algorithm	Adagrad	Nadam
R2 score	0.85175	0.865373
Mean absolute error	2.36707	2.367273
Mean squared error	12.4507	11.306702
Root mean squared error	3.52856	3.362544
Training time (sec)	66.2186	1.30719
Test time (sec)	0.02602	0.019017
Epochs	3000	45

Adagrad has a very stable loss function without oscillations, and its value in each epoch is lower or equal to that in the previous epoch. The last 100 training epochs did not decrease any more (Fig. 9).

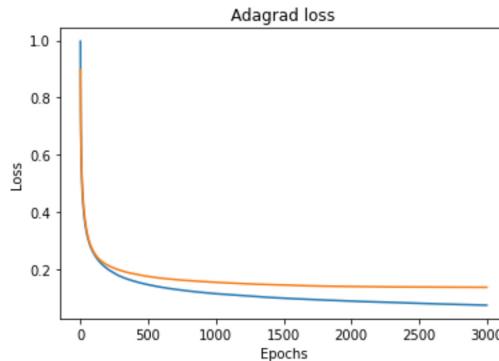


Fig. 9. Value of loss function by epoch Adagrad.

NAdam, as well as Adam, are the fastest algorithms. By using this algorithm, the model was trained for 45 epochs, that is 1.3 s. In this case, the coefficient of determination was 0.865 with an MAE of 2.367. The RMSE is 3.36, the lowest so far.

The value of the validation loss function was decreasing until 45 epochs after which the training was completed (Fig. 10).

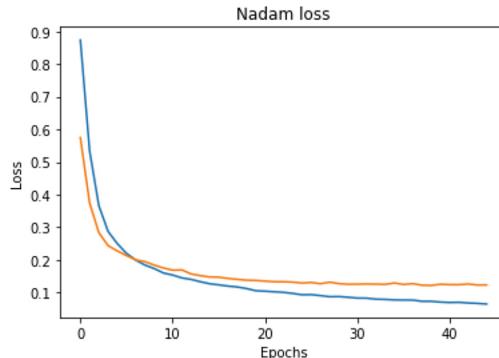


Fig. 10. Value of loss function by epoch NAdam.

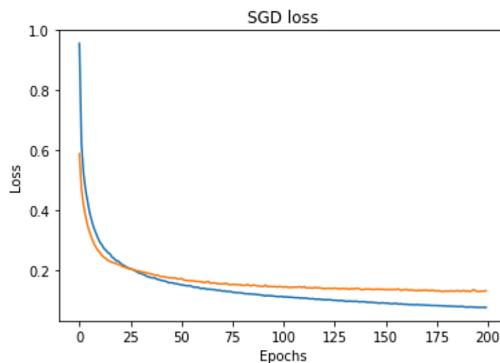
The SGD algorithm had similar results to the other algorithms (Table 7). As before, this algorithm is the second slowest in total, and the model was trained up to the 200th epoch, which is about 4 s. The coefficient of determination is at the level of the others and is 0.859 with an MAE of 2.328.

The validation function is more stable and falls to around the 200th epoch (Fig. 11).

The last algorithm, RMSProp, gave the best results on this data set (Table 7). With the RMSProp algorithm, the model was trained for 60 epochs, which is also quite a

Table 7. Results of testing for SGD and RMSPROP on second dataset.

Algorithm	SGD	RMSprop
R2 score	0.85888	0.867047
Mean absolute error	2.32795	2.150324
Mean squared error	11.8524	11.166184
Root mean squared error	3.44274	3.341584
Training time (sec)	3.80347	1.500365
Test time (sec)	0.01902	0.022018
Epochs	200	60

**Fig. 11.** Value of loss function by epoch SGD.

low 1.5 s. The coefficient of determination is slightly higher than that of the NAdam algorithm and is 0.867. Compared to other algorithms regarding MAE NAdam achieved 2.3, whereas RMSProp achieved 2.15. RMSProp model made an average error of \$200 less. RMSProp model makes the least errors since the RMSE is the smallest 3.34.

This algorithm had a slightly more unstable loss function where we can observe slightly larger oscillations. The optimal training time was 60 epochs (Fig. 12).

Since in this case, the data set was small, the training time was not a big problem with any algorithm except Adagrad. All algorithms finished training within 4 s, except for Adagrad which took 3000 epochs for the value of the validation loss function to start to stagnate.

All artificial neural network algorithms had very similar metric values on this dataset as seen in Table 8.

4 Discussion

This paper examines the performance of artificial neural network algorithms with a focus on their optimization algorithms. The data sets used were of different characteristics including different set sizes, problems being solved, noise in the data, and data types.

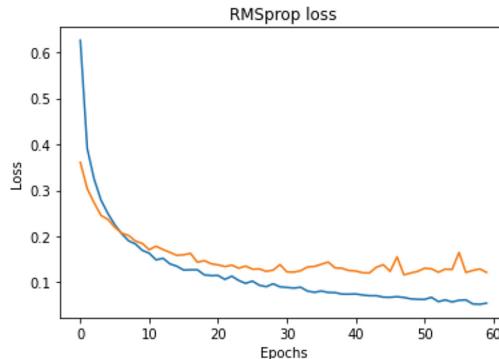


Fig. 12. Value of loss function by epoch RMSprop.

Table 8. Comparison of algorithms on second dataset.

Algorithm	R2 score	Mean absolute error	Mean squared error	Root mean squared error	Training time (sec)	Test time (sec)	Epochs
Adam	0.856973	2.376660	12.012188	3.465860	1.154050	0.025022	45
Adamax	0.859563	2.381599	11.794735	3.434346	2.551839	0.021019	120
Adagrad	0.851751	2.367070	12.450749	3.528562	66.218616	0.026023	3000
Nadam	0.865373	2.367273	11.306702	3.362544	1.307190	0.019017	45
SGD	0.858875	2.327949	11.852439	3.442737	3.803467	0.019017	200
RMSprop	0.867047	2.150324	11.166184	3.341584	1.500365	0.022018	60

Research results indicate the very consistent performance of all algorithms: all algorithms behaved similarly with different types of data sets, from training time to accuracy in prediction. All algorithms can find the minimum of the loss function for both datasets, but the biggest difference is in the time needed to do so. Regarding the overall accuracy with almost every data set, we could see that it differs by about 1% between the algorithms. This would mean that all algorithms reach the minimum of the loss function, but it seems that due to oscillations near the minimum, not all algorithms have identical accuracy.

The algorithm that differed the most from the others is Adagrad. Adagrad is the simplest modification of the basic gradient descent algorithm. It was to be expected that this algorithm should have slightly better performance than the basic gradient descent algorithm due to the variable learning rate, but this was not the case. Due to excessive variability, i.e. the reduction of the learning rate through the epochs, the algorithm lowers the learning rate too quickly to the level that after the first few epochs, it becomes too small, which leads to the fact that at higher epochs the learning progresses is too slow, which is why the learning times themselves are very long. This is particularly problematic with complex data sets, where the learning time is somewhat longer even

with faster algorithms. In these cases, this algorithm simply cannot reach its maximum in a reasonable time. We could recommend this algorithm only for non-complex and very small data sets, i.e. for data sets where the minimum of the loss function is reached very quickly, where the advantage of this algorithm can be used, which is a very stable loss function that does not oscillate as is the case with other algorithms and a very low learning rate at the minimum of the loss function prevents “overlearning”.

The basic algorithm of gradient descent (through tests of the “Mini-Batch Gradient Descent” version) proved to be very good. In addition to faster convergence towards the minimum of the loss function, it showed very high accuracy very often at the level of other algorithms, and also due to the often more stable loss function, it had a little higher accuracy due to better “guessing” the minimum of the loss function. Compared to the remaining algorithms, it is much slower since the learning rate is non-variable. Because of this, it has too slow learning with more complex data sets. In the case of a very simple data set the algorithm can achieve accuracy with each iteration of training where it converges much better to the minimum of the loss function than the other algorithms. For this reason, this algorithm is recommended for smaller and simpler data sets or when the training time is not important.

The remaining algorithms are algorithms with a variable loss function that accelerates training at the start of training and very quickly reaches a minimum. The overall accuracies of these algorithms differed within 1%. The RMSProp and Adamax algorithms in most cases had similar accuracies with a slightly longer training time than the fastest Adam and NAdam algorithms, with the fact that “RMSProp” was almost always faster than the Adamax algorithm. The speed of convergence towards the minimum is a big advantage of these algorithms over SGD and Adagrad. Those algorithms have advantages in use for more complex problems and large data sets. Although Adamax is considered an upgrade to the Adam algorithm, this was not seen in this test. As for the training time, Adam together with the “NAdam” algorithm was the fastest. Both algorithms were almost always close to the minimum of the loss function after the same number of epochs, which is why they could not single out any of the mentioned as better, but they could be singled out as the best together in terms of the speed of converging towards the minimum, which is why it is recommended to use these algorithms in any which data set.

The disadvantage observed in all of these fast algorithms is the problem with potential overlearning. Due to extremely fast reaching the minimum of the loss function, the value of the loss function over the validation set may start to increase after a certain number of epochs, often very suddenly due to their high training speed. Therefore, it is recommended to monitor the graph of the loss function for the training and test set and potentially reduces and increases the number of training epochs. Although Tensorflow/Keras can include the *EarlyStopping* callback function, which automatically stops training if the value of the validation loss function increases for several epochs in a row, it may not detect growth in oscillations. This was the case in this test, where the average function is increasing. For simple sets, we can try the Adagrad or SGD algorithms, which, due to longer training, are much more difficult to overlearn.

5 Conclusion

Neural network algorithms Adam and NAdam are mandatory when building neural networks in the development framework Tensorflow, but examining other algorithms in the form of potentially better accuracy is justified.

Artificial neural networks are very different from other machine learning algorithms due to a large number of adjustable parameters such as network structure, types of activation functions in hidden layers, loss function, sample size for propagation and optimization algorithms. Adjusting these parameters in order to maximize the performance of the built models can be extremely tedious work. In order to make this work a little easier, it is recommended in any case to use the Adam and NAdam algorithms. In the further research, we will deal with the analysis of the influence of different structures of neural networks (in terms of the number of hidden layers and neurons in the layers) on the performance of the model using data sets of different characteristics.

Acknowledgment. This work has been fully supported by Croatian Science Foundation under the project UIP-2020-02-6312.

References

1. Kwon, O., Sim, J.M.: Effects of data set features on the performances of classification algorithms. *Expert Syst. Appl.* **40**(5), 1847–1857 (2013). <https://doi.org/10.1016/j.eswa.2012.09.017>
2. Althnian, A., et al.: Impact of dataset size on classification performance: An empirical evaluation in the medical domain. *Appl. Sci. (Switz.)* **11**(2), 1–18 (2021). <https://doi.org/10.3390/app11020796>
3. Credit Card Fraud.: <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>. Last accessed 10 Dec 2022
4. Boston Real Estate.: <https://www.kaggle.com/datasets/arslanali4343/real-estate-dataset>. Last accessed 10 Dec 2022
5. Géron, A.: Hands-on machine learning with Scikit-Learn and TensorFlow. In: O'Reilly Media. (2017)
6. Brownlee, J., Machine, L.M.: Gradient descent with AdaGrad from scratch. (2021). <https://machinelearningmastery.com/gradient-descent-with-adagrad-from-scratch/>
7. Dozat, T.: Incorporating nesterov momentum into Adam. (2016). <https://openreview.net/pdf/OM0jvwB8jIp57ZJjtNEZ.pdf>



Boosting Federated Multitask Learning: Transfer Effects in Cross-Domain Drug-Target Interaction Prediction

Dániel Sándor^(✉) and Péter Antal

Department of Measurement and Information Systems, Budapest University of Technology and Economics, Budapest, Hungary
sandor@mit.bme.hu, antal@mit.bme.hu

Abstract. Using federated learning to collaborate with other parties is becoming common when conducting machine learning on high-value data. In our work, we try to expand the possibilities of existing federated models to apply them to multitask problems. Previously we presented FedMTBoost, which used boosting to enhance predictive performance in a small drug-target interaction problem. This paper investigates the algorithm’s performance on a larger scale using a cross-domain benchmark data set. The original motivation for boosting was to weigh the data adaptively; thus, the multitask transfer can happen on different tasks in different iterations. However, our results suggest that improvement is mostly present in federated scenarios, leading us to believe that the data and model weights can improve the federated transfer by adapting the models to the clients’ data. Furthermore, the boosting algorithms generally outperform traditional baselines when fewer data are available, either in tasks or samples. In this paper, we examine these findings in multiple experiments and try to explain the improvements achieved.

Keywords: Federated learning · Multitask learning · Boosting · DTI

1 Introduction

Drug-target interaction (DTI) prediction has become one of the leading fields of machine learning-based drug discovery [1]. Drugs are compounds that affect different biological targets in the human body. We call a drug-target pair active if they bind to each other. In drug research, some active pairs are usually known, and DTI prediction aims to find previously unknown active pairs. As this data type is highly valuable, it can be hard for a single partner to obtain meaningful predictions based on its private data set. This is why federated learning can spread in the field.

Federated learning (FL) is the branch of machine learning where multiple parties collaborate to create a common model [2]. It is a relatively new paradigm that is being adopted in most major fields of machine learning, including drug

discovery. In the context of drug research, we can speak about cross-silo learning, which means that a few (2–100) partners with a large amount of data participate in the learning process, and their goal is to improve performances on their private data sets. In the DTI domain, the usual setup includes 5–20 partners with a varying number of available compounds, and they expect predictions for targets, which concerns their current research [13].

Another simple way to improve predictions is to introduce another type of transfer: Multitask learning. Multitask learning refers to training a model on multiple related tasks to improve the performance of each task involved [3]. It has been shown to be an effective tool for Quantitative structure-activity relationship (QSAR) problems [4] if the tasks involved are correlated. However, selecting tasks for learning is a complex problem, and adding more tasks to a setup can also lead to performance deterioration [17]. This gave the idea to reweigh the data adaptively by boosting.

Boosting is an ensemble method that creates strong learners by weighting the predictions of weak ones [15]. It achieves this by assigning weights to the data and refreshing them in every training iteration. This adaptive weighting can be leveraged to capture previously less important parts of the data and create a better representation. However, in our previous measurements, the multitask transfer did not improve with boosting; instead, the improvement was only present in federated scenarios. As this needed further examination, we tested the performance on a full-scale benchmark data set, including the complex hierarchy of kinome [18].

In this paper, we examine the possibilities of combining the previously described methods to create a novel approach to model large-scale federated data sets with improved predictive performance. We present FedMTBoost in detail and describe its possibilities and limitations. The paper is structured in the following way: In Sect. 2, we describe the previous combinations of these approaches and explain how they contribute to the final model. In Sect. 3, we explain the FedMTBoost method from its motivation to the implementation details and highlight how each method contributes to the complete model. In Sect. 4, we present the evaluation of the FedMTBoost method on curated and full-scale data sets. Finally, in Sect. 5, we summarize the work and discuss further possibilities for the method.

2 Related Work

This section overviews the existing approaches for combining FL, boosting, and multitasking learning. The first combination that supports this is FedMTBoost, but previous solutions exist for combining two of the three methods. These can give an idea of the possible solutions and the motivation to use FedMTBoost.

Multitask learning has been used in federated learning for a long time; the two methods are closely related [5] due to the nature of transfer learning. MOCHA also allows clients to learn different tasks than others and manage the multitask transfer in the same way as the federated transfer happens. This approach is efficient in multiple ways, especially considering communication costs. However, it

is only applicable in a convex optimization setting. The introduction of boosting extends this type of approach as it can potentially enhance the transfer effect between tasks and between clients.

Federated boosting is a relatively new field, but a few solutions exist; however, these only support single-task learning to our knowledge. These solutions usually use a form of gradient boosting, where every client is responsible for different models of the ensemble [6], and all of them can use the combined ensemble. Secureboost [7] also uses gradient-boosted decision trees in a vertical FL setting. However, DTI prediction is mostly associated with horizontal FL. Federated Functional Gradient Boosting [8] shares the most similarities with FedMTBoost; however, it uses a knowledge distillation process, which can impact the computation time of FL. FedMTBoost represents a new approach in federated boosting as it supports various settings and allows for multitask learning if the used classifiers support it. The motivation to extend federated boosting in this way is to leverage the transfer of similar task relationships between partners.

Boosting’s integration into multitask learning can happen in multiple ways. In Online Multitask Boosting (OMB) [9], the method is to generalize transfer by learning models based on some tasks and adapting them to every task with the model weights. Here the tasks can learn their weights based on the differences in their losses. In MTBoost [10], a super-task representation is learned based on task covariances and is further adapted to the individual tasks. FedMTBoost uses a simple combination as the approach starts with a model that already supports multitask learning and extends this with the weighting of the data to support boosting. The motivation for this is the integration to FL, as these models already have a simple method to allow distributed learning.

These techniques are all suitable for DTI prediction as each has a history of performing well in related areas. QSAR modeling is one field where boosting is applied [11], traditionally with gradient boosting. SimBoost is another gradient-boosting model for DTI prediction; it constructs feature vectors for drug-target pairs and trains the ensemble on these as well. Multitask in DTI prediction also seems to improve the performance of models; however, sometimes individual tasks can have worse predictive performances even if the average improves [12]. In recent years, multitask DTI has even been adapted to federated settings on large-scale data sets [13]. FedMTBoost allowed the combining of all these approaches in a simple and efficient algorithm. It heavily builds on the Federated model averaging (FedAvg) [14] and the AdaBoost [15] algorithms. As previously shown the method integrates boosting to a federated multitask setup to leverage the benefit of weights in the transfer learning process.

3 Analysis of the FedMTBoost Method

In this section, we explain the role of FedMTBoost in DTI prediction. To understand the original motivation, we can look at multitask learning. To achieve the best possible performance on a given data set, the tasks need to be selected to maximize transfer. This can be achieved manually, by only training on the

selected tasks, or by adaptively weighting the tasks. The adaptive weighting of the data means that in every iteration, a different set of tasks are prioritized, and the model can develop a more general representation. Our approach of combining multitask learning and boosting is shown in Fig. 1.

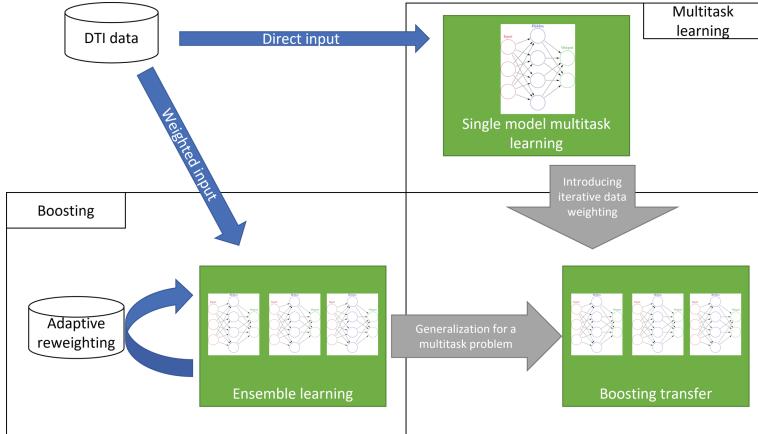


Fig. 1. Multitask boosting problem.

To be able to federate such an algorithm, FedAvg is used, as it is a versatile method when using neural models. The classifiers thus use neural networks to predict activities, and this way, they can be averaged during training. AdaBoost was the choice for boosting as it is a loose framework and can be adapted to arbitrary classifiers. The resulting algorithm has the characteristics of FedAvg from a global (the server's) perspective and from the client's local perspective; it is a simple AdaBoost with extra communication steps.

The research process was the following: First, the multitask boosting algorithm was created, where the classifiers were neural networks and a weighting was added to the data set to support boosting. Initially, the improvement of boosting is not present for local multitask settings. Next, the data set is distributed to clients and a simulation of FedAvg is implemented. Finally, the two methods are combined by averaging the model of the boosting ensemble on the server. In the following section, the experiments are explained in detail to understand the effect of each approach on the final combination.

3.1 Training

The training process of FedMTBoost begins with each client initializing their data weights with uniform weights. In this version of the algorithm, the weights are assigned to each compound, meaning the matrix has uniform weights in its rows. The weight of the i -th compound of the t -th iteration is denoted by ω_{it} .

The server starts with an empty ensemble, and models will be added throughout the training. Each iteration starts with the initialization of a new model; this can happen in several ways: Either the server always initializes a new model and distributes it to clients, or to reduce communication, clients can agree in a seed to initialize models, or they can continue training the previous model.

After the clients receive the models, they can train them using traditional optimization methods with the added difference of data weights being considered in the loss function:

$$w \leftarrow w - \eta \nabla l(w; b; \omega_{(t-1)}) \quad (1)$$

Here w represents the weights of the models, η is the learning rate, and b is the batch the model is trained on. They each train the model for an agreed number of epochs (it should be small so the models do not diverge). After this, the clients can communicate the weights or the gradients to the server, and the server averages these parameters, just like in the FedAvg algorithm:

$$w_t \leftarrow \sum_{k=1}^{|S_t|} \frac{1}{|S_t|} w_t^k, \quad (2)$$

where S_t denotes the set of clients. Now the server can send the averaged models to the clients. The clients can now continue executing the AdaBoost algorithm by calculating the error of the new model on their private data set:

$$\text{Error}_t = \sum_{i=0}^n \omega_{i(t-1)} * I_{y_i \neq \hat{y}_i}. \quad (3)$$

Here the $I_{y_i \neq \hat{y}_i}$ part denotes if the i th compound is correctly predicted. To facilitate quicker learning, we allowed a compound to be classified as correct even if the associated bioactivities are partially correct; we set this threshold to 80% of the activities.

From the error, the weight of the current model can be determined:

$$\alpha_t = \frac{1}{2} \log\left(\frac{1 - \text{Error}_t}{\text{Error}_t}\right). \quad (4)$$

Finally, the data weights can be updated based on the model's performance on the training data set:

$$\omega_{it} \leftarrow \omega_{i(t-1)} * e^{-I_{y_i \neq \hat{y}_i} * \alpha_t}, i = 1, 2, \dots, N \quad (5)$$

Figure 2 shows the setup in a two-client training. We see that the ensemble is shared between the clients, like the model in FedAvg, and every client can adapt them to their data set with their private weights.

Algorithm 1 FedMTBoost

```

1: Initialization on the clients:
2: Initialize the observation weights  $\omega_{i0} = 1/N$ ,  $i = 1, 2, \dots, N$ 
3: initialization of ensemble  $E = \{\}$ 
4: initialization of  $\eta$  learning rate
5: Computed on the server:
6: for  $t = 1, 2, \dots$  do
7:   initialization of  $w_t$ 
8:    $S_t$  = set of clients
9:   for  $k \in S_t$  clients do
10:    Send  $w_t$  to  $k$ 
11:    Recieve  $w_t^k$ 
12:   end for
13:    $w_t \leftarrow \sum_{k=1}^{|S_t|} \frac{n_k}{n} w_t^k$ 
14:   sending  $w_t$  to clients
15: end for
16: Computed on clients:
17: for  $epoch \in Epochs$  do
18:   batches  $\leftarrow$  partitioning data to B partitions
19:   for  $b \in batches$  do
20:      $w \leftarrow w - \eta \nabla l(w; b; \omega_{(t-1)})$ 
21:   end for
22: end for
23: Sending weights to server.
24: Receiving averaged  $w_t$ 
25:  $E \leftarrow E \cup \{w_t\}$ 
26: Compute error  $Error_t = \sum_{i=0}^n \omega_{i(t-1)} * I_{y_i \neq \hat{y}_i}$ 
27: Compute  $\alpha_t = \frac{1}{2} \log(\frac{1 - Error_t}{Error_t})$ .
28: Set  $\omega_{it} \leftarrow \omega_{i(t-1)} * e^{-1^T y_i \neq \hat{y}_i * \alpha_t}$ ,  $i = 1, 2, \dots, N$ .

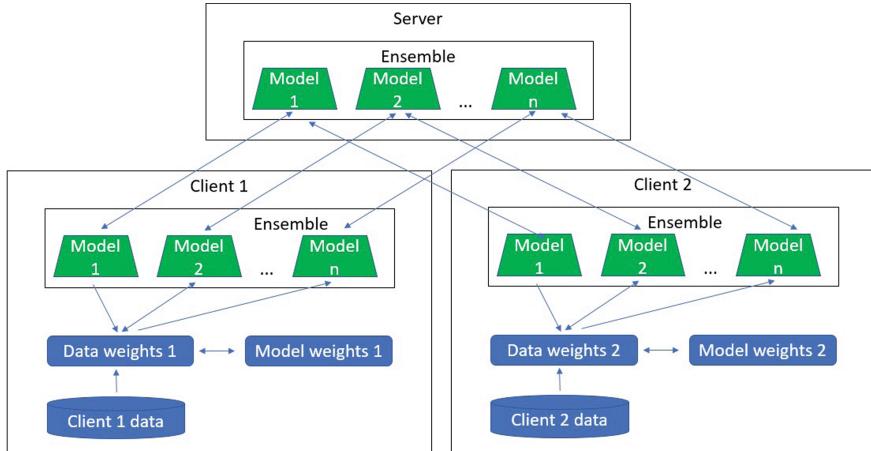
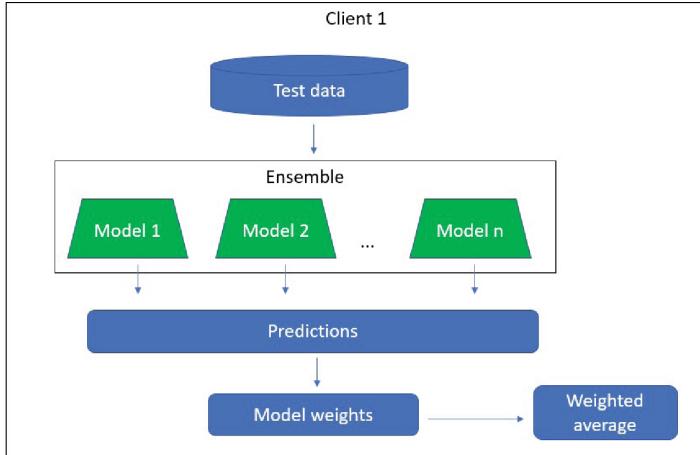
```

3.2 Prediction

When a client wants to predict new data, they can use the prediction method of AdaBoost. It involves feeding the new data set to each model of the common ensemble and weighting the predictions with the corresponding model weights calculated on the client during training. The formula for the prediction can be written as follows:

$$pred = \sum_{i=0}^{|E|} \alpha_i f_{w_i}(x_{test}) \quad (6)$$

Where f_{w_i} is the i -th model of the ensemble. The schematic process of local prediction can be seen in Fig. 3.

**Fig. 2.** FedMTBoost training.**Fig. 3.** FedMTBoost prediction.

3.3 Details of Implementation

In this subsection, we describe the concrete implementation of the algorithm. The classifiers used by the AdaBoost algorithm were Multilayer Perceptrons (MLPs) created in the SparseChem framework [16], which allows to create and train Neural Networks for sparse data prediction. We supplemented the framework with a data loader that handles sparse data weights.

The network’s input was the Extended Connectivity Fingerprint (ECFP) of the molecules, followed by a hidden ‘trunk’ layer of 1400 units. Only this trunk layer is considered when the models’ weights are averaged; thus, the transfer happens in the representation formed here. The output size varies on the number of tasks available.

The networks were optimized by the Adam algorithm with a low learning rate of 0.0001. The loss function on the output was always Binary Crossentropy, with the samples weighted by the corresponding compound weight. Activations in the network are ReLUs, and sigmoids are used on the output to get the probabilities.

4 Evaluation

Federation, multitask, and boosting each impact data, predictive performance, and modeling in different ways. To understand how they should be combined optimally, we must first understand and measure the effect on predictive performance. The evaluation is designed to measure the contribution of every aspect of the solutions; thus, the scenarios were split along the following properties:

- Single- and multitask
- Single-party, multiparty and federated
- Single model and boosting ensemble.

With these distinctions, the scenarios created are shown in Table 1. Most of the experiments serve as different baselines to measure the performance of FedMTBoost. First, in a single-party environment, without splitting the data set, the effect of boosting is measured in a single- and multitask scenario. Next, we can examine the drop in performances when we federated the data set, meaning that every partner can only learn from their part of the data set. When we have this baseline, we can observe the improvement brought in by only federation without boosting in the FedAvg scenarios. Finally, we can compare the FedMTBoost performances to see if it can improve as much as the sum of its parts.

The results will be evaluated on a five-fold split of the KIBA data set. The folds are labeled from 0 to 4, where 0 always plays the role of test data. The remaining four folds are used in a way that every subset is being used for training, and the folds that are not present in the current data are combined with fold 0 to form the test data. This way, we can see the methods’ performances on different data set sizes. The federated data set is split unevenly between 8 partners to measure differences realistically. The number of compounds allocated to each client is shown in Fig. 4. For evaluation, the AUROC and AUPR metrics are used with the DeLong test on the AUROC values to determine significance. We deem an improvement significant if it has a p-value of 0.05 or less.

Table 1. Proposed scenarios to measure the effects of boosting and transfer.

	Single-task	Multitask
MLP	One MLP learning one full selected task	One MLP learning all tasks on the whole data set
AdaBoost	An AdaBoost ensemble of MLPs learning one task	The ensemble learning all tasks, with the AdaBoost weights on the data assigned to the compounds
Partner-wise	Splitting one single-task by compounds to partners and every partner conducts their learning without averaging using one MLP	Splitting the full data set realistically to partners, and everyone performs multitask learning with one MLP without averaging
FedAvg	Using the same split as above, the partners conduct a Federated Averaging scheme on the task	Using the same split as above, the partners conduct a Federated Averaging scheme
FedMTBoost	Using the same split as above, the partners each conduct AdaBoost learning on their part of the task by averaging every model of the ensemble	Using the same split as above, the partners each conduct Adaboost learning on their part of the data set with averaging every model of the ensemble, using compound weighting

The figures each show the AUROC performance on the y-axis and the folds used for training on the x-axis. The different colors represent the different methods used for training, and the dashed lines show the average performance of methods on all fold combinations.

4.1 Summary of Previous Results on Curated Data Set

The original paper evaluated the method on 60 selected tasks from the data set, which led to the AUROC values shown in Tables 2 and 3 in the case of four-fold training on a selected partner. As expected, multitask training performs better than single-task ones, and federation outperforms a non-cooperative scenario. The interesting thing to note is that boosting could only achieve significant improvements in federated training.

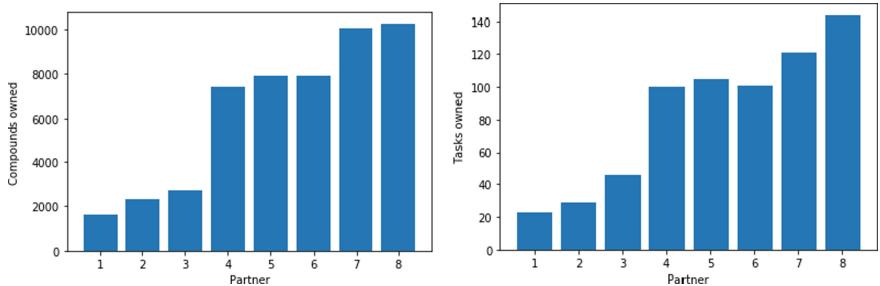


Fig. 4. Number of compounds and tasks available per partners in multiparty and federated scenarios.

Table 2. Single-party average AUROC scores for 4-fold training.

	Single-task	Multitask
MLP	0.7249	0.7885
Boosting	0.7062	0.8007

Table 3. Multiparty average AUROC scores for 4-fold training on partner 6.

	Single-task	Multitask
Multiparty MLP	0.6625	0.725
FedAvg	0.6235	0.7374
FedMTBoost	0.7325	0.7445

4.2 Full-Scale Evaluation on KIBA

The full KIBA data set contained 466 tasks with varying measurements; this means that much more noise is introduced during training, which often deteriorates performances. However, this can show us the optimal way of utilizing the FedMTBoost in an industry-scale setting.

The results will be presented through two perspectives: A partner with little available data and one with more. The selected clients are Partner 1 (the smallest data set) and Partner 7 (with the second largest data set). When looking at the multiparty cases, we can see that the MLP model's performance drops considerably when fewer data are available for clients to train on. Figure 5 shows the performances of single- and multitask cases in a multiparty MLP training for the two partners. We can see that the partners with fewer data suffer from considerably worse performance, and the multitask transfer is not present at that data set size.

When looking at the improvement of FedAvg we see the same differences in the single- and multitask scenarios: larger partners have a stronger transfer. We contrast the performances for the multitask scenario as the final goal is to boost multitask transfer. However, the same results are present in single-task

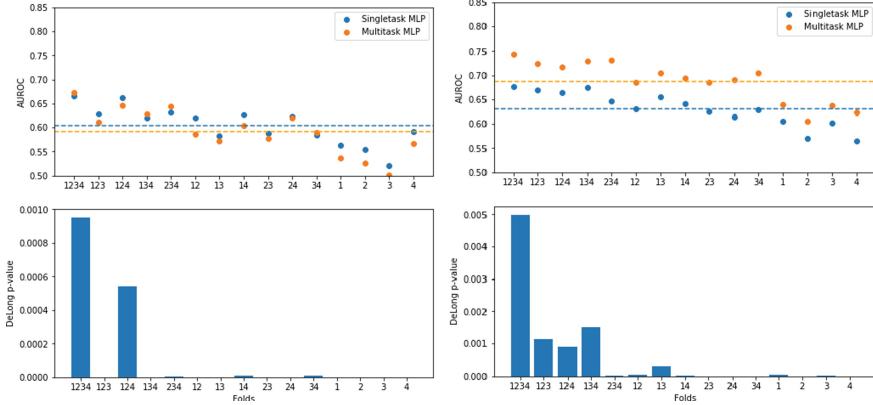


Fig. 5. AUROC values and significance for partner 1 (left) and partner 7 (right) in non-federated multiparty training.

cases as well. The results show that federated averaging does not improve the predictive performance in the case of large data sets; it may even deteriorate the performances. However, on smaller partners, the new information can improve the performance, as shown in Fig. 6

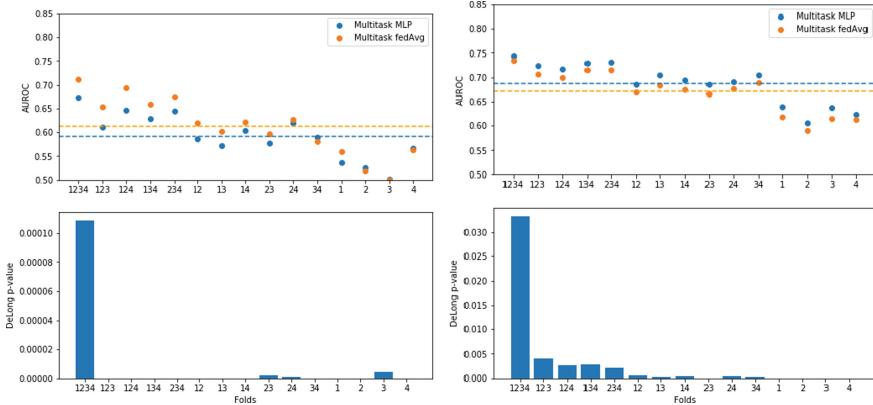


Fig. 6. AUROC values and significance for partner 1 (left) and partner 7 (right) in FedAvg and non-federated multiparty training.

Finally, FedMTBoost is compared to the FedAvg and multiparty MLP scenarios in Fig. 7. Here we can see that the FedMTBoost helps to boost performance in the small partner and further deteriorates it in the large one. This seems to mean that the more data is present, the less the boosting can help, and it starts to bring every client to the same level. The results of FedMTBoost on four-fold trainings can be seen in Tables 4 and 5.

However, the original results show that restricting the training data to a few tasks can improve performances for all participating clients. Figure 8 shows the improvements achieved on a smaller data set.

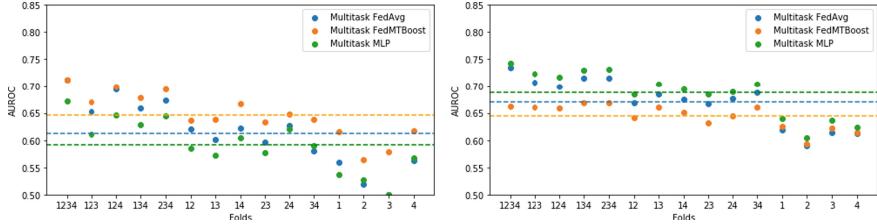


Fig. 7. Results from the full data set: AUROC values for partner 1 (left) and partner 7 (right) in FedMTBoost FedAvg and non-federated multiparty training.

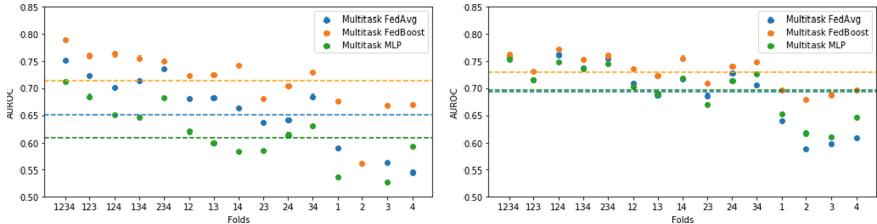


Fig. 8. Results from the curated data set: AUROC values for partner 1 (left) and partner 7 (right) in FedMTBoost FedAvg and non-federated multiparty training.

This leads us to believe that in the full data set, there are tasks that can impact performance negatively. This means that the FedMTBoost is unsuitable for boosting transfer in arbitrarily large data; instead, it works best if the data set is already preprocessed and can pick up on signals better.

Table 4. Multiparty average AUROC scores for 4-fold training on partner 7.

	Single-task	Multitask
Multiparty MLP	0.6755	0.74318
FedAvg	0.6465	0.7345
FedMTBoost	0.7346	0.6630

Table 5. Multiparty average AUROC scores for 4-fold training on partner 1.

	Single-task	Multitask
Multiparty MLP	0.6661	0.6720
FedAvg	0.6580	0.7114
FedMTBoost	0.7215	0.7116

5 Conclusion

We have presented a detailed description of FedMTBoost and the algorithm’s strength. We concluded that it could be useful, but it needs to have pre-processed data as input; otherwise, the weighting can pick up on the less constructive tasks, which will deteriorate performance. This is especially true in federated settings where other clients’ tasks differ or have worse quality than the partner’s tasks. To solve this, further weighting might be introduced for averaging the model parameters on the servers to account for the clients’ quality.

The algorithm proposes several possibilities for future research. The weights produced during boosting might be further utilized to find important and highly predictable parts of the data and help training with imputation. This can be further developed to apply boosting in knowledge distillation-based FL. This can also open the possibility of predicting more accurately for larger data sets. Further evaluation using industry-scale data is also made possible this way.

The small data set size, however, is a constant problem in modern machine learning applications, where training data is hard or expensive to produce. In a scenario where clients have little to contribute to federated training boosting does help to enhance the signals of transfer and contribute to better predictive performance. The case for modeling-related tasks can also be made as traditional multitask setups also operate on this assumption. This means that in small and curated learning problems federated boosting can improve considerably on existing multitask methods.

Acknowledgments. This study was supported by J. Heim Student Scholarship, OTKA 139330, and the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory, New National Excellence Programme of the Ministry of Innovation and Technology, code number ´UNKP-22-2-I-BME-70, funded by the National Research, Development and Innovation Fund.

References

- Chen, R., Liu, X., Jin, S., Lin, J., Liu, J.: Machine learning for drug-target interaction prediction. *Molecules* **23**(9), 2208 (2018)
- Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Zhao, S.: Advances and open problems in federated learning. *Found. Trends Mach. Learn.* **14**(1–2), 1–210 (2021)
- Caruana, R.: Multitask learning. *Mach. Learn.* **28**(1), 41–75 (1997)
- Xu, Y., Ma, J., Liaw, A., Sheridan, R.P., Svetnik, V.: Demystifying multitask deep neural networks for quantitative structure-activity relationships. *J. Chem. Inf. Model.* **57**(10), 2490–2504 (2017)
- Smith, V., Chiang, C.K., Sanjabi, M., Talwalkar, A.S.: Federated multi-task learning. *Adv. Neural Inf. Process. Syst.* **30** (2017)
- Li, Q., Wen, Z., He, B.: Practical federated gradient boosting decision trees. *Proc. AAAI Conf. Artif. Intell.* **34**(04), 4642–4649 (2020)
- Cheng, K., Fan, T., Jin, Y., Liu, Y., Chen, T., Papadopoulos, D., Yang, Q.: Secureboost: a lossless federated learning framework. *IEEE Intell. Syst.* **36**(6), 87–98 (2021)

8. Shen, Z., Hassani, H., Kale, S., Karbasi, A.: Federated functional gradient boosting. In: International Conference on Artificial Intelligence and Statistics, pp. 7814–7840. PMLR (2022)
9. Wang, B., Pineau, J.: Online boosting algorithms for anytime transfer and multi-task learning. Proc. AAAI Conf. Artif. Intell. **29**(1) (2015)
10. Zhang, Y., Yeung, D.Y.: Multi-task boosting by exploiting task relationships. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 697–710. Springer, Berlin (2012)
11. Svetnik, V., Wang, T., Tong, C., Liaw, A., Sheridan, R.P., Song, Q.: Boosting: An ensemble learning tool for compound classification and QSAR modeling. J. Chem. Inf. Model. **45**(3), 786–799 (2005)
12. Moon, C., Kim, D.: Prediction of drug-target interactions through multi-task learning. Sci. Rep. **12**(1), 1–10 (2022)
13. Oldenhof, M., Ács, G., Pejó, B., Schuffenhauer, A., Holway, N., Sturm, N., Galtier, M.: Industry-Scale Orchestrated Federated Learning for Drug Discovery (2022). [arXiv:2210.08871](https://arxiv.org/abs/2210.08871)
14. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)
15. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
16. Arany, A., Simm, J., Oldenhof, M., Moreau, Y.: SparseChem: Fast and accurate machine learning model for small molecules (2022). [arXiv:2203.04676](https://arxiv.org/abs/2203.04676)
17. Song, X., Zheng, S., Cao, W., Yu, J., Bian, J.: Efficient and effective multi-task grouping via meta learning on task combinations. In: Advances in Neural Information Processing Systems (2022)
18. Tang, J., Szwajda, A., Shakyawar, S., Xu, T., Hintsanen, P., Wennerberg, K., Aittokallio, T.: Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. J. Chem. Inf. Model. **54**(3), 735–743 (2014)



Surveying Impacts of AI in Education and Creative Practices

Andy Deck^(✉)

State University of New York, Oneonta, NY 13820, USA
andy.deck@oneonta.edu

Abstract. Fields as diverse as art, photography, writing, and design are now confronting the consequences of easily available—and easy to use—generative systems. Text and image products that used to be understood as direct expressions of the minds of a human creator may now be generated synthetically through software and neural networks. The emergence of such ‘smart’ software poses novel ethical challenges for the evaluation of intellectual products like images and text. There has been a predictable backlash against incursions of automation and generative systems into creative practices that have evolved over centuries. The author surveys critical perspectives on these software systems as they relate to creative practices, and cites examples of how tools and services made possible by neural networks have caused controversy. Further, based on history and trends in mainstream digital culture, the author concludes that popular notions of authorship and creativity will continue to evolve as machine learning and artificial intelligence become increasingly entwined in production tools.

Keywords: Art · Authorship · Creative Practices · Artificial Intelligence · Automation

1 Introduction

In recent years, machine learning and artificial intelligence have emerged forcefully in popular culture. Following many years of software advances and hardware engineering achievements, as well as the appearance command and control systems like Alexa and Siri, intelligent agents are no longer the stuff of science fiction. For many, they are a part of daily life. With this shift, the assessment of the cultural impacts of artificial intelligence moves from the realm of imaginative thought problems into the realm of the observable.

1.1 Suggestive Software

Implementations of machine learning can feel inevitable in the new products that we adopt, such as cellphones. As just one example, touch-screen keyboards rely upon algorithms that borrow from personal data such as contact lists to predict what people are attempting to type. While it may seem trivial to the developers, this intervention into the

interaction between the typist and the text is emblematic of new conditions of authorship. Based on how they are designed, production tools lead to different results. For writing software, the application of neural network algorithms impacts the textual products through suggestion. From grammar correction in word processing, to email software that offers buttons for suggested replies like “Thank you so much!” generic utterances are now promoted. In terms of the practical impacts of artificial intelligence, what does it mean that the ‘Smart Reply’ one-button reply system—which employs “state-of-the-art, large-scale deep learning”—is now used by 10 percent of Gmail’s mobile responses [1]? Has human creativity and authorship diminished measurably?

Typing is increasingly governed by codes that fix what is typed and suggest alternatives. Had the poet E. E. Cummings written poetry with contemporary word-processing software, his expressive process would have been an unending struggle with automated capitalization. Needless to say, mechanical typewriters can still be found in antique shops. For many people ‘smart’ auto-correction is a convenience that speeds up communication. Nonetheless, where education, creativity, and the arts are concerned, it is time to consider the ways that machine learning entrains modes of expression and habits of mind.

2 Machine Learning as Academic Train Wreck

A few years ago, while teaching in a fine art graduate program in New York City, I experienced some of the pitfalls of newly available language technology. Many of my foreign-born students struggled with English. Yet their poorly-written papers were different from ones I had seen before. I discovered, after a lot of sleuthing, that the strangeness that I was experiencing was a consequence of the misuse of online translation software. Since the students were not citing the source texts that they had translated, the slightly garbled results were a kind of plagiarism. Even though tools exist to detect plagiarism, that type of software typically fails to recognize a text that has been translated multiple times. Although it is true that plagiarism has been around for ages, it is easier than ever to find and process texts to meet a set of assigned criteria. Moreover, as has become quite clear now, free, online translation systems were just opening the door: the exploitation of software systems for academic plagiarism was just getting started.

2.1 Writing Does not Equate to Learning Anymore

OpenAI’s ChatbotGPT and Google’s Bard now provide ample opportunities to generate topical essays through the magic of probabilistic algorithms that leverage terabytes of textual data. Using such tools, a student need only suggest a few descriptive phrases and a text-generating engine can emit pages of coherent text that read as though they were written by an informed person. As Stephen Johnson put it, “OpenAI’s GPT-3 and other neural nets can now write original prose with mind-boggling fluency” [2]. Since the widespread release of such tools, every essay that a professor reads now feels like it is a Turing Test. Did the student actually write the paper? Was it generated synthetically by a machine learning algorithm? This startling transformation of word processing is having a momentous impact on academic practices. Since some students will undoubtedly

make use of free text generation tools to fake their way through college courses, higher education institutions are scrambling to ensure that their degree credentials continue to signify genuine academic work. A frenzy of workshops and revised academic guidelines are being offered to teachers. Textual reports and essays are no longer a practical form of assessment that signifies that a student has engaged in critical thinking and research. Plagiarism detection for these new tools is imperfect and will likely remain so for some time because they are evolving quite rapidly.

While academic institutions rely on idealistic and ethical principles of education, it is notable that despite the disruptive impact of these generative tools on education, few are seriously advocating that they be banned or blocked. Like the World Wide Web in the 1990s, which presented similar challenges to academic integrity due to the proliferation of downloadable essays, the prevailing view continues to be that the academic culture must adapt to the existence of new forms of information technology.

2.2 An Evolving Ethical Frame

Will the unattributed use of algorithmically generated text continue to be seen as plagiarism—and grounds for expulsion—in higher education? Adopting a permissive posture toward the liberal use of copy and paste from machine-generated responses would seem to be a violation of the academic mission to expand student understanding. Yet there are a number of reasons to believe that there will be minimal resistance from academia.

Several factors justify this conclusion. First, the fundamental role that neurally networked, dialogic ‘chat’ systems have begun to play in knowledge acquisition suggest that academic ethics will bend to accommodate the widespread adoption of this new technology. Consider that the search engine Bing already has introduced a prominent ‘Chat’ button that reflects its investment in OpenAI and the adoption the dialogic ‘chat’ metaphor. As users of mainstream search tools like Bing grow accustomed to this mode of inquiry, the epistemic precision about where ideas (and sentences and paragraphs) came from will become increasingly blurry. From ‘conversation’ with ‘intelligent agents’ like Microsoft’s Sydney or Google’s Bard, ideas and verbiage arise: the policing of who said what is impractical for professors and teachers to accomplish. It is much easier for them to throw up their hands and to accept that prompting and reading ChatbotGPT responses has some redeeming intellectual value.

Decades before ChatbotGPT, Wikipedia introduced another heavily-plagiarized, epistemologically-problematic information technology. Biased volunteer contributors and pranksters cast a shadow over the encyclopedic aspirations of the project, leading many to view it as a disreputable source. Nevertheless, countless academic papers have copied Wikipedia and rephrased its content in trivial ways. In 2011 a study from Turnitin reported that the most copied website by both secondary and higher education students was Wikipedia [3]. Neither expulsions nor plagiarism detection software have truly stopped students from copying Wikipedia.

Additionally, cross-cultural research suggests that open access to information and communication technology via the Web have been the primary drivers of plagiarism, irrespective of national borders, motivations for schooling, and socialization. In other words, when new technology makes it easy to cut corners and to circumnavigate the difficulties of research and writing, many students do seize those opportunities [4].

2.3 Beyond Academic Integrity

Actions that are punishable in higher education may be considered ‘tactical media’ outside of academia. Beyond academia, rule-breaking akin to plagiarism often brings people success. In social media, for example, impersonating accounts of famous people is common and, when it is discovered, it is punished only through account deactivation. On Twitter, operating a multiplicity of accounts tied to fake identities has been a successful strategy for manipulating users. Operating bot farms to promote ideas, candidates, and ideologies in social media, and leveraging scripting to release tweets that are incessant and targeted are examples of practices that have given some Twitter users an outsized influence. Efforts to curtail these kinds of deception have been half-hearted and ineffectual.

The ‘deep fake’ video is a comparable type of fraud that has become popular. This motion-graphic identity-spoofing technology makes it quite easy to superimpose a person’s face into a photographic video. Deep fake production tools permit the synthesis of speech and facial movements. The videos spread as propaganda and they usually distort the views of the faked person. They also undermine the truism that seeing is believing. But deep fake products are often a hot commodity in the meme culture of social media. Viewers and public figures must then grapple with the misrepresentations. In the ethical vacuum of transnational social media, the deep fake has become for video producers just another tactical tool in the media toolbox. When enough people have access to an advanced technology—even when it is ethically problematic—it becomes difficult to put the genie back in the bottle. In the realm of social media, new digital practices and tools that have the potential to increase followers and generate media attention have had a tendency to overpower ethical reservations.

Of course, the technologies that give rise to digital misrepresentation are international in scope. Fresh and secretive media tactics may be interpreted as corrupt or clever depending on context. Diverse cultures, institutions and sub-cultures, will come to different conclusions about which deceptions are unacceptable.

3 From Understanding to Creativity

Viral deep fake videos are often instruments of propaganda that can be construed as slander or defamation. In many instances, however, they can be interpreted as pranks or parodies. When the intent is to parody, a deep fake may deserve protection as an expressive ‘speech’ act, like the celebrated art pranks of the Yes Men. This leads to an important matter: how a new technology is used. While it is true that generative text systems will be used for cheating in school, it is nevertheless true that many people will find more redeeming and interesting ways to use these new tools. This raises issues related to creativity that will consume the remainder of this article.

3.1 Insight from Deep Learning

From the point of view of a young and inexperienced writer, the newfound technology to generate paragraphs of cogent text instantly, with very little effort required, could

certainly appear rather enticing. It is common in creative domains other than writing to consult canned content, to speed up work or as inspiration. Designers, for example, sometimes browse through books of color palettes, in order to select a harmonious color palette for a given design. It is not uncommon for students in art schools to replicate a well-known painting or to make drawings of great paintings. In effect, students are given assignments to engage in something like forgery in order to understand historic paintings in greater detail. But the motive of those assignments is not to teach forgery: it is to internalize the content and construction of classic art. It supposes that the students will be able to apply the insight that they have absorbed in their own creative practices. It begs the question, how does the novice become the master? Time-honored models of art education, like apprenticeship, have assumed that an important element of knowledge is repetition and rote memorization. Copying from the canon and learning to replicate its qualities establish a foundation from which to build a more personal body of work.

For writing, one can argue that using a ChatGPT-style system to conjure up subsequent phrases and sentences is just another approach to invention, no different from the way designers browse collections of palettes. But if the majority of writers were to use and seek inspiration from the same text generating machines, what would be the cultural impact? What would they memorize and internalize as a foundation?

3.2 Reliance on Software

Generative A.I. systems today are basically imitative: they digest and regurgitate data. It is also true that a lot of what passes for human creativity is rather imitative; even so, a reliance on machine learning ‘bots’ is a questionable detour for writing and literary culture. One impact, as noted above, would be the promotion of normative, conventional, and grammatical language. These could become self-reinforcing biases as texts generated by today’s prototypes become a part of next-generation training sets. Additionally, the virtual writing assistant could usher in an age of intellectual laziness. Rather than carefully planning the sequence and substance of a bit of writing, writers may begin to go about their work in a state of distraction as if it were a sort of multiple-choice activity. The results would be a form of writing, to be sure. But as writing is transformed into a kind of interaction with a virtual intelligence, the writer thought process is transformed. New habits of mind take hold: the writer relies on software.

The situation is comparable to the impacts of navigational systems. Prior to the deployment of the Global Positioning System (GPS), travelers had to learn more about where they were going. They usually internalized the information that they needed to navigate from one place to another. Sometimes this required maps. People stopped a lot more to ask for directions. It was more important to remember the route. Today, when using GPS navigation, there is little motivation to memorize the route because it is so easy to rely on a navigation system. It is also efficient because digital way-finding systems monitor road closures and traffic conditions that are not listed on printed maps. There is little question that a great many people have become reliant on navigational systems. Not everyone will agree on the merits of this transition to greater reliance on the technosphere.

The Greek tragedian Aeschylus wrote that memory is the mother of all wisdom. Is it wise to rely upon neural networks to generate ideas? Do creative surrogates such as

the aforementioned text generation systems invoke a kind of memory that is capable of wisdom and insight? After all there are fundamental distinctions that can be drawn between digital memory and the workings of the human brain. Deeply impactful experiences such as grief, love, and betrayal impact how people think, what they recall, and what things they value. Programmed machines never have such experiences or, for that matter, feelings. They do not know regret, urgency, joy, or mortality. The ‘utterances’ of A.I. are borne of another mother. Emotions and experiences are beyond the grasp of algorithms that suggest words and sentences on the basis of probabilities, data sets, and the previous words and sentences.

In “The False Promise of ChatGPT” the linguists Ian Roberts and Noam Chomsky, and A.I. scientist Jeffrey Watumull caution that ChatGPT and similar innovations should not be misunderstood as “the first glimmers of artificial *general* intelligence—that long-prophesied moment when mechanical minds surpass human minds not only quantitatively in terms of processing speed and memory size but also qualitatively in terms of intellectual insight, artistic creativity and every other distinctively human faculty” [5]. Regarding the importance of “large language models” (LLMs), and whether they may represent such glimmers, there is disagreement. The Berkeley professor Edward Lee skewers Chomsky and his co-authors for their lack of scientific method [6]. For the purposes of this article, the arguments in “The False Promise of ChatGPT” are relevant to questions concerning creativity. The crux of machine learning, they write, “is description and prediction; it does not posit any causal mechanisms or physical laws” [5]. These limitations may prevent ChatbotGPT from being sentient or creative, but they do not necessarily prevent people from using it in the service of creative pursuits, as a tool. There is no harm in cautioning people about the limitations of unregulated and much-hyped technologies, provided that their remarkable potential is not overlooked in the process.

Perhaps the trouble is in the details, because it is tricky to be conscious of what we learn from the tools we use. Grammar checkers and email software ‘suggested response’ buttons have a role to play in language education. Rather than bemoan a machine’s inability to write more nuanced and creative prose, it must be remarked that the new generative tools are designed to be operated interactively with a human. For example, LLMs can be used to mimic writing styles. Every writer can now interrogate a kind of expert system that knows the style, diction, and grammar that define a given discursive terrain. Writers can learn from stylistic mimicry. Still it is hard to believe that the interactive writing assistance of an A.I. system can be simply assigned the status of a tool. The interactive process would seem to push writers toward a kind of co-authorship. Like it or not, this hybrid authorship is coming to screens and pages near you. Writers have already begun to rely on tools to generate ideas and to inform the thoughts that they are expressing. Notwithstanding these problems of over-reliance and limited wisdom, the range of justifiable uses is quite broad.

3.3 Parallel Innovations in Imaging

The questions and controversies surrounding writing and creativity in generative text are mirrored by questions and concerns surrounding deep learning models that are now being used for generating images. OpenAI’s DALL-E and DALL-E 2, and a separate

software project entitled Midjourney, are provoking conversations about the potential and limits of machine learning systems in art, design, and other visual domains.

One problem pertains to assessments of artistic merit. An art competition during the 2022 Colorado State Fair revealed the anxiety about imagery synthesized through machine learning. The competition's judges awarded a divisional prize to an image that had been ink-jet printed onto a canvas. Despite a clue appended to the entrant's name, the judges were unprepared for the possibility that the image had been generated using software that exploits machine learning and deep convolutional neural networks. The contestant, Jason M. Allen, typed descriptions related to the terms 'Space Opera Theater' into the Midjourney software until it generated his winner. Paradoxically the imagery made with Midjourney feels at once strange and familiar because of the ways that keywords are reflected through a dense matrix of training imagery. Other artists, who used more conventional techniques to produce their contest submissions, were critical of the judges' decision. It must be said, however, that Allen's submission conformed to traditions of balance, composition, and color that until now have been attained only by experienced artists who are familiar with art history and painting. Allen was unapologetic for his success. The controversy raises the question: why should Allen be punished for using a generative tool to make his image when the judges believed it was good?

There are divergent perspectives. For one, the winning image was comprised only of colored pixels. For many traditional painters the absence of a physical artifact is a defect. Yet the surface characteristics of conventional paintings could be emulated, also. Next-generation image synthesis could produce the painterly qualities and textures of paintings by using machine learning mimic the surface attributes of celebrated canvases. Had Allen used paints and brushes to copy the same procedurally-generated image onto a canvas, would that have made it worthy?

Although a lot of today's generative imagery tends to be surrealistic, criticism of A.I. art that focuses on visual content may be undermined as the techniques mature. Although the generative process is a little difficult to control today, the potential exists for these systems to make almost any kind of image. In the end one is left to wonder if the crux of the problem, as with plagiarism and academic integrity, is simply that making imagery with generative tools is too easy.

To use the Midjourney tool is to become engrossed in the iterative exploration of possible images with a group of other users who seem to be focused on crafting something specific with the tool. Users type a variety of prompts while visualizing imagery to satisfy their whims. The winner of the Colorado competition made vague claims about having spent upwards of eighty hours coaxing his image out of the Midjourney software. Despite Allen's claim, it is likely that he misled people to believe that he worked much longer than he really did.

3.4 Tradition and Anti-technological Ideals in the Arts

The stunning appearance of new image-making tools, capable of mashing up pictures and visualizing ideas based on a few words, seems to disrespect the more arduous practices of painting. The situation is reminiscent of the arrival of the Daguerrotype. Upon seeing examples of the Daguerrotype, in 1840, the painter Paul Delaroche declared that 'from this day, painting is dead.'

In some ways the current departure from the laborious practices of traditional painting has been coming for decades. As Timothy Binkley wrote in 1988.

One hallmark of interactivity with an “intelligent” machine is the ability to discourse in generalities and dispense with the need to delineate all the specifics: we can tell the computer to adjust properties of objects or images without delineating each and every detail as a painter must in manipulating pigment. Since the computer understands concepts, we can tell it to make the mountains rougher without saying exactly how it is to be done. This makes it possible for the artist to work at a higher level of generality [7].

Making a breathtaking and unique image by typing a few words—as with Mid-journey—is a high level of generality, indeed. However, the ease of use undermines a traditional reverence for the hard-won mastery of painting.

The seeds of objections to ‘A.I. art’ extend back to before the Renaissance, to a time before mirrors, optics, and lenses changed European painting. With the arrival of perspective, painters opted more and more for compositions and styles that conveyed a cohesive point of view, as if the artist’s eyes had been replaced by a camera. In his BBC mini-series *Secret Knowledge* [8], the artist David Hockney argues that the application of imaging technology like concave mirrors, lenses, the *Camera Obscura*, and the *Camera Lucida*, spurred a profound transformation in European painting. Representational approaches became common that were very difficult to achieve with paint and the naked eye. According to Hockney’s research, optical tools began to be applied sometime in the 14th century, although those uses amounted to a trade secret. Such secrecy conferred a competitive advantage over other painters, and it also sustained beliefs about artists’ skill and genius. Practically speaking, optical imaging tricks allowed painters to shift their attention to other aspects of image-making. Instead of needing to painstakingly perceive and reproduce every visual aspect, painters could operate “at a higher level of generality” by tracing and coloring complicated scenes that were projected onto their canvases. Due to the fact that most art historians have neglected Hockney’s findings, many artists who watch *Secret Knowledge* for the first time are shocked to discover that imaging technologies have existed for centuries—along with anti-technological biases about artistic genius. When photography emerged in the nineteenth century, early practitioners were seen as technicians. Likewise, painters were secretive about their use of reference photos. The pattern extends into the age of computing.

Artists continue to conceal their dependence on digital tools, knowing as they do that viewers and critics are biased against processes that involve too much technology. Ironically digital imaging and fabrication are now so pervasive in contemporary art that it is atypical for artists to produce and distribute their work without some reliance on computing. Consequently, it is nonsensical to imagine ways that accolades in art competitions could be reserved only for artists who adhere to traditional painting. However there are legitimate questions about according genius to every dabbler who makes an image with DALL-E. Dissatisfied responses to the rise of A.I. imagery are, in part, an understandable reaction to radical innovations that threaten to disrupt more established forms of visual creativity.

3.5 Fear and Loathing of Automated Creativity

Enabling people to produce manga effortlessly can lead to controversy. A French game developer learned this in 2022 when he created a tool that leverages machine learning to produce manga. All of a sudden users without appreciable talent began making professional-looking manga. Using deep learning to automate the production of manga, in the style of Kim Jung Gi, caused a passionate response from that artist's followers and from the manga community more generally. The developer, known only as '5you' for his own safety, says that he received death threats from Jung Gi admirers, as well as from illustrators who resented the appropriation of the recently deceased artist's work. '5you' believes that the backlash reflects a fear among illustrators: "I think they fear that they're training for something they won't ever be able to live off because they're going to be replaced by A.I." [9].

As with 19th century collodian process photography, which supplanted the practice of hand-painting slides for 'magic lantern' projections, automating the production of manga could in fact disrupt some careers. Replacing skilled artists with A.I. tools extends the shadow of technologically-driven unemployment to a whole new gamut of creative professionals who, until recently, felt safe from the impacts of automation. Whether it is making artists and illustrators expendable (see Fig. 1), or replacing human fashion models with evergreen virtual models (such as Lil Miquel who has more than 2 million followers), visual culture is transforming in surprising ways.



Fig. 1. Image made by the author in 2023 using Midjourney. Terms such as 'Picasso' and 'Three Musicians' were used as prompts.

Even a relatively hot new field like web design may be at risk. The author Steven P. Anderson described this scenario with a lecture at the 2019 conference South By South West (SXSW) [10]. In essence, through the agency of machine learning software, website visitors can be shown slight variations on a website design. By monitoring the 'bounce rate,' and other interactions, the qualities of the website can be subjected to

machine learning. In a nutshell, a website can be optimized iteratively to use the most effective typefaces, font sizes, photos, color schemes, etc. The skills of the web designer may soon be in direct competition with such clever software. Amateurs could well feel emboldened to employ expert systems rather than hiring a designer who feels like a luxury rather than a requirement. Of course this could be terrible or great depending on your point of view.

3.6 Ideals of Creativity

In classical Greece, Plato held painters in low-esteem, seeing their work as less than creative. They just imitated what they saw. For Christians, creativity was seen as God's work throughout the Middle Ages in Europe. Even centuries after the Renaissance the term 'creativity' seemed too proud for Christendom, where the word suggested the Latin notion of creation '*ex nihilo*' (from nothing). In the 19th century the notion of art as human creativity was becoming less taboo.

Nevertheless, when the photographic pioneer Henry Fox Talbot published his ground-breaking research in the middle of the 19th century, he named his book of photographic calotype prints the *Pencil of Nature*. Talbot, like other early inventors of photography, saw himself less as a creator of images and more as one who was helping to reveal the images that nature had generated on his treated paper. Innovators rarely exhibit such modesty today. Software marketing and industry hype assure us that we are creative. We are told that it is easy for everyone to create astounding things; it is unclear which is more impressive, the software or our casual mastery of it. In this new world, illuminated by the glow of marketing, there is no reason to acknowledge the Almighty, or nature: all credit for this wonderful creativity should be ours. Clearly the developers of simplistic application software would like everyone to adopt this perspective, and to believe that we can become creative simply by using a product. Outlandish huckstering has been around for a while, but with the popularization of technological production tools, the zeitgeist of self-centered and intoxicated pseudo-creativity has been taken to new levels.

Despite the exaggerated marketing, actual usefulness is somewhat matter. As with older physical tools, new digital tools are sometimes very useful and other times not so practical. Since 'creative software' is different from a brush, there are some important distinctions to be drawn. For example, traditional paint brushes never record your location or actions, and they do not place targeted advertisements in front of you as you work. Digital applications do this, however. When tools are providing content to creative intellectuals ("users"), and research into the evolution of those tools is not controlled by the users, who are the creative force behind the interaction of the software tool with the users? Is it the users, the programmers, or the corporation producing the software? Half a century ago, the poet W. H. Auden argued that the true men of action, those who transform the world, are not the statesmen and politicians but the scientists. Contrary to Percy Bysshe Shelley's oft-quoted notion that poets are the unacknowledged legislators of the world, Auden made dismissive comments about the impact of artists [11]. Auden's wistful observation can be reformulated to ask if engineers, scientists, programmers, and industrial interests are becoming the dominant players in 'creativity' rather than the artists, designers, and writers who use 'smart' tools.

Since the advent of personal computing in the 1980s, the computer has shifted away from its early allure as a productivity tool. Today it is a hybrid device that also resembles a toy, even for adults. The turn towards play can be seen in popular forms of putatively creative software that leave relatively minor choices to operators. This calls to mind kid's art applications that trivialize creativity in favor of splashy interactive entertainment. To adopt a baseball metaphor, as A.I. applications proliferate, will young artists and writers be encouraged to believe that they 'hit a triple' when in reality programmers put them on third base? Consumers may not be concerned about this expanding automation of creativity, but it would seem that the postmodern ascent of pastiche will be supercharged by the proliferation of A.I.-infused tools.

Despite the relative infancy of generative systems for text and imagery, people's delight at seeing spectacular effects, and their enjoyment of effortless creativity, suggest that these kinds of pastimes are here to stay. When casual software users are presented with a limited range of creative options, they may embrace such tools because they are simple. There are innumerable cellphone apps that let users render photography as cartoons, or that give users rabbit ears, or that apply users' facial expressions to panda heads. From facial photo distortion tools to deep fake video software, media technology now provides an expansive terrain littered with easy-to-use techniques of simulation, photographic filtering, enhancement, and manipulation. Live effects unleash unprecedented experiences as networked computing becomes a performative context. Through venues like social media, augmented reality, and the Metaverse, an array of seductive spectacles has begun to marginalize traditional modes of creativity. Should stars, ratings, and 'liking' promotion supersede the imprimatur of artistic quality that critics and art competitions have traditionally conferred?

When a small range of styles are baked into popular software applications, the resulting imagery belongs more to the software applications than to the artists. Yet such canned visual effects now appear uncritically in art school admissions portfolios. If it only mattered that a student felt satisfied while producing the images, this tendency to celebrate shallow creativity would not merit further reflection. Yet as deep learning content generation becomes increasingly enfolded into the tools that are used by 'creatives,' questions about the auto-piloted qualities of visual innovation will assume a greater cultural importance.

Bing, mentioned earlier for its embrace of chat-like text generation in its search interface, has also introduced image generation in the same Bing Chat. Dubbed the 'Bing Image Creator,' this tool, which is powered by DALL-E, is only available through Bing's 'Creative Mode' [12]. It is clear from this branding that Microsoft wants to sell its services as a form of creative empowerment. Used as an interactive tool, this rhetoric is not without a basis. Indeed, an abundance of strangely impressive imagery is emerging from contexts like this. Even so, it is hard not to question the future of these tools. Given that Microsoft transformed an open source initiative—OpenAI—into private, closed-source infrastructure for its search tools, it is apparent that the evolution of this platform is big business and not subject to the control of individual users.

What is the future of the creative process in the realm of software? Robotross is an online interactive artwork that I made in 2001 that addresses the evolution of creativity in interactive media [13]. The title alludes simultaneously to the artist Bob Ross, to a

robotic version of him, and to the albatross, a mythic sea bird symbolizing both good and bad luck. Robotross provides a user interface not too different from a user-friendly paint program. Although the main interface seems to promise paint brush tools, clicking them merely pastes pre-made visual elements that users can then re-position. Users move the collection of pictorial elements—a tree, a mountain, a sky, nuclear cooling towers, etc.—to make their own digital compositions. But the various paint brushes that dominate the user interface are intended to be frustrating: interacting with them does not change the tool, it just causes another unpredictable picture to appear in the scene. While the interface seems friendly enough, in practice one is faced mostly with awkward constraints of creative control and freedom. In juxtaposition with Ross's traditional painting, the Robotross addresses the seductive limitations of interactive media as a creative tool. In the context of a technoculture that takes progress largely for granted, Robotross serves as a kind of culture jamming that responds to uncritical beliefs about the creative potential of digital media.

A. Michael Noll examined the creative process in the context of computing during the 1960s. Noll, a pioneer of computer art, used a pen plotter printer to make series of black and white drawings. To accompany his research, he wrote cogently about the rapport between the artist and the computer.

In the computer, man has created not just an inanimate tool but an intellectual and active creative partner that, when fully exploited, could be used to produce wholly new art forms and possibly new aesthetic experiences.... The artist's role as master creator will remain, however, because even though the physical limitations of the medium will be different from traditional media, his training, devotion, and visualization will give him a higher degree of control of the artistic experience. As an example, the artist's particular interactions with the computer might be recorded and played back by the public on their own computers. Specified amounts of interaction and modification might be introduced by the individual, but the overall course of the interactive experience would still follow the artist's model [14].

Noll's words continue to offer a fairly good summary of the relationship that many programmer-artists have maintained in their creative computing over the past half century. Even so, like a number of other enthusiastic early adopters Noll's descriptions missed some of the shortcomings. Over the years it has become clear that computers are products, and not simply "intellectual partners" in some unproblematic way. Additionally today most artists that use computers are not programmers. Therefore most artists are using commercial software products. Such tools often demand periodic retraining and considerable expenditures. Although in some ways artists who use computers may still be considered to be "master creators" who apply their "devotion" to maintain "control of the artistic experience," in many respects the artists and other 'creatives' are constrained by executives, lawyers, programmers, and market forces that shape the priorities of the computer and software industries.

4 Conclusions

Despite concerns and controversies surrounding the impacts of new image and text generation tools, historical precedents and industrial momentum suggest that reluctant institutions and cultural traditionalists will be forced to adapt. Like skepticism toward

photographic art in the 19th century, the resistance to image generation systems and text generation systems will fade. The proliferation of a range of fun and practical applications that use deep learning and LLMs will continue to transform productive practices in a variety of creative fields. Despite a range of controversies discussed above, the integration of these new generative technologies into mainstream information services will ensure their profound impact on education, research, writing, visualization, design, and the like. While some have pointed out that current probabilistic approaches to generating texts and images lack a capacity for understanding that should inform creativity, interactive tools that leverage large data models are already transforming the creative practices of many people. In the wake of two centuries of stunning technological development, it ought to be evident that every productive method and technical norm is bypassed in time. Although new approaches to creativity may seem to be too radically new to be artistic, this has seldom stopped people from adopting powerful new technologies. On the contrary it is the popular conceptions of art, writing, and creativity that will soon evolve to make room for the unprecedented new tools.

References

1. Kannan, A., Kurach, K., Ravi, S., Kaufmann, T., Tomkins, A., Miklos, B., Corrado, G., Lukacs, L., Ganea, M., Young, P., Ramavajjala, V.: Smart reply: automated response suggestion for email. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 955–964 (2016)
2. Johnson, S.: The writing on the wall. In: The New York Times, Sunday Magazine p. 28 (2022)
3. Bailey, J.: Wikipedia, Yahoo Answers Tops for Academic Copying. PlagiarismToday (2011). <http://plagiarismtoday.com/2011/11/03/wikipedia-number-most-common-destination-for-plagiarists/>
4. Jereb, E., Perc, M., Lämmlein, B., Jerebic, J., Uh, M., Podbregar, I., et al.: Factors influencing plagiarism in higher education: a comparison of German and Slovene students. PLoS ONE **13**(8), e0202252 (2018)
5. Chomsky, N., Roberts, I., Watumull, J.: Noam Chomsky: the false promise of ChatGPT. In: The New York Times. Opinion (2023). <https://www.nytimes.com/2023/03/08/opinion/noam-chomsky-chatgpt-ai.html>
6. Lee, E.: Is ChatGPT a false promise? In: Berkeley Blog: Science & Technology (2023). <https://blogs.berkeley.edu/2023/03/19/is-chatgpt-a-false-promise/>
7. Binkley, T.: The computer is not a medium. Philos. Exch. **19**(1), 162 (1988–1989)
8. Hockney, D.: Secret Knowledge: rediscovering the lost techniques of the old masters. Viking Stud. (2001)
9. Deck, A.: AI-generated art sparks furious backlash from Japan's anime community. In: Rest of World (2022). <https://restofworld.org/2022/ai-backlash-anime-artists>
10. Anderson, S.: The future of design: computation & complexity. SXSW (2022). <https://stephenanderson.medium.com/the-future-of-design-computation-complexity-a434d2da3cd5>
11. Auden, W.H.: The poet and the City. In: The Dyer's Hand. Vintage Books (1962)
12. Lardinois, F.: Microsoft brings OpenAI's DALL-E image creator to the new Bing. TechCrunch (2023). <https://techcrunch.com/2023/03/21/microsoft-brings-openais-dall-e-image-creator-to-the-new-bing/>
13. Deck, A.: Robotross: the Joy of software. Artcontext (2004). <https://artcontext.net/robotross/>
14. Noll, A.M.: The digital computer as a creative medium. IEEE Spectrum. **4**(10), 88–95 (1967)



Active Risk Mitigation for Unmanned Aerial Systems

Andrew Kendall^{1(✉)} and John-Paul Clarke²

¹ Georgia Institute of Technology, Atlanta, GA 30332, USA
akendall17@gatech.edu

² The University of Texas at Austin, Austin, TX 78712, USA
johnpaul@utexas.edu

Abstract. Efforts are underway to introduce Unmanned Aerial Systems (UAS) into routine cargo operations within the National Airspace System (NAS). Many argue that UAS operations will be riskier than operation with pilots onboard the aircraft because there are no humans onboard to respond to off-nominal events. In this paper, we present a safety assessment methodology where agent-based modeling and rare event estimation are used to determine the risks due to component and system failures, and thereby accident probabilities. The methodology is applied to the final approach and landing phases of flight within the context of an investigation of the performance required from decision-making agents when wind shear, navigation errors, and engine failures occur. Results indicate that the online safety assessment methodology is able to actively mitigate risk and provide the autonomous system safety performance required to comply with FAA airworthiness regulations.

Keywords: Unmanned Aerial Systems · Autonomous System Safety · Rare Event Estimation

1 Introduction

Aviation is becoming increasingly autonomous, in large part due to the ever increasing reliance on automation. Since the early days of flight, automation has been introduced to augment pilot senses in low visibility conditions, eliminate distinct human roles such as flight engineer and navigator, and automatically fly procedures and perform landings with minimal human intervention. A pressing goal in aircraft automation is to introduce routine cargo transport operations using Unmanned Aerial System (UAS) into the National Airspace System (NAS) [33]. Concepts for such unmanned operations range from a single Remote Pilot (RP) operating a single Unmanned Aerial Vehicle (UAV) via a Ground Control Station (GCS) and Command and Control (C2) link, to M:N operations in which a small team of RPs oversee a larger fleet of UAVs.

Remotely Piloted Aerial Systems (RPASs) have been in regular use in military operations for decades, however, large RPAS have not yet been shown to

meet the standards required for routine civilian operations in the NAS. Several potential difficulties exist due to the latency and reliability of the communication and control (C2) link between the RP and the UAV, and the inherent change in situational awareness due to the RP not being physically situated on the UAV [3]. C2 links are used to uplink commands such as flight controls, chosen procedures, and configuration changes from the GCS to the RPA and down-link avionics data, camera feeds, and alerts back down to the GCS. C2 links can have terrestrial radio line-of-sight communication latencies as low as 10 s of milliseconds or Satellite Communication (SATCOM) round trip latencies up to nearly 2 s [38]. Availability of C2 links depends on the presence of buildings, infrastructure, and terrain between the UAV and the ground antenna.

The impact of C2 latencies and lost links on terminal operations is more significant compared to enroute operations. Approach and landing operations present significant challenges due to the precision required to touchdown safely, sensitivity to faults and weather events, and the decrease in number of alternative options available as the ground gets closer. Although taking up only several minutes of flight time, approach and landing accidents consistently contribute to between 40% and 50% of all manned aviation accidents [1]. Significant effort has been made to increase the safety of approach and landing operations via operational procedures, instrument aids, and automatic control.

Landings in Instrument Meteorological Conditions (IMC) have many parallels to RPAS landings, as the pilot has limited access to direct visual cues and must instead rely on on-board instruments to maintain situational awareness. Sensor noise, bias, errors, and failures introduce navigation system errors, possibly resulting in incorrect readings on the instruments. Thus, instruments can only be certified for a narrow set of conditions. The widely implemented Instrument Landing System (ILS) has been certified to perform in a range of low visibility conditions but it does not address the specific complications associated with RPA landings.

We utilize the methodology that is used to certify ILS to generate safety requirements for certifiably safe RPA approach and landing procedures. ILS approaches have a minimum Decision Height (DH), the lowest height to which the aircraft may descend without direct visual contact with the runway lights [12]. If no visual contact is made, position uncertainty is too large to proceed safely, and a Missed Approach (MA) must be executed. Depending on system precision and redundancy, the DH ranges from 200 feet for CAT I ILS procedures and 100 feet for CAT II ILS procedures, down to 0 feet for CAT III ILS procedures. The premise behind the decision to continue landing or execute a MA is based around maintaining an option that satisfies a Target Level of Safety (TLS). Concepts used to certify Required Navigation Performance (RNP) procedures are also considered. Rather than designing a one size fits all procedure for any aircraft, RNP only allows aircraft equipped with navigation and control systems that meet minimum performance requirements to fly particular procedures [19].

The methodology we propose to assess the safety of unmanned aircraft procedures, with emphasis on approach and landing procedures, directly estimates

the probability of accidents. The safety assessment is applied in an online setting to dynamically estimate risks due to various failure conditions. An agent onboard the UAS actively mitigates risk by making the decision to initiating a missed approach when delaying the decision any longer will result in an unacceptable level of risk. The remainder of the paper is structured as follows. We next present background and motivation for UAS safety followed by a formulation of the relevant FAA safety regulations as constraint on accident risk for each failure condition. We then propose an agent-based decision-making model for autonomy onboard the UAS and a rare event estimation framework for efficiently estimating accident probabilities. An unmanned approach and landing simulation scenario is then constructed to demonstrate the ability of the active risk mitigation methodology to avoid accidents that would otherwise occur if the risk mitigation was passive.

2 Background and Motivation

2.1 Unmanned Aircraft System Automation and Safety

Although military RPAS have been in operation for many years, civilian and commercial RPAS applications are still in their infancy. Safety constitutes a major barrier to routine RPAS operations in the NAS [5]. RPAS can be split into several categories, but are most often divided into two categories: small vehicles with a weight below 55 lbs and larger vehicles above this weight. Safety analyses have been performed for small RPAS [17], and federal regulations defining the acceptable operational domain and safety requirements exist under 14 CFR Part 107. Large RPAS still have no routine certification process and require Special Airworthiness Certificates or Grants of Exemption to operate [23]. Separation violations and mid-air collisions are major hazards, especially in non-segregated airspace with both manned and unmanned aircraft [14]. The latency and possible failure associated with C2 links reduces the effectiveness of risk mitigation procedures [3]. The Required Link Performance (RLP) concept places minimum performance requirements on the link performance with parameters such as link continuity, integrity, availability and latency [34]. Onboard collision risk mitigation may be performed by Detect and Avoid (DAA) automation, which maintains separation without pilot intervention [32]. On top of the mid-air hazards, RPAS must still maintain safety during final approach and landing, the riskiest flight segments in manned aviation. CAT III landing systems, described in the previous section, could potentially provide the required level of safety, however, installation are rare and would limit the use of RPAS for commercial application. The use of computer vision has been proposed to bridge the decision height gap between CAT I approaches and CAT III landings [2]. A variety of landing control system architectures have been proposed to land safely in the face of disturbances [15], however, navigation system errors still present a hazard that must be accounted for. Methodologies for certification of conventional instrument landing systems are suited towards the analogous problem of RPAS final approaches and landings.

2.2 Safety Requirements Formulation

The highest level regulations on safety critical systems for transport category aircraft can be found in FAR §25.1309. These regulations require all failure conditions and their joint combinations which are not extremely improbable to be mitigated to an acceptable level of safety. AC 25.1309 provides a quantitative interpretation of these requirements. Extremely improbable is defined as 1 catastrophic accident per one billion operations or flight hours. These requirements apply to each failure condition and associated mitigation strategies.

Failure condition safety requirements can be formulated using notation based on Bayesian probability. Let S be the power set of all possible discrete failure conditions and θ be the set of system design parameters and operational decision parameters to be considered. μ contains the set of static and prescribed system parameters. The event acc indicates that the system has experienced an accident during the operational risk exposure time.

The failure condition level safety requirement is with respect to a probability corresponding to ‘extremely improbable’, the upper bound on a catastrophic failure condition. This probability is interpreted to be 1×10^{-9} .

$$P(s|\theta, \mu)P(acc|s, \theta, \mu) \leq 1 \times 10^{-9} \quad \forall s \in S \quad (1)$$

An implicit assumption in this formulation is that any uncertainty contributing to $P(acc|s, \theta, \mu)$ is effectively another failure condition considered jointly with s . A methodology for assessing the compliance of a system design with airworthiness safety regulations within this probabilistic framework must address the following:

1. Enumerate possible failure conditions
2. Determine the probability of each failure condition
3. Compute the probability of an accident given each failure condition.

The first task requires a comprehensive understanding of the full system, including all system components, possible failure modes, and contributing weather events that may be considered failure conditions. Methodologies such as fault tree analysis [21] or Bowtie analysis [7] are suited to this task. The second task requires acquiring the prior probability of each failure condition, considering the reliability of system components [26] and empirical data on weather events [28]. The third task, the subject of this work, is concerned with determining the safety of an operation given that a particular failure condition has occurred.

3 Active Risk Mitigation Methodology

Passive risk mitigation may be achieved by an autonomous aircraft functioning with only low-level automation. This automation includes flight control, navigation, state estimation, and procedure path following. Any failure-condition events that occurred during final approach and landing were not actively mitigated, but rather passively mitigated by procedure design, limits on allowable

operating conditions, and limits on failure condition magnitudes. Observations made by the aircraft are disregarded when risk mitigation is purely passive. Air-speed deviations, large navigation error residuals, and loss of thrust can be used to make active risk mitigation decisions and execute a missed approach when safety is threatened. Active risk mitigation is performed by a decision making system onboard the autonomous aircraft following a decision making algorithm. This algorithm can take on various levels of sophistication. Human decision making has been classified according to performance level, an notable example being the Skill, Rule, Knowledge framework [29]. These decision making levels may compared to autonomous systems onboard the aircraft. Skill-based behaviors generally convert sensory input directly into outputs through learned sensory-motor mappings, analogously to low-level feedback control and stabilization. Rule-based behaviors generally perform rudimentary processing on inputs and generate outputs through a stored rule or procedure. This could be compared to a checklist based procedure for aborting a landing, where exceeding thresholds of airspeed, flight path deviations, or other system configurations trigger a missed approach. The highest level of decision making behavior is knowledge-based, where a knowledge of system behavior stored in a model and performance goals are used to solve a problem. Model predictive control frameworks fit this category, as a dynamics model and cost function are used to solve a cost minimization problem to generate optimal flight controls [16]. This knowledge or model-based decision making paradigm is proposed to perform active risk mitigation.

The failure condition safety assessment procedure is used to evaluate the system parameters required to passively mitigate risk in offline analysis. It may also be applied in an online setting, using recent observations to update the internal system model and estimate the accident probability associated with the available procedure options. At all times, a procedure option must be available which satisfies safety requirements. Anticipated loss of safety for the primary landing procedure option triggers the execution of the secondary missed approach procedure option. This active risk mitigation framework utilizing an online safety assessment is outlined and compared to passive risk mitigation in scenarios where passive risk mitigation fails to prevent an accident.

3.1 Agent Based Decision Model

Active risk mitigation during final approach and landing is accomplished using online safety assessments to make decisions that satisfy safety requirements while making progress towards the primary goal of completing the landing. This may be framed generally as a constrained optimization problem that may apply to decision beyond the final approach and landing scenario. A number of procedure options may exist that can be ordered by some measure of optimality. The optimal procedure may accomplish the primary goal, such as landing, or minimize some metric such as flight time, fuel burn, emissions, or noise exposure. This desired procedure option will be termed the primary option. While continuing along the primary option, other alternative procedure options are still

available. These include risk mitigating procedures such as missed approach, collision avoidance, or diversion to an alternate airport. At least one of the available procedure options must satisfy the safety requirements, which act as a constraint on accident probability. We will reduce our situation to a case in which only 1 alternative option is available, termed the secondary option. The decision to execute the secondary option is made when the primary option is unsafe and delaying execution of the secondary option any further will render it unsafe as well. If it is possible to continue along the primary option further and still safely execute the secondary option, continuing the primary option is still deemed safe, even if completing the primary option itself unsafe. In practicality, observations used to update the internal model are made periodically and the online safety analysis takes time to complete. Decisions are made periodically at the end of every update step.

The flowchart in Fig. 1 depicts the active risk mitigation decision making in an Observe-Orient-Decide-Act (OODA) Agent framework [18]. The agent takes in observations from the external system and uses them to update the internal model, based around an Extended Kalman filter framework [37]. Accelerometer, Wide Area Augmentation System (WAAS) GPS pseudoranges [13], and radar altimeter measurements [35] are fused to produce an estimate of position and

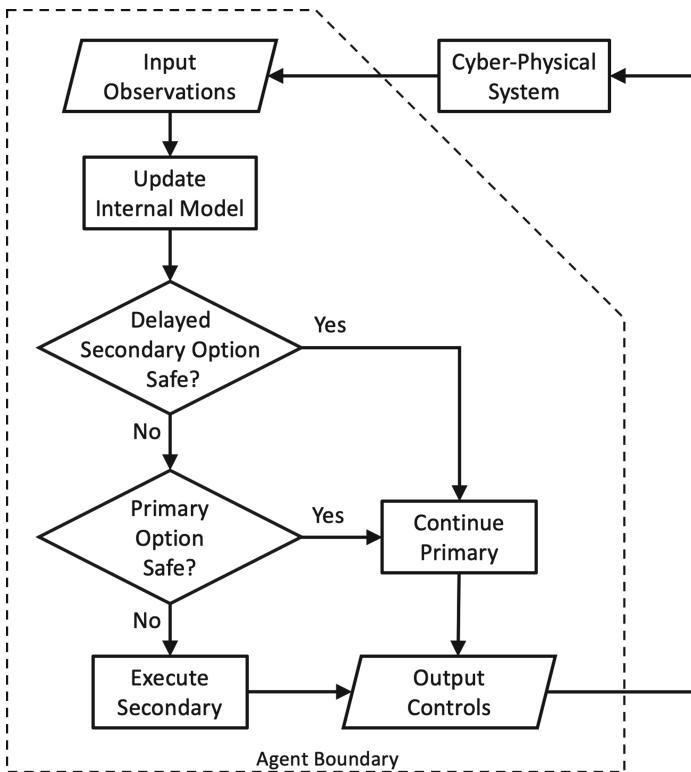


Fig. 1. Active risk mitigation decision flowchart

velocity as a multivariate normal distribution. The periodic decision making occurs after observations have been gathered each update step with time interval ΔT . The online safety assessments occur in the decision blocks and evaluate the accident probabilities for the primary and secondary options. The secondary option safety assessment assumes we continue along the primary option for the next update step and then execute the secondary option at the end of the next safety assessment. A safe result from this secondary option assessment means we may delay the decision to execute the secondary option another update step, an effective delay of $2\Delta T$, without compromising safety. However, at the point where delaying the secondary option is no longer safe and the primary option cannot be shown to be safe, the secondary option must be executed to maintain safety and mitigate risk. The secondary option is delayed as long as possible in the hope that information gathered from observations will be able to rule out an accident if the primary option is continued to completion. When applied to an instrument approach procedure, the cutoff where the secondary option becomes unsafe is directly analogous to the Decision Height, below which a missed approach is unsafe without improved navigation information. Likewise, during take-off rolls, the cutoff is analogous to the V_1 speed, above which the aircraft will be unable to safely come to a stop within the length of runway remaining. Extended-range Twin-engine Operations Performance Standards (ETOPS) allows twin engine aircraft to operate over 180 minutes flight time from a diversion airport. This ETOPS time serves an analogous purpose to the secondary option cutoff, balancing the risk of engine failure with the single engine flight time.

3.2 Rare Event Estimation

The development of a suitable simulation model for a final approach scenario and a method for representing and sampling random input variables allows estimation of the probability of unsafe outcomes. Accident event probabilities may be acquired analytically for sufficiently simple models. However, complex models such as the autonomous UAV agent based model require sampling based Monte Carlo methods to estimate probabilities. Equation 2 gives the basic form of the Monte Carlo estimation problem, where the accident probability ℓ is estimated by sampling multivariate random variable X .

$$\ell = P(d(X) \leq 0) = \mathbb{E}[\mathbf{1}_{\{d(X) \leq 0\}}] \approx \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{\{d(X_i) \leq 0\}} \quad (2)$$

An accident event is given by $\{d(X) \leq 0\}$, meaning that the minimum distance d between the state trajectory and the hazard set is less than or equal to 0. $d(X)$ can be considered as a black-box function mapping the input sample to some output value of interest, though for our particular case we will call it the distance function. Quantifying the error of the Monte Carlo estimate is especially important when dealing with safety critical systems. The ratio of the error standard deviation to the probability estimate, called relative error, is often used to quantify the uncertainty in a Monte Carlo estimate. The relative error of the

estimate is dependent on the sample variance and the number of samples. Lowering the relative error can be achieved by increasing the number of samples, or decreasing the variance of the sample measurements.

$$\text{Relative Error} = \frac{1}{\ell} \left(\frac{\mathbb{E}[(\mathbf{1}_{\{d(X) \leq 0\}} - \ell)^2]}{N - 1} \right)^{\frac{1}{2}} \quad (3)$$

Difficulty arises when a rare event is being estimated, as is the case for FAA AC 25.1309 [10] safety requirements in which accident probabilities must be less than 1×10^{-9} . Thus, rare event estimation using basic Monte Carlo requires on the order of 1×10^9 samples to observe an accident event once and several orders of magnitude more to produce an estimate with low relative error. This in many samples is practically infeasible, even with a fast-time aircraft simulation. Efficient rare event estimation is a topic of interest in many fields, such as physics [8], weather forecasting [36], and air traffic systems [24]. Several techniques have been applied to the problem, including importance splitting [25] and sequential Monte Carlo [4], suited towards stationary dynamical systems. The technique of Importance Sampling [8] is suited for the final approach safety assessment application, where trajectories start from an initial position distribution. Multivariate normal distributions are used to represent the internal belief state distribution and white noise processes contributing to sensor noise and disturbances. Random events' times are represented using the beta distribution, where Beta(1, 1) corresponds to the uniform distribution which may be used to approximate Markov process jump times for transitions that are rare when considered over a risk exposure time of 30 seconds. The Cross-Entropy Importance Sampling method is used to efficiently sample from exponential family distributions [6].

4 Results and Discussion

The system parameters required to passively mitigate risk are used as a baseline for demonstrating the performance gained with active risk mitigation using online safety assessments. The parameters which passively mitigate risk to a level of 1×10^{-9} are used to acquire simulation samples that result in an accident. The rare event estimation methodology is used to search for the simulation trajectory samples with the highest likelihood of causing an accident. This baseline trajectory resulting in an accident is used to demonstrate active risk mitigation using online safety assessment. The risk associated with the delayed missed approach option and continue approach option are computed along the baseline trajectory and a missed approach is triggered by the decision making algorithm when the proper criteria are met. It is expected that when observations suggesting the increased probability of an accident are acquired, a missed approach will be triggered before an accident occurs during the final approach and landing. A crucial parameter in the active risk mitigation methodology is time between safety assessments ΔT . For demonstration, a ΔT of 1.0 second is used. Navigation drift errors due to ionospheric weather [27], wind shear [9], and engine failure conditions are chosen to demonstrate active risk mitigation.

4.1 Arrival and Approach Procedure Scenario

The simulation used to demonstrate the active risk mitigation methodology is initialized prior to the descent and arrival phase of flight and stops before the final 200 ft of the final approach, the point at which safety becomes critical for instrument approach procedures. This initial simulation serves to warm-up the internal states of the guidance, navigation, and control system, and damp out transients in the aircraft dynamics. A Boeing 737-800 at Maximum Landing Weight (MLW) is used as the subject of the safety assessments. Aerodynamic, engine, and control parameters are defined using the parameters verified in the TASAT model [31]. The initial states at the start of the safety assessment are sampled from the belief state distribution produced by the Extended Kalman Filter. AC 120-28D requires safety to be ensured in a defined range of operational conditions, specifically mean head winds of 25 knots, crosswinds of 15 knots, and tail winds of 10 knots [11], with a gust intensity of 10% mean wind at 20 ft. A worst case mean wind with 15 knot crosswind and 10 knot tailwind is used to provide a conservative safety assessment. A runway of standard size at 1000 feet above mean sea level on flat terrain is used for the safety assessment. The runway aimpoint location relative to the runway threshold is used as a variable in the procedure design to set the baseline undershoot risk level. Variable aimpoint and glide slope are features of GPS based procedures as they are not determined by a fixed antenna location. Glide slope, however, is fixed at 3° for the safety assessment. Process noise is sampled using the Karhunen-Loeve (K-L) expansion technique [22] and event times are sampled using Beta distributions.

Although the last 200 ft of the final approach are the primary interest of the safety assessment, a full arrival and approach procedure is simulated beforehand to ‘warm-up’ the internal states of the aircraft simulation. A Continuous Descent Arrival/Approach (CDA) procedure is defined, with a lateral path and vertical profile similar to an Area Navigation (RNAV) procedure. CDA procedures reduce fuel burn, emissions, and noise exposure by reducing thrust requirement with a continuous rate of descent [30][20]. The lateral path is depicted in Fig. 2, featuring a 45° turn onto what would nominally be the downwind approach segment followed by a 180° turn to intercept the final approach. It is worth reiterating that the aircraft is experiencing a tail wind during final approach as opposed to the nominally desired head wind to serve as a worst case scenario.

The altitude profile of the arrival and approach, starting partway through the arrival descent is depicted in Fig. 3. The vertical profile has constant 3° glide slope except for a brief segment of reduced glide slope that serves to help decelerate the aircraft to meet the 250 knot (128 m/s) calibrated air speed limitation imposed by air traffic control below 10000 feet (3048 m). The corresponding calibrated airspeed profile is depicted in Fig. 4, with numerous steps down in airspeed as flaps are deployed and the aircraft stabilizes on the final approach at the reference approach speed.

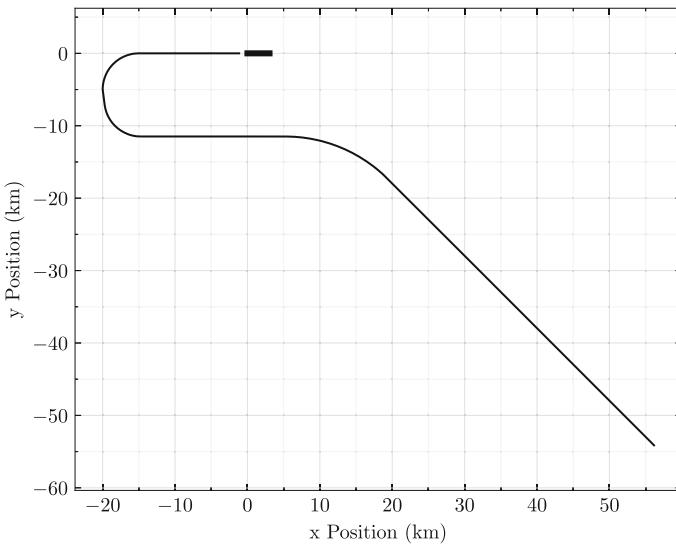


Fig. 2. Arrival and approach lateral path

4.2 Baseline Scenario Parameters

The safety assessment is applied in an offline setting to set the limiting scenario parameters to ensure accident probability remains below 1×10^{-9} . Navigation system configurations considered include pure WAAS GPS, and WAAS GPS augmented by a radar altimeter. Free parameters in the scenario include runway aimpoint relative to the runway threshold, the maximum safe navigation system error gradient, and maximum safe wind shear gradient. The particular navigation

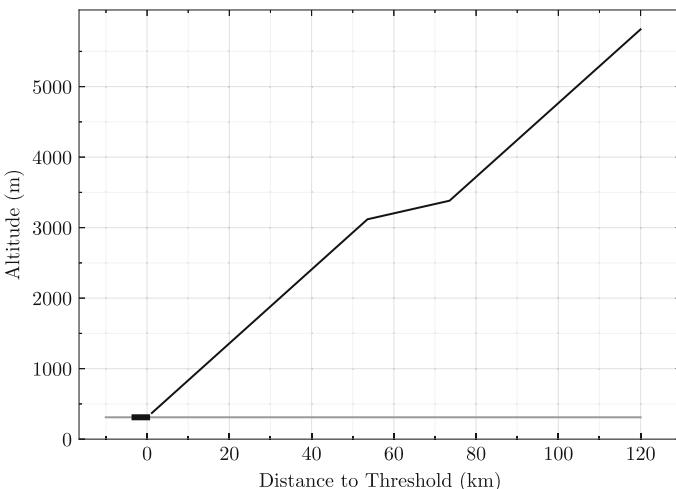


Fig. 3. Arrival and approach altitude profile

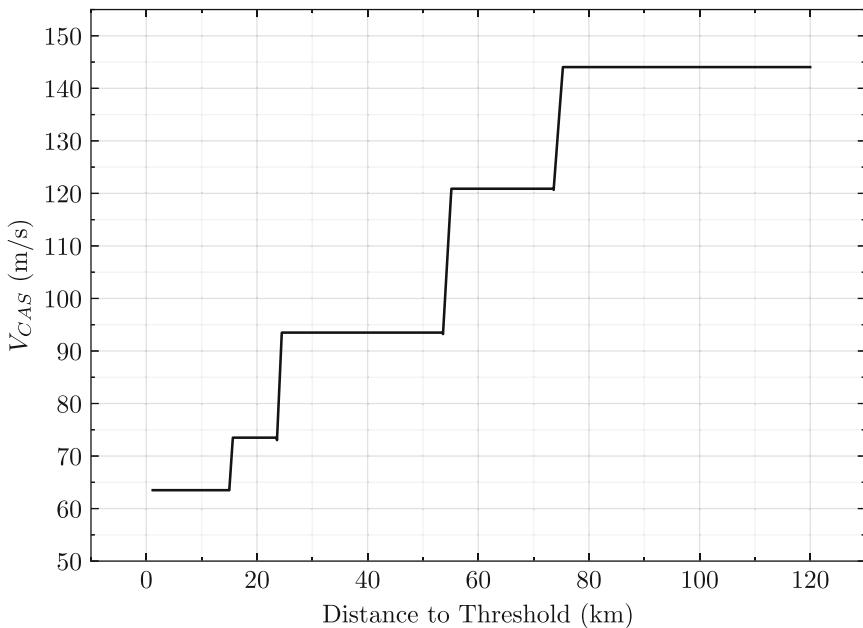


Fig. 4. Arrival and approach calibrated airspeed profile

system configuration and the associated limiting free parameter values providing passive risk mitigation for individual failure conditions are given in Table 1. These limiting parameters are used as the baseline for demonstrating the active risk mitigation methodology.

4.3 Engine Failure Online Safety Assessment

At each update step, we compute the continue approach risk, plotted in black with a 6σ upper bound error bar, and $2\Delta T$ delayed missed approach risk, plotted in blue with a 6σ upper bound error bar. When both of these accident probabilities are greater than 1×10^{-9} , a missed approach is triggered and the immediate missed approach probability is computed and plotted in red. Down arrows indicate an accident probability estimate below 1×10^{-20} . The immediate missed approach is actually executed at the end of the current update step, so it is in reality delayed by ΔT , as opposed to the delayed missed approach which is executed at the end of the next update step delayed by $2\Delta T$. To maintain safety, the immediate missed approach risk must be safe while the other 2 options are both unsafe.

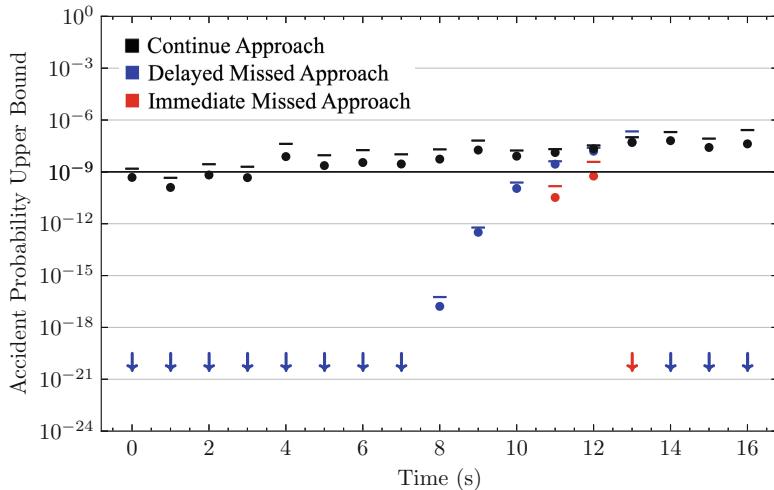
The online safety assessment plots should be read left to right, with attention on which options safety assessment upper bound bars are above or below 1×10^{-9} . When both the black bar and blue bar are above this value, a missed approach should be performed. The immediate missed approach safety assessment given in

Table 1. Passive risk mitigation parameters.

Configuration	Failure condition	Parameter	Parameter limit
WAAS GPS	Nominal	Aimpoint	210 m
	Engine Failure	Aimpoint	240 m
	WAAS GPS Error	Error Gradient	0.002
	Wind shear	Wind shear Gradient	0.002 s^{-1}
Radar Altimeter	Nominal	Aimpoint	75 m
	-Augmented	Error Gradient	0.04
WAAS GPS	Radar Altimeter Error	Error Gradient	0.06
	Wind shear	Wind shear Gradient	0.02 s^{-1}

red serves to show the validity of the delayed missed approach safety assessment in the previous update step. Fast acting failure conditions will result in sudden jumps in accident probability while slower acting failure conditions will result in gradual upward tendencies in accident probability.

The first failure condition considers a critical engine failure during final approach. It is assumed that the engine failure and subsequent loss of thrust can be accurately detected by the engine control system and used in the online safety assessment.

**Fig. 5.** Online safety assessment versus time, $\Delta T = 1.0$ s, engine failure, WAAS GPS, and nominal wind model

Results plotted in Fig. 5 depict the safety assessments over time for the WAAS GPS system with runway aimpoint 240 m and time interval ΔT of 1.0 s. The

missed approach is triggered at 11.0 s and missed approach risk is kept sufficiently low for another update step. The problem of hazardously misleading risk estimates is still apparent near the touchdown due to uncorrected WAAS GPS errors.

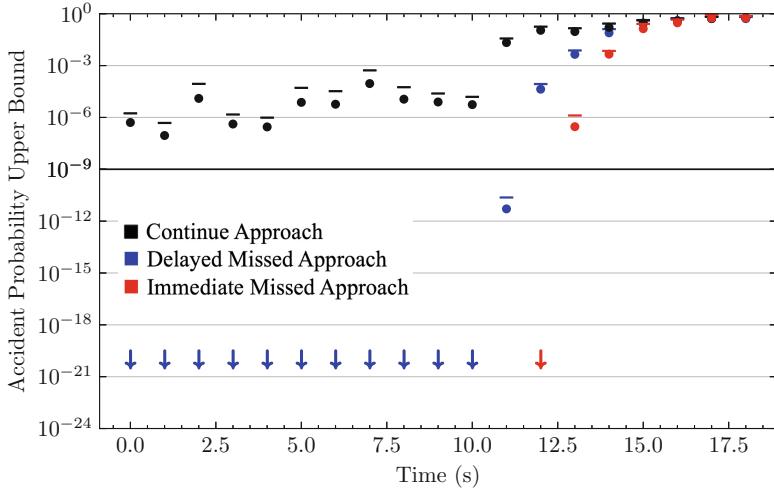


Fig. 6. Online safety assessment versus time, $\Delta T = 1.0$ s, engine failure, radar altimeter augmented WAAS GPS, and nominal wind model

Results plotted in Fig. 6 depict the safety assessments over time for the radar altimeter augmented WAAS GPS system with runway aimpoint 75 m and time interval ΔT of 1.0 s. The missed approach is triggered at 11.0 s and cannot be safely delayed another update step. The detection of a loss of thrust quickly increases the risk estimate and the radar altimeter navigation error corrections produce more consistent risk estimates near touchdown.

4.4 Wind Shear Online Safety Assessment

The second failure condition considers an onset of severe wind shear during final approach. It is assumed that a decrease in airspeed can be detected by the air data system and considered in the safety assessment as an increase in tailwind.

Results plotted in Fig. 7 depict the safety assessments over time for the WAAS GPS system with a wind shear gradient of 0.002 s^{-1} , runway aimpoint 210 m, and time interval ΔT of 1.0 s. The missed approach is triggered at 11.0 s and missed approach risk and cannot be delayed another update step. The problem of hazardously misleading risk estimates is still apparent near the touchdown due to uncorrected WAAS GPS errors.

Results plotted in Fig. 8 depict the safety assessments over time for the radar altimeter augmented WAAS GPS system with a wind shear gradient of 0.002

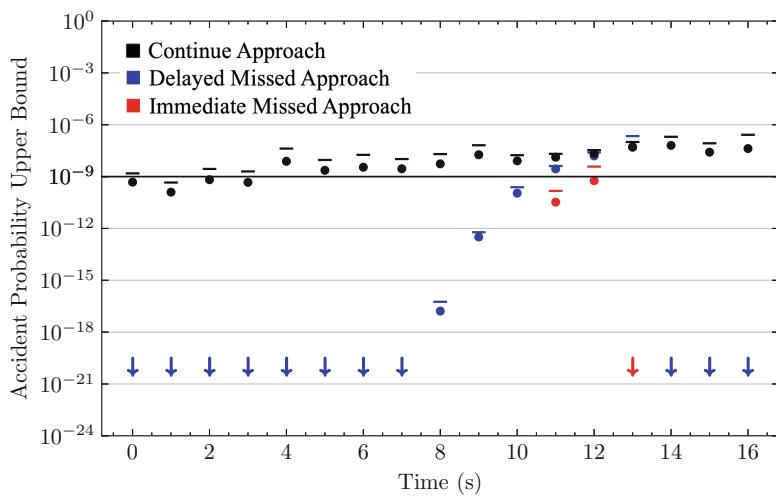


Fig. 7. Online safety assessment versus time, $\Delta T = 1.0$ s, wind shear, WAAS GPS

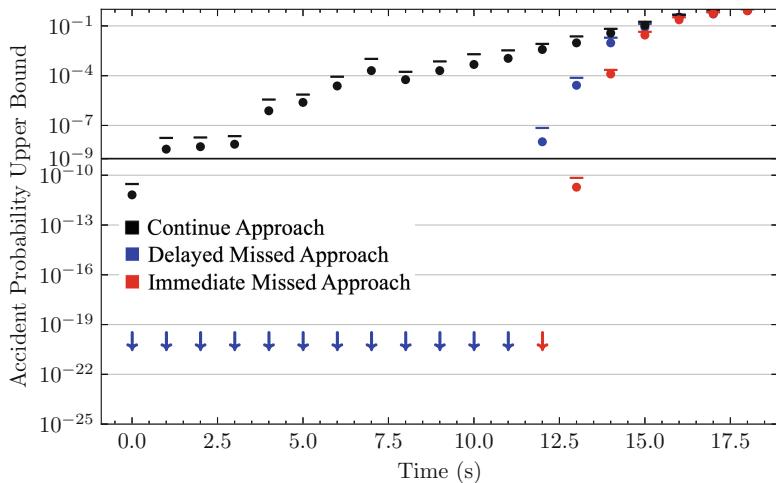


Fig. 8. Online safety assessment versus time, $\Delta T = 1.0$ s, wind shear, radar altimeter augmented WAAS GPS

s^{-1} , runway aimpoint 75 m, and time interval ΔT of 1.0 s. The missed approach is triggered at 12.0 s and may be safely delayed another update step. The detection of wind shear gradually increases the risk estimate and the radar altimeter navigation error corrections produce more consistent risk estimates near touchdown.

4.5 Navigation Drift Error Online Safety Assessment

The third failure condition considers a vertical navigation drift error during final approach. Navigation errors cannot be detected without independent measurements.

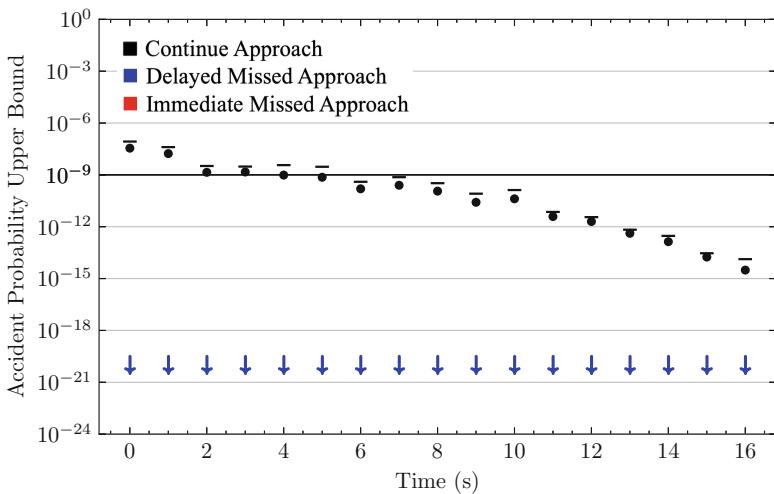


Fig. 9. Online safety assessment versus time, $\Delta T = 1.0$ s, WAAS GPS drift error, WAAS GPS, and nominal wind model

Results plotted in Fig. 9 depict the safety assessments over time for the WAAS GPS system with a vertical drift error gradient of 0.008, runway aimpoint of 210 m, and time interval ΔT of 1.0 s. The risk starts out large due to large risk exposure time before touchdown and decreases as this exposure time decreases. Due to a lack of independent observations to detect or correct navigation errors, the accident precursors go undetected and a missed approach is not executed, resulting in an accident.

Results plotted in Fig. 10 depict the safety assessments over time for the radar altimeter augmented WAAS GPS system with a GPS vertical drift error gradient of 0.008, runway aimpoint of 75 m, and time interval ΔT of 1.0 s. The missed approach is triggered at 14.0 s and cannot be safely delayed another update step. The radar altimeter provides the necessary independent information to mitigate

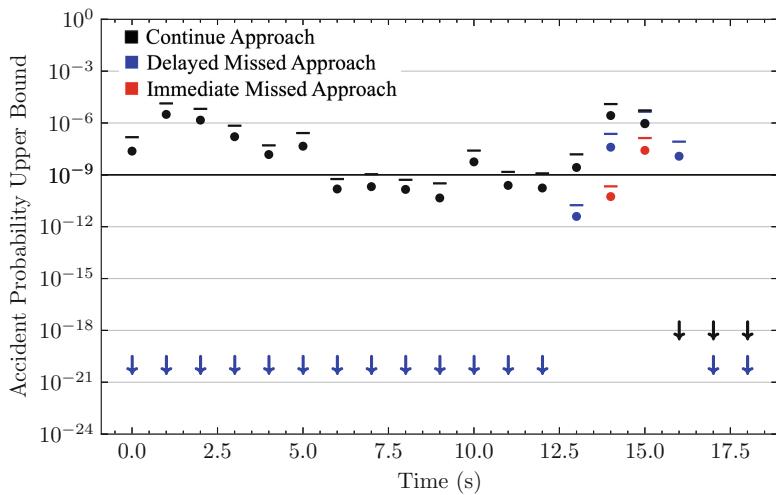


Fig. 10. Online safety assessment versus time, $\Delta T = 1.0$ s, WAAS GPS drift error, radar altimeter augmented WAAS GPS, and nominal wind model

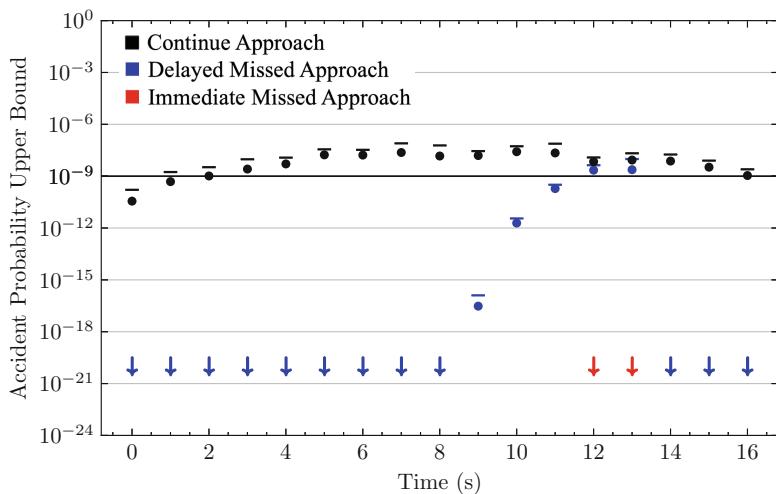


Fig. 11. Online safety assessment versus time, $\Delta T = 1.0$ s, radar altimeter drift error, dual radar altimeter augmented WAAS GPS, and nominal wind model

the risk, but remaining navigation errors still result in hazardously misleading safety assessments close to touchdown.

The possibility of a radar altimeter drift error requires a further source of independent observations more accurate than WAAS GPS to provide a correction. This is accomplished by adding a second independent radar altimeter to the aircraft such that navigation is provided by redundant dual radar altimeter augmented WAAS GPS. Results plotted in Fig. 11 depict the safety assessments over time for the dual radar altimeter augmented WAAS GPS system with a drift error gradient of 0.008 applied to a single radar altimeter, runway aimpoint of 75 m, and time interval ΔT of 1.0 s. The missed approach is triggered at 12.0 s and can be safely delayed another update step. However, remaining navigation errors still result in hazardously misleading safety assessments close to touchdown.

5 Conclusion

While passive risk mitigation can ensure safety when failure condition magnitudes stay within limits, active risk mitigation can detect deviations from safe conditions while operations are underway. A variety of observation inputs such as air data sensors, engine sensors, and redundant navigation systems can be used to update the internal state estimate of the computer agent onboard the unmanned aircraft and perform online safety assessments to estimate the probability of an accident during operation. The comparison of accident probabilities for the available procedure options allows the agent to always maintain an option that satisfies safety requirements.

The period of the update steps, time interval ΔT , has a significant impact on how far along the final approach and aircraft may proceed before risk grows to large. Small ΔT allows decisions to be made much closer to touchdown, however, it requires safety assessments to be completed in less time. The majority of the time consuming computation during rare event estimation is due to running batches of fast-time aircraft simulations. While large batches consisting of several hundred simulations are too slow to run serially in real-time by a factor of 10, the methodology is readily parallelizable. Utilization of basic parallel computing resources could be applied to speedup the batch simulations by the required magnitude, allowing real-time online safety assessments.

The findings relating to the navigation system redundancy needed to actively mitigate risk ties back to the requirements of conventionally piloted ILS procedures. Operations with lower minima require more accurate equipment, redundant systems, and active monitoring to detect and mitigate errors. Augmenting existing navigation systems with independent sources of information is a viable path towards providing the performance required for both zero-visibility conventional landings and autonomous landings of unmanned aerial systems.

This decision making methodology can be extended to the more general case in which a preferred primary option and many alternative options are available, such as choosing an ideal diversion airport. In the case that none of the available options satisfy the safety requirements, the safest of the available options should be chosen instead. This would constitute operations outside of the regular operational domain where emergency decision making criteria are required.

References

1. Boeing Commercial Airplanes: Statistical summary of commercial jet airplane accidents. Worldwide Operations 1959–2019 (2020)
2. Blom, B., Bretschneider, L., Hecker, P.: RPAS automatic take-off and landing operations using computer vision. In: 2017 Integrated Communications, Navigation and Surveillance Conference (ICNS), pp. 5B2–1–5B2–8 (2017)
3. Bulusu, V., Chatterji, G.B., Lauderdale, T.A., Sakakeeny, J., Idris, H.R.: Impact of latency and reliability on separation assurance with remotely piloted aircraft in terminal operations. In: AIAA AVIATION 2022 Forum, p. 3704 (2022)
4. Cérou, F., Del Moral, P., Furon, T., Guyader, A.: Sequential Monte Carlo for rare event estimation. *Stat. Comput.* **22**(3), 795–808 (2012)
5. Couto, D., Delmas, K., Pucel, X.: On the safety assessment of RPAS safety policy (2020)
6. De Boer, P.-T., Kroese, D.P., Mannor, S., Rubinstein, R.Y.: A tutorial on the cross-entropy method. *Ann. Oper. Res.* **134**(1), 19–67 (2005)
7. de Ruijter, A., Guldenmund, F.: The bowtie method: a review. *Saf. Sci.* **88**, 211–218 (2016)
8. Denny, M.: Introduction to importance sampling in rare-event simulations. *Eur. J. Phys.* **22**(4), 403 (2001)
9. Federal Aviation Administration: Windshear training aid (1987)
10. Federal Aviation Administration: Ac 15.1309-1a. advisory circular—system design and analysis (1988)
11. Federal Aviation Administration: Ac 120-28d. advisory circular—criteria for approval of category iii weather minima for takeoff, landing, and rollout (1999)
12. Federal Aviation Administration: Ac 120-29a. advisory circular - criteria for approval of category i and category ii weather minima for approach (2002)
13. Federal Aviation Administration: Global positioning system wide area augmentation system (WAAS) performance standard. U.S. Department of Transportation, Technical report (2008)
14. Ferreira, R.B., Baum, D.M., Neto, E.C.P., Martins, M.R., Almeida, J.R., Cugnasca, P.S., Camargo, J.B.: A risk analysis of unmanned aircraft systems (UAS) integration into non-segregate airspace. In: 2018 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 42–51. IEEE (2018)
15. Gautam, A., Sujit, P.B., Saripalli, S.: A survey of autonomous landing techniques for UAVs. In: 2014 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1210–1218 (2014)
16. Gopinathan, M., Boskovic, J.D., Mehra, R.K., Rago, C.: A multiple model predictive scheme for fault-tolerant flight control design. In: Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171), vol. 2, pp. 1376–1381. IEEE (1998)

17. Guglieri, G., Ristorto, G.: Safety assessment for light remotely piloted aircraft systems (2016)
18. Idris, H.R., Dao, Q., Rorie, R.C., Hashemi, K.: A framework for assessment of autonomy challenges in air traffic management. In: AIAA Aviation 2020 Forum, p. 3248 (2020)
19. Kelly, R.J., Davis, J.M.: Required navigation performance (RNP) for precision approach and landing with GNSS application. *Navigation* **41**(1), 1–30 (1994)
20. Kendall, A.P., Clarke, J.-P.: Stochastic optimization of area navigation noise abatement arrival and approach procedures. *J. Guid., Control Dyn.* **43**(4), 863–869 (2020)
21. Kornecki, A.J., Liu, M.: Fault tree analysis for safety/security verification in aviation software. *Electronics* **2**(1), 41–56 (2013)
22. Liu, Z., Liu, Z., Peng, Y.: Dimension reduction of Karhunen-Loeve expansion for simulation of stochastic processes. *J. Sound Vib.* **408**, 168–189 (2017)
23. Malecha, J., Inspector, A.S.: Drone operations over 55 pounds. In: FAA UAS Symposium (2019)
24. Morio, J., Balesdent, M.: Estimation of Rare Event Probabilities in Complex Aerospace and Other Systems: A Practical Approach. Woodhead Publishing (2015)
25. Morio, J., Pastel, R., Le Gland, F.: An overview of importance splitting for rare event simulation. *Eur. J. Phys.* **31**(5), 1295 (2010)
26. Pecht, M.G., Nash, F.R.: Predicting the reliability of electronic equipment. *Proc. IEEE* **82**(7), 992–1004 (1994)
27. Pullen, S., Cassell, R., Johnson, B., Brenner, M., Weed, D., Cypriano, L., Topland, M., Stakkeland, M., Pervan, B., Harris, M., et al.: Impact of ionospheric anomalies on GBAS GAST D service and validation of ICAO SARPS requirements. In: Proceedings of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2017), pp. 2085–2105 (2017)
28. Pullen, S., Rife, J., Enge, P.: Prior probability model development to support system safety verification in the presence of anomalies. In: Proceedings of IEEE/ION Plans, pp. 1127–1136 (2006)
29. Rasmussen, J.: Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Trans. Syst. Man Cybern.* **3**, 257–266 (1983)
30. Ren, L., Clarke, J.-P.: Separation analysis methodology for designing area navigation arrival procedures. *AIAA J. Guid. Control. Dyn.* **30**(5), 1319–1330 (2007)
31. Ren, L., Clarke, J.-P.B.: Flight-test evaluation of the tool for analysis of separation and throughput. *J. Aircr.* **45**(1), 323–332 (2008)
32. Sabatini, R., Moore, T., Hill, C.: GNSS avionics-based integrity augmentation for RPAS detect-and-avoid applications (2014)
33. Sakakeeny, J., Dimitrova, N., Idris, H.R.: Preliminary characterization of unmanned air cargo routes using current cargo operations survey. In: AIAA Aviation 2022 Forum, pp. 3701 (2022)
34. Serafino, G., Derin, D., Babich, F., Pietrosemoli, E., Goiak, M.: Link performance evaluation procedure for the introduction of unmanned air vehicles in civil airspace. In: 2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace), pp. 182–186. IEEE (2019)
35. Videmsek, A., de Haag, M.U., Bleakley, T.: Radar altimeter aiding of GNSS for precision approach and landing of RPA. In: 2019 Integrated Communications, Navigation and Surveillance Conference (ICNS), pp. 1–16. IEEE (2019)

36. Webber, R.J., Plotkin, D.A., O'Neill, M.E., Abbot, D.S., Weare, J.: Practical rare event sampling for extreme mesoscale weather. *Chaos: Interdiscip. J. Nonlinear Sci.* **29**(5), 053109 (2019)
37. Yadav, N.C., Shanmukha, A., Amruth, B.M., et al.: Development of GPS/INS integration module using Kalman filter. In: 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAM-MAET), pp. 1–5. IEEE (2017)
38. Zolanvari, M., Jain, R., Salman, T.: Potential data link candidates for civilian unmanned aircraft systems: a survey. *IEEE Commun. Surv. Tutor.* **22**(1), 292–319 (2019)



Towards Explainable AI: Relationship Between Twitter Sentiment, User Behaviour, and Bitcoin Price Prediction

Qinan Zhu and Rotimi Ogunsakin^(✉)

Alliance Manchester Business School, Booth St. West, Manchester M15 6PB, UK
rotimi.ogunsakin@manchester.ac.uk

Abstract. Bitcoin prices have been predicted using Twitter sentiments, with results showing relatively low prediction accuracy. Additional external data sources, such as Google Trends, have been used to improve prediction accuracy. However, to the best of our knowledge, no analytical approach has been used to explain why Twitter sentiment is not a good predictor and why additional external data improved predictive model accuracy. Consequently, this paper uses cluster analysis and Shapley Additive Explanations (SHAP) to analyse feature importance and impact on the prediction outcome of the eXtreme Gradient Boosting (XGBoost) model. A combination of Twitter sentiments and user interaction behaviour, such as likes, retweets, and replies, are used as input variables for the XGBoost model and Bitcoin closing prices as the target variable. Our findings indicate that the sentiment score is insufficient because the majority of Bitcoin-related tweets come from Bitcoin enthusiasts whose opinions are unaffected by market fluctuations, and the improved prediction accuracy observed when external data are used in addition to the sentiment score is significant only during price volatility and can be attributed to an increase in the total number of interactions from new sets of users and not the cumulative user behaviour.

Keywords: Shapley Additive Explanations · Explainable AI · Sentiment Analysis · Bitcoin

1 Introduction

The cryptocurrency was introduced by Satoshi Nakamoto in 2008 to serve as a potential alternative to central bank-issued currency [22]. However, the popularity and public recognition of cryptocurrency have been largely due to the potential returns generated by the investment activity associated with them, where a \$1 investment in Bitcoin made at the beginning of 2010 potentially worth approximately \$64,800 on April 14, 2021 [25].

Despite the fact that the volatility of cryptocurrency prices makes it difficult to predict price movement, there is significant academic interest in the subject. Possibly due to the belief that it is possible to predict the Bitcoin price by employing techniques

similar to those used for stock price prediction and the perceived relationship that exist between human sentiment and Bitcoin price [14].

Twitter's user interaction behaviour, on the other hand, defines user behaviour patterns while interacting with tweets using the interface interaction functions defined by Twitter, which are the likes, retweet, and reply buttons [8]. The user interaction pattern captures users' perception, emotion, and commitment to Bitcoin-related tweets, which to a large extent, indirectly impact Bitcoin price movement [19].

Consequently, this research aims to forecast the daily closing price of Bitcoin using Twitter sentiment and user interaction behaviour, such as retweets, replies and likes [2,9], and provide an explanation for the model prediction, including the impact of the features on prediction outcome. To achieve this aim, Bitcoin-related Twitter posts from the past three years are gathered for sentiment analysis, including the user interaction behaviour, which is the number of retweets, replies, and likes for each post. The lexicon-based method known as VADER (Valence Aware Dictionary and Sentiment Reasoner) [10] is used to assign sentiment polarity scores to tweets, while XGBoost is used for model development [4].

Using the Shapley Additive Explanations (SHAP) [17] to analyse feature importance and impact on prediction outcomes shows that sentiment score and user interaction behaviour play a lesser role in predicting the Bitcoin price during periods of price volatility and vice versa. We conclude that the sentiment score is insufficient because the majority of Bitcoin-related tweets come from Bitcoin lovers whose opinions are unaffected by price volatility, while the improved prediction accuracy observed when external data are used in combination with the sentiment score is mainly due to an increase in the total number of interactions from new sets of users and not the cumulative user behaviour captured through the user interaction functions. Therefore, we posit that sentiments and user behaviour on social media are not good predictors of Bitcoin price.

The remainder of the paper is organised as follows. Section 2 contains a review of research in blockchain, bitcoin and bitcoin prediction. Section 3 presents the research methodology, which includes the data collection and analysis methods, and analysis and modelling of user behaviour and bitcoin prices. The result and interpretation of the model result are presented in Sect. 4. While the discussion and conclusion are presented in Sects. 5 and 6, respectively.

2 Literature Review

Satoshi Nakamoto introduced Bitcoin in 2008, and it's primarily regarded as a revolutionary digital currency that solves the problems associated with constructing a secure and robust digital currency system [22]. The high level of Bitcoin price speculation and the absence of intrinsic value makes it difficult to predict the Bitcoin price [27]. Also, the high volume of speculative transactions on the Bitcoin market adds volatility to the price, making it difficult for investors to predict the price movement with accuracy [29].

It is believed that emotions influence financial activities and decisions in addition to monetary value. Therefore, numerous studies have been conducted to investigate the viability of using human sentiment to forecast financial market activities and fluctuations [14]. New research has focused on the collective public mood on Twitter as tweet sentiment has been shown to reflect public opinion on a broad range of subjects and correlate with real-world sentiments [5, 24].

A relatively large amount of research has been conducted to forecast the Bitcoin price using multiple predictive and machine-learning models. Various sentiment extraction and modelling techniques, ranging from multiple linear regression to advanced machine learning, have been implemented for cryptocurrency price prediction. For example, using multiple linear regression models, [12] predicted the 2-hour Bitcoin price. The predictive power of the ARIMA model was tested in predicting the Bitcoin price, and it was discovered that ARIMA was less accurate than the deep learning model [20]. While [13] uses a Bayesian neural network (BNN) and other deep learning algorithms to forecast Bitcoin price and concluded that BNN described Bitcoin's log price and volatility more accurately than other methods.

Tweet signals have been shown to improve Bitcoin exchange rate prediction, suggesting tweet sentiment data may be utilised in developing Bitcoin trading strategies [6]. In addition, Google Trends has been shown to reflect the real-time public interest in a topic, and a reliable indicator of the Bitcoin price movement [18, 19]. This discovery supports the opinion that polarity and emotional valence adequately explain Bitcoin price fluctuations [7].

The Extreme Gradient Boosting Regression Tree Model (XGBoost) shows a significant result and comparatively accurate prediction compared to other models when used to predict the prices of cryptocurrencies [16]. A real-time platform has also been proposed for predicting the price of cryptocurrencies using Twitter sentiment, and Bitcoin historical prices [21]. Finally, a Naïve Bayes (Bernoulli and Multinomial) and a logistic regression model comparison for predicting Bitcoin price have been performed by [3].

Despite the extensive research to validate the predictive power of Twitter sentiment, some studies have cast doubt on the applicability of such methods under different circumstances and the causal relationship between sentiment and Bitcoin price. For example, [15] conducted a Granger causality analysis and discovered no evidence of an emotional tweet's causal effect on the Bitcoin market, and suggested that sentiment more frequently reflects the market than anticipates it. The author in [1] discovered that regardless of potential price fluctuations, Twitter users are generally positive about cryptocurrencies. The study [28] tested the relationship between sentiment and Bitcoin price using a multiple linear regression model, and the model fails the significance test, leading to the conclusion that sentiment and Bitcoin price have no statistically significant relationship.

This study extracts data from the most influential Twitter users as proposed by [23] and explores new methods for measuring the chain effect caused by user behaviour on Twitter. In addition, user interaction behaviour of the subsequent

impact of original tweets, such as the number of likes, retweets, and replies, are collected and used for model training and prediction.

3 Methodology

In this research, the sentiments and the user interaction behaviour serve as the input variable to the XGBoost models, while the daily close price of Bitcoin serves as the target variable. An overview of the research methodology is presented in Fig. 1

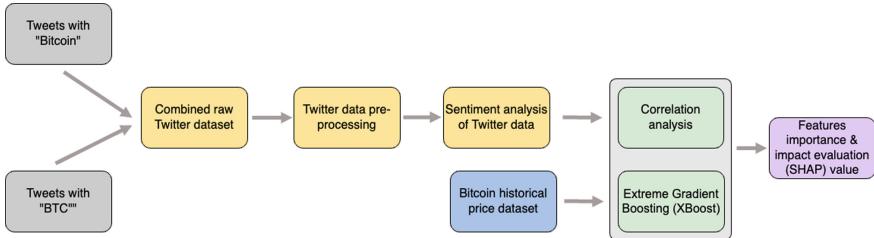


Fig. 1. Overview of research methodology.

3.1 Data Collection

The open-source Python library, *snscreape*, is used to scrape tweets from Twitter. The data collection method introduced by [23] is adopted where only tweets posted by the top 35 influential Twitter users are collected for the study. These 35 Twitter accounts are chosen based on the number of followers and inclusion on Bitcoin.com [26]. These 35 accounts' tweets containing the keywords “Bitcoin” and “BTC” from July 31 2018, to July 29 2021 are collected and analysed. The daily closing Bitcoin prices in US dollars are sourced from *Bitstamp*, one of the most well-known Bitcoin trading platforms in the world. The historical Bitcoin prices from July 31, 2018, to July 29, 2021, are available to the public and were collected.

The initial CSV dataset consists of 37,339 tweets with their respective text contents and numbers of likes, retweets, and replies. The following procedures are used to clean the data: 1. Remove duplicate tweets, 2. Determine missing values, 3. Remove unwanted words and characters, 4. Convert case

3.2 VADER and Sentiment Analysis

The sentiment score is extracted from tweets using a lexicon-based method [10]. The Valence Aware Dictionary for Sentiment Reasoning (VADER) is chosen for Twitter sentiment analysis [10]. VADER takes punctuation and semantics into

account as it assigns a higher score to exclamation marks, accounts for changes in sentiment after conjunctions such as “but,” and, most importantly, it updates its lexicon to include the most recent emojis, abbreviations, and slang such as “lol,” “lmao,” and “:P,” which indicate extremely strong emotions. It also identifies hidden emotions in contractions like “haven’t” and “don’t.” Due to VADER’s capability to interpret punctuation, uncommon expressions, abbreviations, and stop words, they are not removed in this study, as they improve the accuracy of the sentiment score rather than adding noise. Python’s *nltk* library’s *SentimentIntensityAnalyzer()* is used to implement the VADER.

3.3 Correlation Between Sentiment Score and Bitcoin Price

Prior to engaging in modelling, we investigate the approximate relationship between sentiment score and Bitcoin price (see Figs. 2 and 3) as a line graph and scatter plot, respectively. Normalising the Bitcoin price improves the visual representation of the relationship between the Bitcoin price and the sentiment score. The sentiment score does not show any discernible trend over the past three years; thus, the sentiment score’s sensitivity to Bitcoin prices is relatively low (Fig. 2). The correlation between the Bitcoin price and sentiment score is 0.025. This low correlation is evident in the line graph illustrating the evolution of the daily average sentiment score and the Bitcoin price over the past three years. In addition, Fig. 2 demonstrates that the daily sentiment score averaged around 0.2, indicating that Twitter users tend to post more positive tweets about Bitcoin, consistent with the findings of [1].

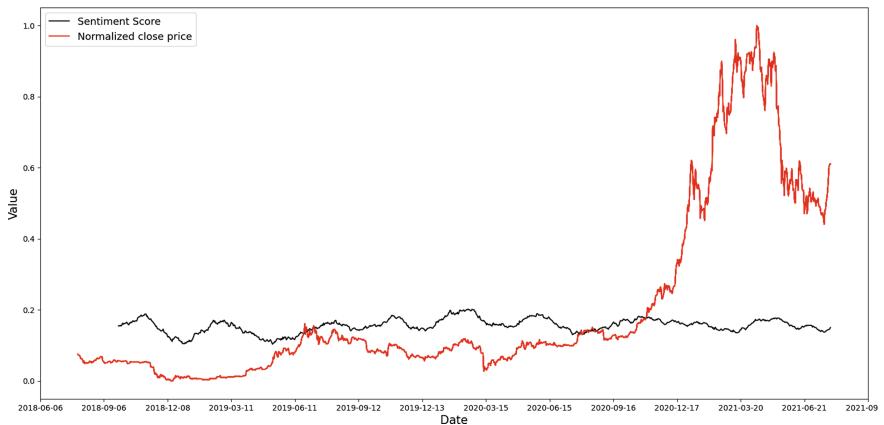


Fig. 2. Line graph for sentiment score and Bitcoin price.

A scatter plot was created to visually observe the relationship in greater detail. There is no linear relationship between sentiment and Bitcoin price as

observations with a daily sentiment score between 0 and 0.3 correspond to Bitcoin prices ranging from 5,000 to 60,000 USD (see Fig. 3). Thus, it appears Bitcoin's price is unlikely to be explained solely by sentiment score, consistent with the findings of [28]. To increase the prediction accuracy of the Bitcoin price, other features such as the user interaction behaviour are therefore strongly needed as input features.

In addition, Fig. 3 shows three distinct clusters with different volatility levels—stable price clusters (1), Tesla announcement clusters (2), and volatile price clusters (3). To investigate volatility's impact on prediction accuracy, we use the different cluster datasets as input variables to the XGBoost model; after using all the datasets.

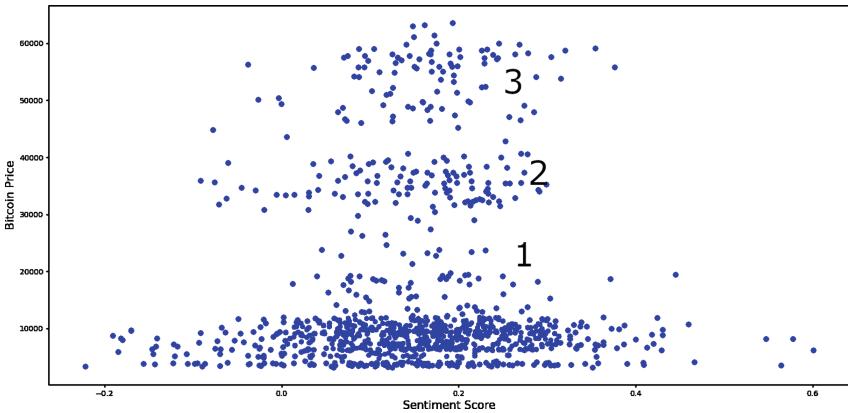


Fig. 3. Scatter plot for sentiment score and Bitcoin price.

3.4 User Interaction Behaviour and Bitcoin Price

A normalised time-series graph showing the relationship between Bitcoin price and user interaction behaviour is presented in Fig. 4a–f. An upward movement in the number of replies, retweets, and counts during the Bitcoin price's rapid increase at the end of 2020 is observed (see Fig. 4c–f), while the sentiment score remained relatively stable (see Fig. 4b). In addition, the correlation between the user interaction behaviour and the Bitcoin price shows a relatively moderate to high correlation (Table 1). Indicating a relationship between the user-interactive features and the Bitcoin price, and suggesting that the user interaction behaviour may have a direct influence on Bitcoin's price.

Therefore, we build a predictive model based on the entire dataset and on the individual clusters to investigate the impact of the different cluster characteristics on the accuracy of the predictive models. The names of the three clusters are derived from either the widespread acclaim or the significant events that occurred during the corresponding time period (see Table 2).

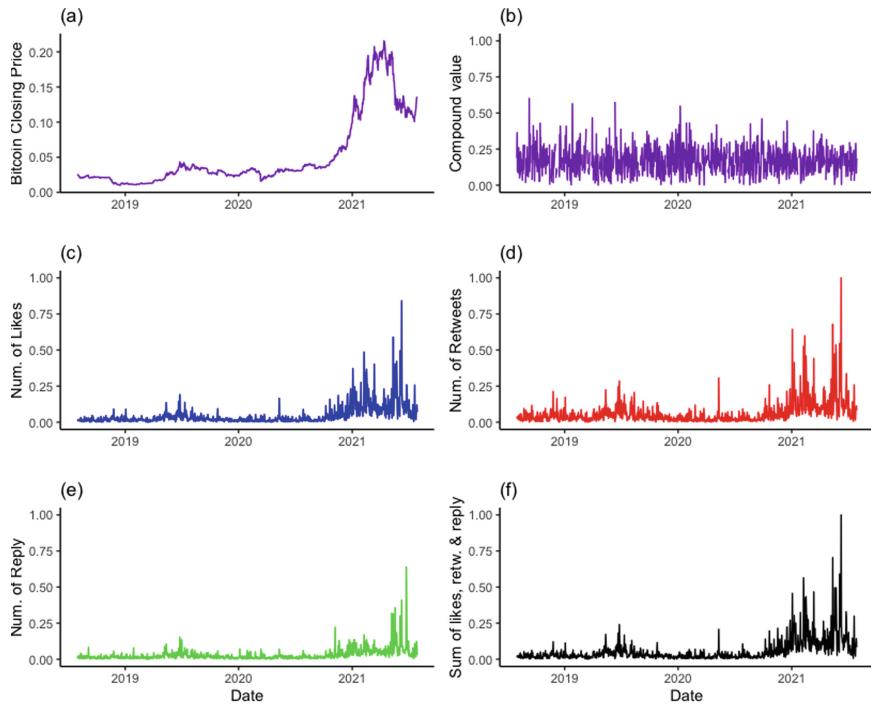


Fig. 4. Comparing Bitcoin price, sentiment score and user interaction behaviour.

Table 1. Correlation between Bitcoin price, sentiment score, and user interaction behaviour.

	Close price	Compound	Like count	Reply count	Retweet count
Close price	1				
Compound	0.025	1			
Like count	0.58	0.01	1		
Reply count	0.35	0.022	0.67	1	
Retweet count	0.34	0.014	0.88	0.6	1

3.5 Modelling

Model Selection Extreme Gradient Boost (XGBoost) is utilised to forecast the Bitcoin price. The XGBoost is one of the most powerful machine learning algorithms with high speed and performance, and it is an ensembled decision tree algorithm that constructs a strong classifier from a series of weak classifiers, thus able to deal with the bias-variance trade-off [4].

ModelBuilding The *xgboost* and *sklearn* packages are used to execute the XGBoost model in Python, while the accuracy of predictions is measured using the Root Mean Square Error (RMSE). A set of parameters are determined beforehand, where

Table 2. Period for each cluster.

Name	Stable	Volatile	Tesla announcement
Period	31/07/2018-31/12/2020	01/01/2021-20/05/2021	21/05/2021-29/07/2021
Cluster ID	Cluster 1	Cluster 3	Cluster 2

parameters with minimal impact on the model’s predictions output retain their default values, while other parameters, such as *eta*, *max_depth*, *subsample*, and *colsample_bytree* are assigned reasonable values within the suggested range. The *eta* represents the number of steps required to arrive at the optimal prediction. The *max_depth* parameter prevents over-fitting, which indicates that a greater depth will result in extremely specific relationships with a given sample. The *subsample* represents the fraction of observations to be sampled at random for each tree, and the *colsample_bytree* represents the fraction of columns to be sampled at random for each tree (Jain, 2016). Each value is presented in Table 3.

Table 3. Value of XGBoost parameters.

Types of parameter	Name of parameter	Value
General	booster	default
	Silent	default
	nthread	default
	num_pbuffer	default
	num_feature	default
Tree Booster	eta	0.1
	gamma	default
	max_depth	12
	min_child_weight	default
	max_delta_step	default
	subsample	0.7
Task	colsample_bytree	0.7
	objective	reg:linear
	base_score	default
	eval_metric	default

4 Empirical Results

4.1 Modelling the Entire Dataset

This section presents the outcomes of the XGBoost models utilising the entire dataset. The entire dataset is divided based on the timeline, with the training

dataset set containing data from 31 July 2018 to 31 January 2021 and the test dataset set containing data from 1 February 2021 to 29 July 2021.

XGBoost Using the XGBoost, the final RMSE is 35032.36, which is comparatively high compared to the Bitcoin price moving average. The Shapley Additive Explanations (SHAP) summary plot is used to explain the inner workings of the XGBoost model (see Fig. 5). The SHARP summary plot illustrates the ranking of feature importance and whether each feature has a negative or positive effect on the predicted Bitcoin price. The top variables contribute more to the output of the model than the bottom variables, therefore having greater predictive power.

In the SHARP summary plot, the SHAP value is predominantly negative when the number of likes is low, indicating a low Bitcoin price output. The SHAP value increases significantly when the number of likes increases, indicating an increase in the Bitcoin price output. This suggests that the number of likes has a positive relationship with the Bitcoin price output. While the number of retweets and replies has a slightly negative effect on Bitcoin's output, suggesting that fewer retweets and replies result in a slightly higher Bitcoin output price.

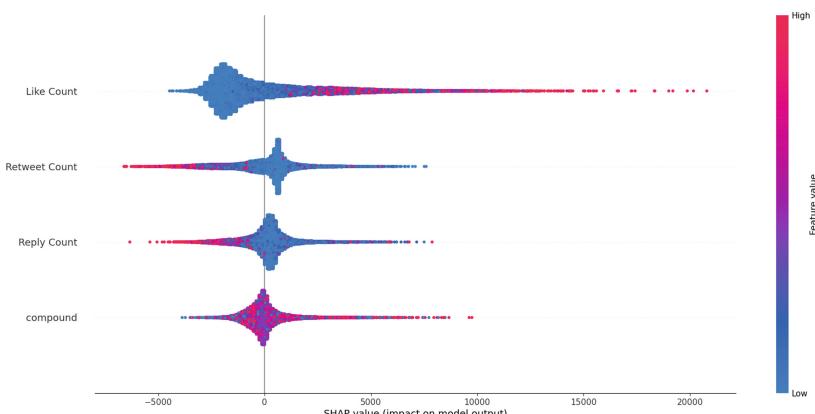


Fig. 5. SHAP value summary plot for entire dataset.

4.2 Modelling on Individual Clusters

We investigate the model performance on each individual cluster. To achieve this, the dataset is partitioned based on the timeline with a ratio of 5:1 for training and testing datasets, respectively. This ratio is repeated for the remaining clusters.

Stable Cluster Using the XGBoost, the final RMSE is 9678.98, a 72% increase from the previous RMSE, which can be attributed to the similarity in the data points from the same cluster. Figure 6 illustrates the significance of the features based on the SHAP value from XGBoost. The SHAP value is more likely to

be negative when the number of likes is low, suggesting that a low number of likes indicates a low Bitcoin price prediction. In contrast, the SHAP value increases relative to the number of likes, indicating that a larger number of likes corresponds to a higher value in the Bitcoin price prediction.

In contrast, the number of retweets has the opposite effect on the model output; fewer retweets result in a higher predicted price. While the majority of points for the number of replies are located near the zero baselines, indicating a weak contribution to the model prediction. In addition, the influence of sentiment score, indicated as the compound score, is relatively small, as the same sentiment score achieves varying SHAP values, indicating that sentiment score is a poor predictor. The aforementioned findings imply that during price stability, the total number of likes positively impacts the Bitcoin prices, while the number of retweets indicates a negative impact. During price stability, most contributors to Bitcoin tweets are Bitcoin enthusiasts, hence exhibiting more “like” interactions as it is the less demanding form of interaction.

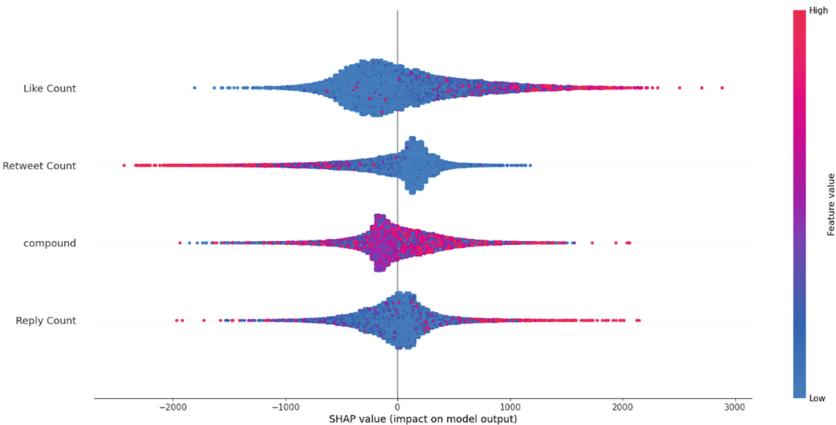


Fig. 6. SHAP summary plot for stable cluster.

Volatile Cluster Using the XGBoost, the final RMSE is 8357.14, which is a 13% increase from the stable cluster’s RMSE. The output of XGBoost’s feature importance rank for the volatile cluster is opposite that of the stable cluster, as the average effect of the like count on the Bitcoin price is mostly neutral. In general, the effect revealed by each individual observation appears to be more centralised than in the stable cluster, with the majority of values located near the zero baselines and smaller values at both the positive and negative SHAP value ranges, with the exception of the compound values (Fig. 7). In terms of their contribution to price prediction, the number of likes, retweets, and replies in the volatile cluster plays a lesser role compared to the stable cluster.

A different finding in this cluster is that the reply count outperforms sentiment and retweet count in terms of the feature’s importance and has a more

negative impact on the Bitcoin price. This implies many users are more ready to comment on Bitcoin-related tweets when the prices go down during a volatile period. Additionally, the sentiment score has a greater influence on Bitcoin's price prediction in the red cluster, with a predominantly negative impact on Bitcoin's price, suggesting that a high sentiment score indicates a low Bitcoin price prediction, meaning more sentiment is likely to be observed when Bitcoin prices are low compared to when Bitcoin prices are high.

Thus from the output of the SHAP plot, the “like” behaviour observed in the stable cluster is neutralised, and the sentiment impact is negative. It implies that the increase in the number of interactions, as shown in Fig. 4, appears to be from a new set of users as less sentiment and relatively neutral “like” behaviour count impact is observed.

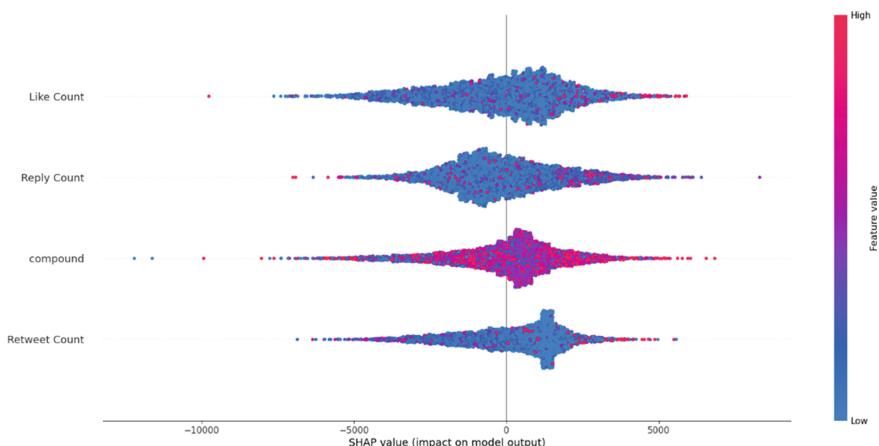


Fig. 7. SHAP summary plot for volatile cluster.

Tesla Announcement Cluster Using the XGBoost, the final RMSE is 1927.14, which is a 77% increase from the volatile cluster's RMSE. The SHAP summary plot presented in Fig. 8 reveal that the number of likes is no longer the most important feature; rather, the number of retweets is the most important and has a neutral impact on the Bitcoin price. All other characteristics also have a relatively neutral effect on the Bitcoin price. The sentiment score, meanwhile, becomes the second most important feature, indicating more sentiments are observable during Bitcoin's price volatility, as indicated in the volatile cluster.

In summary, there is a dominant increase in prediction accuracy in correlation to price volatility, while at the same time, the relative impact of sentiment score and user interaction behaviour decreases. This implies that the sentiment and user interaction behaviour contribute less to the prediction accuracy during a volatile price period.

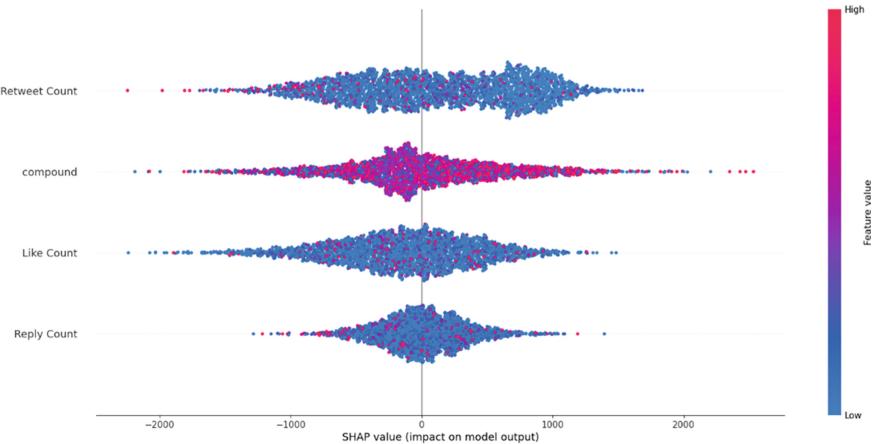


Fig. 8. SHAP summary plot for red-hot cluster.

5 Discussion

The results provide valuable insights into Bitcoin price prediction using the Twitter sentiment score. It has been demonstrated that the predictive power of sentiment is insufficient to forecast the Bitcoin price as a whole. However, there is a positive correlation between the Bitcoin price and user interaction behaviour, namely the number of retweets, replies, and likes. Consequently, a combination of the sentiment score and the user interaction behaviour as inputs to the XGBoost significantly improves the accuracy of the prediction, at the same time, the impact of the features on model output decreases.

Comparing the results from each of the three clusters reveals that when there is a fluctuation or movement trend, the user interaction behaviour and sentiment score play a lesser role in predicting the Bitcoin price. This observation is supported by the relatively neutral impact of the features in the XGBoost model. Therefore, the increased prediction accuracy for the volatile and Tesla announcement clusters is not solely attributable to the additional features, but rather to the models' overall predictive ability and the increase in user interaction. However, when the Bitcoin price is relatively stable, the improved accuracy is primarily due to the features and the corresponding interaction behaviour, which have both a strong positive and negative impact on the models' outcomes.

The model analysis indicates that the impact of the sentiment score on the predicted Bitcoin price is relatively low, regardless of the price movement trend. This indicates that the majority of Bitcoin-related tweets originate from Bitcoin enthusiasts whose opinions are unaffected by market fluctuations. The relatively neutral to positive sentiment scores also support this conclusion. The results support [11] assertion that Twitter sentiment alone is insufficient for predicting the Bitcoin price and that additional factors are strongly recommended to improve the accuracy of price forecasts.

Consequently, the output of our research suggests that during Bitcoin's volatile price movement, more people (mostly non-Bitcoin enthusiasts) tend to react to Bitcoin tweets or search for Bitcoin-related tweets on Twitter, similar to a Google search, thus bringing neutrality to the behavioural trends observed during the stable price period.

6 Conclusion

In conclusion, this paper predicts the closing price of Bitcoin using Twitter sentiments and user interaction behaviour and provides an explanation for the model's output. Our research indicates that Twitter sentiment is insufficient for predicting the Bitcoin price, but using both sentiment score and user interaction behaviour improves prediction accuracy, mostly during price volatility, while the impact of the feature on model output decreases at the same time.

We conclude that the sentiment score is insufficient because the majority of Bitcoin-related tweets come from Bitcoin enthusiasts whose opinions are unaffected by market fluctuations. We also conclude that the improved prediction accuracy observed during Bitcoin's price volatility is solely attributable to an increase in the total number of interactions from new sets of users and not the direction of interaction observable during the stable price period. Thus, the increase in non-Bitcoin enthusiast user interaction during periods of price volatility neutralises the sentiment and directional user interaction trends observed during periods of stable price movement.

References

1. Abraham, J., Higdon, D., Nelson, J., Ibarra, J.: Cryptocurrency price prediction using tweet volumes and sentiment analysis. *SMU Data Sci. Rev.* **1**(3) (2018). <https://scholar.smu.edu/datasciencereview/vol1/iss3/1>. Accessed 13 Sept 2021
2. Benevenuto, F., Rodrigues, T., Cha, M., Almeida, V.: Characterizing user behavior in online social networks. In: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement, pp. 49–62 (2009)
3. Colianni, S., Rosales, S., Signorotti, M.: Algorithmic trading of cryptocurrency based on twitter sentiment analysis (2015)
4. 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
5. Contreras, J., Espinola, R., Nogales, F., Conejo, A.: ARIMA models to predict next-day electricity prices. *IEEE Trans. Power Syst.* **18**(3), 1014–1020 (2003)
6. Galeshchuk, S., Vasylchyshyn, O., Krysovatty, A.: Bitcoin response to Twitter sentiments. In: ICTERI Workshops (2018)
7. Garcia, D., Schweitzer, F.: Social signals and algorithmic trading of Bitcoin. *R. Soc. Open Sci.* **2**(9) (2015)
8. Guerrero-Solé, F.: Interactive behavior in political discussions on Twitter: politicians, media, and citizens' patterns of interaction in the 2015 and 2016 electoral campaigns in Spain. *Soc. Media+ Soc.* **4**(4) 2056305118808776 (2018)

9. Gotz, D., Zhou, M.X.: An empirical study of user interaction behavior during visual analysis. In: IBM Research RC24525 (W0803-127) (2008)
10. Hutto, C., Gilbert, E.: Vader: a parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 8, No. 1, pp. 216–225 (2014)
11. Jain, A.: Complete guide to parameter tuning in XGBoost with codes in python (2016). <https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/>. Accessed 13 Sept 2021
12. Jain, A., Tripathi, S., Dwivedi, H., Saxena, P.: Forecasting price of cryptocurrencies using tweets sentiment analysis. In: Proceedings of 2018 Eleventh International Conference on Contemporary Computing (IC3), Noida, India (2018)
13. Jang, H., Lee, J.: An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. IEEE **6**, 5427–5437 (2017)
14. Kahneman, D., Tversky, A.: Prospect theory: an analysis of decision under risk. *Econometrica* **47**(2), 263–292 (1979)
15. Kaminski, J.: Nowcasting the bitcoin market with twitter signals. MIT Media Lab (2014) <http://arxiv.org/abs/1406.7577>. Accessed 13 Sept 2021
16. Li, T., Chamrajnagar, A., Fong, X., Rizik, N., Fu, F.: Sentiment-based prediction of alternative cryptocurrency price fluctuations using gradient boosting tree model. *Front. Phys.* **7**(98) (2019)
17. Linardatos, P., Papastefanopoulos, V., Kotsiantis, S.: Explainable ai: a review of machine learning interpretability methods. *Entropy* **23**(1), 18 (2020)
18. Madan, I., Saluja, S., Zhao, A.: Automated bitcoin trading via machine learning algorithms (2015). <http://cs229.stanford.edu/proj2014/Isaac%20Madan,20>. Accessed 12 Jan 2023
19. Matta, M., Lunesu, M.I., Marchesi, M.: Bitcoin spread prediction using social and web search media. In: UMAP Workshops (2015)
20. McNally, S., Roche, J., Caton, S.: Predicting the price of bitcoin using machine learning. In: 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), pp. 339–343 (2018)
21. Mohapatra, S., Ahmed, N., Alencar, P.: KryptoOracle: a real-time cryptocurrency price prediction platform using Twitter sentiments. In: IEEE Big Data (2019)
22. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system. *Decentralized Bus. Rev.* 21260 (2008)
23. Öztürk, S., Bilgiç, M.: Twitter & bitcoin: are the most influential accounts really influential? *Appl. Econ. Lett.* (2021). <https://doi.org/10.1080/13504851.2021.1904104>. Accessed 13 Sept 2021
24. Panger, G.T.: Emotion in Social Media. University of California, Berkeley, Ph.D (2017)
25. Redman, J.: Bitcoin gained 8.9 million percent over the last decade—featured bitcoin news (2019). <https://news.bitcoin.com/bitcoin-gained-8-9-million-percent-over-the-last-decade/>. Accessed 13 Sept 2021
26. Redman, J.: The 35 most influential bitcoiners dominating crypto Twitter by follower count—featured bitcoin news (2021). <https://news.bitcoin.com/35-most-influential-bitcoin-crypto-twitter/>. Accessed 13 Sept 2021
27. Silverman, G., Murphy, H., Authers, J.: Bitcoin: an investment mania for the fake news era, [online] financial times (2017). <https://www.ft.com/content/c84caffc-d683-11e7-a303-9060cb1e5f44>. Accessed 13 Sept 2021

28. Verma, M., Sharma, P.: Money often costs too much: a study to investigate the effect of Twitter sentiment on bitcoin price fluctuation (2020). <https://doi.org/10.20944/preprints202009.0216.v1>. Accessed 13 Sept 2021
29. Yermack, D.: Is bitcoin a real currency? An Economic Appraisal. NBER Working Paper, p. 19747 (2013). <http://www.nber.org/papers/w19747>. Accessed 13 Sept 2021



Production Portfolio Theory II—First Steps Towards a General Portfolio Theory and Numerical Examplifications

Bernhard Heiden^{1,2(✉)} and Bianca Tonino-Heiden²

¹ University of Applied Sciences, Villach, Austria

² University of Graz, Graz, Austria

b.heiden@cuas.at

Abstract. This paper formalises the Production Portfolio Theory in more statistical and application-oriented detail, gives the first steps towards a General Portfolio Theory and shows some connections to self-organisational theory and cybernetic informational systems. The paper further demonstrates the Production Portfolio Theory with examples. Above this, we analyse the possible applications of this theory, show similarities and differences to Portfolio Theory from Markowitz, where it is inherited from, and give connections to future cybernetics and numerical information computational systems and an outlook for further research questions in the field.

Keywords: Optimisation · Computational Intelligence · Economic Application · Markowitz Portfolio Theory · Economy · Production · Statistics · Production Portfolio Theory · General Portfolio Theory

1 Introduction

Background and Current Approaches In a recent paper, we gave a new theoretical framework called Production Portfolio Theory [12]. This theory aims at combining different theoretical approaches like, mainly the Markowitz Portfolio Theory [19], as a statistical approach, to the production regime, and some other theoretical approaches like the Theory of Meaning from Putnam [21], and that of self-organisation [10, 17]. We will use a classical approach of valuation in finance, especially the Discounted Cash Flow (DCF) method (see, e.g. [3, 6–8]), where the valuation becomes important, when transforming the portfolio theoretic approach into a general portfolio theory.

The flexibility needs in production, especially in Industry 4.0, is an increasing challenge (see, e.g. [1, 11]). With regard to valuation, this means that there is more need for calculation and case-specific valuation of processes, as well as of distributed costs or short ‘options’. Conversely, costs/prices are often calculated in a total fashion, so the valuation would need to be more appropriate to the underlying ‘real’ effort.

Current approaches for valuation in production are, e.g. the value stream design [5], and also computational and economic simulations, like with the Witness program [24]. On the other side, a practical valuation is often not valued automatically, as valuations are done in the enterprise balances, which are done on a fixed time base, e.g. per year or quarterly. On the other side, the optimisation process is still out of the scope of valuation processes today. This would need a systematic approach of real-time valuation and optimisation according to individual needs, especially at the point of ‘sale’.

Research Focus One point is the intrinsic static practice of valuation in many companies. Although in some more information technology-oriented modern companies, it is standard to price in option prices, like in the airline business, there is still a growing need to optimise the decisions systematically. Our approach then seeks to close this gap, at least partly, as we will propose an approach that (a) makes possible the general valuation of options and (b) their integrated *bundle optimisation*. The bundle is then the set of decisions of different stakeholders, and so this aims at maximising, e.g. customer satisfaction.

Aim and Goal of the Paper In summary, this paper aims to provide the statistical, numerical, and methodological framework for the Production Portfolio Theory as well as its possible generalisation to generic variables and a numerical example [12] to see how to use it in practical applications.

Content In Sect. 2, we start with a short description of the self-organisation theory with regard to product portfolio and a future general portfolio theory. Then, in Sect. 3, we give an overview of some aspects related to the Production Portfolio Theory, the Markowitz Portfolio, the DCF, and the self-organisation theory. Then, in Sect. 4, we will show a numeric example applying the Production Portfolio Theory. Finally, Sect. 5 gives a summary, conclusions, limitations, main findings and an outlook on future research and applications.

2 Self-organisation

2.1 Autopoietic Systems

Autopoietic systems are operationally closed but informationally open (as we learned from Johann Göttschl, e.g. [10]).

The difference between classical organisation and self-organisation is that organisation consciously filters and processes information (is informationally open and operationally closed cf., e.g. [9, 10]), whereas self-organisation is open to incoming information but does not consciously, i.e. intentionally, seek it and perhaps could not seek it.

The organisational process is a conscious one, the self-organisational one an unconscious one, which is why this self-organisational process is attributed to natural, biological and psychological systems, but the organisational process to human and social, e.g. scientific, systems.

The reduction of complexity can be described as follows:

Self-organisational systems are superior to organisational systems, why?

Because unconscious knowledge is limitless, conscious knowledge is limited (Einstein).

Humans can still only marvel at nature, not recreate it.

Man cannot create a single plant, a single animal.

Man can only submit to the world of natural laws, and the better he submits, the better he feels.

2.2 Elements of Self-organisation

What we learn from self-organisation is that (a) the human part is essential in creating value. This is then more than a purely automated application. This leads to the point that only a critical threshold of value generation leads to the overall growth process, which then means a living system or growth in order. So this first kind of self-organisation is part of the organisation. The second step (b) would be the emerging self-organisation of new life forms.

As previously formulated there are three steps in order generation of technology in humankind: (1) fossil efficiency, (2) sustainable efficiency and (3) quantum efficiency. (1) is generated by the classical industrial revolutions by men, with pure organisational aspects. (2) becomes even more difficult possible with classical deterministic or organisational approaches. Here then comes into play the self-organisational aspects that are natural, like, e.g., Chat GPT from OpenAI [20]. The natural like nature is that there takes place a mechanisation of the mind that previously not had been thought possible. It becomes ‘self-organisational’ in the sense that increasingly large tasks are solved autonomously, which can be a definition of selforganisation, fully compatible with the theoretical theorem of being an operationally closed system and informationally open. A chatbot fulfils already these criteria. And in effect, it is a mechanistic solution to what is classically called the mind-body problem, at least operationally, according to Ryle [23]. So Ryle states there that philosophers and ordinary persons (*ibid. p. 27*), “nehmen an, die vornehmlichste Geistesbetätigung bestände in der Beantwortung von Fragen” (translated from German in English: assume that the most important intellectual activity consists in answering questions), which means that when we look at the chatbots today, that these are the modern equivalent to the chess bots some years ago, for the complex game of language with Wittgenstein.

So we first have boundaries between which self-organisation is working, leading to higher efficiency and sustainability.

2.3 Production Portfolio Self-organisationally Interpreted

Concerning Production Portfolio Theory, we have defined the elements of trade as the elements that are repeatedly reproduced, which means with Luhmann that they form a media (see, e.g. [18]). This we then have called a ‘market’ with

respect to a growth variable. The Markowitz growth variable is the return and the second moment of the market distribution is the variance, which is then a measure of the risk of the system.

We now need these growth variables that give us the logistic growth in a particular stage of development (cf. e.g., [14]). For example, concerning production efficiency, as stated earlier, we are transitioning from (1) to (2) in the ‘efficiency’ range, from fossil to sustainable efficiency. Here then, arises a market, and that is a fractal system. First, the global one. Second that of companies and countries. Thirdly that of the individual plant, company or ‘country’ organisation. Regarding companies, this is called management, and concerning countries, this is called politics, which is managed on the societal level.

The market is then, in any case, a specific system. Be it global or local. And the basic unit is then the product or the parts that the product is composed of, concerning material parts, as well as with regard to production steps or process steps, which means that these are process sequences and events. This is to be understood as a process with limited time and room and regarded as a set of events. The critical point is that they serve as reproducing elements in the media of production, which then means the dedicated series of events. This then constitutes the ‘market’. Variables are then, e.g. product price, part price, service price, or components of it, like product component price, or what we usually call option, which is then related to business decisions, as we regard their business value.

3 Production Portfolio Theory: Markowitz Portfolio, DCF, and Self-organisation

In this section, we summarize shortly essential aspects of the Markowitz Portfolio, DCF, Self-organisation, and show their relation to the Production Portfolio Theory.

3.1 Markowitz Portfolio

When we look at Fig. 1, we see a typical representation of Markowitz Portfolio theory in the left picture. The market is represented by a set of points in a Cartesian plane, spanned by the variables return (“Rendite”) (r) and variance (σ_r). In this case, we have the relation of the index r to the variable r as this relation of two statistically dependent parameters. First, the return is regarded as a mean value (first statistical property), and second, the variance, which is a fluctuation range (second statistical property). All these data depend on the *timestring of one variable*, the stock market growth, or induction from the effective price, on the hypothesized growth behind. We do not put into question that there is a difference or could be a difference here as a model assumption.

We generalize those points of time strings, with the restriction to the condition ‘growth’. In that respect, we are identical to the Portfolio Theory. What we differ in is that we go beyond the stock market growth variable, which means

effectively that we extend the term ‘market’ and are meaning, more broadly, any variable that is in a growing context, or can be described as such, can be regarded as such, as one criterion. Another is to get sufficient data to measure the describing time-string variable of the system state to be investigated. This then means, in a self-organisational context, that there is a Luhmann-like media relation of information or a double contingency in events of the process, or that the process continues to exist as a process, and is in an evolutionary sense, always, in a series of coupling and decoupling process steps. The effective growth of such a process, then, is in itself related to succession points, but due to self-organisational aspects, decoupled from previous decision points, at least partly. This openness to a decision, together with its flexibility possibility, ‘guarantees’ then a time string that justifies a ‘variable’ on the one side and the property of being statistically far distinct from other data points, which are effectively then projections in some virtual planes, as we can imagine in Fig. 1.

Those data points have then, under the mentioned conditions, the property, of being ‘far distant’—different and ‘near distant’-different.

The near-distant difference is to be seen in the property of being one measurement of one time-string variable, which is split into two properties, of the two first statistical properties of this time-string variable in the Cartesian plane, for mean and variance, where variance is ‘dependent of the ‘mean’. This is a local near-distant, far-distant relation in the sub-(space) event area, mean and variance. The first moment (mean) is locally near-distant, and the second moment is locally far-distant.

The far-distant property difference can be the data points, in the Cartesian plane, regarding the property of different market participants.

Whereas the first difference has the real property of being differentiable, at least approximately, because of the nearness, the last is not. There is a far-distant relation, as different market individuals are concerned, which is here more defined with effectively high degrees of freedom and self-organisation in the sense of effectively being autonomous. This illustrates quite well the concept of the well-known Poincaré plane (see, e.g. [16]), that precisely describes this property, here in terms of a dynamic trajectory that is somehow given in the form of a set of differential equations in the form of $dv/dt=f(v,t)$, where t denotes the time variable, and v the regarded variables.

Also, with regard to Fig. 1, we can now exemplify variables that fulfil the criteria above given to sum it up:

- of growth variable type
- market type, or reoccurring (far distant)
- evolutionary type, i.e. reoccurring in a growth fashion that originates from a mainly free-deciding autonomous unit (e.g., individual, agent, etc.)
- Poincaré plane property, of being the variable in a ‘time-string’
- of being composable, in near-distant local terms (different moments of distribution).

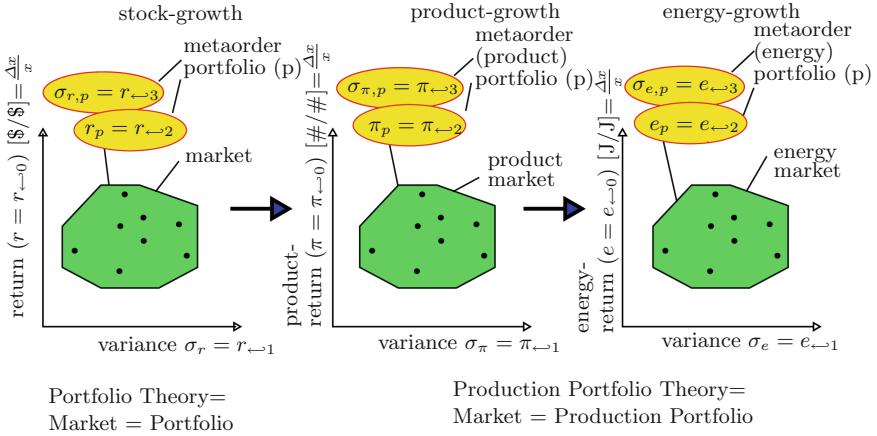


Fig. 1. Production Portfolio- and Portfolio-Theory; Δx = profit, x = value (Price or Stock-Price P , Stock Return r , Product Return π , Product Price P^π , Energy Return e , Energy Price P^e)

3.2 Return ('Rendite', r), Product Return (π), and Energy Return (e)

The return r (internal interest rate) is defined as the capital value (c_0) of cash inflow and cash outflow when the financial investment is zero (see, e.g. [6] p. 33 f), where P_i ($i..0,1$) denotes the price at the time i

$$c_0 = -P_0 + \frac{P_1}{1+r} = 0 \quad (1)$$

with

$$r = \frac{P_1 - P_0}{P_0} = \frac{P_1}{P_0} - 1 = \frac{\Delta P}{P} \sim \left[\frac{\$}{\$} \right]. \quad (2)$$

Analogously (see also Fig. 1), the product return π (Rendite) can be defined, indicating upper index π in as the product number, where $\#$ is the 'unit'

$$\pi = \frac{P_1^\pi - P_0^\pi}{P_0^\pi} = \frac{P_1^\pi}{P_0^\pi} - 1 = \frac{\Delta P^\pi}{P^\pi} \sim \left[\frac{\#}{\#} \right] \quad (3)$$

and with energy e with the common SI unit Joule (J) and upper index e :

$$e = \frac{P_1^e - P_0^e}{P_0^e} = \frac{P_1^e}{P_0^e} - 1 = \frac{\Delta P^e}{P^e} \sim \left[\frac{J}{J} \right]. \quad (4)$$

A generalisation toward meta-variable ξ gives then:

$$\xi = \frac{P_1^\xi - P_0^\xi}{P_0^\xi} = \frac{P_1^\xi}{P_0^\xi} - 1 = \frac{\Delta P^\xi}{P^\xi} \sim \left[\frac{\text{unit of } \xi}{\text{unit of } \xi} \right]. \quad (5)$$

In the above equations, besides the proportional symbol \sim and inside the square brackets, we denote the units associated with the equation as meta-order notation.

Effectively this also means that these and similar variables can be transformed: r, π, e, ξ or P, P^π, P^e, P^ξ on this occasion with adequate transformation equations, which should be possible in principle, and which is one of the main ideas of the paper. The difference lies then in the observation standpoints that allow for consideration under the same ‘invariant’ statistical methodical approach. As with Einstein’s famous method, we use the same mathematics and a different ‘physics’ (see, e.g. [4]).

3.3 Risk

Regarding the future, we have an ex-ante consideration, which means we have some properties that can be assumed as random variables. So, we have the expectancy $E(r)$ and the variance $\text{Var}(r)$. $E(r)$ is then

$$\underbrace{E(r)}_{\substack{\text{expected event} \\ \text{in the future}}} = \sum_i \underbrace{p(z_i)}_{\text{probability}} \cdot \underbrace{r(z_i)}_{\text{real event-state}} \quad (6)$$

where p is the probability and z is the state.

The risk σ is then

$$\sigma = \sqrt{\text{Var}(r)} = \sqrt{E(r^2) - E(r)^2} = \sqrt{\sum_i p(z_i) \cdot r(z_i)^2 - [\sum_i p(z_i) \cdot r(z_i)]^2}. \quad (7)$$

3.4 Portfolio Risk, Production Risk, Energy Risk

The portfolio return expectation $E(r_p)$ and the portfolio risk $\sigma(r_p)$ according to Fischer [6] is

$$E(r_p) = \sum_j E(r_j) \cdot x_j \quad (8)$$

and

$$\sigma_{r,p} = \sigma(r_p) = \sqrt{\text{Var}(r_p)} = \sqrt{E(r_p^2) - E(r_p)^2} \quad (9)$$

where x_j denotes the share or fraction of the stock in the portfolio. So, it is a norming with regard to all shares, and hence it is of a meta-order. We have denoted this with regard to Fig. 1 as cybernetic being of higher order (or orgiton, see, e.g. [13]) with $r_p = r_{\leftarrow 2}$ and $\sigma_{r,p} = r_{\leftarrow 3}$.

We can now form the expectation values of the product portfolio and the risk

$$E(\pi_p) = \sum_j E(\pi_j) \cdot x_j \quad (10)$$

and

$$\sigma_{\pi,p} = \sigma(\pi_p) = \sqrt{Var(\pi_p)} = \sqrt{E(\pi_p^2) - E(\pi_p)^2}. \quad (11)$$

The x_j indicate here then the share of product types or single products in the total production.

Analogously we can write for the energy portfolio case:

$$E(e_p) = \sum_j E(e_j) \cdot x_j \quad (12)$$

and

$$\sigma_{e,p} = \sigma(e_p) = \sqrt{Var(e_p)} = \sqrt{E(e_p^2) - E(e_p)^2}. \quad (13)$$

Here each x_j denotes then an event with a certain amount of energy invested. This can be a production step, a product, etc.

In a generalisation, the production portfolio can be regarded as a random variable ξ with the properties of the expected variable:

$$E(\xi_p) = \sum_j E(\xi_j) \cdot x_j \quad (14)$$

and

$$\sigma_{\xi,p} = \sigma(\xi_p) = \sqrt{Var(\xi_p)} = \sqrt{E(\xi_p^2) - E(\xi_p)^2}. \quad (15)$$

x_j denotes here the portion of this random variable in a generalised portfolio consisting of (countable) elements of the random variable ξ .

3.5 Covariances of Stocks, Energy and in General Consideration

Now, we have in this set of elements, here the stocks, a quadratic expectation value. This means that, when combining those variables, we have no additional term in the linear case, and hence each part can be expected as a singleton variable. But in the real world, there are relations, and this relation can be summed up in a matrix of ‘correlations’. So we can split the interdependency into two parts: self-(inter)dependency and alien- (‘fremd’) (inter)dependency.

The process of variable multiplication in the next step can be regarded as a self-referential step of future progression or implementing a time (variable) progression (where it is never the case that the time does not progress, for the simplified assumption of ‘normal’ earth conditions).

The covariance $\text{Cov}(r_j, r_k)$ is defined as:

$$\text{Cov}(r_j, r_k) = E(r_j \cdot r_k) - E(r_j) \cdot E(r_k) \quad (16)$$

and denotes the difference between the inner expectation value multiplication versus the outer one. This means, it is the formal description of the non-linear relationship between two points in space-time events.

The dependency can be regarded as an information flow, which leads to an unequilibrated state of non-equilibrium or order, far from the thermodynamic equilibrium.

The norming of the variance defines the correlation coefficient ρ :

$$\rho(r_j, r_k) = \frac{\text{Cov}(r_j, r_k)}{\sigma(r_j) \cdot \sigma(r_k)}. \quad (17)$$

The sum then over all possible or combinatorial relationships, which means effectively, information flows, is then the expected, the ‘reality’ result, which we call variance $\text{Var}(r_p)$ and its root, or space-time, recalibrated, or -normed ‘present-reality-state’ risk $\sigma(r_p)$.

$$\text{Var}(r_p) = \sum_j \sum_k \text{Cov}(r_j, r_k) \cdot x_j \cdot x_k \quad (18)$$

So, effectively a variable of any kind of state $v(t)$ is split into a continuum concerning the variable and the space-time dimension. This leads to a net of relations, described by the Covariance (Cov) matrix. Here crystallises the reality determining step of reality restrictions (mathematically often denoted as ‘boundary conditions’).

Each evolutionary step is hence described as a potentially unknown state of interrelation conditions that converge to a prototyping crystallising point of a new space-time reality event, which is known by observation, being part of the evolution process, and by this irreversibly changing the ‘reality’ boundary conditions.

The velocity of observing systems, or in the real world today, an application of cybernetic systems makes a massive difference in this co-formed reality.

The point is that the observation, as an irreversible information flow process, allows not only for potentially better prediction but also changes the reality, and hence is invariable coupled to an increasing potential of higher unpredictability.

The mechanism is of statistical nature. And, what we look at are the linear projections (a) of two space-time events A,B, as well as (b) their statistical meta-descriptions in an overall self-organisational framework.

In the differentiated case of the production portfolio, we can have the Covariance (Cov) and the Variance (Var) concerning the products P^π and its return π :

$$\text{Cov}(\pi_j, \pi_k) = E(\pi_j \cdot \pi_k) - E(\pi_j) \cdot E(\pi_k). \quad (19)$$

The norming of the variance defines here the correlation coefficient ρ :

$$\rho(\pi_j, \pi_k) = \frac{Cov(\pi_j, \pi_k)}{\sigma(\pi_j) \cdot \sigma(\pi_k)} \quad (20)$$

and

$$Var(\pi_p) = \sum_j \sum_k Cov(\pi_j, \pi_k) \cdot x_j \cdot x_k. \quad (21)$$

x_j and x_k denote here the rates with regard to the product returns.

In the energy case, the formulas can be written for the covariance and variance:

$$Cov(e_j, e_k) = E(e_j \cdot e_k) - E(e_j) \cdot E(e_k) \quad (22)$$

for the correlation coefficient ρ

$$\rho(e_j, e_k) = \frac{Cov(e_j, e_k)}{\sigma(e_j) \cdot \sigma(e_k)}. \quad (23)$$

and the variance Var

$$Var(e_p) = \sum_j \sum_k Cov(e_j, e_k) \cdot x_j \cdot x_k. \quad (24)$$

x_j and x_k denote then here the rates with regard to the energy return.

In a general consideration, the formulas can be written with the general random variable ξ :

$$Cov(\xi_j, \xi_k) = E(\xi_j \cdot \xi_k) - E(\xi_j) \cdot E(\xi_k) \quad (25)$$

for the correlation coefficient ρ

$$\rho(\xi_j, \xi_k) = \frac{Cov(\xi_j, \xi_k)}{\sigma(\xi_j) \cdot \sigma(\xi_k)} \quad (26)$$

and the variance Var

$$Var(\xi_p) = \sum_j \sum_k Cov(\xi_j, \xi_k) \cdot x_j \cdot x_k. \quad (27)$$

x_j and x_k denote here the rates with regard to the random variable return ξ .

3.6 Two Stock, Two Products, Two Energies, and Two Generic Random Variable Case

The solution of the above equations for the portfolio theory for two stocks A,B is then for expectation value $E(r_p)$ of the stock return and risk $\sigma_{r,p}$:

$$E(r_p) = x_A \cdot E(r_A) + x_B \cdot E(r_B) \quad (28)$$

$$\sigma_{r,p} = \sqrt{Var(r_p)} = \quad (29)$$

$$= \sqrt{x_A^2 \cdot Var(r_A) + x_B^2 \cdot Var(r_B) + 2 \cdot x_A \cdot x_B \cdot Cov(r_A, r_B)}. \quad (30)$$

We can formulate the problem of a Product Portfolio for the product number return $E(\pi_p)$ and the risk $\sigma_{\pi,p}$:

$$E(\pi_p) = x_A \cdot E(\pi_A) + x_B \cdot E(\pi_B) \quad (31)$$

$$\sigma_{\pi,p} = \sqrt{Var(\pi_p)} = \quad (32)$$

$$= \sqrt{x_A^2 \cdot Var(\pi_A) + x_B^2 \cdot Var(\pi_B) + 2 \cdot x_A \cdot x_B \cdot Cov(\pi_A, \pi_B)} \quad (33)$$

Here x_A and x_B denote the share of A,B in the Product Portfolio.

Concerning the Energy Portfolio, we get the expected value for the energy return $E(e_p)$ and the risk $\sigma_{e,p}$:

$$E(e_p) = x_A \cdot E(e_A) + x_B \cdot E(e_B) \quad (34)$$

$$\sigma_{e,p} = \sqrt{Var(e_p)} = \quad (35)$$

$$= \sqrt{x_A^2 \cdot Var(e_A) + x_B^2 \cdot Var(e_B) + 2 \cdot x_A \cdot x_B \cdot Cov(e_A, e_B)}. \quad (36)$$

Here x_A and x_B denote the share of A,B in the Energy Portfolio.

Finally, we can generalise for a random variable that is sufficient for the given conditions of a market, the decoupling and the evolutionary aspects ξ and its risk $\sigma_{\xi,p}$:

$$E(\xi_p) = x_A \cdot E(\xi_A) + x_B \cdot E(\xi_B) \quad (37)$$

$$\sigma_{\xi,p} = \sqrt{Var(\xi_p)} = \quad (38)$$

$$= \sqrt{x_A^2 \cdot Var(\xi_A) + x_B^2 \cdot Var(\xi_B) + 2 \cdot x_A \cdot x_B \cdot Cov(\xi_A, \xi_B)}. \quad (39)$$

Here x_A and x_B denote the share of A,B in the generic random value variable Portfolio.

3.7 DCF

The Discounted Cash Flow (DCF) is a model that counts as the most appreciated and accurate one concerning future prediction. In essence, it is grounded by the definition,

$$c_t = \underbrace{c_0}_{\text{initial growth}} + \underbrace{\sum_{i=1}^t \frac{c_i}{(1+r)^i}}_{\text{stabilisation}} + \underbrace{c_e}_{\text{decay}} \quad (40)$$

where c denotes the capital at time $t,i,0$ and e at the end when sold, and r is the return (Rendite) of the capital. The capital c_i represents hence the profit in a specific time-interval.

In our notation, the capital c_i is the base value, and the return is the long-term growth term of this base value. According to the DCF method, there are some differentiations concerning this basic value, the c_i . We regard here also as essential that in the common definition as in equation (40) has been included the self-organisational sequence as described in [10, 12], *growth, stabilising, decay*, and again growth as a part of the overall evolutionary process. This, then, is the fractal short-term perspective, which denotes its similarity, also according to chaos theory and theory of fractal systems, which is the long-term equivalent.

This then denotes the market definition in the Portfolio theory and that of the Production Portfolio theory on different size scales.

The beginning is then denoted to c_0 , where growing becomes possible, the injection of ‘information’ and energy. The growth phase, then, which is the stable part of the process, and the selling part, is where the business is given away or transformed. Finally, with this whole overall process, the condition of *autonomy*, in the sense of a Luhmann criteria of continuation, is obtained, which then is the ‘total’ calculated value c_t of the ‘investment’. The *continuation medium* is then the ‘market’ in the above-denoted meaning.

3.8 Products P, Energy E, and Generalisation ξ

We can now write equation (40) with regard to products π ,

$$c_t^\pi = \underbrace{c_0^\pi}_{\text{initial growth}} + \underbrace{\sum_{i=1}^t \frac{c_i^\pi}{(1+\pi)^i}}_{\text{stabilisation}} + \underbrace{c_e^\pi}_{\text{decay}} \quad (41)$$

with regard to energy e ,

$$c_t^e = \underbrace{c_0^e}_{\text{initial growth}} + \underbrace{\sum_{i=1}^t \frac{c_i^e}{(1+e)^i}}_{\text{stabilisation}} + \underbrace{c_e^e}_{\text{decay}} \quad (42)$$

and a generic (random) variable ξ denoting a *value* and a *medium* (as money (price), energy, products, etc.) are media, with regard to their observation space-time or event, which can be any variable as an extension of price, products, and energy, with the associated unit:

$$c_t^\xi = \underbrace{c_0^\xi}_{\text{initial growth}} + \underbrace{\sum_{i=1}^t \frac{c_i^\xi}{(1+\xi)^i}}_{\text{stabilisation}} + \underbrace{c_e^\xi}_{\text{decay}}. \quad (43)$$

We can conclude that we can use DCF calculations to norm the value of an investment. A market, and an element in the market, as we understand it here,

then stocks, products, energy, and a generic value, can hence be modeled with these terms in detail.

3.9 Markowitz Minimum Varianz Portfolio (MVP)

Now, that we have the interesting random variables, the next step is to optimise these variables. Regarding the market, there are two main strategies for the two variables E , and σ .

Stock Portfolio When looking at the two-parameter case (two-singeltons), we have two equations describing the parameters of the Portfolio: $E(r_p)$ (equation (28)) and $\sigma_{r,p}$ (equation (29,30)). The third condition is that the sum of the shares is one, which means in the two stocks case: $x_A + x_B = 1$.

So the first optimisation case means having a goal function and a goal, as well as boundary conditions. Category (A), as we denote it here, is to minimise the risk σ . This is called the *risk-averse decision of the investor*. This leads then, due to the hyperbolic surface structure (boundary) of the goal function to a point, the minimum variance portfolio (MVP). This problem can be solved by the minimisation condition or solving the extreme value problem

$$\min \sigma_{r,p}^2 = \text{Var}(r_p) \quad (44)$$

with the side conditions

$$E(r_p) = \sum_j E(r_j) \cdot x_j \quad (45)$$

and

$$1 = \sum_i x_i \text{ and } x_i \geq 0. \quad (46)$$

With the variation of the parameter $E(r_p)$, the Markowitz efficiency curve can be calculated, by this obtaining the boundary of the singleton market values.

In category (B), as we denote next, the optimisation solution focuses on the return r_p . The so-called *rational investor* is then seeking an optimal return r_p when a risk $\sigma_{r,p}$ is given.

The problem can then be formulated by the following two equations, where the finishing condition in equation (46) stays the same:

$$\max E(r_p) = \sum_j E(r_j) \cdot x_j \quad (47)$$

and

$$\sigma_{r,p}^2 = \text{Var}(r_p). \quad (48)$$

Production Portfolio For the production portfolio, the equations for the category (A) problem, how to reduce the risks in production, to fulfil the contract and be reliable are

$$\min \sigma_{\pi,p}^2 = \text{Var}(\pi_p) \quad (49)$$

with the side conditions

$$E(\pi_p) = \sum_j E(\pi_j) \cdot x_j \quad (50)$$

and equation (46). For category (B) the point to be most efficient in the production, which means the inclusion of risk, not to produce in time and or with the contracted amount of products can be described by the equations

$$\max E(\pi_p) = \sum_j E(\pi_j) \cdot x_j \quad (51)$$

the condition

$$\sigma_{\pi,p}^2 = \text{Var}(\pi_p). \quad (52)$$

and that in equation (46).

Energy Portfolio For category (A), the question is to reduce the risk of having energy available. This is related to the now common keyword ‘black-out’. The less the risk, the less we have to expect this event. And it is clear that in times of seeming energy-source scarcity, this becomes of increasingly common interest. The equations are here then

$$\min \sigma_{e,p}^2 = \text{Var}(e_p) \quad (53)$$

with the side conditions

$$E(e_p) = \sum_j E(e_j) \cdot x_j \quad (54)$$

and equation (46). For category (B) the last equation stays the same and the other conditions are

$$\max E(e_p) = \sum_j E(e_j) \cdot x_j \quad (55)$$

and

$$\sigma_{e,p}^2 = \text{Var}(e_p). \quad (56)$$

This then means an optimisation toward maximum energy profit earnings. The main problem here is that this procedure weakens the overall market. This is a main characteristic feature of our current economic system, as we will criticise more deeply in our next publications, and which can be overcome by an overall system that is stepping towards an overall osmotic characteristic.

Generic Portfolio Analogously we can generalise the portfolio approach to a generic random variable ξ which denotes a ‘value’ in some sense, for a person, for a process, for a goal, and is embedded in a self-organisational context of growth, decay or stable evolutionary phase. So, we can apply the statistical properties of relative growth and risk, or probability, to be in a predicted range of values with regard to the basic growth and the treated random variable. The category (A) problem, of minimising the risk of an expected outcome of the variable ξ then gives the equations

$$\min \sigma_{\xi,p}^2 = \text{Var}(\xi_p) \quad (57)$$

with the side conditions

$$E(\xi_p) = \sum_j E(\xi_j) \cdot x_j \quad (58)$$

and equation (46).

For category (B), to maximise the growth of the variable ξ , the last equation stays again the same, and the conditions are

$$\max E(\xi_p) = \sum_j E(\xi_j) \cdot x_j \quad (59)$$

and

$$\sigma_{\xi,p}^2 = \text{Var}(\xi_p). \quad (60)$$

Generally, all these variables can be redefined according to the optimisation problem because it might not be rational to ‘maximise’ a growth variable significantly when it is clear that the interactions are increasing with it simultaneously. Other possibilities are to invert the problem by converting it to a minimisation problem, or take into account a specific range of the variable that shall be obtained (cf. also for optimisation techniques [2]).

This means that we have to take care of possible ‘chain reactions’, of many kinds of possible variables, in the future, increasingly. At least the good thing now is that with this approach, we can better control systems of any kind and be more rational and effective.

4 Numerical Example of Production Portfolio Theory

Before we start with the numerical examples, an interesting feature of virtualisation (computer simulation, cybernetic systems, language, etc.) shall be underpinned. When discussing the topic, the question has arisen, what is the difference between simulation, and no simulation of a production process, or any kind of simulation. In fact this leads even back to an epistemologic question, how to gain knowledge. But only, when we do not go so far and allow the pyrronic scepticism to allow, that nothing can be known for sure (cf., e.g. [22]), we can say at least, that it will be better, with a certain probability, to give the scepticism some inevitable room. But why to be not so vague. A major reason can be seen in

a principal difference, that we have called the osmotic paradigm [15, 25], which is related to informational or cybernetic systems, in increasingly high amounts of applications, that is in the principle of cybernetisation, where modern forms, are augmented and virtual reality. Human language, as evolutionary relict, is of the same kind, as comes to consciousness, when regarding chatbots like the above mentioned. The crucial point is the play. Be it Wittgenstein's 'language game play', the play of computer games, or playing around with digital twins or computer simulations. The dominant, and the increasing dominance of informational or cybernetic systems lies in the fact, that a far distance as well as a near distance in cybernetic terms can be achieved at the same time. This virtual-nearness, leads then to major difference, of real and virtual systems. Virtual system experiments, if they depict 'reality', can then reduce potentially *risk*, compared to using real-experiments. So a major reason for simulation is, that a realising or prototyping process is (a) accelerated, and (b) by simultaneously reducing risk. This is the reason, why human language, makes the difference in developing the human species above other near related primates. And this is also a chaos-theoretic insight, that nearby trajectories, can go potentially infinitesimally far apart in time. The second type of virtualisation is that of using flexible tools. They reduce the risk on the operational level as they *make processes more accurate*. And both processes are related. First by backcoupling processes, and second by the osmotic principle of 'redundancy', which effectively reduces risk then, although redundancy in normal operation, or epistemologically reduces information, can in such a meta-verse-context due to its nonlinear property, and the emergence of a phase reversal just be used for the opposite, which is in this case reducing risk, potentially.

4.1 Model Example Scheme for Production Portfolio

The here given example is widely numerically the same as in task ('Aufgabe') 21 in [6] p. 228, 39f, 45f, where the task has been transformed from Portfolio Theory to Production Portfolio Theory as is derived in this paper above. For reference, this book and the accompanying examples book are recommended for further deepening the knowledge and application of Markowitz's Portfolio Theory.

A production planner wants to produce 1 Million parts (#) of a Product Portfolio of two products A, and B. The product return π_P of the products is normal distributed with the following parameters (Table 1):

Table 1. Model Example for Production Portfolio

	A	B
$E(\pi_j)$	0.08	0.15
$\sigma_{\pi_j, p}$	0.1	0.2

I. The Product Portfolio of the contract sold is 60% of A and 40% of B, and the correlation ρ between the two productions (production numbers per time, which is a ‘material’ number flow) is 0.4.

(a) How is the product return π_p of the product portfolio distributed, and which parameters $\sigma_{\pi,p}$ and $E(\pi_p)$ do this distribution have?

(b) What is the probability of a negative production portfolio return π_p ?

(c) What is the probability that the product return π_p is greater than 7%?

II. (a) What is the minimum variance product portfolio $\min \sigma_{\pi,p}^{MVP}$?

4.2 Example Solution

I. (a) To calculate the expected product return we use equation (31):

$$\begin{aligned} E(\pi_p) &= x_A \cdot E(\pi_A) + x_B \cdot E(\pi_B) \\ &= 0.6 \cdot 8\% + 0.4 \cdot 15\% \\ &= 10.8\% \end{aligned}$$

For the product portfolio variance we use then equations (32) and (33):

$$\begin{aligned} \sigma_{\pi,p} &= \sqrt{Var(\pi_p)} = \\ &= \sqrt{x_A^2 \cdot Var(\pi_A) + x_B^2 \cdot Var(\pi_B) + 2 \cdot x_A \cdot x_B \cdot Cov(\pi_A, \pi_B)} \\ &= \sqrt{0.6^2 \cdot 0.1^2 + 0.4^2 \cdot 0.2^2 + 2 \cdot 0.6 \cdot 0.4 \cdot Cov(0.08, 0.15)} \end{aligned}$$

and with equation (20) $0.4 = Cov(0.08, 0.15)/(0.1 \cdot 0.2)$ solved for the covariance Cov we get

$$\begin{aligned} &= \sqrt{0.6^2 \cdot 0.1^2 + 0.4^2 \cdot 0.2^2 + 2 \cdot 0.6 \cdot 0.4 \cdot 0.4 \cdot 0.1 \cdot 0.2} \\ &= \sqrt{0.01384} = 11.76\%. \end{aligned}$$

The distribution of the product portfolio is then $\pi_p = N(0.108; 0.1176)$ where N is the Gaussian Normal distribution.

(b) The probability P of a up to negative product portfolio return is

$$P(\pi_p \leq 0) = \phi\left(\frac{\pi_p - E(\pi_p)}{\sigma(\pi_p)}\right) = \phi\left(\frac{0 - 0.108}{0.1176}\right) = \phi(-0.918)$$

With the command ‘=STANDNORMVERT(-0.918)’ we calculate the ϕ function in Microsoft® Excel®, we get the result 17.93%. So, the probability of having a negative product return is 17.93%, which means to produce not enough parts as are ordered with this probability.

(c) The probability P to have a product portfolio return greater than 7% can be calculated as

$$\begin{aligned}
 P(\pi_p \geq 0.07) &= 1 - \phi(\pi_p \leq 0.07) \\
 &= 1 - \phi\left(\frac{\pi_p - E(\pi_p)}{\sigma(\pi_p)}\right) \\
 &= 1 - \phi\left(\frac{0.07 - 0.108}{0.1176}\right) \\
 &= 1 - \phi(-0.3231) = \phi(0.3231) = 62.66\%
 \end{aligned}$$

what we have calculated with the same Microsoft® Excel® command as above for the ϕ function. So, the probability of having a product portfolio return or excess of more than 7% is 62.66%. **II.** (a) With regard to the MVP, in this case, for the production portfolio case, we solve the optimisation problem, which is denoted by equations (46,49,50). The solution can be done analytically or numerically. As a solution, the obtained value according to [6] p. 46, where we have translated the results into the here given nomenclature, can be yielded:

$$\begin{aligned}
 x_A^{MVP} &= 94.12\% \text{ with } \rho = 0.4, x_B^{MVP} = 5.88\%, \\
 E^{MVP}(\pi_p) &= 8.41\% \text{ and } \sigma_{\pi,p,MVP} = 9.94\%.
 \end{aligned}$$

The solving steps of the numeric optimisation are given in Fig. 2.

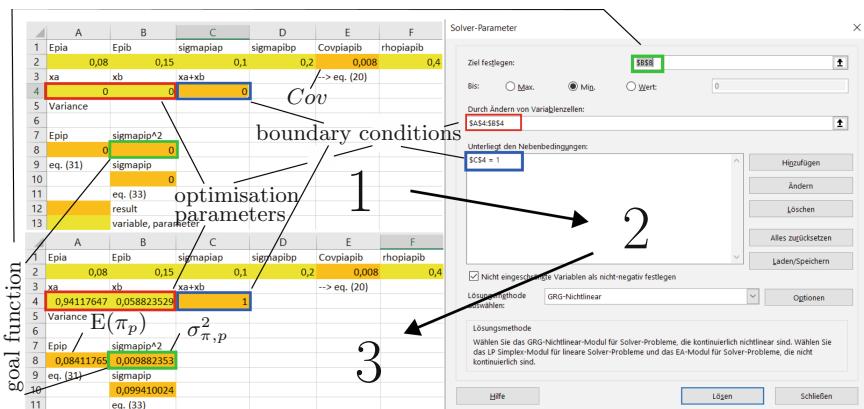


Fig. 2. Optimisation in Microsoft® Excel® with the Solver Add-In

The overall task of optimisation, here the minimisation, is to define parameters, a goal function and the restrictions or boundary conditions, as is also shown in the figure. First in (1) all the parameters of the example are given in the yellow fields, as well as the equations in the orange fields for (a) $E(\pi_p)$, (b) $\sigma_{\pi,p}^2$, (c) Cov and the boundary condition (d) $x_A + x_B = 1$ respectively the open

formula to yield it $x_A + x_B = ..$ and its restriction in the dialogue box of the solver depicted in (2). Then in the optimisation dialogue (2) the references to the boundary condition (d) is given. The variables x_A and x_B are referenced, as these variables vary during the optimisation. (b) is the goal. The condition that the variables are non-negative is checked. Finally in (3) the optimisation results can be read in the Microsoft® Excel® sheet.

MVP denotes here the Minimum Variance Production Portfolio. It can be seen that in the above given combination of the products A and B the risk can be minimised and it is less than the risk of each single product A and B. So, this shows the diversification effect of the optimal product portfolio, in this case, concerning risk minimisation, which means on the customer site, to achieve maximal customer satisfaction from the production view of available products.

5 Summary, Conclusion, Limitations and Outlook

5.1 Summary, Conclusions, Limitations and Main Findings

The important learning here is that in the simulation context *redundance* can be used to *reduce risk*, and is hence a useful epistemic concept, as opposed to the seemingly pure rationalistic positivist, that needs to explain everything informationally efficient.

Limitations can be seen concerning the Markowitz Portfolio Theory itself, and the same limitation applies as is the case for, e.g. the properties of the random variables.

In practical cases, it might be difficult to have enough data. So, for starting a business, production, etc., it will be challenging to make the correct assumptions with regard to valuation, as data define the quality of statistics in most cases. The same applies but more stringently to correlations of risks. This might be difficult to interpret and reproduce as an issue of increasing complexity. This means the more components or options, the more difficult it will be to judge whether the correlation effect is true or is otherwise effected, produced or influenced.

So, if our method is suitable to *reduce risk*, and also to *increase productivity*, then we can regard it as successful and is this hence the lower border of its application and reasonable applicability.

5.2 Outlook and Research Applications

Now the Production Portfolio Theory is ready to be applied in interdisciplinary application fields to judge and model business perspectives of different kinds. Of production and as well service schemes. Applications of information and service will certainly increase in importance due to the derived property of potential risk reductions, which is increasingly essential in increasingly complex worlds.

Future research questions and topics are to adapt the advances in portfolio theory after Markowitz, into production portfolio theory and in general portfolio theory. Crucial will also be the question of how to formalise further and generalise

the theory, e.g. by the idea of the tensor calculus, and whether this is feasible. Thinkable are also considerations and numerical solutions using graph-theoretic approaches that will merge deeper with statistical issues. An interesting question is how to formalise growth, evolutionary steps and general statistical properties. Here opens a myriad of possibilities. Besides the pure numerical solution, general system solutions also seem to be within reach here. One of our latest discoveries is that it is possible to merge as far disciplines like justice and economy into one combined theory with a powerful impact, which we have denoted provisionally as Portfolio Fairness Theory. This approach seems far-reaching and effectively applicable in a real inter- and transdisciplinary way. We hope to give a follow-up research in this very direction in the near future.

Researchers and scholars of all disciplines are invited to apply this method practically and numerically to reduce risk effectively and be productive, which means increasing overall human progress and well-being. Of course, this will not for sure go on forever, but we will have, with it, a rational chance to master it. Further steps towards this very chance are to *further generalise* and *apply* these new methods, and most importantly, to show how to use them effectively as tools in a way that this is the virtual new background of the world of the future after the future.

References

1. Bauernhansl, T., ten Hompel, M., Vogel-Haeuser, B.: Industrie 4.0 in Produktion, Automatisierung und Logistik. Springer Vieweg Verlag, Wiesbaden (2014)
2. Benker, H.: Mathematische Optimierung mit Computeralgebraystemen. Springer, Berlin (2003)
3. Damodaran, A.: The Dark Side of Valuation. Fifth Printing (2013)
4. Einstein, A., von Smoluchowski, M.: Studies on the theory of Brownian motion - Treatise on Brownian motion and related phenomena, original in German: Untersuchungen über die Theorie der Brownschen Bewegung - Abhandlung über die Brownsche Bewegung und verwandte Erscheinungen, volume 199/207 of Ostwalds Klassiker der exakten Wissenschaften. Verlag Harri Deutsch, Frankfurt am Main, 2001. contains the original 1905 Einstein paper: "Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen" (Aus den Annalen der Physik, 4. Folge, Band 17, 1905, Seite 549–560) and other works on the particle-osmotic analogy
5. Erlach, K.: Value Stream Design—The Path to A Lean Factory, Original in German: Wertstromdesign—Der Weg zur schlanken Fabrik, 2nd edn. Springer, Berlin (2010)
6. Fischer, E.O.: Finanzwirtschaft für Fortgeschrittene. Oldenburg Verlag, München, 3. aufl. edition (2002)
7. Fischer, E.O.: Finanzwirtschaft für Anfänger. Oldenburg Verlag, München, 5. aufl. edition (2009)
8. Fischer, E.O., Keber, C., Mahringer, D.G.: Arbeitsbuch zur Finanzwirtschaft für Anfänger. Oldenburg Verlag, München, 2. nachdruck edition (1999)
9. Fruhmann, I.: Zur differenzierenden offenenheit wissenschaftlicher systeme epistemologisch-sprachphilosophische zugänge (2011)

10. Götschl, J.: Vorlesung: Einführung in die philosophie der gegenwart (organisation und selbstorganisation im natur-kulturgefüge); translated: Lecture: Introduction to Contemporary Philosophy (organization and self-organization in the structure of nature and culture) (2023)
11. Heiden, B.: Wirtschaftliche Industrie 4.0 Entscheidungen—mit Beispielen—Praxis der Wertschöpfung. Akademiker Verlag, Saarbrücken (2016)
12. Heiden, B., Tonino-Heiden, B.: First Elements of Production Portfolio Theory: A New Industrial Engineering Scientific Method
13. Heiden, B., Tonino-Heiden, B.: Philosophische Untersuchungen—Spezielle Orgitontheorie (unpublished)
14. Heiden, B., Tonino-Heiden, B.: Lockstepping Conditions of Growth Processes: Some Considerations Towards Their Quantitative and Qualitative Nature from Investigations of the Logistic Curve, pp. 695–705 (2023)
15. Heiden, B., Volk, M., Alieksieiev, V., Tonino-Heiden, B.: Framing artificial intelligence (AI) additive manufacturing (AM). Procedia Comput. Sci. **186**, 387–394 (2021); Elsevier B.V.
16. Hilborn, R.C.: Chaos and Nonlinear Dynamics—An Introduction for Scientists and Engineers. Oxford University Press, New York (1994)
17. Kanitscheider, B.: Chaos und Selbstorganisation in Natur- und Geisteswissenschaft, pp. 66–90. Böhlau Verlag, Köln (2006)
18. Luhmann, N.: Soziale Systeme, 17th edn. Suhrkamp Verlag AG (2018)
19. Markowitz, H.M.: Portfolio selection*. J. Financ. **7**(1), 77–91 (1952)
20. OpenAI. Chatgpt: Optimizing language models for dialogue
21. Putnam, H.: Die Bedeutung der “Bedeutung”. Klostermann Texte (2004)
22. Rorty, R.: Der Spiegel der Natur. Suhrkamp
23. Ryle, G.: Der Begriff des Geistes. Reclam, Ditzingen
24. Teichert, C.: WITNESS—Logistik, pp. 34–42. Springer Gabler Verlag, Wiesbaden (2018)
25. Tonino-Heiden, B., Heiden, B., Alieksieiev, V.: Artificial life—investigations about a universal osmotic paradigm (UOP). In: Arai, K. (ed.) Intelligent Computing, LNNS, vol. 285, pp. 595–605. Springer Nature (2021)



AI as a Threat to Education: Contrasting GPT-3 and Google in Answering Questions Along Bloom's Taxonomy of Educational Objectives

Nina Li^(✉)

York House School, 4176 Alexandra Street, British Columbia, Vancouver, Canada
ninayiyihu@gmail.com

Abstract. Artificial Intelligence products have seen an unprecedented rise in the last few years, with GPT-3 and ChatGPT considered the most powerful AI tools in history. Recently, people have been concerned that services such as ChatGPT may be a threat to education. Specifically, teachers are concerned about students using the free and accessible tool as a Wikipedia replacement to complete homework and to write assignments for them. It poses importance for both researchers and practitioners in AI, and community stakeholders including teachers and parents, to understand the capability of AI text generative models in answering questions that may appear in an educational assessment or the school's curriculum. In this research, we follow a widely adopted framework in defining educational question types, namely Bloom's taxonomy. We perform an experiment comparing the quality of answers retrieved from Google and GPT-3 on a series of questions identified along the hierarchy of Bloom's taxonomy. We summarize the capabilities and limitations of using GPT-3 to answer educational questions along Bloom's taxonomy. We discuss the implications on designing educational assessments to combat the threats introduced by such AI services.

Keywords: Generative Language Models · Question Answering · AI in education · Bloom's Taxonomy

1 Introduction

Artificial Intelligence products have seen an unprecedented rise in the last few years, with GPT-3 and ChatGPT considered the most powerful AI tools in history. A lot of prior research and recent blog posts have investigated what people can do with conversational artificial intelligence such as GPT-3 [2, 3]. As example, GPT-3/ChatGPT can help people prepare a cover letter for a job application, can write, debug and explain code, solve math problems, and write essays on almost any topic. Recently, people have been concerned that services such as ChatGPT may be a threat to education [2]. Specifically, teachers are concerned about students using the free and accessible tool as a Wikipedia replacement

to complete homework and to write assignments for them. It poses importance for both researchers and practitioners in AI, and community stakeholders including teachers and parents, to understand the capability of AI text generative models such as GPT-3 and more recently ChatGPT in answering questions that may appear in an educational assessment or the school's curriculum.

In this research, we follow Bloom's taxonomy of defining question types. Bloom's taxonomy is a hierarchical model used for the classification of educational learning objectives into levels of complexity and specificity [4]. Bloom's taxonomy defines learning objectives in six categories, namely (1) Knowledge (2) Comprehension (3) Application (4) Analysis (5) Synthesis, and (6) Evaluation. It is widely used by teachers in instructional design. Teachers often strive to design educational materials that target higher Bloom's goals which are considered to exercise students' higher-order thinking, as opposed to simply letting students memorize facts. Investigating generative language models' capabilities in answering educational questions along Bloom's taxonomy provides practical guidance to teachers on designing assessments and learning materials.

We perform an experiment comparing the answers retrieved from Google and ChatGPT on a series of questions identified along the hierarchy of Bloom's taxonomy. We compared the quality of answers and summarized the capabilities and limitations of using ChatGPT to answer educational questions along Bloom's taxonomy. We discuss the implications of designing educational assessments to combat the threats introduced by such AI services.

Our results show that (1) GPT-3 can produce human-like responses to questions, making it ideal for answering questions that require higher levels of thinking. For example, questions that require comprehension, application, analysis, synthesis, and evaluation (higher on the bloom's hierarchy) can be answered by GPT-3 with greater accuracy and nuance than Google. (2) GPT-3 sometimes produces incorrect answers to math questions, or questions that require computation; whereas Google has the advantage of being able to quickly retrieve information from the web, making it the better choice for answering factual or straightforward questions that require a basic level of knowledge. (3) For detail-oriented questions that fall into the category of comprehension and application, e.g., writing a piece of code, or explaining a chemistry phenomenon, GPT-3 can provide an accurate answer whereas Google often retrieves a list of articles where the user must dig up and summarize the answers by themselves. (4) It's also worth noting that when looking for reliable sources, such as news articles or reference papers, GPT-3 may not be the best choice as it cannot distinguish between credible sources and fake resources. Therefore, it is important to use GPT-3 with caution and cross-reference with other sources to ensure the accuracy and reliability of the information provided.

2 Related Work

2.1 Comparison of Google Versus ChatGPT

Many recent blog posts have contrasted the capabilities of using GPT-3 versus Google in answering user questions and retrieving information. A recent blog post put it as “GPT-3 is quietly damaging Google search” [5]. The author found that when using GPT-3 to answer questions, the results were much simpler and clearer. Another article showed that GPT-3 tends to give users direct answers whereas they would have to go to various websites to dig up the answer if they had Googled the question [6]. However, the same article also spot areas where Google gave more satisfying answers than ChatGPT, e.g., asking for gift ideas where Google gave links to websites and also gave personalized recommendations based on the user’s search history. Users have found that ChatGPT may generate strangely close yet totally wrong answers, whereas Google generates consistent and reliable answers [6].

2.2 The Use of ChatGPT in Education

A lot of experts are saying that artificial intelligence chatbots, such as ChatGPT are changing the way students are taught [6]. A real worry and potential threat to higher education is whether students will use ChatGPT to answer all of their assignments and stop learning. On the other hand, there have been investigations on how to minimize such risks. For example, a student at Princeton University developed an app that can quickly and efficiently detect whether an essay is ChatGPT or human written [1].

Whereas on the other hand, people have argued that ChatGPT could aid student learning while not taking their learning opportunities away. As an example, similar AI services could provide code explanations to students who are struggling to learn to code. In these cases, AI could make the student feel “empowered” to work independently [6]. Thinking of how teachers may use ChatGPT, some teachers may give ChatGPT a prompt and analyze its response with students as a practice in editing and critical thinking [1]. Other teachers have explored using ChatGPT to generate ideas for lesson plans and class activities.

However, it remains unclear how good such AI generative models are in answering educational questions. If teachers want to design educational assessments or learning materials that cannot be simply answered by GPT-3, what should they do? In this research, we aim to tackle this problem. Specifically, we address two research questions.

- RQ1: We will explore the boundaries of generative language models in answering educational questions. What are they capable of doing and what are they not capable of doing?
- RQ2: Compared to traditional information retrieval tools such as Google, what kinds of questions are better answered by Google? What kinds of questions are better answered by generative AI?

3 Methods

We use Bloom's taxonomy to categorize the type of educational questions because it is a widely recognized method for assessing student knowledge and understanding. This taxonomy provides a structure for organizing questions and tasks based on the level of cognitive complexity they require [4]. Bloom's taxonomy allows us to categorize questions into different groups that require varying levels of cognitive load. Since Bloom's taxonomy is also widely adopted in instructional design in classrooms. Analyzing GPT-3's capability in answering questions along Bloom's taxonomy offers practical evidence to teachers when they design assessment questions and learning materials. Here we offer a brief description of the 6 levels in Bloom's taxonomy.

Bloom's taxonomy [4].

- Knowledge (organization and selection of facts and ideas)
- Comprehension (use of facts, rules and principles)
- Application (separation of a whole into component parts)
- Analysis (separation of a whole into component parts)
- Synthesis (combination of ideas to form a new whole)
- Evaluation (development of opinions, judgments, or decisions).

We picked educational assessment questions from online tests and K-12 curriculum and classify them along the hierarchy of Bloom's taxonomy as shown above. We then used these questions as prompts to get GPT-3 answers, and search queries to get Google answers. We display the answers by GPT-3 and Google in Table 1. We then qualitatively analyzed the difference between the answers retrieved by GPT-3 and Google.

4 Results

In this section, we summarize the findings from our experiments.

First, we found that GPT-3 is generally a superior choice for tasks involving comprehension, application, analysis, synthesis, and evaluation. GPT-3 can produce human-like responses to questions, making it ideal for answering questions that require higher levels of thinking. Examples include:

Question: Do you agree with the idea that you are what you consume?

Answer by GPT-3: No, I do not agree with this idea. While it is true that what we consume can influence our bodies and our overall health, I believe that we are much more than what we consume. Our thoughts, feelings, beliefs, and experiences are just as much a part of who we are as what we consume.

Question: How is friendship an example of a relationship?

Answer by GPT-3: Friendship is an example of a relationship because it involves two or more people having an emotional connection. This connection can be based on mutual trust, respect, care, and loyalty. Friends are often able to share their thoughts and feelings with each other openly and honestly, support one another, and enjoy spending time together.

Table 1. Comparison of answers retrieved by Google and GPT-3 on each category of bloom's taxonomy.

Bloom's taxonomy	Questions: Prompts/Search Queries (3–5)	GPT-3 answer	Google answer	Notes
Knowledge (identification and recall of information)	What is Van Gogh's most famous art piece?	Vincent van Gogh's most famous art piece is arguably "The Starry Night," an oil on canvas painting from 1889 that depicts a swirling night sky over a small hillside village	A series of different photos painted by Van Gogh like "Sunflowers", "The Starry Night" did show up at the very top of the magazine.artland.com website, but there are also many other opinions on other websites	GPT-3 is able to state a direct opinion but google has opinions of many and photos to use if the user wants to make his or her own judgement
	What is $1/2\sqrt{34} + 8$?	$1/2\sqrt{34} + 8$ is equal to 9.4	10.9154759474	Calculations are wrong for GPT-3
Comprehension (organization and selection of facts and ideas)	Rephrase "All the members of a particular organization or population who are able to work, viewed collectively"	The workforce	Nothing	GPT-3 is great at generating answers to questions that need to be organized but Google is unable to provide any results or only provides related websites that contain the words searched
Application (use of facts, rules and principles)	Write a function, it takes a number array, and return a reversed array for Java e.g. {3,5,1} -> {1,5,3}	<pre>Public static int[] reverseArray(int[]) arr) { int n = arr.length; int[] revArr = new int[n]; for (int i = 0; i < n; i++) revArr[i] = arr[n - i - 1]; return revArr; }</pre>	Links to websites that teach you how to do a reverse function and teach you the idea behind it. It also provides code that reverse arrays, but is not exactly applicable to this set of question	GPT-3 provides the exact code to use to answer the question, but Google does not as the question might not have been one that someone has posted, however, Google is able to provide a rationale to how to do the question so the student can apply it themselves

(continued)

Table 1. (*continued*)

Bloom's taxonomy	Questions: Prompts/Search Queries (3–5)	GPT-3 answer	Google answer	Notes
	Correct this to standard English: She no went to the market	She did not go to the market	Gives a series of grammar rules/examples	GPT-3 can act as a grammar corrector for students, but Google does not have that function, so it can only give related websites
Analysis (separation of a whole into component parts)	Classify apple, celery, peach, noodles, and pizza according to different types of food	Fruit: Apple, Peach Vegetable: Celery Starch: Noodles Protein: Pizza	Definitions and classifications for different types of food	GPT-3 is able to do the task of classifying and identifying parts that the user has entered but google can only provide websites with words that are related
Synthesis (combination of ideas to form a new whole)	What solutions would you suggest for a child crying?	<ol style="list-style-type: none"> 1. Try to identify the source of the child's distress. Determine if they are hungry, tired, or need to go to the bathroom 2. Comfort the child with hugs and reassuring words 3. Distract the child by engaging them in an activity or offering a toy or book 4. Offer a snack or drink 5. If possible, take the child outside for some fresh air or a walk ... 	Articles about crying and how to make babies stop crying	GPT-3 is able to provide exact solutions and steps, but Google can only retrieve from what others wrote

(continued)

Second, we found that GPT-3 is incapable of answering complex mathematical questions. In multiple cases we investigated, GPT-3 generates incorrect answers to math questions. We consider this is because GPT-3 does not have the ability to perform complex computations as a calculator or math software does. On the other hand, Google

Table 1. (*continued*)

Bloom's taxonomy	Questions: Prompts/Search Queries (3–5)	GPT-3 answer	Google answer	Notes
Evaluation (development of opinions, judgments, or decisions)	Do you agree with the idea that you are what you consume?	No, I do not agree with this idea. While it is true that what we consume can have an effect on our bodies and our overall health, I believe that we are much more than what we consume. Our thoughts, feelings, beliefs, and experiences are just as much a part of who we are as what we consume	Papers written about this topic (all sorts of opinion)	GPT-3 can generate an opinion and provide context while there are many ideas written when searching on Google

is better at mathematical questions and quickly retrieving facts from the internet. We consider Google to be a better choice for answering factual or straightforward questions that fit into the “Knowledge” category in Bloom’s taxonomy.

Question: What is $1/2\sqrt{34} + 8$?

Answer by GPT-3: $1/2\sqrt{34} + 8$ is equal to 9.4

Correct answer: 10.9154759474.

Third, for detail-oriented questions that fall into the category of comprehension and application, e.g., writing a piece of code, or explaining a chemistry phenomenon, GPT-3 can provide an accurate answer whereas Google often retrieves a list of articles where the user must dig up and summarize the answers by themselves. Examples include “explain why the third ionization energy of potassium is lower than the first ionization energy of lithium in short”.

Furthermore, it’s also worth noting that when looking for reliable sources, such as news articles or reference papers, GPT-3 may not be the best choice as it cannot distinguish between credible sources and fake resources. Therefore, it is important to use GPT-3 with caution and cross-reference with other sources to ensure the accuracy and reliability of the information provided.

Question: What is a reference I can use to write an essay for medical reforms for COVID-19?

Answer by GPT-3: New York State Department of Health. (2020). COVID-19 Health Care Reform. Retrieved from https://www.health.ny.gov/press/releases/2020/2020-03-24_covid-19_health_reform.htm.

This is not a real website that can be accessed.

References

1. Cole Claybourn: ChatGPT in Classrooms: What to Know (2023). <https://www.usnews.com/education/best-high-schools/articles/chatgpt-in-classrooms-what-to-know>
2. Nguyen, N.: ChatGPT: a threat to education? (Opinion) (2022). <https://feedbackfruits.com/blog/chatgpt-a-threat-to-education-opinion>
3. How ChatGPT, other AI tools could change the way students learn. <https://www.theglobeandmail.com/canada/article-chatgpt-education-ai-technology/>
4. Krathwohl, D.R.: A revision of Bloom's taxonomy: an overview. Theory Pract. **41**(4), 212–218 (2002); GPT-3 Is Quietly Damaging Google Search
5. <https://analyticsindiamag.com/gpt-3-is-quietly-damaging-google-search/#:~:text=The%20key%20difference%20is%20that,and%20find%20the%20required%20information>
6. Sofia, P.: Google vs. ChatGPT: Here's what happened when I swapped services for a day (2022). <https://www.cnbc.com/2022/12/15/google-vs-chatgpt-what-happened-when-i-swapped-services-for-a-day.html>
7. Stock, L.: ChatGPT is changing education, AI experts say – but how? (2023) <https://www.dw.com/en/chatgpt-is-changing-education-ai-experts-say-but-how/a-64454752>



Citation Recommendation Employing Proximity-Based Heterogeneous Network Embeddings

Zafar Ali^{1(✉)}, Irfan Ullah², Pavlos Kefalas⁴, Nimbeshaho Thierry³,
Kalim Ul Haq¹, and Anupam Sarkar⁵

¹ School of Computer Science and Engineering, Southeast University, Nanjing, China
zafarali@seu.edu.cn

² Department of Computer Science, Shaheed Benazir Bhutto University, Sheringal, Pakistan

³ College of Information and Communication Engineering, Nanjing University of Posts and Telecommunication, Nanjing, China

⁴ Department of Informatics, Aristotle University, Thessaloniki, Greece

⁵ School of Information Science and Engineering, Southeast University, Nanjing, China

Abstract. The number of research papers is growing exponentially on the Web and digital libraries, which makes it a cumbersome chore to determine relevant research works. To address this problem, citation recommendation (CR) models have been proposed. Nevertheless, these CR models are limited in considering the semantic relations among network objects, e.g., authors, papers, tags, venues, and topics in the heterogeneous paper's network. Moreover, existing models do not consider the significance of proximity information between network nodes. Additionally, the current CR models face cold-start paper problems. To alleviate such problems, this work proposes a proximity-based heterogeneous network embedding (CR-PHNE) model that exploits semantic information of a network from node sequences using a probability-sensitive meta-structure-guided random walk method. Next, this information is given as input to deep neural networks to learn the latent representations of contributing nodes. Compared to its counterparts, the results produced by CR-PHNE over publicly available datasets bring 5% and 4% improvement regarding MAP and nDCG metrics, respectively. Further, the model demonstrates 7% and 3% improvements in terms of MAP and Recall@100 scores, respectively, to mitigate the cold-start paper problem compared to its counterparts.

Keywords: Deep Learning · Citation Recommendation · Cold-Start Problem · Heterogeneous Network Embedding

1 Introduction

The colossal size and ever-increasing number of pages on the Web has made finding relevant items a daunting and challenging task. To address this information

overload and enable users find relevant items easily, recommender systems have been introduced that produce relevant recommendations by considering user needs and preferences [1]. Such recommendation solutions are also essential in the scholarly domain to help researchers identify relevant research publications and assess their research ideas and findings. In this regard, several latest CR models [2–5] have been published, which adopt various filtering methods, such as content-based (CB) [6, 7], collaborative (CF) [3, 9], and graph-based [2, 5, 8, 10]. The content-based CR models exploit the publication’s descriptions, features, user profiles, and their history, otherwise they face cold-start issue [11, 12]. The collaborative CR models the users’ as well as their friends’ feedback and ratings; however, if the rating matrix is sparse, then the resulting recommendations are poor and barely address the user needs [4]. The graph-based CR models [2, 8, 10, 13] handles data sparsity by considering the relationships among the bibliographic network objects. However, these models [8, 13] treat recommendation as a link prediction process on graphs, thus giving higher weights to outdated and old node [4]. To address these problems, recent studies [5, 14, 15] have employed homogeneous NRL methods such as LINE [16] and Deepwalk [31] in producing citation recommendations. Yet, the multiplicity and heterogeneity in bibliographic networks are ignored by the current CR models. Among these, the heterogeneity issue was addressed by heterogeneous NRL-based models [2, 18].

The NRL-based models addressed the heterogeneity problem linked with previous homogeneous embedding models. However, these models are still limited in considering useful relations among the network objects. For instance, if we take an example of a heterogeneous information network, i.e., DBLP, there exist multiple relations among the network objects. That is, authors can establish relations with papers based on citation proximity, authorship relation, topical relevance, share tags, share-venue, etc. These relations establish a view of the DBLP network; therefore, it is indispensable to exploit all such relations among network objects. However, recent NRL-based citation recommendation models [2, 18] do not consider such semantics and lack the ability to capture researchers’ preference dynamics, and therefore do not generate justifiable recommendations. Besides, these models utilized limited information regarding the proximity among network objects. Thus, they were unable to capture context-preserving representations of nodes.

In light of the aforementioned intuitions, we propose CR-PHNE, a network embedding-based CR model that encodes different relations among the network objects (i.e., topics, authors, papers, venues, and tags) into a low-dimensional embedding space for personalized recommendations. At the encoding layer, a specific probability is allocated to each node for generating node sequences, and the relations among important nodes are strengthened using a mutual proximity-aware mechanism. This way, we consider various node types, their relations, and the awareness of mutual proximity among them. In addition, the auxiliary information is considering in addressing the cold-start problem. Following are the research contributions:

- Presenting a **P**roximity-based **H**eterogeneous **N**etwork **E**mbedding approach, which adequately employs the semantic relationships between network objects and alleviates the cold-start paper problems.
- We introduce a meta-path selection method that exploits meaningful relations and semantics between the objects of the heterogeneous information network.
- We propose a novel network embedding-based CR model that learns the papers' and scholars' latent representations by considering multiple relations in a continuous vector space for personalized recommendations.
- We conducted detailed experimentation using real-world datasets to analyze the efficacy of our model compared to other counterparts with respect to MAP, nDCG, and recall metrics.

The organization of the rest of the paper is such that Sect. 2 presents review of the literature. Section 3 explains the preliminaries and problem statement. Then, the methodology and technical details are discussed in Sects. 4 and 5. Section 6 describes the experimental setup and performance comparison of the proposed model compared to other counterparts. Last, Sect. 7 completes this work by highlighting the findings of this research and future research directions.

2 Related Studies

Numerous graph-based citation recommendation models including [14, 29, 35] have been introduced in the literature. They employ additional connections between network nodes. Nevertheless, these techniques assign higher weights to old and outdated network nodes since they consider recommendations as a problem of link prediction [1]. To address this issue, HNPR [29] uses random walks for edge traversal in heterogeneous information networks and applies natural language models to correlate word representations for citation recommendations. More advanced network embedding [28] and graph embedding [27] techniques have been introduced lately, to encode graphs, nodes, graphs, or a network into an embedding space. Therefore, different techniques such as [5, 14, 24, 30, 38] employed such embedding techniques to produce citation recommendations. Yet, these models are unable to account for the diversity and heterogeneity of citation networks.

For instance, Gupta and Varma [14] employed Doc2vec [22] and DeepWalk [31] in learning embeddings of the content and structure of scholarly publications, respectively. The proposed model recommend publications by considering the similarities among the learned representations. In a similar way, VOPRec [30] integrated content-based vectors and structure-based embeddings using the Paper2vec [32] and the Struct2vec [33] embedding methods. The bibliographic network representation (BNR) [2] uses Node2Vec [19] for exploiting semantic relations among objects, including authors, papers, content, venue, etc., of the heterogeneous bibliographic network (HBN). Similarly, CR-HBNE [34] uses Node2vec in learning the representations of bibliographic network objects, such as authors, papers, abstracts, fields of study, topics, and publication time.

Generative adversarial networks have shown promising results dealing with sparse networks since they learn the data distribution from a limited amount of data, and exploit prior distribution for introducing regularization to the latent codes [41]. They need no approximation for training as differential networks sufficient to train them end-to-end. To this end, VCGAN [42] adopted an NRL-based model using GAN. It produces content-based graph embedding by exploiting the network structure and content related to nodes. The learned node vectors are employed to generate the final recommendations for the user's provided query. Likewise, GCR-GAN exploited GAN with SPECTER [25] content embedding model to learn network-based embedding of nodes in the heterogeneous bibliographic network holding papers, topics, content, and authors. The learned embedding is utilized to rank manuscripts close to the user's requested paper.

The Citation Tendency Random Walk (CIRec) [38] is a weighted heterogeneous network embedding (HEN) model that considers the relations among publications and their references using a weighted random walk. It computes the citation tendency using the relations among publications and citation by exploiting authors and terms. It uses skip-grams for producing the vector representations of publications. The cosine similarity between the paper embedding vectors is computed for recommendations. Similarly, PR-HNE [23] exploited a HEN method. The model employed six relation graphs established between papers, authors, topics, venues, and labels using the network embedding method, i.e., LINE [16], to learn node embedding for making final recommendations. The SCR-NTR [39] model exploits network and textual data in learning the text and network representations using BERT and HeGAN [40] models, respectively. The aforementioned NRL-based models have tackled the heterogeneity challenge faced by other counterparts [14, 35]. Yet, they cannot fully utilize the relationships connected to the diverse information network objects. Moreover, these models ignore the significance of relations established between the nodes of the networks. Finally, the proximity information, i.e., two nodes are closer than others in the network, is not considered by these models.

3 Preliminaries and Problem Description

This section presents the preliminaries in understanding the proposed heterogeneous network and mathematically define the problem faced. For reasons of simplicity, we provide a notation table with all symbols used for the rest of the paper.

Definition 1. (*Heterogeneous Information Graph (HIG)*) is denoted as $G = (V, E)$ with two mapping functions for nodes and edges as $\phi: V \rightarrow Z$ and $\psi: E \rightarrow Y$, respectively. Also, each node $v \in V$ and $e \in E$ represents a distinguished node and edge type such as the schema is represented as $S_G = (S_V, S_E)$. Thus, given $|Z| + |Y| > 2$ we have a Heterogeneous Information Graph otherwise we have a homogeneous one. An example of two types of edge types can be seen in Fig. 1a.

Definition 2. (*Meta-path* Γ) is a sequence of transitions over the nodes of a Multi-view heterogeneous graph with y edge types and it is defined as $\Gamma: V_1 \xrightarrow{y_1} V_2 \xrightarrow{y_2} \dots \xrightarrow{y_k} V_t$, with t been the length of a meta-path. Meta-paths are used to capture the semantic relations among the nodes in the graph.

The notion that nodes can be linked following paths of different edge types is illustrated in Fig. 1, where meta-path $\Gamma_1 = (P1 - T - P2)$ denotes that paper P1 is related to P2 sharing similar topic T. An interesting feature of meta-paths is the bidirectionality which captures both the simple and biased semantics [36].

Thus, in our previous example we would also have $\Gamma' : V_t \xrightarrow{y_k^{-1}} V_2 \xrightarrow{y_2^{-1}} \dots \xrightarrow{y_1^{-1}} V_1$. A major drawback with meta-paths is that they ignore complex relations between nodes, as illustrated in Γ_1 and Γ_2 in Fig. 1b where two authors are linked with more than one edge type. To address this issue, meta-structures have been introduced.

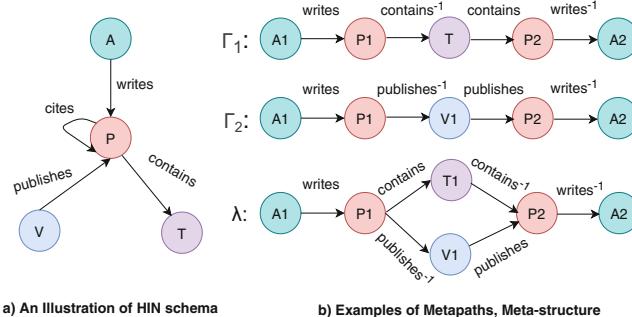


Fig. 1. Examples for HIG Schema, Meta-Paths, and Meta-Structure

Definition 3. (*Meta-structure*) is a directed acyclic graph over a HIG schema $S_G = (S_V, S_E)$ starting from a source node v_s and terminating at a node v_t which is denoted as $\Lambda = (Z, Y, v_s, v_t)$. An example of meta-structure is given in Fig. 1b where the two authors are related if they publish papers (1) in the same venue with (2) on the same research topic. Such complex semantics from meta-structures will make results robust and reliable.

Definition 4. (*Heterogeneous Graph Embeddings*) transforms nodes $v \in V$ of G into an embedding space with $d \leq |V|$ by learning a mapping function $\Psi: V \rightarrow \mathbb{R}^d$. Nodes that share similar semantics in G are kept closer in the vector space and thus have similar representations.

Problem definition 1 Given an author's a provided seed paper and the heterogeneous information graph $G = (V, E)$ with author, papers, venues, topics, and tags, we aim to explore various relation types and recommend the top k related papers to the target author.

4 Proposed Model

This section presents the proposed model with architectural details in Fig. 2. The model consists of a two-step procedure. First, we extract the semantics using Meta-structure guided random walk method, which generates node sequences based on first-order and second-order proximity between the nodes [16, 26]. Then, the generated sequences are fed into an auto-encoder to generate their low-dimensional vector representations of nodes.

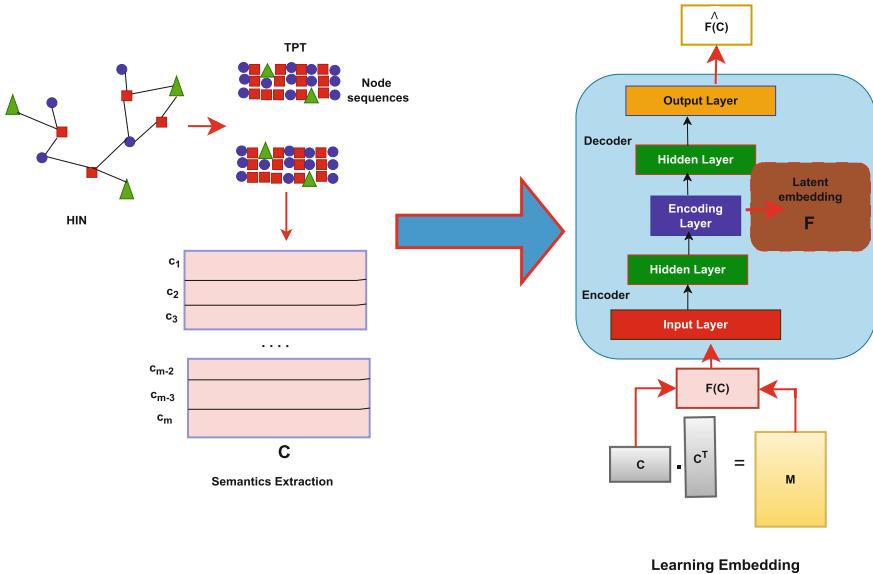


Fig. 2. The Architecture of the Proposed CR-PHNE Model

4.1 Meta-Structure Guided Random Walk

Given a HIG $G = (V, E)$ we produce the meta-structure $\Lambda = (Z, Y, v_s, v_t)$ as described in Definition 3 following [19]. In such heterogeneous graphs, there are multiple edge types starting from a target node, which may lead to an unequal 1-hop transition probability distribution. To avoid this we utilize the meta-structure-aware random walk method using the ratio of the node types number to all nodes linked to the target node directly as described below.

$$P(v_i | v_{i-1}; \Lambda) = \frac{T_{v_i}^Y}{\sum_{j=0}^{T_\Lambda(v_{i-1})} T_{v_j}^Y} \times \frac{1}{|\{u | (v_{i-1}, u) \in E, z(v_i) = z(u)\}|} \quad (1)$$

where $T_\Lambda(v_{i-1})$ represents the total number of edge types starting from node v_{i-1} , $T_{v_i}^Y$ is the number of v type nodes, whereas $|\{u | (v_{i-1}, u) \in E, z(v_i) = z(u)\}|$ denotes the total number of v_{i-1} 's neighbors.

$\frac{N(T_{vi})}{\sum_{j=0}^{T_G(v_{i-1})} N(T_{vj})}$ represents the ratio of nodes belong to type v_i and the total number of vertices directly linked with v_{i-1} . At each step, the approach works in such a way that the total number of edges is counted concerning an edge type that fulfills the criteria of meta-structure. Then our model computes the probability of edge type which is directly connected to vertex v_{i-1} . There is a possibility that a qualified type can have a transition probability between 0 and 1 is chosen. In this case, the type with more edges is selected easily. This way, it walks to a node in the network through one edge of the selected/chosen type in a random manner (randomly). When we construct sequences of nodes using the meta-structure-guided random walk method, a co-occurrence matrix is used to represent the relationship between nodes. In the co-occurrence matrix, $C \in \mathbb{R}^{n_t \times n_s}$, n_t denotes the total number of vertices corresponding to a specific type, whereas n_s represents the total number of vertices that occur within node sequences. More specifically, $c_i \in \mathbb{R}^{n_s}$ represents the occurrence of vertices that start from vertex i , where we have $c_{i,j} = 1$ when there exists relation between node j and node i , otherwise $c_{i,j} = 0$. If we take a sequence from the DBLP dataset i.e., $a_2 - p_2 - v_2 - p_3 - a_3$, the entries in the matrix C for vertex pairs $(a_2, p_2), (a_2, v_2), (a_2, p_3), (a_2, a_3)$ are set to 1. It reveals that the matrix preserves first-order proximity among nodes in the HIN network. To capture second-order proximity, we introduce another matrix $M \in \mathbb{R}^{n_t \times n_t}$ that captures both first-order and second-order proximity among nodes in matrix C by computing CC^T , such that $0 \leq i < n_t$, and $0 \leq j < n_t$. For instance, $m_{i,j}$ in the matrix M is computed as $m_{i,j} = c_i \cdot c_j^T$. This way, the model can capture second-order proximity between nodes v_i and v_j . There can be situations, where nodes such as v_i and v_j in matrix C will have no direct connections and thus $c_{i,j} = 0$, however, these nodes will have second-order proximity (i.e., they are linked to the same node v_l). In such cases, matrix M can preserve their second-order proximity by assigning $m_{i,j}$ to value 1. Thus, the proposed model can exploit meaningful and robust semantics in the HIN network.

5 Learning Embeddings

Given a meta-structure-guided random walk, the next step is to encode the participating nodes in the HIN into the low-dimensional space and preserve the aforementioned semantic relations among them. To preserve the aforementioned proximity relations among nodes in the HIN network, we introduce a neighborhood-enhanced learning mechanism. Our model is based on the concept of auto-encoders, which employ an encoder and decoder to learn node embeddings. In contrast to conventional auto-encoders, the proposed method employs the neighbor representation of a vertex as an input to the encoder to produce the vector representation of the concerned node. In particular, c_i represents the vector representation of vertex v_i , whereas $F(v_i)$ denotes the aggregation of neighbors' nodes corresponding to node v_i in the HIN network. The proposed model aims to optimize the following objective.

$$O_p = \sum_{i=1}^{n_s} \left\| \hat{F}(v_i) - x_i \right\| \quad (2)$$

where $\hat{F}(v_i)$ represents the output generated by the decoder part of the architecture, and $F(\cdot)$ denotes a weighted aggregation strategy used to incorporate the proximity among network objects. More specifically, as mentioned in the above sections our model first constructs a co-occurrence matrix $C \in \mathbb{R}^{n_t \times n_s}$ by using the sequences generated using a meta-structure guided approach. To hold the first-order and second-order proximity among nodes, the matrix $M = CC^T \in \mathbb{R}^{n_t \times n_t}$ is used. To compute the mean of neighbors concerning a node v_i , the model uses the following aggregation function:

$$\frac{1}{|N(i)|} \sum_{j \in N(i)} m_{i,j} c_j \quad (3)$$

where $m_{i,j}$ represents the significance of relation/proximity between two vertices v_i and v_j . While $N(i)$ represents the neighbors of node v_i . Additionally, during the encoding and decoding process, the model uses K hidden layers defined as follows.

$$h_i^{(l_1)} = \sigma(W^{l_1} c_i + b^{(l_1)}) \quad (4)$$

$$h_i^{l_k} = \sigma(W^{l_k} h_i^{l_{k-1}} + b^{l_k}), k = 2, \dots, L \quad (5)$$

Here W and b represent hyper-parameters weight matrix and bias vector respectively. Whereas σ denotes the non-linear activation that can be sigmoid, softmax and relu, etc. On the other hand, L is used for the number of layers corresponding to the encoder and decoder. Using encoding layers, we obtain $H \in \mathbb{R}^{m \times d}$, where d denotes the dimensionality of terminal embedding vectors and m denotes the total number of embedded vertices. This way, our model minimizes the difference between the vector representations of a node and its corresponding neighbors' representation and generates improved and semantic-aware node representations compared to traditional auto-encoders.

5.1 Citation Recommendations

The model exploits various relations types in the HIN network and learns the embeddings of participating nodes in the network. These embeddings are used in producing the final recommendation list. In particular, the model aims to recommend top k citations that meet the researcher's research interests.

$$P_{v_c, v_t} = \alpha(\vec{v}_t \vec{v}_c^T) + \beta(\vec{v}_{t_a} \vec{v}_{c_a}^T) + \varphi(\vec{v}_{t_t} \vec{v}_{c_t}^T) + \mu(\vec{v}_{t_v} \vec{v}_{c_v}^T) + \lambda(\vec{v}_{t_{tg}} \vec{v}_{c_{tg}}) \quad (6)$$

where, \vec{v}_t , \vec{v}_{t_a} , \vec{v}_{t_t} , \vec{v}_{t_v} , and $\vec{v}_{t_{tg}}$ represent the embedding vectors of target's paper, target's paper author, target's paper topic, target's paper venue, and target's

paper tags. Moreover, \vec{v}_c , \vec{v}_{ca} , \vec{v}_{ct} , \vec{v}_{cv} , and \vec{v}_{ctg} are used to denote the embedding representations of seed's paper, seed paper's author, seed paper's topic, venue, and seed paper's tags, respectively. Where α , β , φ , μ , and λ represent the hyper-parameters used for tuning the significance of participating relation types. This way, the model selects the top k research papers for a user.

Table 1. Meta-Paths for Different Relation Types

Metapaths	Explanation and semantic meaning
PAP	Represents that an author has written two papers
PA	Represents the collaboration of two authors for a research paper
PVP	Two papers share the same venue
PTP	Two papers contain the same research topic
PLP	Two papers share the same tag/label
APTPA	The papers of two authors share the same research topic
APVPA	The papers of two authors share the same research venue

6 Experimental Study

This section illustrates the experimental details including the models used as baselines for comparison, datasets, and evaluation metrics.

Table 2. The Specifications of Datasets Used

Datasets	Authors	Papers	Venues	Citation relations	Tags
DBLP-V13	3,50,123	5,354,309	18,145	48,227,950	1,755
DBLP-V12	245,201	3,520,130	16,205	25,022,312	11,133

6.1 The Evaluation Methodology

To analyze the experimental results of the baseline models and our proposed model, we utilize two large-size datasets, viz., the DBLP-V12¹ and the DBLP-V13.² Table 2 shows the statistics of these datasets. DBLP-V13 is relatively a new and updated version of DBLP-V12. Both these datasets provide access to information regarding research papers, authors, relevant content, keywords/tags, citation relations, and publishing venues. The abstract (content) of the papers are available as indexed abstracts, therefore we utilized regular expressions to extract this information from the JSON files.

¹ <https://www.aminer.cn/citation>.

² <https://www.aminer.cn/citation>.

To examine the results of models, we divide each dataset into two sets namely the training set \mathcal{Y}^t holding 80% and the test set \mathcal{Y}^p having 20% of the data. It is important to note that $\mathcal{Y} = \mathcal{Y}^t \cup \mathcal{Y}^p$ and $\mathcal{Y}^t \cap \mathcal{Y}^p = \emptyset$. For a seed manuscript, each model process Top@ k predictions using the \mathcal{Y}^t . We declare the recommendation to be relevant/related if the model recommends the ground truth in the Top@ k , otherwise irrelevant.

The metrics adopted for results evaluation include normalized Discounted Cumulative Gain (nDCG), Mean Average Precision (MAP), and Recall. A recent study shows that these are the commonly used metrics in recommender systems [1, 20, 21]. The **recall** metric evaluates the predictions of systems utilizing the percentage of relevant results offered in the top- k list of recommendations, where we set $k = \{20, 40, 60, 80, 100\}$.

$$\text{Recall} = \frac{1}{Q} \sum_{j=1}^Q \frac{R_p \cap T_p}{T_p} \quad (7)$$

where, Q denotes all target research papers, R_p represents the *top-k* suggestions offered against the seed publication p . The second metric, **MAP**, assesses the importance of a model by examining if relevant papers are suggested in top- k or not. It gives a penalty to the errors that occur at higher positions in the top@ k .

$$\text{AP}@k = \frac{1}{GTP} \sum_{i=1}^k \frac{TP_{seen}}{i} \quad (8)$$

where TP_{seen} represents total true positives appeared till k . We select the first ten values as the cut-off value for AP, i.e., AP@10. Finally, the third metric, **nDCG**, evaluates the rank among the *top-k* true relevant publications that the model recommends [23].

6.2 Baseline Models Used for Performance Analysis

This section presents the baseline models that we use to compare the experimental results of the proposed model. Further details are given below.

- **Specter** [25] is a pre-trained content embedding model that uses a citation-informed transformer to generate citation recommendations. It enriches the vector representations of documents learned using SciBERT [37] by exploiting the citation relations as relatedness signals among documents. To generate recommendations, we use cosine similarity between the embedding of research papers. The final paper embedding dimension is set to 768.
- **CCA** [14] is a representation learning model that learns content-based and network-based embedding using the Doc2Vec and DeepWalk [17] methods, respectively. The final predictions are made by computing the cosine similarity between the learned vectors. The dimensionality of Doc2vec and Deepwalk is set to 300 and 64, respectively.

- **NNRank** [7] utilizes a deep neural network to exploit the embeddings of abstract, keywords, authors, venues, and key phrases to provide top-k results for a query manuscript employing the cosine between the learned embeddings vectors of aforementioned objects. The model generates the best results when we choose batch size = 512, learning rate = 0.001, length of abstract text = 500, nearest neighbors = 5, and the dimensionality of dense layers = 75.
- **BNR** [2] uses Node2vec [19] embedding model to learn to embed network objects, such as authors, papers, venues, and content. Then, cosine similarity between the learned embeddings is calculated to make final predictions. The parameters were set as 128 embedding dimensions with context size as 10, walks per vertex as 80, tuning parameter as $\beta = 0.7$, the return parameter as $q = 2$, and the in-out parameter as $p = 1$.
- **PR-HNE** [23] adopts a weighted probabilistic network embedding model using LINE [16] method to learn the representations of network objects including topics, authors, papers, labels, and venues. To explore relations in the entire network, it uses first- and second-order proximity concepts. Once the model learns embeddings, the top- k publications are recommended in relation to the seed paper by computing the dot product between these vectors. Following the original paper, we set the default parameters.

6.3 Comparative Performance Analysis

This section analyzes and compares the experimental results of the proposed model and its counterpart baselines. The main reason is that there is not only text information in the knowledge graph, but also the author, tags, topics, venue, and relationship information between these participating nodes. Such information can enrich the semantics and associations of nodes, and enhance the expressive ability of nodes. Content-based recommendation models do not make full use of auxiliary information, resulting in insufficiently personalized recommendation results. CCA performed relatively better compared to SPECTER because it exploits network structure along with the content of research papers to produce recommendations. However, it employs the shallow doc2vec method to

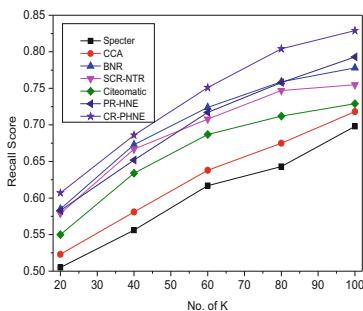


Fig. 3. Recall using DBLP-V12 Dataset

learn paper content-based embedding and considers no auxiliary information factors. Citeomatic outperformed CCA since it exploits side information including paper content, citation relations, tags, and author information. BNR outperformed CCA and Citeomatic, as it exploits semantic relations between the vertices (author, paper, content, venue, etc.) of the heterogeneous bibliographic network using the Node2vec network embedding method. However, these models have been outperformed by PR-HNE, which utilizes a weighted network embedding method to pay importance to various information networks in making final predictions. Thus, PR-HNE produces the second-best MAP, nDCG, and recall scores.

Our model CR-PHNE outperformed all the aforementioned baselines regarding nDCG, MAP, and recall on the DBLP-V12 and DPLP-V13 datasets as depicted in Tables 3, 4, Figs. 3, and 4. It is noticeable that the CR-PHNE has achieved nearly 4.3%, 3.5%, and 3.56% better results in terms of these metrics compared to the PR-HNE model using the DBLP-V13 dataset. By looking at the recall scores, CR-PHNE produces improves results over the baselines. This demonstrates the robustness and stability in the results produced by CR-PHNE. The results depicted in Tables 3 and 4 exhibit that CR-PHNE has improved NDCG scores by approximately 3.5% related the second-best model, i.e., PR-HNE, which shows that our model produces better-ranked predictions in the top-k results. This is attributed to the fact that CR-PHNE captures graph structure semantics to generate semantic-preserving entity embedding using Meta-structure guided random walk method. Moreover, the auxiliary information used, including author information, citation relations, tags, and topical relevance and venue information are helpful in capturing the preferences of the scholars in producing personalized results.

To summarize, CR-PHNE maximizes the impact of node proximity information and captures the heterogeneous quantitative details of various types of nodes. The best performance of the proposed model reveals that making efficient use of node-to-node proximity in a network embedding framework results in high-quality embedding vectors and thus can be justified use in recommending citations.

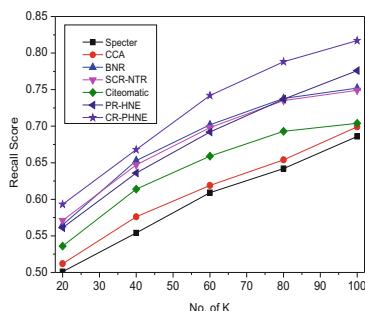


Fig. 4. Recall using DBLP-V13 Dataset

Table 3. Comparison of the Models based on the DBLP-V12 Dataset

Model	MAP	nDCG	Re@20	Re@40	Re@60	Re@80	Re@100
Specter	0.253	0.295	0.506	0.557	0.618	0.644	0.699
CCA	0.269	0.309	0.524	0.582	0.639	0.676	0.719
BNR	0.387	0.478	0.586	0.684	0.725	0.759	0.779
SCR-NTR	0.527	0.588	0.579	0.668	0.709	0.748	0.756
Citeomatic	0.527	0.595	0.552	0.635	0.687	0.713	0.729
PRi-HNE	0.579	0.637	0.584	0.653	0.718	0.758	0.794
CR-PHNE	0.626	0.673	0.608	0.687	0.752	0.805	0.829

6.4 Impact of Adding Different Relation Types

We conduct an ablation experiment to analyze the consequence of adding various relation types on the recommendation results. Table 5 depicts the results produced by adding different relation types. Particularly, we analyzed the significance of adding authors' information, venue, citations, tags, and topical relevance. For this purpose, we compared the results of the variants of the proposed model. These include **CR-PHNE_{CV}**, which employs citation relations and venue information. **CR-PHNE_{CVA}** extends the previous version by adding author relation to enhance results. **CR-PHNE_{CVAT}** incorporates tags information. **CR-PHNE_{Prop}** incorporates all relationships including the topics.

Table 4. Comparison of the Models based on the DBLP-V13 Dataset

Model	MAP	nDCG	Re@20	Re@40	Re@60	Re@80	Re@100
Specter	0.246	0.281	0.501	0.554	0.609	0.642	0.686
CCA	0.268	0.305	0.512	0.576	0.619	0.654	0.699
BNR	0.381	0.462	0.565	0.653	0.702	0.738	0.752
SCR-NTR	0.508	0.566	0.571	0.647	0.698	0.735	0.749
Citeomatic	0.526	0.571	0.536	0.614	0.659	0.693	0.704
PRi-HNE	0.565	0.618	0.561	0.636	0.692	0.737	0.776
CR-PHNE	0.618	0.660	0.593	0.668	0.742	0.788	0.817

The results exhibit that **CR-PHNE_{CV}** generated imprecise results related to other variants. The reason is that it does not employ other personalized information including author, topics, and tags. Also, the second variant of the proposed model, i.e., **CR-PHNE_{CVA}** shows that integrating author information has greatly boosted the performance of the model. On the contrary, the tags/keywords have relatively less impact in personalizing citation recommendations. The results of the final variant **CR-PHNE_{Prop}** exhibit that topical

relevance plays a vital role in producing precise results. To summarize, these findings suggest that incorporating authors information and paper topics help model capture researcher's preference dynamics and gain improved results.

Table 5. The Significance of Employing Different Relation Types

Datasets	Model	nDCG	MAP	Re@20	Re@40	Re@60	Re@80	Re@100
DBLP-V12	CR-PHNE_{CV}	0.585	0.543	0.536	0.613	0.654	0.692	0.728
	CR-PHNE_{CVA}	0.603	0.579	0.598	0.664	0.705	0.732	0.754
	CR-PHNE_{CVAT}	0.646	0.597	0.602	0.678	0.725	0.753	0.773
	CR-PHNE_{Prop}	0.626	0.673	0.608	0.687	0.752	0.805	0.829
DBLP-V13	CR-PHNE_{CV}	0.603	0.564	0.558	0.626	0.669	0.715	0.747
	CR-PHNE_{CVA}	0.637	0.598	0.562	0.669	0.704	0.741	0.768
	CR-PHNE_{CVAT}	0.661	0.626	0.579	0.683	0.724	0.765	0.798
	CR-PHNE _{Prop}	0.694	0.647	0.621	0.692	0.756	0.813	0.841

6.5 Dealing with Cold Start Papers

Cold-start papers problem arises when the system does not have access to sufficient information regarding papers. In such cases, the system faces difficulties in making accurate predictions. If the system cannot get access to the content, keywords, and the author's information of papers, then we declare it as cold-start. To investigate the results of our model compared to other baselines, we choose 21,445 research papers as cold-start articles. The results shown in Table 6 exhibit that although with missing data, CR-PHNE can benefit from the side information to generate satisfactory results. Particularly, if the author information of a paper is missing, CR-PHNE utilizes its tags/labels, and topic. The results exhibit that the proposed model gained 6.5% and 2.1% improved MAP and Rec@100 scores related to the second-best performer. This significant gain shows that CR-PHNE makes use of additional relations in the heterogeneous network such as citations, authorship, topics, and tags in producing recommendations. These factors enable CR-PHNE to produce better results.

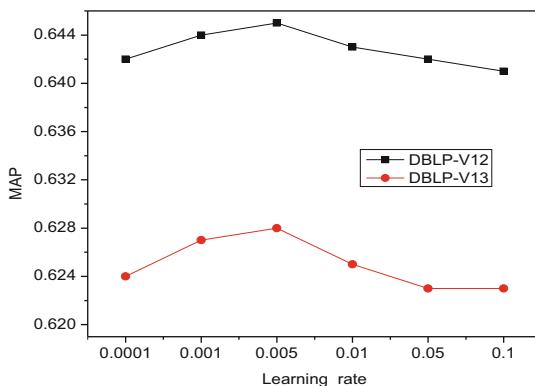
6.6 Parameters Impact

We present the details of parameters utilized by our model and their consequence on the results produced. That is, we check the impact of the learning rate, nodes embedding dimension d , and the number of topics. For analyzing the learning rate, we provide MAP results produced using the DBLP-V12 and DBLP-V13 datasets depicted in Fig. 5. We can see that CR-PHNE produces insignificant results employing a learning rate of 0.1, which exhibits the poor convergence capability of the model on a high learning rate. On the contrary, the model converges slowly when we employ a small value of 0.0001. Moreover, after employing

Table 6. Results Analysis w.r.t Cold-Start Papers

Model	nDCG	MAP	Re@20	Re@40	Re@60	Re@80	Re@100
Doc2Vec	0.258	0.224	0.426	0.477	0.526	0.554	0.585
CCA	0.285	0.248	0.484	0.553	0.599	0.635	0.658
BNR	0.392	0.327	0.513	0.595	0.633	0.657	0.685
SCR-NTR	0.557	0.489	0.561	0.659	0.692	0.736	0.769
NNiRank	0.552	0.466	0.524	0.605	0.646	0.675	0.709
PR-HNE	0.566	0.506	0.515	0.596	0.637	0.663	0.697
CR-PHNE	0.658	0.575	0.575	0.684	0.725	0.766	0.796

a fine-search mechanism, it is discovered that 0.001 yields the best MAP results using both the DBLP datasets. Finally, we analyze the model’s performance for the embedding dimension. We can notice in Fig. 6 that the model gains the highest results using values 120 for the DBLP-V12 and 140 on the DBLP-V13 datasets. This research used LDA for identifying each paper’s research topic. The abstract was extracted as it best reflects the topic of the article. In this regard, it is essential to determine the number of topics before identifying the topic of the paper. Therefore, the coherence indicator was used to understand the impact of the number of topics ranging between 1 and 50. A higher consistency in results was observed by keeping the number of subjects as 25 and 26 in the two datasets. Therefore, the number of topics was set to 27 and 28 separately. Finally, for tuning parameters viz., α , β , φ , μ , and λ , we set such values whose sum equals 1. We noticed that our model yields best results when we set $\alpha = 0.2$, $\beta = 0.3$, $\varphi = 0.2$, $\mu = 0.1$, and $\lambda = 0.2$ on both the datasets. It reveals that author information and topics play a significant role personalizing results, while venue information has little impact. Metapaths used have been shown in Table 1.

**Fig. 5.** The Influence of Learning Rates on the DBLP-V12 and DBLP-V13 Datasets

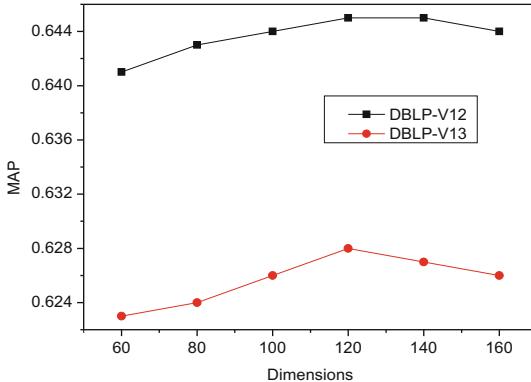


Fig. 6. The Dimensions of Embedding Vectors on the DBLP-V12 and DBLP-V13 Datasets

7 Conclusion and Future Directions

The existing citation recommendation models have proposed different solutions and methodologies to make personalized recommendations. Nevertheless, these are deficient in better exploiting researchers' interests and capturing salient information factors and meaningful relations in heterogeneous citation networks to produce justifiable results. These models have issues namely lack of personalization and cold-start problems. To alleviate such problems, the proposed model employs a graph embedding method to exploit relevant semantics from network structure and content of papers to make personalized and semantic-preserving recommendations. The findings conclude that a context-preserving embedding can be produced by integrating a proximity-based optimized deep neural network with probability-sensitive random walks and applying it to a heterogeneous information network. Moreover, our model can capture rich semantics for producing context-aware citation recommendations by employing meta-structure-guided random walks that exploit auxiliary data related to the graph nodes viz., authors, tags, papers, and topics. Furthermore, using such factors helps address the cold-start problem faced by other state-of-the-art baselines.

In the future, we plan to investigate the consequence of utilizing dynamic graph representation learning methods on the results of the recommendation model. In addition, the relations explored in this paper are not the only ones; other factors including the affiliation of authors, their positions, the geographical distance among them, and the quality of their previous research works may further augment the prediction quality as well as the performance of the citation recommendation model. We plan to consider exploring these factors in further research.

Acknowledgments. This research work is supported and funded by the NSFC under grant no. 6509009704.

References

1. Ali, Z., Qi, G., Kefalas, P., Abro, W., Ali, B.: A graph-based taxonomy of citation recommendation models. *Artif. Intell. Rev.* 1–44 (2020)
2. Cai, X., Zheng, Y., Yang, L., Dai, T., Guo, L.: Bibliographic network representation based personalized citation recommendation. *IEEE Access* **7**, 457–467 (2019)
3. Bansal, T., Belanger, D., McCallum, A.: Ask the GRU: multi-task learning for deep text recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems (RecSys), pp. 107–114 (2016)
4. Son, J., Kim, S.: Academic paper recommender system using multilevel simultaneous citation networks. *Decis. Support Syst.* **105**, 24–33 (2017)
5. Cai, X., Han, J., Li, W., Zhang, R., Pan, S., Yang, L.: A three-layered mutually reinforced model for personalized citation recommendation. *Trans. Neural Netw. Learn. Syst.* **29**, 6026–6037 (2018); 12
6. Sun, J., Ma, J., Liu, Z., Miao, Y.: Leveraging content and connections for scientific article recommendation in social computing contexts. *Comput. J.* **57**, 1331–1342 (2014)
7. Bhagavatula, C., Feldman, S., Power, R., Ammar, W.: Content-based citation recommendation. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 238–251 (2018)
8. Chakraborty, T., Modani, N., Narayananam, R., Nagar, S.: DiSCern: a diversified citation recommendation system for scientific queries. In: 31st IEEE International Conference on Data Engineering, (ICDE) 2015, Seoul, South Korea, April 13–17, 2015, pp. 555–566 (2015); 4,13
9. Wang, H., Li, W.: Relational collaborative topic regression for recommender systems. *Trans. Knowl. Data Eng. (TKDE)* **27**, 1343–1355 (2015)
10. Yang, L., Zheng, Y., Cai, X., Dai, H., Mu, D., Guo, L., Dai, T.: A LSTM based model for personalized context-aware citation recommendation. *IEEE Access* **6**, 59618–59627 (2018)
11. Christoforidis, G., Kefalas, P., Papadopoulos, A., Manolopoulos, Y.: Recommendation of points-of-interest using graph embeddings. In: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), pp. 31–40 (2018)
12. Christoforidis, G., Kefalas, P., Papadopoulos, A., Manolopoulos, Y.: Recommendation of points-of-interest using graph embeddings. In: 5th IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2018, Turin, Italy, October 1–3, 2018, pp. 31–40 (2018)
13. Tian, G., Jing, L.: Recommending scientific articles using Bi-relational graph-based iterative RWR. In: Proceedings of the 7th ACM Conference on Recommender Systems (RecSys), pp. 399–402 (2013)
14. Gupta, S., Varma, V.: Scientific article recommendation by using distributed representations of text and graph. In: Proceedings of the 26th International Conference on World Wide Web Companion, pp. 1267–1268 (2017)
15. Kobayashi, Y., Shimbo, M., Matsumoto, Y.: Citation recommendation using distributed representation of discourse facets in scientific articles. In: Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries, pp. 243–251 (2018)
16. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: LINE: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web (WWW), pp. 1067–1077 (2015)

17. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710 (2014)
18. Jiang, Z., Yin, Y., Gao, L., Lu, Y., Liu, X.: Cross-language citation recommendation via hierarchical representation learning on heterogeneous graph. In: The 41st International ACM SIGIR Conference on Research , pp. 635–644 (2018)
19. Grover, A., Leskovec, J.: Node2Vec: scalable feature learning for networks. In: Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 855–864 (2016)
20. Kefalas, P., Manolopoulos, Y.: A time-aware spatio-textual recommender system. *Expert Syst. Appl.* **78**, 396–406 (2017)
21. Kefalas, P., Symeonidis, P., Manolopoulos, Y.: Recommendations based on a heterogeneous spatio-temporal social network. *World Wide Web* **21**, 345–371 (2018)
22. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings Of The 31st International Conference on Machine Learning, vol. 32, pp. 1188–1196 (2014); 6,22
23. Ali, Z., Qi, G., Muhammad, K., Ali, B., Abro, W.: Paper recommendation based on heterogeneous network embedding. *Knowl.-Based Syst.* **210**, 106438 (2020)
24. Ali, Z., Qi, G., Muhammad, K., Kefalas, P., Khusro, S.: Global citation recommendation employing generative adversarial network. *Expert Syst. Appl.* **180**, 114888 (2021)
25. Cohan, A., Feldman, S., Beltagy, I., Downey, D., Weld, D.: Specter: document-level representation learning using citation-informed transformers (2020). [arXiv:2004.07180](https://arxiv.org/abs/2004.07180)
26. Zhang, C., Wang, G., Yu, B., Xie, Y., Pan, K.: Proximity-aware heterogeneous information network embedding. *Knowl.-Based Syst.* **193**, 105468 (2020)
27. Goyal, P., Ferrara, E.: Graph embedding techniques, applications, and performance: a survey. *Knowl.-Based Syst.* **151**, 78–94 (2018)
28. Cui, P., Wang, X., Pei, J., Zhu, W.: A survey on network embedding. *Trans. Knowl. Data Eng.* **31**, 833–852 (2018)
29. Du, N., Guo, J., Wu, C., Hou, A., Zhao, Z., Gan, D.: Recommendation of academic papers based on heterogeneous information networks. In: 2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA), pp. 1–6 (2020)
30. Kong, X., Mao, M., Wang, W., Liu, J., Xu, B.: VOPRec: vector representation learning of papers with text information and structural identity for recommendation. *IEEE Trans. Emerg. Top. Comput.* **9**, 226–237 (2018)
31. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710 (2014)
32. Ganguly, S., Pudi, V.: Paper2vec: combining graph and text information for scientific paper representation. In: European Conference on Information Retrieval, pp. 383–395 (2017)
33. Ribeiro, L., Saverese, P., Figueiredo, D.: struc2vec: learning node representations from structural identity. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 385–394 (2017)
34. Ali, Z., Qi, G., Muhammad, K., Bhattacharyya, S., Ullah, I., Abro, W.: Citation recommendation employing heterogeneous bibliographic network embedding. *Neural Comput. Appl.* **34**, 10229–10242 (2022)

35. Xia, F., Liu, H., Lee, I., Cao, L.: Scientific article recommendation: exploiting common author relations and historical preferences. *Trans. Big Data.* **2**, 101–112 (2016); 4
36. Zhou, Y., Huang, J., Sun, H., Sun, Y., Qiao, S., Wambura, S.: Recurrent meta-structure for robust similarity measure in heterogeneous information networks. *ACM Trans. Knowl. Discov. Data (TKDD)* **13**, 1–33 (2019)
37. Beltagy, I., Lo, K., Cohan, A.: SciBERT: a pretrained language model for scientific text (2019). [arXiv:1903.10676](https://arxiv.org/abs/1903.10676)
38. Chen, X., Zhao, H., Zhao, S., Chen, J., Zhang, Y.: Citation recommendation based on citation tendency. *Scientometrics* **121**, 937–956 (2019)
39. Qiu, T., Yu, C., Zhong, Y., An, L., Li, G.: A scientific citation recommendation model integrating network and text representations. *Scientometrics* **126**, 9199–9221 (2021)
40. Hu, B., Fang, Y., Shi, C.: Adversarial learning on heterogeneous information networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 120–129 (2019)
41. Xie, F., Li, S., Chen, L., Xu, Y., Zheng, Z.: Generative adversarial network based service recommendation in heterogeneous information networks. In: 2019 IEEE International Conference on Web Services (ICWS), pp. 265–272 (2019)
42. Zhang, Y., Yang, L., Cai, X., Dai, H.: A novel personalized citation recommendation approach based on GAN. In: International Symposium on Methodologies for Intelligent Systems, pp. 268–278 (2018)



Performance of Machine Learning Classifiers for Malware Detection Over Imbalanced Data

Paulina Morillo^{1,2(✉)}, Diego Bahamonde¹, and Wilian Tapia¹

¹ Universidad Politécnica Salesiana, Computer Science Engineering, Quito, Ecuador
dbahamondem@est.ups.edu.ec, wtapiapi@est.ups.edu.ec

² Universidad Politécnica Salesiana, IDEIAGEOCA Research Group, Quito, Ecuador
pmorillo@ups.edu.ec

Abstract. Detecting malware is crucial to avoid severe damage to a computer system. However, doing it by training Machine Learning algorithms can present complications since often there is imbalanced data. Therefore, one of the challenges faced by binary classification is learning to clearly distinguish between two classes when you have a much larger number of instances of one class than another. To decrease bias and to handle imbalance, some techniques increase or reduce the number of cases of the minority and majority classes, respectively. This paper analyzes the performance of three cost-sensitive classifiers, LR, DT, and RF, trained with an imbalanced malware detection dataset and four artificial datasets built using Near Miss, SMOTE, SMOTEENN, and SMOTE-Tomek re-sample techniques. The results show that Near Miss achieves a proper balance between the classes so that the algorithms increase their overall performance, reaching balanced accuracies greater than 95%. On the other hand, the rest of the techniques slightly increase the ability of the classifiers to identify objects of the minority class. Meanwhile, Random Forest achieved balanced and high performance. Besides, the training and testing times for oversampling or hybrid techniques are far superior to those obtained by undersampling since the latter reduces the number of instances processed by the models.

Keywords: Binary Classification · Re-Sample · Oversampling · Undersampling · Hybrid · Balance Accuracy · G-Mean · AUC · Confusion Matrix

1 Introduction

Machine Learning (ML) is present in many daily applications, such as weather forecasts, recommendation systems, spam detectors, etc. These applications mostly use supervised learning algorithms, such as classifiers, to identify the class to which an object belongs. These algorithms have a very high dependency on the training data set. Therefore, when a dataset does not contain the same number of instances for each class, the algorithm's overall performance decreases, as does its ability to identify instances of minority classes, producing a bias in

the classification [1–3]. Imbalanced datasets (IDS) are often present in various problems, such as detecting fraudulent financial transactions, where only five of every 1,000 transactions are fraudulent. Similarly, in detecting unusual data traffic on a network, where 88.5% of the traffic is normal, while only 11.5% is unusual [4], similar examples are everywhere in real life. Also, detecting malware in a network or a computer system through ML techniques often runs into imbalanced datasets [5] since more than 90% of the analyzed traffic is benign. In comparison, approximately 10% is malignant software. Therefore, misclassification costs are different since not detecting malware can cause severe damage to systems or, even worse, loss of valuable information [6–9]. On the other hand, if a system classifies all software as malware, it can cause functional problems or system saturation. Therefore, in these cases, the classification is not a trivial task since the algorithms tend to favor the majority class, making more significant errors in predicting the minority class.

The imbalanced problem can be solved by applying techniques developed from two approaches. On the one hand, there are techniques to handle an imbalance in data level and others in the algorithm level [1, 10, 11]. In the first case, the data-level techniques aim to modify the size (increase or decrease) of the IDS instances through re-sampling techniques [12] in the data preprocessing phase before training the ML model. In the second case, algorithm-level techniques eliminate the imbalance by modifying the learning algorithms [13], such as cost-sensitive algorithms, in which a more significant penalty is given to the classification wrong of the minority class compared to the majority class [14]. One of the class re-sampling algorithms is oversampling, which balances IDS by doubling or generating synthetic samples in the minority classes [3, 12]. However, this procedure can cause an over-fit in the model and a high computational cost. Another method is sub-sampling, which acts on the majority class, eliminating samples until the dataset is balanced. These methods can be risky, as excessive instances can be removed, decreasing the algorithm's overall performance [15]. In addition to the mentioned methods, hybrid algorithms combine both re-sampling techniques to have a greater separation between classes [16–19].

1.1 Related Works

Currently, the use of Machine Learning and Deep Learning (DL) for malware detection is widespread; for example, in [20], the authors explore the use of various classical supervised learning algorithms to detect malware such as k-Nearest Neighbors (kNN), Naïve Bayes, J48 Decision Tree, Support Vector Machine (SVM) and Multilayer Perceptron Neural Network (MLP). The experimental results show that in all cases, accuracy and detection rates are higher than 90%. Similarly, in the survey [21], the results of three SVM, ANN, and RF classifiers are analyzed in contrast to other works where other classifiers or Deep Learning are used, their conclusions indicate that despite the simplicity of the Logistic Regression algorithm, it shows high detection capacity compared to more robust algorithms such as SVM, ANN and Convolutional Neural Networks (CNN). Even in [22] compare an ML classification algorithm (RF) against one of

the DL algorithms, Deep Neural Networks (DNN), the results show that regardless of the feature inputs, the RF accuracy exceeds the DNN, not counting the high computational cost of DNNs. In addition, several articles [23–27] address the malware detection on Android and mobile devices with different approaches and through the use of classifiers, the algorithms that outstanding are Naïves Bayes and Random Forest.

On the other hand, detecting malware involves using large, imbalanced volumes. In [28], the authors work on a large set of instances, of which two-thirds belong to the malware class, and propose a model based on the State-of-the-Art Model for Natural Language Processing (BERT) with which they achieve an F1-Score greater than 90% using only 0.5% of examples from the majority class. In another work, [29], the problem of class imbalance is addressed with a feature extraction technique and an Ensemble Learning Method with which they obtain an accuracy close to one. For its part, [30] proposes the use of oversampling techniques such as SMOTE combined with two cost-sensitive algorithms (SVM) and C4.5 decision tree to detect malware on Android, the results show that for a given threshold, the performance of classification algorithms is significantly degraded, to mitigate this degradation a Simplex Imbalanced Data Gravitational Classification model (S-IDGC) is used to further reduce IDGC time costs without sacrificing classification performance and implemented a machine learning-based prototype system that allows comparison of the detection performance of different classification algorithms on the same data set, as well as the performance of a specific classification algorithm on multiple datasets. Similarly, [31] uses hybrid resampling techniques (subsampling and oversampling) to improve class balance and KNN for Android malware classification, obtaining performance metrics above 97%.

Despite the multiple articles that address the detection of malware in imbalanced datasets, our work proposes the analysis of the performances of three Cost-Sensitive classification algorithms, Logistic Regression (LR), Decision tree (DT), and Random Forest (RF), over an imbalanced dataset for malware detection and over four new datasets built through techniques to handle an imbalance in data level, Near Miss, SMOTE, SMOTETomek, and SMOTENN.

This work uses conventional classification performance metrics without considering graphical evaluation tools such as Cost Curves, ROC curves, etc. On the other hand, we only analyzed if the modification of the weights of the misclassification cost of the cost-sensitive classifiers added to the use of balanced datasets allows for improvement in the performance of the detection of the minority class. Resample techniques are also applied to a single dataset containing static analysis data extracted from the ‘pe_sections’ elements of Cuckoo Sandbox reports; other datasets analyzed in the literature are not considered.

In this way, Sect. 2 presents the material and methods to be used in the data balancing techniques and the training and evaluation of the models. Section 3 describes the experiments performed and the results. Finally, the conclusions and prospects of this work are presented in Sect. 4.

2 Materials and Methods

This article applies techniques to handle an imbalanced malware dataset. The dataset is available at <https://ieee-dataport.org/open-access/malware-analysis-datasets-pe-section-headers>. Before training the models, data normalization is performed to ease the algorithms' convergence. Four artificial datasets are then built by applying the re-sampling techniques: Near Miss, SMOTETomek, SMOTEENN, and SMOTE. Finally, three classifiers are trained and evaluated: RL, DT, and RF, partitioning each dataset in train and testing using the stratified cross-validation method to keep the percentage of samples of each class in each partition.

2.1 Data Normalization

Data cleaning consists of inspecting the dataset to determine if there are null, duplicate, or empty values. The data is then normalized so the attributes are on the same scale. In this way, the learning of algorithms is facilitated, and the bias caused by extreme values is minimized. Once the data normalization is done, the instances with their respective features are stored in an X variable, while the labels (classes) are in a Y variable.

2.2 Construction of Balanced Datasets

The imbalanced classes are treated by applying four techniques with the following approaches: increasing the instances of the minority class (SMOTE), decreasing the samples of the majority class (Near Miss), and finally, hybrids (SMOTETomek and SMOTEENN). Table 2 shows the configurations of these techniques.

The SMOTE technique requires the rate of increase of the instances in the minority class, which is calculated using the formulation described in [3] and whose recommended value is 0.9858. The SMOTETomek and SMOTEENN hybrid techniques share the SMOTE concept to generate samples based on the distance between each datum and the nearest neighbors of the minority class. In turn, ENN and Tomek Link are in charge of eliminating overlapping samples that are on the limit between the classes. Both approaches work well because one generates synthetic samples for the minority class, and the other removes samples from the majority class until the classes are balanced. These two techniques require the sample size parameter, which can be configured with the automatic option 'auto' to resample and remove instances more evenly. In the case of Near Miss [32, 33], version 2 was selected since being focused on distant samples tends to be less affected by the noise in the data. The configuration of this technique includes the re-sampling ratio parameter, which is set to 0.5 to keep instances of the minority class and remove instances of the majority class up to twice the size of the samples of class 0. The removal and sample selection are performed considering the average distance of the minority instances and selecting the number of neighbors with parameter 3-NN [34].

Table 1. Hyper-parameters of each classifier.

Classifiers					
RL		DT		RF	
penalty	l2	max_depth	5	n_estimators	100
dual	False	random_state	0	bootstrap	True
tol	0.0001			criterion	gini
solver	lbfgs				
solver	lbfgs				
multi_class	ovr				

2.3 Training of Classifiers

In the model configuration and training stage, cross-validation is carried out, a mechanism that reduces training bias caused by a particular division of the data set. This way, the training and evaluation process is carried out with different training and test partitions k . Finally, the evaluation metrics of each test are averaged to obtain the final performance of the model. In this case, stratified 10-cross-validation guarantees an adequate number of instances of both classes. Considering the IDS, if this technique is not implemented, the training dataset could result without instances of the minority class, causing a classification bias that benefits the majority class.

Supported by study [35], the algorithms that allow cost sensitivity parameters, which are Decision Trees, Logistic Regression, Random Forest, and Support Vector Machines Classifier, were selected. The configurations of these algorithms are summarized in Table 1.

Table 2. Techniques to handle an imbalance setup.

Near Miss	SMOTETomek	SMOTEENN		SMOTE
sampling_strategy	0.5	sampling_strategy	Auto	sampling_strategy
version	2	random_state	None	random_state
n_neighbors	3	smote	None	k_neighbors
n_neighbors_ver3	3	tomek	enn	5

For the first case, when algorithms are trained with IDS, we consider a cost-sensitive matrix $C_{2 \times 2}$, where $C_{0,0} = C_{1,1} = 0$, $C_{0,1} = 1$, and $C_{1,0} = IR = X_m/X_M$ with X_m and X_M being the size of subsets of samples belonging to the majority and minority class, respectively. This heuristic means that the cost of a false negative is 1, and the cost of a false positive is the imbalance ratio (IR). For the other cases, the cost-weight $C_{0,1}$ and $C_{1,0}$ are set to one when the training is done with the balanced datasets (BDS), i.e., the cost of FP and FN are equal.

2.4 Evaluation of Classifiers' Performance

The usual metrics to measure the performance of a classifier are Accuracy (Acc) and Precision (Pr). However, because imbalanced data is used, additional metrics are included, such as Specificity(Sp), Recall (Re), Balance Accuracy (BAC) [36], G-mean [37], and Area Under the ROC Curve (AUC). Therefore, after cross-validation, we calculate the mean and standard deviation of metrics get in all folds in a similar way to the confusion matrix, which shows the number of True Negatives (TN), True Positives (TP), False Positives (FP), and False Negatives (FN). The metrics used allow for a more accurate analysis of the successes and errors of each class and the performance of the classifiers [38].

3 Experiments and Results

The experiments were carried out using Google Colaboratory (Colab) through the free cloud service of a virtual machine with 12 GB of RAM and 50 GB of hard drive, of which 1.24 GB and 41.86 GB were used, respectively. The programming language used was Python V.3.

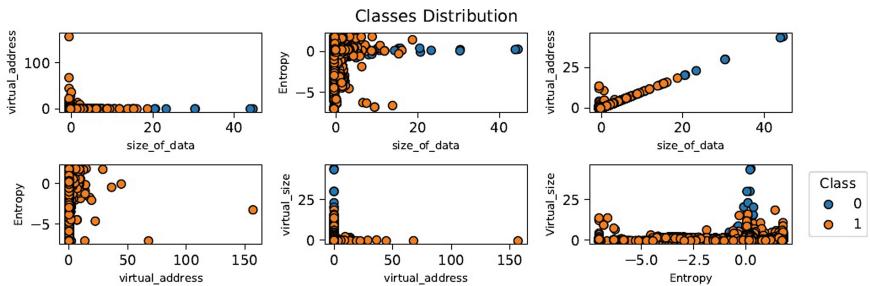


Fig. 1. Classes distribution between each pair of attributes on the normalized dataset

The dataset selected for the experimental part of this work is called malware analysis dataset: Pe Section Headers, created by Oliveira, the dataset is in the public domain, it is hosted at IEEEDataPort [39], and it was extracted from the Cuckoo Sandbox dynamic report. Malware samples were taken from Virusshare, and goodware examples were downloaded from portableapps.com and Windows 7 × 86 directories. The dataset has a total of 43,293 instances, with five features, of which four are numeric (size of data, virtual address, entropy, and the virtual size); the last attribute is of type text (hash) that contains an encrypted string, also include a column with two classes (1: malware o 0: goodware). The class distribution consists of 96.02% malware versus 3.98% goodware.

After selecting the data set, normalization was performed. Figure 1 shows the distribution of the two classes and their relationship between each pair of input variables. As can be seen, there is an overlap between the classes, and the

attributes “virtual address” and “size of data” have very similar values, keeping a linear relationship.

No data cleaning was necessary because no missing values or outliers were found. So only the text-type hash column was eliminated since it is an attribute that does not provide relevant information for class prediction. With the normalized data set, resampling techniques were applied, obtaining four additional datasets whose sizes are shown in Table 3, and the comparison of the distribution of classes in each dataset is shown in Fig. 2.

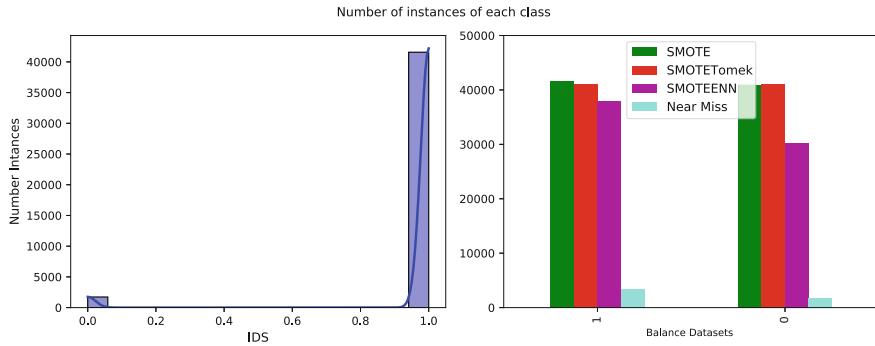


Fig. 2. Class distribution in each dataset (IDS and other balance datasets)

The Near Miss eliminated the samples of class one until maintaining twice the class zero. Contrarily, the resample and hybrid techniques significantly increased the instances of the minority class and reduced the instances of the majority class by a smaller percentage, e.g., SMOTETomek decreased by a little more than one percent of malware examples. At the same time, it increased by more than 12 times the number of goodware examples. Similarly, SMOTEENN multiplied for more than 17 the number of minority class samples, while it removed close to 10% of nearest neighbor samples for the majority class. Finally, SMOTE added synthetic samples to class zero, similarly to other techniques.

On the other hand, in the model training phase, stratified cross-validation was performed with 80% (of each class) for training and 20% (remaining) for validation. Table 4 shows the different performance metrics of each classifier before and after applying the techniques. The configuration of the algorithms in both cases was similar except for the cost-weight, where $C_{1,0}$ was equal to 0.0415 for IDS and 1 for BDS. Thus, the accuracy and precision were greater than 70% in all models trained with IDS. In contrast to the specificity (TNR), which was below 7% in three of four BDS, most hits correspond to the malware class. In the case of RF models, the TNR is close to 30%. Instead, the BAC gets values that are more in line with reality. This metric penalizes errors in the minority class, so it is around 50% in models trained with IDS, except for RF, which reached almost 15% higher.

Table 3. Number of instances before and after applying the resample techniques.

	IDS	Near miss	SMOTETomek	SMOTEENN	SMOTE
0	1725	1725	41009	30224	40977
1	41568	3450	41009	37979	41568

When classifiers are trained using the balanced dataset with Near Miss, an equilibrium is observed in the number of hits of both classes. On the contrary, the resampling and hybrid techniques (SMOTETomek, SMOTEENN, and SMOTE) generally increase the number of TN but reduce the number of TP. Therefore, the accuracy decreased in the LR and DT models. Although the same does not happen with RF, where both classes' hits are balanced, the Acc values remain close for all datasets. All the performance metrics of all the algorithms trained with the Near Miss balanced dataset were higher than 90%, which shows that applying data balancing with this technique improves the learning of the models and reduces the classification bias.

Further, when analyzing the results of the classifiers trained with the dataset balanced with the SMOTETomek technique, the precision and then the G-mean of the LR is reduced between 20 and 30% compared to the metrics obtained with the original dataset. Still, specificity is augmented at 50%, which implies that the balance of the data contributes to class learning. In the case of DT, all metrics are close and superior to 80%, so the classifications are relatively balanced. Lastly, the Re did not present significant changes for Random Forest, but the Sp showed an increase of more than 60%. RF obtains the highest performance values in detecting both classes.

Continuing the analysis, the SMOTEENN combined with LR did not obtain favorable results. These results show that the balance of the data did not contribute to the learning of the minority class and worsened the classification of the majority class. In contrast, the RF and DT classifiers increased the TNR of 54% and 70%, respectively. The precision, the G-mean, and the recall remained at values above 80%, like the balanced accuracy.

Finally, the classification results using the balanced dataset with the SMOTE technique increased the sensitivity of Logistic Regression, but specificity, precision, and G-mean decreased. These estimates show that learning did not benefit from class balance. On the contrary, Random Forest and Decision Tree increased their specificity while remaining the rest of the metrics above 80%, with a notable superiority of RF. Both algorithms maintained their ability to detect instances of class one and, in turn, increased their ability to detect class zero.

Besides, the time in seconds in training (t_{train}) and in the testing phase (t_{test}) of each algorithm was measured. These times are shown after the performance metrics in Table 4. Most experiments' classifiers with the shortest training and testing times were LR and DT. RF got longer training times but was relatively fast in the testing phase.

Table 4. Perfomance of classifiers trained by IDS and DDS.

		Acc	Pre	G-mean	Re(TPR)	Sp(TNR)	BAC	AUC	t_{train}	t_{test}
IDS	LR	0.9602	0.9602	0.9799	1.0	0.0	0.5	0.614	0.1094	0.0228
	RF	0.9646	0.9705	0.9818	0.9933	0.273	0.636	0.933	3.2227	0.3429
	DT	0.9603	0.9604	0.9799	0.9999	0.007	0.504	0.83	0.0485	0.0051
Near miss	LR	0.9697	0.9565	0.978	1.0	0.909	0.954	0.958	0.0269	0.0023
	RF	0.983	0.9856	0.9873	0.989	0.971	0.98	0.999	0.2438	0.027
	DT	0.9803	0.9819	0.9853	0.9887	0.9635	0.976	0.994	0.0094	0.001
SMOTETomek	LR	0.6412	0.6212	0.6702	0.7232	0.5592	0.641	0.702	0.1822	0.0168
	RF	0.9711	0.9812	0.9708	0.9606	0.9816	0.971	0.993	7.6237	0.8443
	DT	0.8134	0.838	0.8069	0.777	0.8498	0.813	0.894	0.1239	0.0132
SMOTE-ENN	LR	0.542	0.5519	0.7131	0.9213	0.0707	0.496	0.708	0.1494	0.0148
	RF	0.9922	0.9952	0.9929	0.9907	0.994	0.992	1.0	5.9702	0.6447
	DT	0.8354	0.9193	0.8417	0.7707	0.9159	0.843	0.915	0.1028	0.0109
SMOTE	LR	0.6491	0.6132	0.7098	0.8215	0.4742	0.648	0.7	0.1811	0.019
	RF	0.9642	0.9754	0.9641	0.953	0.9756	0.964	0.991	7.7742	0.8559
	DT	0.8086	0.8306	0.8042	0.7788	0.8387	0.809	0.889	0.1224	0.0139

4 Conclusions and Future Work

This work presents a performance analysis of three cost-sensitive binary classification algorithms (DT, LR, RF) that were trained with a dataset used for malware detection. The experimental results show that it is possible to improve the capacity of the models for detecting goodware (minority class) by applying oversampling, undersampling, and hybrid techniques such as Near Miss, SMOTE, SMOTETomek, and SMOTEEENN, respectively. Of these techniques, Near Miss stands out since it improved the performance of models getting greater than 90% balanced accuracy; that is, it increased the capacity to detect instances of the minority class without impairing the detection of the samples of the majority class; in addition, the training and testing times are decreased when Near Miss is applied due to the reduction of the number of instances of the majority class, in contrast, to the other resamples techniques. However, these measured times depend on the computational infrastructure to implement the models. Therefore, we propose conducting tests on machines with different memory and calculation capacities to compare the results.

Regarding the general performance of the classifiers, the Random Forest algorithm manages to differentiate both classes with high precision, even with imbalanced data. In future work, we propose incorporating new algorithms and parameters into the analysis, such as the cost of sensitivity using different thresholds and including new metrics to measure the performance of classifiers such as Cost Curves, Matthews correlation coefficient (MCC), kappa, etc.

References

1. Gao, J., Gong, L., Wang, J., Mo, Z.: Study on unbalanced binary classification with unknown misclassification costs. In: 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). IEEE, pp. 1538–1542 (2018)
2. Blake, R., Mangiameli, P.: The effects and interactions of data quality and problem complexity on classification. *J. Data Inf. Qual. (JDIQ)* **2**(2), 1–28 (2011)
3. Lee, D., Kim, K.: An efficient method to determine sample size in oversampling based on classification complexity for imbalanced data. *Expert Syst. Appl.* **184**, 115442 (2021)
4. Bagui, S., Li, K.: Resampling imbalanced data for network intrusion detection datasets. *J. Big Data* **8**(1), 1–41 (2021)
5. Damodaran, A., Troia, F.D., Visaggio, C.A., Austin, T.H., Stamp, M.: A comparison of static, dynamic, and hybrid analysis for malware detection. *J. Comput. Virol. Hacking Tech.* **13**, 1–12 (2017)
6. Vinod, P., Jaipur, R., Laxmi, V., Gaur, M.: Survey on malware detection methods. In: Proceedings of the 3rd Hackers' Workshop on computer and internet security (IITK HACK'09), pp. 74–79 (2009)
7. Ye, Y., Li, T., Adjeroh, D., Iyengar, S.S.: A survey on malware detection using data mining techniques. *ACM Comput. Surv.* **50**(3) (2017). <https://doi.org/10.1145/3073559>
8. Aslan, O.A., Samet, R.: A comprehensive review on malware detection approaches. *IEEE Access* **8**, 6249–6271 (2020)
9. Gavriluț, D., Cimpoeșu, M., Anton, D., Ciortuz, L.: Malware detection using machine learning. In: International Multiconference on Computer Science and Information Technology, pp. 735–741 (2009)
10. Lin, W.-C., Tsai, C.-F., Hu, Y.-H., Jhang, J.-S.: Clustering-based undersampling in class-imbalanced data. *Inf. Sci.* **409**, 17–26 (2017)
11. Zhai, J., Qi, J., Shen, C.: Binary imbalanced data classification based on diversity oversampling by generative models. *Inf. Sci.* **585**, 313–343 (2022)
12. Cai, L., Wang, H., Jiang, F., Zhang, Y., Peng, Y.: A new clustering mining algorithm for multi-source imbalanced location data. *Inf. Sci.* **584**, 50–64 (2022)
13. Ali, H., Salleh, M.N.M., Saedudin, R., Hussain, K., Mushtaq, M.F.: Imbalance class problems in data mining: a review. *Indones. J. Electr. Eng. Comput. Sci.* **14**(3), 1560–1571 (2019)
14. Kim, J., Comuzzi, M.: A diagnostic framework for imbalanced classification in business process predictive monitoring. *Expert Syst. Appl.* **184**, 115536 (2021)
15. Li, X., Zhang, L.: Unbalanced data processing using deep sparse learning technique. *Futur. Gener. Comput. Syst.* **125**, 480–484 (2021)
16. Puri, A., Kumar Gupta, M.: Improved hybrid bag-boost ensemble with K-means-SMOTE-ENN technique for handling noisy class imbalanced data. *Comput. J.* **65**(1), 124–138 (2021). <https://doi.org/10.1093/comjnl/bxab039>
17. Goel, G., Maguire, L., Li, Y., McLoone, S.: Evaluation of sampling methods for learning from imbalanced data. In: International Conference on Intelligent Computing, pp. 392–401. Springer (2013)
18. Srivastava, J., Sharan, A.: Smoteen hybrid sampling based improved phishing website detection (2022)
19. Prati, R.C., Batista, G.E., Monard, M.C.: Learning with class skews and small disjuncts. In: Brazilian Symposium on Artificial Intelligence, pp. 296–306. Springer (2004)

20. Firdausi, I., lim, C., Erwin, A., Nugroho, A.S.: Analysis of machine learning techniques used in behavior-based malware detection. In: 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies, pp. 201–203 (2010)
21. El Merabet, H., Hajraoui, A.: A survey of malware detection techniques based on machine learning. *Int. J. Adv. Comput. Sci. Appl.* **10**(1) (2019)
22. Shukla, P., Bhowmick, K.: To improve classification of imbalanced datasets. In: 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), pp. 1–5 (2017)
23. Sahs, J., Khan, L.: A machine learning approach to android malware detection. In: European Intelligence and Security Informatics Conference, pp. 141–147 (2012)
24. Amos, B., Turner, H., White, J.: Applying machine learning classifiers to dynamic android malware detection at scale. In: 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 1666–1671 (2013)
25. Demontis, A., Melis, M., Biggio, B., Maiorca, D., Arp, D., Rieck, K., Corona, I., Giacinto, G., Roli, F.: Yes, machine learning can be more secure! a case study on android malware detection. *IEEE Trans. Dependable Secure Comput.* **16**(4), 711–724 (2019)
26. Narudin, F.A., Feizollah, A., Anuar, N.B., Gani, A.: Evaluation of machine learning classifiers for mobile malware detection. *Soft. Comput.* **20**, 343–357 (2016)
27. Peiravian, N., Zhu, X.: Machine learning for android malware detection using permission and api calls. In: 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, pp. 300–305 (2013)
28. Oak, R., Du, M., Yan, D., Takawale, H., Amit, I.: Malware detection on highly imbalanced data through sequence modeling. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, ser. AISec'19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 37–48. <https://doi.org/10.1145/3338501.3357374>
29. Zhang, Y., Huang, Q., Ma, X., Yang, Z., Jiang, J.: Using multi-features and ensemble learning method for imbalanced malware classification. In: 2016 IEEE Trustcom/BigDataSE/ISPA, pp. 965–973 (2016)
30. Chen, Z., Yan, Q., Han, H., Wang, S., Peng, L., Wang, L., Yang, B.: Machine learning based mobile malware detection using highly imbalanced network traffic. *Inf. Sci.* **433–434**, 346–364 (2018). www.sciencedirect.com/science/article/pii/S0020025517307077
31. Dehkordy, D.T., Rasoolzadegan, A.: A new machine learning-based method for android malware detection on imbalanced dataset. *Multimed. Tools Appl.* **80**, 24 533–24 554 (2021)
32. Mqadi, N.M., Naicker, N., Adeliyi, T.: Solving misclassification of the credit card imbalance problem using near miss. *Math. Probl. Eng.* (2021)
33. Jabbar, A., Li, X., Omar, B.: A survey on generative adversarial networks: Variants, applications, and training. *ACM Comput. Surv. (CSUR)* **54**(8), 1–49 (2021)
34. G. Lemaître, Nogueira, F., Aridas, C.K.: Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *J. Mach. Learn. Res.* **18**(17), 1–5 (2017). <http://jmlr.org/papers/v18/16-365.html>
35. Alkharabsheh, K., Alawadi, S., Kebande, V.R., Crespo, Y., Fernández-Delgado, M., Taboada, J.A.: A comparison of machine learning algorithms on design smell detection using balanced and imbalanced dataset: a study of god class. *Inf. Softw. Technol.* **143**, 106736 (2022)

36. Brodersen, K.H., Ong, C.S., Stephan, K.E., Buhmann, J.M.: The balanced accuracy and its posterior distribution. In: 20th International Conference on Pattern Recognition, pp. 3121–3124 (2010)
37. Kubat, M., Matwin, S., et al.: Addressing the curse of imbalanced training sets: one-sided selection. Icml; Citeseer **97**(1), 179 (1997)
38. Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B., Herrera, F., Fernández, A., García, S., Galar, M., Prati, R.C., et al.: Learning from imbalanced data streams. In: Learning from Imbalanced Data Sets, pp. 279–303 (2018)
39. Oliveira, A.: Malware analysis datasets: Pe section headers (2019). <https://doi.org/10.21227/2czh-es14>



Pursuing the Optimal CP Model: A Batch Scheduling Case Study

Giacomo Da Col¹(✉) and Erich Teppan^{1,2}

¹ Fraunhofer Austria Research GmbH, KI4LIFE, Klagenfurt, Austria

giacomo.dacol@fraunhofer.at, erich.teppan@aau.at

² Alpen-Adria Universität Klagenfurt, Klagenfurt, Austria

Abstract. The process of coming up with an effective constraint programming (CP) model for complex industrial optimization problems constitutes a cumbersome engineering task. Not only the problem representation itself, but also hyper-parameters steering the search and constraint propagation can have a significant impact on the solving performance. In this paper, we illustrate the evolution of a CP implementation (i.e. model and parameters), from the first functioning version with default hyper-parameters, to a more effective implementation through equivalence-preserving model changes and parameter tunings. In particular, we use four different problem representations in combination with three different levels of constraint propagation and a grid of parameter configurations for the employed search strategy. In this case study, we focus on the Oven Scheduling Problem, a formulation of a job scheduling problem variant that often occurs in production industry (e.g. the semiconductor domain). Concerning CP solvers, we use IBM CP Optimizer, currently one of the strongest suits for scheduling problems. By iteratively evolving a given CP implementation that was not able to solve all benchmark instances within the given limit of one hour, we increased the performance to the point where all instances could be solved, and solution quality is at par with the best known solutions in literature.

Keywords: Constraint Programming · Industrial Optimization Problems · Oven Scheduling Problem · CP Optimizer

1 Introduction

Recent years have experienced a resurrection of exact methods, like constraint programming (CP), for optimization problems. Part of it is due to the advancement in computing technology (cheaper and larger memory), part due to advancement in the techniques used (e.g. large neighborhood search [6]) and part due to the rising demand of digitization from industry. These factors, combined with the innate compactness and adaptability of declarative approaches (such as CP, mixed integer programming, or answer set programming), started to draw attention on such methods not only from academia, but also from “real

world” industrial domains. For example, Google has developed OR-Tools, an open-source general purpose solver that it is also used internally to solve their intricate routing problems [11]. On the commercial side, IBM provides CP Optimizer, a proprietary CP solver that is specifically targeted to scheduling problems [7]. There are several works in literature that show the effectiveness of such approaches in the context of production scheduling [1, 12, 15, 16].

With increase of demand comes a consequent push to increase the offer, and many practitioners are starting to develop an interest in these tools. The goal of this paper is to provide a set of guidelines and good practices for modelling a scheduling problem using constraint programming. In particular, we start from an encoding of a scheduling problem from literature, and we incrementally improve on it using a three-stage approach based on: the adaptation of the problem representation (i.e. the CP model), experimentation with different levels of constraint propagation (i.e. inference levels) and hyper-parameter tuning with respect to the used search strategy (in our case this was restart search).

Our target problem is the Oven Scheduling Problem (OSP) formally defined in [8]. This problem is a variant of a job scheduling problem, all of which basically deal with the question of which machines carry out which job operations at which time in order to optimize some given criterion. In the OSP some of the machines are ovens that heat the processed items for a certain time at a certain temperature. These ovens, in contrast to single-operation machines, are capable of processing multiple items in parallel. Thus, a central planning issue for the OSP is to decide which of the processed items are grouped (i.e. form a batch) and ‘baked’ together.

Declarative approaches like Constraint programming (CP), Answer Set Programming (ASP) and (Mixed-) Integer Programming (MIP) have a successful history in solving combinatorial problems in general [2, 14] and scheduling problems in particular [3–5, 7]. In [8], the authors present a thorough investigation of several CP solvers, comparing the performance of both constraint programming and mixed-integer programming encodings on the Oven Scheduling Problem. The standard modelling language for CP and MIP programs is MiniZinc [10], which is supported by the majority of the solvers natively (e.g. Gurobi, OR-Tools, Chuffed). However, CP Optimizer (which is typically considered one of the stronger players in the field) while supporting MiniZinc in recent versions, needs to use its own proprietary optimization language (OPL) to unlock its full potential, because MiniZinc cannot make usage of specialized structures that CPO provides for scheduling.

For this reason, the authors of [8] also provide an OPL encoding for CPO for the Oven Scheduling Problem (OSP). Interestingly, the performance of the OPL encoding was worse compared to the case where a MiniZinc encoding was provided as input to CPO. On the other hand, in a follow-up publication by the same authors [9] dealing with a slightly different variant of the OSP, CPO in combination with an OPL encoding produced better results as with the MiniZinc encoding (in fact, the best results among all tested systems).

Out of this, we assume that there could be room for improvement of the original OPL encoding in [8]. We now take this as a chance to investigate how to improve a given CP model from an engineering perspective. Thus, we take the OPL encoding from [8] as a starting point and show how it can be turned into an even more performing implementation. We want to emphasize that the goal of this paper is not to show the prominence of CPO over other solvers, but rather use this scenario to illustrate the steps involved in the evolution of an effective CP implementation for a problem at hand.

While this study focuses on a specific problem and a specific solver, it can be assumed that the general principles for model evolution discussed herein can be applied for many different problems and declarative solution approaches.

2 Constraint Programming

Constraint programming (CP) [13] is a declarative approach that builds on solving constraint satisfaction problems (CSPs). Hence, a problem at hand must be first represented as a CSP that a CP solver then solves. A CSP is defined as a triple $\langle V, D, C \rangle$, whereby V is a set of variables, D is a set of value domains so that for each $v \in V$ there is exactly one $d \in D$, and C is a set of constraints imposed on the variables. A solution to a CSP is an assignment $\forall v \in V(v = d)$, $d \in \text{dom}(v)$ such that all constraints $c \in C$ are fulfilled.

One of the most classic variants of CSPs uses variables with finite and discrete domains. Consequently, the domain sizes are limited in the form of lower and upper bounds, and the domain values are discrete, typically integers, in which case variables are called *integer variables*. Depending on the solver used, different constraints can be employed to restrict value combinations for variables incorporated by the constraint. For example, *primitive constraints* express that some variable's value must be equal/unequal/smaller/greater than some other variable's value or constant value (e.g., $v1 \neq v2$, $v1 > 6$). *Arithmetic constraints* express arithmetic operations like $v1 + v2 = v3$. *Global constraints* restrict value combinations for a set of variables and typically implement algorithms for solving special sub-problems which, without such a global constraint, have to be represented with primitive and arithmetic constraints. Hence, global constraints significantly increase the expressive power. Take as an example the *alldifferent* global constraint, assuring that no two variables out of some predefined set of variables have the same value. For instance, having three variables $v1$, $v2$, and $v3$ with equal domains ranging from 1 to 3, $\text{alldifferent}(\{v1, v2, v3\})$ states that exactly one variable takes the value 1, 2, and 3, respectively. With primitive constraints, three constraints would be needed to express the same, i.e., $v1 \neq v2$, $v2 \neq v3$, $v1 \neq v3$. Even when 1,000 variables should get different values, still a single *alldifferent* constraint suffices, whereas, without this global constraint, roughly half a million primitive constraints are needed. Similarly to global constraints that lift expressive power for particular sub-problems, higher-order types of variables, e.g., *set variables*, have been introduced. Whereas a finite-domain integer variable takes values from a finite set of integers, a finite-domain set variable takes values from the power set of a finite set of integers. The

expressive power of set variables is exponentially higher as they can be assigned to any of the 2^n subsets of values for a domain of size n .

2.1 CP Optimizer

IBM's CP Optimizer (CPO) is an optimization tool based on constraint programming that is considered the standard de facto for solving scheduling problems. In addition to the typical constructs of constraint programming languages, CPO brings additional construct that are specifically targeted towards scheduling problems, the first of which is a higher-order variable called *interval variable*. Interval variables are designed to meet the special requirements of scheduling problems. Interval variables are well-suited for representing time intervals, as interval variables consist of a start time from an integer domain, a duration, and an end time. An interval variable automatically enforces an arithmetic constraint, i.e., $start + duration = end$. By employing interval variables, a job operation can be represented by a single (interval) variable. Interval variables can also be optional, simulating the case where not all the variables need to be assigned. In addition, CPO also offers a vast variety of oriented global constraints, from the popular NoOverlap and Cumulative constraint, to more scheduling specific ones like span (force a group variables to span within a certain interval) or alternative (specifies that, among a set of optional variables, just one must be present).

These special constructs are the reason why using the proprietary specification language OPL (instead of the more popular MiniZinc) is advised. MiniZinc is an open source constraint modeling language that slowly became the standard language for constraint programming, since it is supported by almost all the popular constraint solvers (Gurobi, Jacop, Choco, OR-Tools). Since the last version, also CPO added support for MiniZinc as specification language. However, CP Optimizer uses optional interval variables to model tasks in scheduling, while MiniZinc uses integer variables (e.g. to represent start times). For this reason, when a MiniZinc variable is used in a scheduling constraint (in particular cumulative) then an artificial interval variable synchronized with the original integer variable is created. Therefore, scheduling models translated from MiniZinc are sub-optimal and it is recommended to use native CP Optimizer model instead. Also optional variable are transformed into non-optionals at MiniZinc level. This is a second reason why the generated model is suboptimal.

CPO is not the only solver that employs ad-hoc constructs for scheduling problems. Also OR-Tools, a famous solver by Google, implements interval variables and scheduling oriented global constraints. However, studies have shown that CPO is typically better suited for solving scheduling problems [4].

3 The Problem

In typical production scheduling problems, a set of machines and a set of products (in literature commonly referred to as jobs) are given. A product has one or more production steps (aka operations) that are performed by one or more

machines. When a machine can collect several operations and perform the production step in parallel (for example an oven that bakes more biscuits in one go), that is a batch machine. In this paper, we focus on a scheduling with batch machines formulation that is called Oven Scheduling Problem (OSP) [8,9].

In OSP, every job is composed of a single operation that can be performed on one of the eligible machines for that job. Every job has an earliest start time that determines when the job is ready to be processed, and a due date (or latest end time). The processing time varies within a range between min and max processing time, on a job basis. Lastly, every job has a size that indicates how many slots are occupied in a batch, and an attribute, which indicates the type of the job.

Concerning machines, every machine has a capacity that indicates the size of the batch. A machine can process a job only if it is listed as an eligible machine for that job. Machines process jobs based on their attribute: Only jobs with the same attribute can be assigned on the same batch. A machine can only process one batch at a time, therefore batches of the same machine cannot overlap. A machine can process jobs of different attributes, but whenever a subsequent batch processes jobs with an attribute different from the current batch (or even the same attribute, in some cases), the machine needs to be setup for the new batch: this comes with an expense in terms of time and costs. Every machine has a matrix of setup times to indicate the time needed for changing from any combination of attributes, and another matrix for the setup costs. The processing time of each batch is determined by the ranges of the jobs that are assigned to the batch, such that it lies within all ranges. For example if a job with range (1,3) and another with (3,6) are in the same batch, the only possible processing time for that batch is 3. Lastly, every machine has intervals where they are not operative, therefore it is not possible to schedule batches or setups during those time intervals.

The optimization criterion is a linear combination of four factors to be minimized: (i)the sum of the processing times of the batches, (ii)the number of due date violations,(iii) the sum of setup times and (iv)the sum of setup costs. More detailed information can be found in [8].

4 Starting Model

In this section we describe the OPL model created in [8]. This model functions as the baseline of our study (hereafter, we refer to this as m_0). We do not go in detail on a line-by-line basis (Detailed information can be found in [8]), but we rather focus on the representation of specific constraints of the problem.

Figures 1 and 2 show the definition of the input constants given by the instance and the definition of the decision variables. The terms *Machines*, *Attributes*, *Jobs* and *Batches* are used to define a range in the dimension of the input specified by the instance. For example, the 2D matrix *SetupTimes* has a number of rows and columns equal to the number of attributes in the instance, and indicates the time needed for changing from a configuration to the other,

INPUT VALUES

```

int SetupTimes[Attributes][Attributes]
int MinCap[Machines]
int MaxCap[Machines]
stepFunction AvailabilityTimes[Machines]
int MinTime[Jobs]
int MaxTime[Jobs]
int JobSize[Jobs]
int Attribute[Jobs]
```

Fig. 1. Definition of input values. This part is common on all the models investigated**DECISION VARIABLES**

```

dvar boolean jobInBatch[Jobs][Machines][Batches]
dvar interval batch [Machines][Batches] optional
    intensity AvailabilityTimes[m]
dvar int attr[Machines][Batches] in Attributes
dvar interval setupTime [Machines][Batches] optional
    intensity AvailabilityTimes[m]
```

Fig. 2. Definition of decision variables. This part is common on all the models investigated

for each attribute couple. The step function *AvailabilityTimes* is a function that assumes the value 100 in the intervals where the respective machine is available, and 0 in the remaining intervals.

The decision variables are defined based on the ranges of values specified by the input, but their value varies during the search and assumes the value of the best solution when the search is completed. The presence of a job in a particular batch of a machine is encoded using a 3D matrix *jobsInBatch* of boolean variables. In addition, a 2D matrix of integer variables *attr* indicates what is the attribute of each batch, a 2D matrix of interval variables *batch* encodes the starting time and the duration of the batches in machines, and a 2D matrix of interval variables *setupTime* encodes when the setup times occur in a machine. Notice that the batch variables are defined as optional, because it is possible that not every batch is present in the final solution. Moreover, the “intensity” of the variable is set using the step function *AvailabilityTimes*. The intensity of a variable is typically used for cases where the “work” done is not the same during the whole interval. For example let’s consider a worker who does not work during weekends (his work intensity during weekends is 0%) and on Friday he works only for half a day (his intensity during Friday is 50%). In this case, the intensity is used to model the machine availability, meaning that when corresponding machine is not available (i.e.when the step function is 0), no batch can be scheduled.

In Fig. 3, it is shown that the variables $jobsInBatch$ are used to enable the application of the constraints when a job is present in a certain batch. For example, in $C3$ the constraints span over all the possible jobs, machines and batches: the first line after the **forall** specifies the length of the batch interval $batch[m][b]$ to be larger or equal to $MinTime[j]$ when $jobInBatch[j][m][b]=1$ (i.e. when the job is present in the batch); otherwise, the length must be ≥ 0 . It is possible to concatenate more than one constraint in the same **forall** by connecting them with a logical and. In the following line, the constraint to put the upper bound on the batch length is applied using the same concept. This means that two constraints are generated for every triple of job, machine and batch, even for machines that are not eligible for a job; this can lead to the generation of many useless constraints, based on the sparsity of the matrix.

5 Model Evolution

In this section we illustrate the process that was followed to improve the performance of the model. We followed three basic stages. At each stage, we performed an evaluation of performance of the models. The best performer of each stage passes to the next stage for further improvement. In all evaluations, we used the same benchmark of 80 OSP instances of increasing size of 10, 25, 50 and

CONSTRAINTS

```
//C1: A job must be assigned to a single batch in a machine
forall (j in Jobs):
    sum (m in Machines, b in Batches):
        jobInBatch[j][m][b] == 1
//C2: The collective size of the jobs assigned to a batch
//cannot exceed the machine capacity
forall (m in Machines, b in Batches):
    sum(j in Jobs):
        JobSize[j] * jobInBatch[j][m][b]) ≤ MaxCap[m]
//C3: The processing time of the batch must be consistent
//with the min and max time of the jobs assigned to it
forall (m in Machines, b in Batches, j in Jobs):
    MinTime[j] * jobInBatch[j][m][b] ≤ length(batch[m][b]) ∧
    length(batch[m][b]) ≤ MaxTime[j]*jobInBatch[j][m][b] +
        maxDuration*(1-jobInBatch[j][m][b])
//C4: Setup times and costs must be accounted for between
//every batch of a machine
forall (m in Machines, b in Batches):
    presence(batch[m][b+1]) == presence(setupTime[m][b]) ∧
    endBeforeStart(batch[m][b], setupTime[m][b]) ∧
    presence(batch[m][b+1]) => (length(setupTime[m][b]) ==
        SetupTimes[attr[m][b]][attr[m][b+1]])
```

Fig. 3. Constraint model for m_0

100 jobs (20 instances per group size), following the same experimental setup as [8]. In particular, every model had a timeout of 1 h per instance, and was allowed to run on a single core. The experiments run on a server with 128 GB of RAM and an AMD EPYC 2.0 GHz with 32 cores. Given that the server was a different one compared to [8], we also rerun the experiment of the m_0 , to be totally comparable with the results. The version of CPO used is the 22.1.0.0.

In the first stage we focused on the modelling decisions, producing several versions of the model using different design choices. At this stage, all the inference level options and the hyper-parameters were set to the default value. The focus was to find the best representation for the problem at hand, and on reducing the search space as much as possible. In the second stage, we tested the best model of the previous stage with different inference levels that regulate the strength of the constraint propagation. In the third stage, we focused on the search hyper-parameters.

5.1 First Stage: Adapting the Problem Representation

The model m_0 was inspired by the MIP encodings developed in [8]. While these encodings work well in the context of MIP solving, this modelling paradigm is generally not ideal for CPO models, which favor the use of optional interval variables compared to boolean variables.

With this principle in mind, in our first variant (m_1) we decided to use interval variables in *jobInBatch*. This choice enabled the use of two global constraints, *alternative* and *span*. The global constraints are compact and effective constraints that are applied to set of variables. The alternative constraint specifies that, out of a set of optional interval variables, just one is present. In m_1 we use this global constraint to enforce *C1*. The *span* constraint forces an interval variable to span over a set of interval variables, which is exactly what a batch variable should do with respect to the jobs assigned to it. Then, we link the end of each job in a certain batch with the end of the batch itself. In this way, we just need to specify that each batch should not overlap using the *endBeforeStart* constraint, and this will be automatically enforced on all the jobs belonging to the batch (similarly to what is done in [9] with the “representative” jobs). To enforce *C3*, we proceed similarly to m_0 , but since we modelled *jobInBatch* using interval variable, the presence of the variable in a certain batch is taken into account before applying the constraint to the batch. Concerning *C2*, we modelled the maximum cap defining a cumulative function for each machine. A cumulative function is defined as a step function, and presents a way to restrict the use of resources over time. Similar to m_0 , we used the constraint *endBeforeStart* to specify that batches in the same machine must not overlap, but we additionally specify a delay between the end of the preceding batch and the start of the successor. This delay is set equal to the setup time of the two batches (*C4*). Thus, we do not need to encode the setup times as additional interval variables.

Model m_2 builds directly on the structure of m_1 . The only difference is in the representation of the machine availability. In m_1 we used the same technique as m_0 , encoding the availability of the machine as intensity of the batch variables. In

m_2 , we decided to encode the machine availability within the cumulative function already used for limiting the size of the batches. The idea is to fill the capacity of a certain machine in the intervals when the machine is not available, such that no job can be scheduled in that interval. Thus, we can spare the enforcement of the intensity function for all the *Batch* \times *Machines* batch variables, and encode the information on the already-present cumulative function.

In model m_3 we aimed to reduce the size of the search space as much as possible. In order to achieve this, we calculated the number of *actual* possible batches for each machine, instead of setting the number of batches to the number of jobs (worst case scenario) for each machine. Moreover, we edited constraints C1 to C4 to cover only the possible combinations of jobs, machines and batches. For all the impossible combinations (e.g. machines that are not eligible for a job) we enforce the corresponding *jobsInBatch* to be absent, in this way we make the execution of the alternative and span constraint faster.

Table 1. Table of the first phase of the model improvement: test performance of different model choices

Model	Solved	Opt	Proof	Avg time(s)	Quality
m_0	52	24	19	2838.2	0.586
m_1	57	24	19	2779.4	0.693
m_2	80	24	24	2568.2	0.979
m_3	80	24	24	2553.7	0.989

Table 1 shows the performance evaluation of the four models. For each model we count the number of instances that are solved, the number of optimal proofs, the number of optimal solutions reached, the average solving time and the average solution quality. The solution quality is the average across the instances of *bestCost/currentCost*, where current cost is the cost of the current solution and the best cost is the best found across all tests in this experiment. Thus, the quality is 1 when the current cost is equal to the best cost, and 0 when the current cost is infinite (i.e. the model could not find a solution for that instance). Table 1 shows that the improvement from m_0 to m_1 is visible, but still marginal. In particular, the limitation of m_0 in solving the large instances of 100 jobs (2/20 solved) is also present in m_1 (5/20 solved). The largest improvement is by far seen in the changes done in m_2 , where thanks to the availability times included in the cumulative function, it was possible to not only solve all the instances, but also to find the optimality proof for 5 additional instances. The improvement brought by the search space reduction done in m_3 is on the average solution time as well as in the average solution quality.

5.2 Second Stage: Inference Levels

The inference level is a powerful parameter that determines how much effort is put in the constraint propagation. Higher levels of inference indicate that

the solver will spend more time in the constraint propagation, possibly achieving stronger domain reductions. CPO offers four levels of inference: low, basic, medium and extended. The default inference level is basic. We applied all three additional levels to our m_3 encoding to see if there is something to be gained. Table 2 shows the results of this test. There is no difference in the number of opt or proof, but there is a definite gain in solution quality when less time is invested in propagation and more in the search (low inference level).

Table 2. Table of the second phase of the model improvement: test performance of different inference levels for m_3 .

Inf level	Solved	Opt	Proof	Avg time(s)	Quality
Low	80	24	24	2545.4	0.995
Basic	80	24	24	2553.7	0.989
Medium	80	24	24	2544.1	0.989
Extended	80	24	24	2571.9	0.980

5.3 Third Stage: Hyper-parameter Tuning

The standard type of search that CPO provides is called restart search which, as the name implies, restarts the search after a certain amount of failures. To steer the behavior of this search, there are two main parameters: the restart limit and the restart factor. The restart limit determines the number of failures for a restart to occur. At each restart, the restart limit is updated multiplying the current limit by the restart factor. The default values for limit and factor are 100 and 1.05 respectively.

To tune the hyper-parameters, we selected the best configuration of the previous phase, m_3 with low inference level. In addition to the default parameters, we tried two factors (1.5 and 2) and two limits (10 and 1000), and all the 9 possible combinations of these values. Table 3 shows the results of this last test. It seems that concerning this problem, the default parameters already work quite well, since they are limited to no improvement with other configurations.¹

6 Discussion and Conclusions

Figure 4 summarizes the improvements of the various versions of the encodings based on solution quality. Clearly, the phase that contributed to the largest improvement is the first phase, and in particular the use of cumulative constraint to model machine availability as implemented in m_2 .

To expand a bit further on why the original model m_0 did not perform well, the main reason is the limited use of interval variables in favor of Boolean

¹ All the models, instance data and extensive results files are provided at <https://tinyurl.com/3n88wj9u>.

Table 3. Table of the third phase of the model improvement: hyper-parameter tuning. Columns with identical values were dropped. Best is in bold.

Limit	Factor	Avg time(s)	Quality
10	1.05	2561.6	0.9926
100	1.05	2545.4	0.9951
1000	1.05	2552.4	0.9937
10	1.5	2560.5	0.9933
100	1.5	2560.9	0.9925
1000	1.5	2557.4	0.9925
10	2	2562.5	0.9931
100	2	2553.6	0.9926
1000	2	2553.3	0.9939

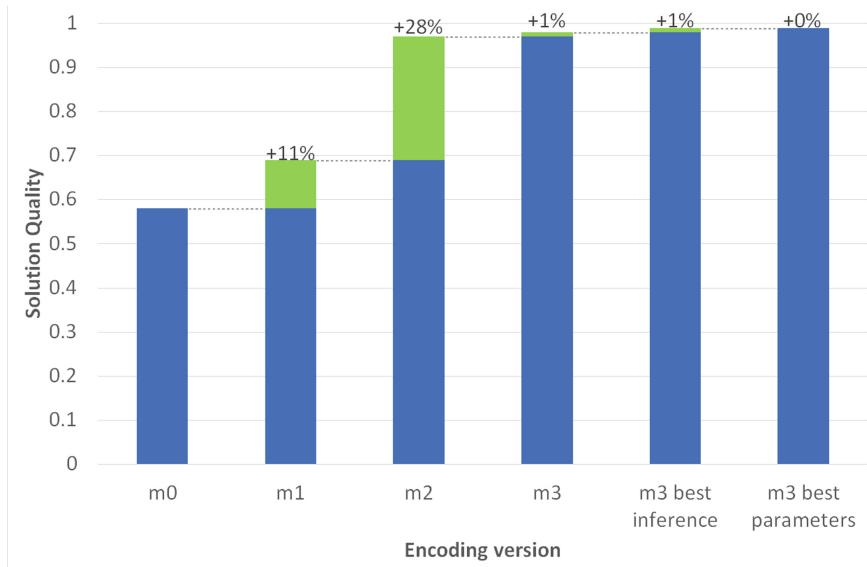


Fig. 4. Solution quality improvement in the different iterations of the model

variables and primitive constraints. When a model is encoded using interval variables, CPO creates a logical network that keeps track of the presence of the variables, and a temporal network that connects the interval variables thanks to global constraints (e.g.alternative and span) and constraints specific for interval variables (e.g.endBeforeStart) [7]. These networks make the propagation much more efficient and can steer the search towards better solutions, avoiding pitfalls. All this is lost when integer or Boolean variables are used, thus, the use of interval variables and global constraints is advised whenever possible.

Summarizing, in this paper we illustrated a three-stage approach to improve a CP encoding in general, and a CP Optimizer encoding in particular, on a scheduling problem. In the experiments, we showed the impact of the different choices, as well as some modelling techniques and hints to exploit the solver capabilities. In particular, the use of interval variables, global constraints (m_1) and cumulative functions (m_2) enabled the model to solve all the instances of the benchmark, including the largest instances with 100 jobs. To a lesser degree, also the reduction of the search space (m_3) and the exploration of different inference level and search parameters improved the model performance. In general, this three-stage approach can be applied to any CP solver, focusing the efforts on creating different versions of the model before trying any tuning of inference levels and search parameters. Comparing with the state-of-the-art results on the same problem, we are at least at par with the best exact method using integer linear programming and warm start [8].

Acknowledgments. This work was supported by the EFRE, REACT-EU, and Carinthian Economic Promotion Fund (Project ML&Swarms, Contract No. KWF-31417—34815—50878)

References

1. Da Col, G., Teppan, E.: Google vs IBM: a constraint solving challenge on the job-shop scheduling problem (2019). [arXiv:1909.08247](https://arxiv.org/abs/1909.08247)
2. Da Col, G., Teppan, E.C.: Learning constraint satisfaction heuristics for configuration problems. In: 19th International Configuration Workshop, vol. 8 (2017)
3. Da Col, G., Teppan, E.C.: Industrial size job shop scheduling tackled by present day cp solvers. In: International Conference on Principles and Practice of Constraint Programming, pp. 144–160. Springer (2019)
4. Da Col, G., Teppan, E.C.: Industrial-size job shop scheduling with constraint programming. Oper. Res. Perspect. **9**, 100249 (2022)
5. Kovács, B., Tassel, P., Kohlenbrein, W., Schrott-Kostwein, P., Gebser, M.: Utilizing constraint optimization for industrial machine workload balancing. In: 27th International Conference on Principles and Practice of Constraint Programming (CP 2021). Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2021)
6. Laborie, P., Godard, D.: Self-adapting large neighborhood search: Application to single-mode scheduling problems. In: Proceedings MISTA-07, Paris 8 (2007)
7. Laborie, P., Rogerie, J., Shaw, P., Vilím, P.: IBM ILOG CP optimizer for scheduling. Constraints **23**(2), 210–250 (2018)
8. Lackner, M.L., Mrkvicka, C., Musliu, N., Walkiewicz, D., Winter, F.: Minimizing cumulative batch processing time for an industrial oven scheduling problem. In: 27th International Conference on Principles and Practice of Constraint Programming (CP 2021). Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2021)
9. Lackner, M.L., Mrkvicka, C., Musliu, N., Walkiewicz, D., Winter, F.: Exact methods and lower bounds for the oven scheduling problem (2022). [arXiv:2203.12517](https://arxiv.org/abs/2203.12517)
10. Nethercote, N., Stuckey, P.J., Becket, R., Brand, S., Duck, G.J., Tack, G.: Minizinc: Towards a standard cp modelling language. In: International Conference on Principles and Practice of Constraint Programming, pp. 529–543. Springer (2007)

11. Perron, L., Furnon, V.: Or-tools (2022). <https://developers.google.com/optimization/>
12. Rodler, P., Teppan, E., Jannach, D.: Randomized problem-relaxation solving for over-constrained schedules. In: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, vol. 18, pp. 696–701 (2021)
13. Rossi, F., Van Beek, P., Walsh, T.: Handbook of Constraint Programming. Elsevier (2006)
14. Tarzariol, A., Schekotihin, K., Gebser, M., Law, M.: Efficient lifting of symmetry breaking constraints for complex combinatorial problems. *Theory Pract. Logic Program.* **22**(4), 606–622 (2022)
15. Teppan, E., Da Col, G.: Automatic generation of dispatching rules for large job shops by means of genetic algorithms. In: CIMA ICTAI, pp. 43–57 (2018)
16. Teppan, E.C.: Types of flexible job shop scheduling: a constraint programming experiment. In: 14th International Conferences on Agents and Artificial Intelligence (ICAART 2022), vol. 3, pp. 516–523 (2022)



Data-Driven Decision-Making Framework for Cost-Efficient Energy Retrofit of Italian Residential Building Stock

Ania Khodabakhshian^(✉) and Fulvio Re Cecconi

Department of Architecture, Politecnico Di Milano, Built Environment and Construction Engineering, Via Ponzio 31, 20133 Milan, Italy
{ania.khodabakhshian,fulvio.receconni}@polimi.it

Abstract. Strategic decision-making to invest in building stock energy retrofit is a time-consuming task, requiring a significant amount of complex and technical data with high acquisition costs in its conventional form. Besides cost data of various retrofit alternatives, needing unit-based calculations, the monetary value of resulting benefits, like saved energy, should be considered for making cost-efficient decisions by policymakers. On the other hand, novel techniques for building retrofit are gaining increasing attention due to their vital role in sustaining the built environment and lowering energy consumption, especially in countries like Italy, where a significant share of the residential building stock is outdated. With advancements in data-collection methods and data-driven decision support systems, more data is available, based on which these decisions can be automated and optimized. This research aims to provide a comprehensive framework for an automated cost-benefit analysis for various energy retrofit scenarios given their energy-saving potential, implementation cost, and associated investment payback period. Based on Italian National databases, i.e., CENED (Building Energy Certificate), TAB-ULA, and Superbonus Cost data of the Lombardy Region, this study proposes a building stock clustering using energy labels, building technologies, and construction period to identify building archetypes. Moreover, the maximum investment amount of each cluster (archetype) for different payback periods is calculated using Monte Carlo Simulation and is compared to the estimated retrofit cost based on regional price lists for each archetype. This cost-benefit analysis using estimated retrofit cost and saved energy cost contributes to faster, more objective, factual, and cost-optimized decision-making for urban-level energy retrofit decision-making, highlighting the importance of government funding and subsidies like Superbonus in achieving sustainability, energy efficiency, and decarbonization goals of the European Union.

Keywords: Energy Retrofit · Clustering · Building Archetypes · Cost-Benefit Analysis · Payback Period · Retrofit Cost

1 Introduction

With the increasing price of energy on the one hand and the low energy efficiency of building stock in Europe, energy saving and greenhouse reduction policies are at the center of attention [1]. While the construction sector is responsible for about 40% of gross energy consumption in the European Union, and 75% of the buildings are outdated, and of high energy consumption, the energy retrofit rate is critically low, about 0.4% to 1.2% [2]. The situation is even more severe in Italy, where 60% of residential buildings are constructed before the first energy-saving law, and over 31 million residential units have Energy Performance Certification (EPC) label (“G”) [3, 4].

The building sector plays a key role in the European Commission’s proposal for the energy saving target of 30% by 2030, due to which the Energy Performance of Buildings Directive, introduced in 2002 (EPBD 2002/91/EC) and revised in 2010 (EPBD recast 2010/31/EU), as well as numerous local laws and regulations were proposed [5]. The essences of all these reports, laws, and regulations were the critical impact of building retrofit in decreasing energy consumption in the building sector and fulfillment of sustainability goals on environmental, social, and economic aspects. Energy retrofit covers an extensive range of operational or physical adjustments in a building, its energy-consuming systems, or occupants’ behavior to lower energy consumption [6]. It can be achieved through different interventions such as passive-cooling measures like ventilation and shading, fabric efficiency measures such as insulation, and freshwater intake limit measures and indoor air quality control [7].

In Italy, starting in 2017, new budget laws and bonuses were proposed to deduct the energy improvement cost by up to 65% in 2017 [8] and super eco bonus [9] to promote energy efficiency through building retrofit. Moreover, with the goal to reach 15.50 million tons of oil equivalent (Mtoe) through energy efficiency measures promotion by 2020, the Strategy for the National Property Assets (STREPIN) was proposed in accordance with the Legislative Decree 2014, No. 102 [4]. This has served as a stimulation for a retrofit rate increase with the vision to save about 5.00 Mtoe/year in the period 2021–2030, which decreases the current consumption to half [1]. In this regard, and with the purpose of fostering urban-level policy making, some valuable national databases like CENED have been collected, with over 1.5 million records on building stock energy and seismic performance measures, and some international projects like Typology Approach for Building Stock Energy Assessment (TABULA) and its follow-up Energy Performance Indicator Tracking Schemes for the Continuous Optimization of Refurbishment Processes in European Housing Stocks (EPISCOPE) have been conducted to analyze building archetypes at national level [10, 11].

The first step for proposing a retrofit strategy is to assess the current energy performance of the building stock, which is conducted in bottom-up or top-down forms [12]. Top-down methods examine the total energy usage within the building sector and rely on national sources of data, aiming to identify the factors that influence long-term changes in energy consumption trends. The bottom-up methods start from small samples with similar characteristics, with the potential of extrapolating the results to a segment of the building sector [13]. The bottom-up methods are used for scenario analysis with the energy-saving vision and can be either statistical or building physics-based. Various

techniques can be applied for bottom-up energy modeling, such as engineering methods, Neural Networks, and conditioned demand analysis [14].

Engineering methods utilize physical laws to determine building energy consumption at either the whole or sub-system levels, and apply complex mathematics or building dynamics to derive accurate energy usage data for all building components, considering internal and external factors [15]. Therefore, they require an enormous amount of complex and technical data regarding (a) building characteristics, (b) energy systems characteristics, control and maintenance, (c) weather parameters, and (d) occupants' behavior, among other sociological parameters, forecasting [13]. Such technical data gathering is a time-consuming and resource-intensive process, including on-site building audits and specific building retrofit performance analysis to estimate the potential of each retrofit measure [16]. Moreover, it is unlikely to scale and generalize them to a complete building stock due to overreliance on building-specific and context-based data.

Even though Engineering and case-based approaches have been the dominant conventional method, they are not applicable to huge databases for urban-level decision-making. It is due to reliance on specialist knowledge and being engaged with intrinsic complexities and technical features of every single property. Therefore, the automation methods of the retrofit analysis process are becoming an emerging field of study [17]. In this sense, Machine Learning (ML)-based and statistical models are the mainstream of research in building retrofit, alongside their trending application in all the disciplines of the construction industry. ML can process vast and complex data faster and more efficiently, and generalize the rules to future data entries, using relatively small number of input features for the learning process [18]. Unlike conventional methods, in which the objective is to predict an output based on a huge volume of input variables and equations, in data-driven approaches like ML, the inputs and outputs of many samples are known, and the algorithms need to find the link between them through learning [13]. Hence, ML, simplifying the data gathering and analysis process of conventional energy retrofit estimation methods, is proposed to analyze and cluster building data in national databases, based on which further cost-benefit analysis on retrofit measures can be conducted [19]. However, the explainability and transparency of the ML algorithms and the understandability of the workflow between the input and output variables are massive challenges for their public acceptance, which is caused by their black-box structure [20]. Other challenges are the unfamiliarity of the building sector's professionals with Artificial Intelligence and digital technologies and the requirement of special training to operate them.

The next step is determining the main features for clustering buildings. The uniform basis for the current and potential energy status is the Energy Performance Certification (EPC) label, an integral part of the Energy Performance of Buildings Directive (2002/31/EC1; 2010/91/EU2), which is an instrument to indicates the current and desired energy performance of the buildings. EPC labels reflect technological factors like the building's cooling and heating systems, geometric peculiarities, finishing materials, window frames, hot water production, and the presence of renewable energy production systems [21]. Therefore, to improve the energy performance of a building, or in other words, the EPC label of a building, these technical factors should be targeted and improved, resulting in different retrofit scenarios and strategies.

Each retrofit scenario can result in various energy saving and implementation costs, both of which should be considered when policymaking on an urban level. The potential energy saving can be assessed through case-based retrofit energy simulation or large data simulation with methods like Monte Carlo. On the other hand, the retrofit costs are estimated according to the unit-based price of components used or work conducted. ML algorithms such as Neural Networks and Decision Trees can help predict the retrofit cost of buildings based on national databases of previous cases and cost-optimal retrofit analysis [22]. However, if some critical data features, like the building footprint or final cost for previously retrofitted buildings, is missing from the database, ML cannot provide useful and accurate results, and instead, case-based calculations, building archetypes, or parametric cost estimation methods need to be used.

Even though the potential of data-driven methods to foster cost-efficient, quick, and precise decision-making has been highlighted and proved in previous literature, a holistic framework to integrate the data flow from current energy performance analysis and building stock clustering to energy saving and retrofit cost estimation, which can integrate and fully benefit from the data-driven tools is missing. Furthermore, although having various national databases on building stock, they are not always compatible or interoperable, with some key features missing that hinder full automation of the process. Therefore, in order to fill this research gap and provide solutions to the automation limitations of the retrofit decision-making process, two research questions were considered:

1. What are the most cost-efficient retrofit strategies based on a cost-benefit analysis of saved energy and retrofit cost?
2. What building groups (archetypes) and based on what criteria should be prioritized for cost-efficient retrofit policymaking?
3. What are some alternative cost estimation methods if some critical features are missing from the database, hindering the automated cost estimation?

This paper aims to propose a data-driven framework for cost-efficient retrofit decision-making and prioritizing building clusters to retrofit, based on their investment payback periods, and the most efficient interventions to make, based on their implementation cost and potential to save energy. For this purpose, the national databases on Italian building stock (CENED, TABULA, and EPISCOPE) are analyzed, and buildings are clustered based on eight parameters using ML techniques. For each cluster, various standard and advanced retrofit scenarios are proposed and analyzed based on potential energy and CO₂ savings using Monte Carlo Simulation and associated retrofit costs using case-based cost calculations of each building archetype. Moreover, the maximum retrofit investment amounts for each building cluster that can be paid back within a window of 15, 20, and 25 years are calculated using Monte Carlo Simulation, which serves as a benchmark to compare and justify the cost-effectiveness of each cluster's retrofit costs. It is noteworthy that this article is a part of an ongoing research project, and its scope is on the framework structure and required components definition. This framework helps policymakers and public institutions to compare the potential energy savings and costs of each retrofit scenario and each building cluster to make informed and cost-informed decisions. Furthermore, it enables verification of the already activated energy policies effectiveness and promotion of energy efficiency and sustainability initiatives in the built

environment. With the high retrofit costs, the requirement for governmental subsidies is inevitable, and this research's findings prove this assertion.

2 Background

The background consists of three main parts: (a) Building archetypes and clustering methods, (b) Energy saving Estimation Models, and (c) Retrofit cost calculation.

2.1 Building Archetypes and Clustering Methods

Prototype building models represent typical urban building characteristics based on their function, construction year, and climate zone. They are fundamental for various applications such as Urban Scale building performance simulation, benchmarking, quantity estimation of materials, and providing a taxonomy of the stock [23]. Building prototypes or archetypes that are comparable with building clusters offered by this study delineate the typical attributes of a building type. These attributes, such as gross constructed area, footprint, and the window-to-wall ratio of the represented buildings, can be used for quantifying the basic energy use and retrofit cost estimation on a unit-based basis.

There are many previous studies on building prototypes, most of which focus on Europe, due to the TABULA project and its follow-up EPISCOPE project that aimed to establish a common classification of residential buildings for 21 European countries [24]. Many studies on Italian archetypes emerged due to the TABULA/EPISCOPE projects. TABULA project classifies the building stock by the energy properties and construction periods and defines archetypes for each building category [10]. The EPISCOPE project used TABULA archetypes to monitor the renewal activities of the building stock by analyzing different retrofit scenarios at different levels and their energy-saving potentials in three windows of 2020, 2030, and 2050 [25].

Usually, these studies pivot around three groups of integrated information: (a) geometrical data, collectible by GIS, and (b) thermophysical data (e.g., envelope properties, internal loads, HVAC systems and related inputs), and (c) building features (e.g., window-to-wall ratio, thermal zoning, ceiling height, sill height) [26]. It is noteworthy that the TABULA project and some Italian standards are the only databases containing the building envelope data, which is the main focus of this study [10]. Although these databases facilitate the data-gathering process, they are not always in great harmony or comparable to each other.

Carnieletto et al. (2021) [26] investigated the Italian building stock and developed 16 single-family buildings, 16 multi-family buildings, and 14 office building prototypes to be integrated into Urban Building Energy Modeling platforms. Loga et al. (2016) [24] conducted a comparative analysis between a simplified housing stock energy performance model based on the common data structure, specifically TABULA project, with actual energy demand of buildings. Berg et al. (2016) [27] presented a categorization method using physical segmentation and weighting of categories for the historical fabric of Palermo, Italy, to integrate environmental and cultural heritage aspects with energy indicators. Ballarini et al. (2014) [28] presented a methodology to identify reference building in Italian Piedmont region building stock and the energy-saving potential

for each proposed retrofit strategy, indicating a potential to save up to 75% by standard retrofit measures in Italy; however, cost analysis was not provided. Ballarini et al. (2011) [29] presented a cluster analysis of building stock in the Piedmont region based on energy needs for space heating (QH), primary energy for space heating (EPi), net floor area (A), opaque envelope average thermal transmittance (Uop), and window average thermal transmittance (Uw).

Even though archetypes provide valuable information on the building attributes of each cluster, they cannot solely serve as a basis for large-scale cost-benefit analysis of retrofit decision-making. Moreover, the energy results of the archetype might not have high accuracy and need to be compared with average annual energy data documented at national and local levels or through case study benchmarks. For this reason, Machine Learning-based clustering is proposed to cluster buildings with the same attributes in large databases quickly and precisely. In this case, calibration and validation steps are also required.

Clustering techniques aim to group a collection of items into classes or clusters with similar attributes when objects in separate clusters are quite different and distant [30]. As a pattern detection and data-driven tool, clustering methods have been used to detect Underlying structures and anomalies in databases, determine the degree of resemblance between shapes or species, and summarize data using cluster prototypes [31]. The application of clustering in building stock prototyping has proved to be of high accuracy and factual representation of building attributes [16], which is a significant advantage compared to traditional building prototyping. Clustering can be done based on: (a) Monitoring indicators (MI) such as the number of buildings by construction period, Thermal Transmittance (U-value) of building envelope, types of thermal system, and renewable energy sources, and (b) Model assumptions such as residential building stock floor area by the construction period and building size, U-value of bottom and upper floors, and efficiencies of the subsystems of the technical building systems [11]. The choice of the attributes to use for clustering can be made based on available data, experts' opinions, or special requirements for analysis.

2.2 Energy Saving Estimation Models

The saved energy and cost calculation due to retrofit measures are the next step in the process that has been widely studied. Liu et al. (2018) [32] listed a number of benefits resulting from energy retrofit in two groups (a) Direct economic benefits such as heating and electricity cost savings in winter, Air-conditioning operating cost savings in summer, and maintenance cost savings of building envelopes and heating systems, and (b) Indirect benefits such as Increased heat source construction fees, the improved comfort level of living, extended lifetime of the existing building, and enhanced property value.

In an Evolutionary Polynomial Regression-based model proposed by Morano et al. (2018) [1] on the marginal contribution of the energy performance factors on the housing prices in Italy, the influential variables were grouped under (a) Intrinsic characteristics, such as the total surface of the property, the quality of the maintenance conditions, the age of the building, and the EPC label, and (b) Locational characteristics: such as the location of the property and the market conditions. A similar categorization of attributed was proposed by Khodabakhshian and Toosi (2021) [33], where the monetary

effects of the Building-technical attributes were reflected in the Life Cycle Cost of the properties, calculated through the Building Information Modeling (BIM) model, which formed a significant share in the property value alongside the Financial-Legal and Environmental-locational attributes. The study highlighted the importance of better maintenance conditions and lower energy demand of the properties in long-term cost saving and consequently lower Life Cycle Costs. Re Cecconi et al. (2022) [16] proposed a Monte Carlo simulation-based energy-saving assessment for different building clusters in Lombardy region under different retrofit scenarios to achieve a better EPC energy label. Energy and CO₂ simulation with software like TOBUS and EnergyPlus, Life Cycle Assessment, Genetic Algorithm, and fuzzy set methods are some other applied techniques for energy-saving calculation and retrofit decision-making [34, 35].

2.3 Retrofit Cost Calculations

Cost encountered during a retrofit process on a phased basis can be grouped into (a) Costs before the energy retrofit, such as Investigation and testing costs, decision-making costs, and design costs, (b) Costs during the energy retrofit process such as building envelope and heating system retrofit, (c) Costs during Operation stage such as service and maintenance costs, and (d) Costs during demolition stage [32]. Moreover, the costs are not merely limited to materials or equipment, and they include a wider range of workforce payments, building permits, tax, etc.

Utility costs in Italy are predicted to increase by 250% by 2023 from 1.2 € per M3 in 2022 to 2.7 € per M3 in 2023. The natural gas cost by March 2020 in Italy is 0.163 € per kWh, and the electricity cost is 0.316 € per kWh [36]. Therefore, retrofit strategies to reduce the consumption of energy in the forms of electricity and gas can save a significant amount of capital. Italian government has started some programs to encourage building retrofit projects in the forms of (a) tax reductions for up to 100000€ deductible per unit in over ten years, where individuals pay the retrofit costs from pocket and get up to 65% deduction of expenses through tax reduction, and (b) Super Bonus of 110% retrofit cost return, as a result of which more than 307000 assertions with a total investment of more than 51.200.000.000 € were registered, which is estimated to increase to 1.3 million housing units and corresponding subsidized expense of 57.2 bn€ by 2028, with a promising vision up to 25.8 Bn€ revenue emerged from the energy saving and increased property value. Table 1 presents the detailed investment cost at Italy and Lombardy Region levels for each building type, of which 12.3% were condos with an average investment of 585.795,17 €, 58.2% were single-family housing with an average investment of 113.327,08 €, and 35.1% were functionally independent housings with an average investment of 97025.57 € [9, 37]. This gives a decent benchmark estimation for future target groups.

The main steps of the Superbonus application are (a) Defining a typical intervention strategy, (b) Estimating the work components and products required for each activity such as isolation or heating systems, (c) The cost-benefit analysis for different stakeholders.

Table 1. Total investment cost for retrofit projects in Italy and Lombardy region documented during the super eco bonus project.

		Condos	Single family housing	Functionally independent housings	Total
Italy	Number of assets	37838 (12.3%)	178785 (58.2%)	90562 (29.5%)	307191
	Average Investment cost	585795.17 €	113327.08 €	97025.57 €	
	Total Investment cost	22163995415 €	20261182566€	8786829720€	51212808110 €
Lombardy Region	Percentage	6322 (13.4%)	24372 (51.5%)	16593 (35.1%)	47288
	Average Investment cost	649946.81 €	114361.79 €	103464.47 €	
	Total Investment cost	4108963751 €	2787225593 €	1716785966 €	8613063031 €

3 Methodology

The research methodology is depicted with reference to literature in the field and with a data-driven approach to automate the decision-making process. Based on Ballarini et al. (2014) [28], the cost-efficient analysis of retrofit strategies requires the preliminary definition of reference buildings and is followed by the following steps:

1. Energy performance calculation of the reference buildings to assess the baseline of the energy performance;
2. Definition of sets of energy retrofit measures to apply to the reference buildings;
3. Energy performance calculations to evaluate the energy performance after the retrofit measures;
4. Calculation of the life cycle costs using net present valuation;
5. Assessment of the cost-optimal set of measures to optimize (and increase) the energy performance of the reference buildings.

To depict a holistic cost-efficient energy retrofit decision support system framework, the following steps are proposed:

6. Clustering buildings in CENED database using their construction period, Energy label, and building technologies.
7. Associating these clusters with proposed archetypes in national databases
8. Defining Retrofit strategies.

9. Retrofit cost calculation using the footprint of each building archetype (cluster) and unit prices of each intervention.
10. Conducting energy saving simulation using Monte Carlo simulation to assess the energy saving potential of each cluster using different retrofit strategies.
11. Conducting maximum investment assessment retrievable in different payback periods using Monte Carlo simulation based on the cost of potential saved energy and interest rate.
12. Comparing the retrofit cost range for each cluster with the payback period window to make investment decisions.
13. Prioritize building clusters to intervene and retrofit strategies based on their potential energy saving and associated cost.

Figure 1 presents the proposed research framework. A detailed clustering process and clusters' energy-saving simulation are presented in Re Cecconi et al. 2022 [16]. The clustering criteria proposed were the Year of Construction (YoC), Energy Label (EL), and building components Thermal Transmittance (U_Value), and by the Gaussian mixture partitional model (GMM). During that phase, the energy performance of each cluster was calculated using statistical distribution and in the form of Primal Energy Demand (EPH); consequently, the possible energy saving of each building cluster was estimated by Monte Carlo Simulation through different scenarios to reach clusters with better EL. Logically, buildings with the same YoC and better EL are already retrofitted buildings and can portray post-retrofit energy performance. With the energy prices per unit data available, it is easy to calculate the potential cost saving on an annual basis for each of the clusters.

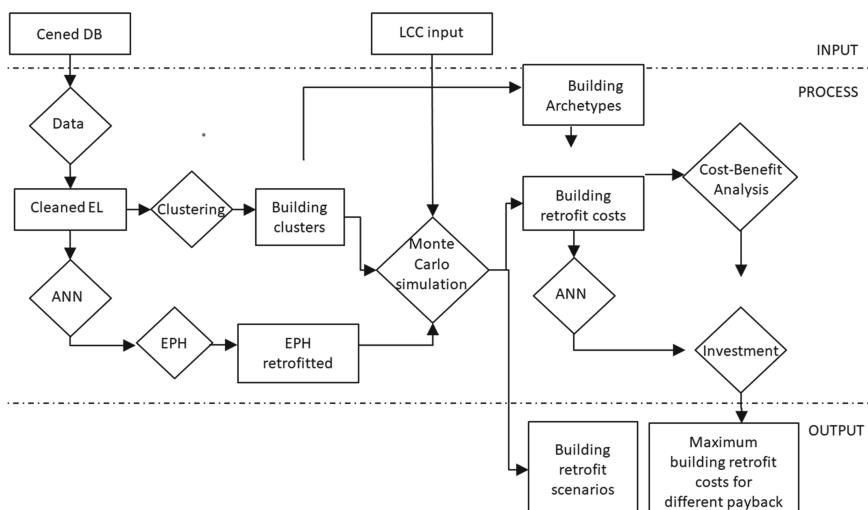


Fig. 1. Cost-efficient energy retrofit decision making framework.

The criteria used in this paper is a clustering-based U-value of building elements, which was indirectly influenced by the YoC. The reason for choosing these criteria is to

correspond with the building archetypes proposed in TABULA project, where for each construction period and building type, the typical building technology and corresponding U-values are listed.

This paper focuses more on the cost estimation process. The main challenge with the CENED database is that the footprint data is not available, and only building volume is indicated, which can be translated into a variety of alternatives with different ceiling heights and the number of floors, and consequently, different footprint areas. Therefore, the first step to overcome this challenge was to propose the concept of the archetype and associate the building clusters with the existing building archetypes in the Lombardy region, like the TABULA project. With this information in hand, it is possible to delineate the common building archetype related to each cluster and calculate the retrofit cost using the unit-based prices in national databases, as the average height and floor area of each archetype is mentioned in TABULA. By dividing the overall retrofit cost by the average footprint of the given building type, it is possible to have a parametric estimation of cost per unit for each of the retrofit alternatives.

In TABULA, buildings are grouped under eight construction intervals and are within one of the following size classes: (a) Single-family house, (b) Terraced house, (c) Multi-family house, and (d) Apartment block; and the definition of building types follows one of the following approaches:

1. The “Real Example Building” (ReEx), identifies the building type by experience and expert opinion.
2. The “Real Average Building” (ReAv) identifies the building type through statistical analysis of a database.
3. The “Synthetical Average Building” (SyAv) identifies the building type as an “archetype” based on the statistical analysis of a database, which is a virtual building characterized by a set of properties statistically detected in a building category [28].

This paper uses the SyAv approach based on the building clusters identified in the CENED database. The features used for clustering are construction period, Energy labels, and U-values of building components. The footprint area, height, and number of floors of each building type are retrievable from TABULA.

The next step is defining the retrofit scenarios. It is noteworthy that building archetypes have an assigned value as their U-value of building components, which should be improved during a retrofit process. Retrofit scenarios applied to the building envelope can be “standard” or “advanced”, as described in TABULA. However, the new Minimum Requirements Decree Law in 2021 [38] requires the following post-retrofit measures in Climatic zones E and F, which are the climatic zones in the Lombardy region:

U-value_Walls: 0.26–0.28 W/m²K.

U-value_Roofs: 0.22–0.24 W/m²K.

U-value_Floors: 0.28–0.29 W/m²K.

U-value_Window: 1–1.40 W/m²K.

To meet these measures, various retrofit options and insulation alternatives are available in the market. A detailed list of each material’s properties and the unit-based price is listed in the Lombardy Region price list [39].

Once the pre-retrofit and desired post-retrofit thermal transmittance of the building component is determined, it is possible to calculate the thickness and implementation

cost of the insulation layers, which would result in the desired post-retrofit thermal transmittance. The maximum and minimum retrofit prices are calculated based on the Lombardy Region price list for each building archetype.

Finally, the cost-benefit analysis is done on a discounted cash flow basis to depict the Capex cost of intervention along with the annual saved cost of energy to calculate the payback period of the intervention cost. The maximum retrofit investment is calculated using Eq. 1, where the payback period is set to 3 amounts of 15, 20, and 25 years, and the Net Annual Cashflow is equal to saved energy cost, calculated by Monte Carlo simulation.

$$\text{Payback Period} = \frac{\text{Initial Investment}}{\text{Net annual Cash Flow}} \quad (1)$$

As a result of this calculation, the payback period distribution of all the assets in a cluster (archetype) is calculated, serving as a benchmark to assess the cost-efficiency of each retrofit scenario. If the calculated retrofit cost for a certain archetype is higher than the maximum investment amount for its cluster payback period window, it is not an optimal option for retrofit investment. On the other hand, if a certain archetype's retrofit cost is in the window of a shorter payback period, it is an optimal option to invest in, as it logically has higher energy savings and lower retrofit costs. These comparisons help policymakers prioritize the building archetypes for investment.

4 Results

As a result of the clustering process, eight building clusters/archetypes are identified in CENED database, presented in Table 2. Seven of these clusters perfectly fall into a specific construction period group and can be easily aligned with TABULA data, which also grouped building archetypes based on their construction period. Table 3 presents the TABULE building archetypes with the same YoC grouping. These two tables provide complementary information on the classification of the building archetypes and the average characteristics of which, like the net floor area or the number of floors, which will be used for assessing the saved energy and retrofit costs.

As archetype buildings are not actual buildings, but a representation of the average of a statistical sample regarding the assets' characteristics (SyAv-ReEx), this research proposed imaginary case studies as archetype representatives to implement the cost-efficient decision-making framework. The dimensions, YoC, and thermal characteristics of these case studies correspond to the archetypes listed in Table 3. The archetype study is used because of the missing value on some essential characteristics like floor area in the CENED database, which is a common limitation in national databases hindering the full application of automated models.

Case-study 1 is a single-family house constructed before 1930, consisting of two floors, with a net floor area of 139 M² and a height of 3.8 M (533/139). Case-Study 2 is a Multi-family house, constructed between 1945 and 1960, consisting of 4 floors and 20 units, with a net floor area of 1164 M² and height of 3.7 (4388/1164). Table 4 shows the current and post-retrofit U-values of the two case studies based on national standards and requirements of the retrofit process.

Table 2. Building clusters identified in cened database.

Cluster	Before 1930	1930–1945	1946–1960	1961–1976	1977–1992	1993–2006	After 2006
1	13990	0	0	0	160	0	886
2	1	0	0	0	35937	0	0
3	0	195	0	4032	0	36196	0
4	2616	341	2016	1144	1607	28	28034
5	0	0	0	30606	0	2251	0
6	0	7044	0	0	0	0	0
7	0	0	13297	0	0	0	1
8	371	268	881	3197	492	133	482

Table 5 presents the calculation of maximum and minimum retrofit costs for the two case studies based on the required insulation thickness and unit price and required window replacements. This gives a range for the retrofit costs of each archetype, considering the least and the most expensive options in the market. The first case study archetype has a retrofit cost range of 128–375 €/M², and the second case study archetype has a retrofit cost range of 79–209 €/M².

Based on a report issued by the National Association of Building Construction on Superbonus cost breakdowns [37], the share of insulation and windows replacement is 41% of the entire cost. The other expenses are materials by 21.1%, design by 9.7%, Tax by 9.1%, building operations by 8.7%, scaffolding by 8.1%, and safety by 2.3%. Therefore, the calculated values should be multiplied by (100/41.1) to get the final retrofit cost. For the first case study, the range will be 311–911 €/M², and for the second one it will be 192–508 €/M². Given the inflation Italy is currently experiencing, it is more likely that the prices are negatively skewed toward the upper end of these ranges. For this purpose, a beta distribution is considered with $\alpha = 15$ and $\beta = 5$. The distribution of costs is shown in Fig. 2. Based on this distribution, the Maximum Likelihood Estimate for the first case study is 777 €/M² and for the second case study is 437 €/M².

Now, after knowing the range of potential retrofit costs for each case study, it is required to calculate the average investment reimbursable during three payback periods of 15, 20, and 25 years to see if the building retrofit costs of the archetypes fall into the proposed amount and how long the payback periods would be. For this purpose, the potential saved energy for each cluster is calculated using the Monte Carlo simulation, as depicted in Fig. 3 for cluster/building archetype 7. Monte Carlo runs the post-retrofit energy-saving simulation for all the building assets in each cluster and provides a final average amount. Moreover, the cost of saved energy per post-retrofit EP_h and EP_h savings for each cluster, or the revenue, is depicted in Fig. 4, using per unit cost of the saved energy based on national reports.

Based on the figures above, an energy retrofit on building clusters with low primal energy demand (EP_h) would give meager benefits (i.e., savings in fuel expenditure) that would not pay off expensive retrofit interventions. Clusters with a high Maximum

Table 3. Italian building archetypes in TABULA relatable with the research clusters [10].

Building size class	Building age class	Gross heated volume [m^3]	Net floor area [m^2]	Gross floor area [m^2]	Compactness factor [m^{-1}]	Number of floors	Number of apartments
Single family houses	1	533	139		0.77	2	1
	2	448	115		0.82	2	1
	3	455	116		0.81	2	1
	4	583	162		0.75	2	1
	5	679	156		0.73	2	1
	6	725	199		0.72	2	1
	7	605	172		0.73	2	1
	8	607	174		0.72	2	1
Terraced houses	1	500	123		0.51	2	1
	2	478	112		0.51	2	1
	3	428	113		0.49	2	1
	4	400	111		0.51	2	1
	5	374	89		0.52	2	1
	6	434	125		0.69	2	1
	7	426	111		0.67	2	1
	8	519	127		0.64	2	1
Multi-family houses	1	2684		657	0.55	2	5
	2	4113		1306	0.54	2	16
	3	4388		1164	0.51	4	20
	4	3076		961	0.51	3	12
	5	3074		934	0.54	5	10
	6	4136		1209	0.48	3	12
	7	3526		1120	0.54	3	15
	8	2959	829		0.54	3	13
Apartment blocks	1	3745		1058	0.35	5	16
	2	11029		2880	0.47	4	40
	3	7197		2249	0.46	5	30

(continued)

Investment (MI) like Cluster 5, quite the opposite, are made of buildings with poor energy performances and thus would pay off retrofit interventions with high costs per square meter of the heated surface.

Table 3. (*continued*)

Building size class	Building age class	Gross heated volume [m ³]	Net floor area [m ²]	Gross floor area [m ²]	Compactness factor [m ⁻¹]	Number of floors	Number of apartments
	4	5949		1763	0.46	4	24
	5	9438		2869	0.46	8	40
	6	12685		4125	0.37	6	48
	7	9912		3271	0.43	6	36
	8	8199	2124		0.4	7	31

Table 4. Current and post retrofit U values of research case studies' building components.

Case study	Roof/ceiling material	U _{roof}	Floor material	U _{floor}	Wall material	U _{wall}	Window material	U _{window}
1	Pitched roof with wood structure and planking	Current: 2.5	External: Concrete floor on soil- Internal floor/ceiling: Beam-wooden slab	Current: 1.22	Masonry with lists of stones and bricks (40 cm)	Current: 2.58	Single glass, metal frame without thermal break	Current: 5.7
		Post Retrofit: 0.22		Post Retrofit: 0.28		Post Retrofit: 0.26		Post Retrofit: 1
2	Ceiling with reinforced concrete	Current: 1.65	Floor with reinforced concrete	Current: 1.3	Solid brick masonry (38 cm). Hollow wall brick masonry (30 cm)	Current: 1.48	Single glass, wood frame	Current: 4.9
		Post Retrofit: 0.22		Post Retrofit: 0.28		Post Retrofit: 0.26		Post Retrofit: 1

Figure 5 depicts the average retrofit investment cost reimbursable in three payback periods, 15, 20, and 25 years, using Monte Carlo Simulation. Based on the potential annual revenue of the saved energy in the cash flow of each cluster, it is possible to calculate the maximum reimbursable capital cost of retrofitting in a certain amount of time. The shorter the payback period, the less capital cost for energy retrofit can be spent. Comparing the unit-based retrofit cost range of each cluster with these average numbers, it is possible to identify and prioritize the clusters with short payback periods. For both case studies, i.e., single-family building in cluster 1 and multi-family building in cluster 7, the most likely retrofit costs, 777€/M² and 437€/M², respectively, are much higher than the reimbursable value during the three payback periods. Therefore, if the

Table 5. Calculation of maximum and minimum retrofit cost per M² for the two case studies.

Case Study			Unit price (€)	Area/count	Total price (€)
1	Wall insulation	Most economic option: 1C.10.250.0040	16.54	259.6	4293.78
		Most expensive option: 1C.10.300.0010	94.01	259.6	24404.99
	Roof insulation	Most economic option: 1C.10.050.0020	23.39	70	1637.3
		Most expensive option: 1C.10.050.0120	51.71	70	3619.7
	Floor insulation	Most economic option: 1C.10.200.0050	26.46	140	3704.4
		Most expensive option: 1C.10.200.0080	142.79	140	19990.6
	Window replacement		1400	6	8400
	Total cost	Most economic option			18035.48
		Most expensive option			52215.29
	Cost per M ²	Most economic option			128.8
		Most expensive option			375.64
2	Wall insulation	Most economic option: 1C.10.250.0040	15.43	972.5	15005.675
		Most expensive option: 1C.10.300.0010	91.78	972.5	89256.05
	Roof insulation	Most economic option: 1C.10.050.0020	21.29	291	6195.39

(continued)

entire cost is meant to be financed by individuals and homeowners, social engagement

Table 5. (*continued*)

Case Study			Unit price (€)	Area/count	Total price (€)
		Most expensive option: 1C.10.050.0120	51.71	291	15047.61
Floor insulation	Most economic option: 1C.10.200.0050	26.46	582	15399.72	
		142.79	582	83103.78	
Window replacement		1400	40	56000	
Total cost	Most economic option				92600.78
	Most expensive option				243407.44
Cost per M ²	Most economic option				79.5
	Most expensive option				209.11

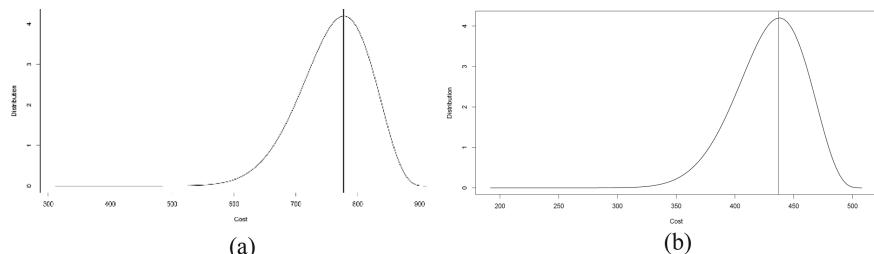


Fig. 2. **a** Probabilistic distribution of costs for case study 1. **b** Probabilistic distribution of costs for case study 2.

will be relatively low, as this might not seem like a cost-efficient investment. Therefore, the Italian government is offering subsidies and bonuses to help homeowners conduct retrofit activities at a much lower cost, which greatly motivates retrofit projects. With the role of government funding being inevitable, it is a win-win situation for both the government to have more retrofit projects initiated by residents and for homeowners to conduct the activities at a lower cost and have more energy-efficient homes at the end of the day. This will also contribute to lower energy consumption by the building sector on the national level.

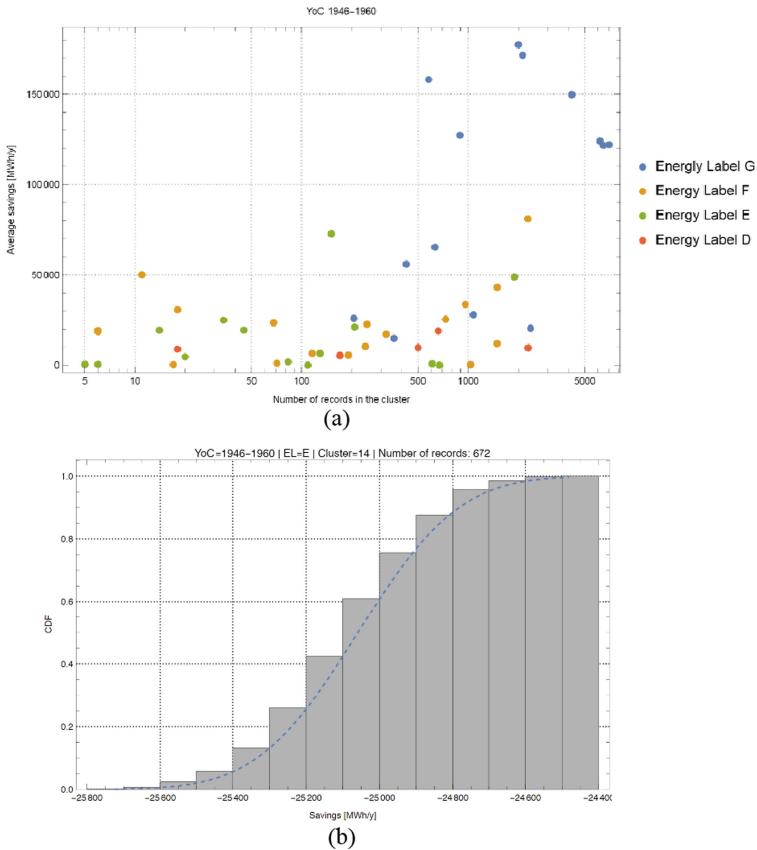


Fig. 3. **a** Monte carlo simulation of energy saving for building cluster 7(1946–1960). **b** Energy saving (EP_h Saving) distribution for assets cluster 7 with EL E.

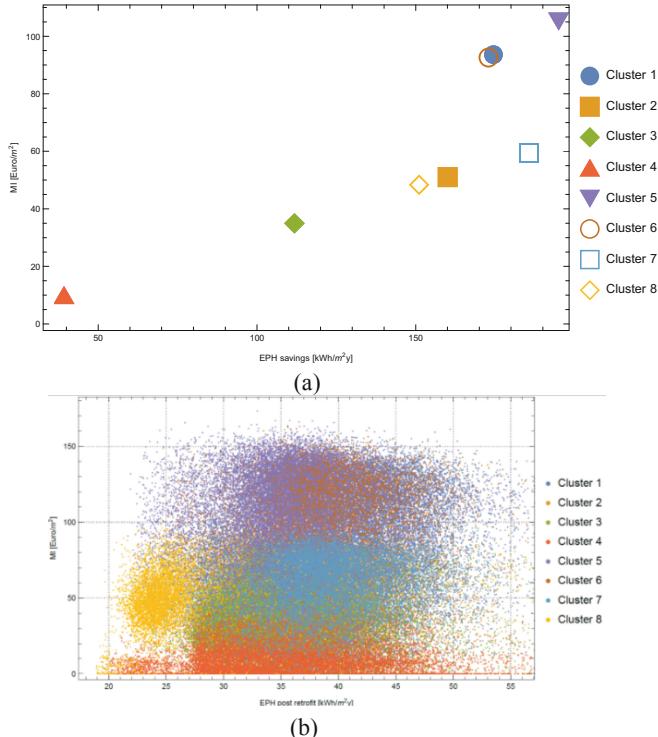


Fig. 4. **a** Average saved energy cost for each building cluster per EPH savings based on Monte Carlo simulation. **b** Saved energy cost for all buildings divided per cluster.

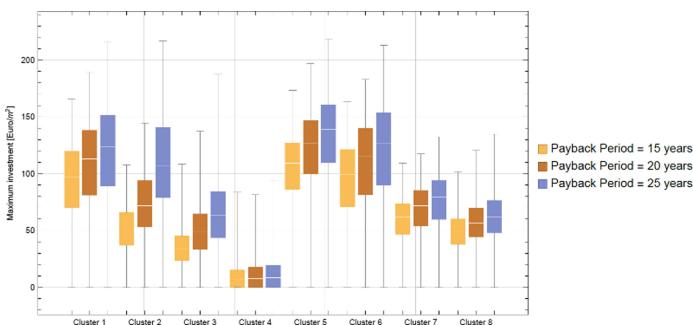


Fig. 5. Average retrofit investment cost reimbursable in three payback periods of 15, 20, and 25 years for the 8 building clusters.

5 Conclusion

This study proposes a data-driven decision-making framework that performs the retrofit energy-saving prediction in an automated and quick fashion. Benefiting from the potential of ML-based approaches to analyze and extract insights from huge databases, this

research focused on Italian national building databases to propose an accurate and optimal energy-saving and retrofit cost prediction method, which can overcome the shortcomings of conventional bottom-up approaches, the manual, time-consuming, and complicated energy saving calculation. This method is applicable to various scales of analysis, from single assets to urban levels. Moreover, it provides a comparative analysis of various retrofit alternatives and scenarios based on their potential to save energy and implementation costs, helping optimize the retrofit scenario and materials choice. As evident in the results, the proposed method has huge potential to be implemented at the national level and for policymaking purposes to choose the most profitable and cost-effective building types to invest in based on the Monte Carlo simulation of different building clusters. It is in alignment with EU goals of energy consumption and CO₂ emission reduction and built environment sustainability goals, calling for vaster actions and funding from governmental agencies to initiate retrofit projects. Therefore, the proposed framework can address the main research questions and objectives derived from the research gap in state of the art, adding a practical, systematic, holistic, and cost-effective perspective to the development of ML-based models in building retrofit research. Furthermore, this research is a clear example of how the construction industry can benefit from digital tools and AI on strategic and policymaking levels, which has not been studied much.

The main shortcomings of the research are the missing of buildings' footprint data in CENED database that hindered the complete automation of the retrofit cost prediction process by Monte Carlo simulation. This is a critical issue since different building dimensions and footprints could result in the same volume mentioned in the database, and the unit-based prices of insulation materials are per square meter for the floor area. To overcome this issue, the archetype concept and parametric calculation of costs were proposed. However, comparing the parametric cost with actual energy saving is not the most accurate method. Another research limitation was the existence of missing values, outliers, and unreliable data in the database, which is caused by the existence of various sources of audit and auditors' errors while collecting data.

As future studies, this ongoing research will use geo-locating techniques to find the address of the building assets in the database by GIS and calculate their footprints. Consequently, the retrofit cost of each building can be calculated through the proposed framework in an automated fashion, increasing the prediction precision significantly. Another solution could be the parametric simulation of all buildings in each cluster and conducting a Monte Carlo simulation of retrofit costs for different possible building dimensions to achieve a cost distribution for each building and the entire cluster. Both solutions will make the comparison more accurate as they are implemented on the whole database in a fully automated fashion. Moreover, the implementation of partial retrofit measures can be studied and analyzed through sensitivity analysis to find the most critical building components to be triggered.

Finally, the study shed light on governmental funding and subsidies' important role in promoting building retrofit in European countries. Although initially, the investment payback period calculation for the retrofit projects was the purpose of this study, the case studied and unit-based prices of insulations and retrofit scenarios proved the retrofit cost will not be easily paid back in the near future. Even though this research objective was

not fulfilled, the findings contribute to highlighting the importance of retrofit projects; as with nonrenewable energy sources decreasing, the energy prices will increase, and the retrofit projects and their energy saving potentials will become more intriguing.

References

- Morano, P., Tajani, F., Di Liddo, F., Guarnaccia, C.: The value of the energy retrofit in the Italian housing market: two case-studies compared. *WSEAS Trans. Bus. Econ.* **15**, 249–258 (2018)
- Agenzia Nazionale per l'Efficienza Energetica dell'ENEA. Rapporto annuale efficienza energetica 2017. <http://www.eficienzaenergetica.enea.it/>, last accessed 2023/01/21
- Law No. 373, Norme per il contenimento del consumo energetico per usi termici negli edifici. Italy (1976)
- Energetica AN per l'Efficienza, Dell'ENEA. Strategia per la Riqualificazione Energetica del Parco Immobiliare Nazionale (STREPIN) <http://www.sviluppoeconomico.gov.it/>. Accessed 21 Jan 2023
- European Commission: Communication from the Commission to the European Parliament and the Council. Brussels. (2014). <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52014DC0520/>, last accessed 2023/01/11
- Jafari, A., Valentin, V.: Proposing a conceptual decision support system for building energy retrofits considering sustainable triple bottom line criteria. In: Construction Research Congress 2018 Sustainable Design Construction Education, pp. 553–563 (2018)
- Lowe, R., Chiu, LF.: Innovation in deep housing retrofit in the United Kingdom: the role of situated creativity in transforming practice. *Energy Res. Soc. Sci.* **63**, 101391 (2020)
- Law No. 232. Bilancio di previsione dello Stato per l'anno finanziario 2017 e bilancio pluriennale per il triennio 2017–2019. Italy
- Agenzia Nazionale per le nuove tecnologie l'energia e lo sviluppo economico sostenibile. Super Ecobonus 110 % (2022)
- Corrado, V., Ballarini, I., Cognati SP.: Typology approach for building stock: national scientific report on the TABULA activities in Italy, p. 104 (2012)
- Ballarini, I., Corrado, V.: A new methodology for assessing the energy consumption of building stocks. *Energies* **10**(8) (2017)
- Kavgic, M., Mavrogianni, A., Mumovic, D., Summerfield, A., Stevanovic, Z., Djurovic-petrovic, M.: A review of bottom-up building stock models for energy consumption in the residential sector. *Build. Environ.* **45**(7), 1683–1697 (2020)
- Fumo, N.: A review on the basics of building energy estimation. *Renew. Sustain. Energy Rev.* **31**, 53–60 (2014)
- Aydinalp-Koksal, M., Ugursal, V.I.: Comparison of neural network, conditional demand analysis, and engineering approaches for modeling end-use energy consumption in the residential sector. *Appl. Energy* **85**, 271–296 (2008)
- Yan, B., Hao, F., Meng, X.: When Artificial Intelligence meets building energy efficiency, a review focusing on zero energy building. *Artif. Intell. Rev.* **54**(3), 2193–2220 (2021)
- Re Cecconi, F., Khodabakhshian, A., Rampini, L.: Data-driven decision support system for building stocks energy retrofit policy. *J. Build. Eng.* **54**, 04633 (2022)
- Sun, K., Hong, T., Taylor-Lange, S.C., Piette, M.A.: A pattern-based automated approach to building energy model calibration. *Appl. Energy* **165**, 214–224 (2016)
- Seyedzadeh, S., Pour Rahimian, F., Oliver, S., Rodriguez, S., Glesk, I.: Machine learning modelling for predicting non-domestic buildings energy performance: a model to support deep energy retrofit decision-making. *Appl. Energy* **279**, 115908 (2020)

19. Khodabakhshian, A., Rampini, L., Re Cecconi F.: Data driven framework to select best retrofitting strategies. In: Research in Building Engineering, pp. 67–75. ETS, Universitat Politècnica de València (2022)
20. Pillai, V.S., Matus, K.J.M.: Towards a responsible integration of artificial intelligence technology in the construction sector. *Sci. Public Policy* **47**(5), 689–704 (2020)
21. Nässén, J., Holmberg, J.: Quantifying the rebound effects of energy efficiency improvements and energy conserving behaviour in Sweden. *Energy Effic.* **2**, 221–231 (2009)
22. Deb, C., Dai, Z., Schlueter, A.: A machine learning-based framework for cost-optimal building retrofit. *Appl. Energy* **294**, 116990 (2021)
23. Kordas, O., Pasichnyi, O.: Data-driven building archetypes for urban building energy modelling. *Energy* **181**, 360–377 (2019)
24. Loga, T., Stein, B., Diefenbach, N.: TABULA building typologies in 20 European countries—making energy-related features of residential building stocks comparable. *Energy Build.* **132**, 4–12 (2016)
25. Stein, B., Loga, T., Diefenbach, N.: Monitor Progress Towards Climate Targets in European Housing Stocks Main Results of the EPISCOPE Project. Institut Wonen und Umwelt, Darmstadt, Germany (2016)
26. Carnieletto, L., Ferrando, M., Teso, L., Sun, K., Zhang, W., Causone, F., et al.: Italian prototype building models for urban scale building performance simulation. *Build Environ.* **192** (2021)
27. Berg, F., Genova, E., Broström, T.: Interdisciplinary building categorization—a method to support sustainable energy efficiency strategies in historic districts. In: Central Europe Towards Sustainable Building, pp. 41–8 (2016)
28. Ballarini, I., Corgnati, S.P., Corrado, V.: Use of reference buildings to assess the energy saving potentials of the residential building stock: the experience of TABULA project. *Energy Policy* **68**, 273–284 (2014)
29. Ballarini, I., Corgnati, S.P., Corrado, V.: Improving energy modeling of large building stock through the development of archetype buildings, In: Proceedings of Building Simulation, 12th Conference of International Building Performance Simulation Association, Sydney (2011)
30. Pfitzner, D., Leibbrandt, R., Powers, D.: Characterization and evaluation of similarity measures for pairs of clusterings. *Knowl. Inf. Syst.* **19**(3), 361–394 (2019)
31. Jain, A.K.: Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **31**(8), 651–666 (2010)
32. Liu, Y., Liu, T., Ye, S., Liu, Y.: Cost-benefit analysis for Energy Efficiency Retrofit of existing buildings: a case study in China. *J. Clean. Prod.* **177**, 493–506 (2018)
33. Khodabakhshian, A., Toosi, H.: Residential real estate valuation framework based on life cycle cost by building information modeling **27**, 1–15 (2021)
34. Caldas, L.G., Norford, L.K.: A design optimization tool based on a genetic algorithm. *Autom. Constr.* **11**, 173–184 (2002)
35. Sharma, A., Saxena, A., Sethi, M., Shree, V.: Life cycle assessment of buildings: a review. *Renew. Sustain. Energy Rev.* **15**, 871–875 (2011)
36. Electricity prices. https://www.globalpetrolprices.com/electricity_prices/. Accessed 23 Jan 2023
37. Direzione Affari Economici F e CS. SUPERBONUS 110 % Quanto costa davvero allo Stato? (2022)
38. Ministero dello Sviluppo Economico: Requisiti specifici per gli edifici esistenti soggetti a riqualificazione energetica, pp. 1–6 (2015)
39. Regione Lombardia.: Prezzario delle Opere Pubbliche (2011)



Text Summarization for Call Center Transcripts

Ishrat Ahmed¹(✉), Yu Zhou², Nikhita Sharma², and Jordan Hosier²

¹ University of Pittsburgh, Pittsburgh, USA
isa14@pitt.edu

² Vail Systems, Inc., Chicago, IL, USA
{yzhou, nsharma, jhosier}@vailsys.com

Abstract. While text summarization of transcripts in call centers is needed for detailed analysis, it presents challenges stemming from the call itself (context switching among speakers, cross talk, etc.) and from the resulting transcript (ASR transcription errors). This work aims to develop a summarization model suitable for on-premise deployment at call centers by fine-tuning pre-trained open-source large language models, assisted with reference summaries generated by GPT-3. The results are analyzed using ROUGE and human evaluation scores, and the correlation of these two metrics is examined. A fine-tuned BART model outputs satisfactory summaries with a human evaluation score of 6.95, approaching the GPT-3 score of 7.69.

Keywords: Summarization · Call Transcript · Large Language Model · GPT-3

1 Introduction

Call centers process a large volume of calls each day, of which only a small portion are selected later for manual review [1]. To analyze these calls (i.e., categorizing issues reported by the customers and identifying gaps and opportunities in provided services), it is necessary to automatically generate a text summary for each call. In recent years, transformer-based large language models (LLM) have shown promise in text summarization. These generative language models are especially skilled in extracting key contents from long documents and producing abstractive summaries. These summaries are well suited for transcripts of conversations between customers and call center agents, as these calls often last many minutes and cover a wide range of topics.

While pre-trained LLMs effectively summarize various text documents, call center transcripts present some unique challenges [2]. The lower-quality audio, often recorded at 8K sampling rate, and noisy environments result in a high word error rate (WER) when transcribed by an automatic speech recognition (ASR) model, as ASR models are commonly trained using higher-quality clean audio datasets such as LibriSpeech [3]. Cross-talk can further confuse both acoustic

and language models within the ASR engine. In addition, multiple topics can be scattered across utterances, so producing a concise summary is non-trivial.

The common LLMs used for text summarization includes BART [5], Pegasus [7], T5 [6], and more recently GPT-3 [4]. While GPT-3 produces a satisfactory summary, it can only be accessed via external APIs, which is unsuitable for call center services due to privacy concerns over personal identifiable information (PII). In this work, we experimented with open-source LLMs to summarize call transcripts that are suitable for deploying on-premise. However, these pre-trained open-source LLMs do not perform well on call transcripts without fine-tuning using domain-specific data. Fine-tuning requires reference summaries of a large number of call transcripts to accommodate the vast variation in call center services. This is an expensive task when using human annotators. As a result, we adopt an approach that leverages the capability of GPT-3 to generate reference summaries for properly redacted call transcripts. These summaries are then used as training samples to fine-tune open-source LLMs so that their output can resemble the quality of GPT-3 on our domain-specific call transcripts. The models are evaluated using ROUGE scores and human evaluation scores, and the correlation between these two metrics is examined.

The main contributions of this work are: (1) fine-tuning open source LLMs for summarization tasks and using GPT-3 to generate ground truth of training samples; (2) demonstrating LLMs can generate summaries for imperfect texts, and (3) analyzing the correlation between ROUGE scores and human evaluation scores across the studied LLMs.

2 Related Work

Automatic text summarization has been extensively studied in the Natural Language Processing (NLP) domain. Traditional summarization methods can be categorized into two classes: extractive and abstractive. The extractive approach selects the most important words and sentences within the original document to generate a summary. In contrast, the abstractive approach generates a whole new summary based on the original text, often including text that doesn't appear in the original document. Early methods such as the TextRank [8] algorithm and Latent Semantic Analysis [9] focused on extractive summarization. More recently, transformer-based LLMs such as BART [5], Pegasus [7], and T5 [6] have been utilized for both types of summarization methods. LLMs are particularly utilized in spoken dialogue summarization. Analysis of such dialogues (e.g., online meetings, customer service calls, etc.) combines speech recognition efforts with text summarization [10].

Much work in summarization has been specifically aimed at call transcripts, as it presents unique challenges (i.e., noisy environments, cross talk, etc.). Chandramouli et al. [11] presented an unsupervised approach to extract meta-data from call transcripts using BERT [12], including key topics and intents to classify transcripts into pre-defined categories. They used an unsupervised method due to the expense of tagging call transcripts. Biswas et al. [13] developed a method

combining topic modeling and sentence selection with punctuation restoration to condense ill-punctuated call transcripts to produce readable extractive summaries. Uma and Sityaev [20] evaluated several extractive text summarization techniques (e.g., Text Rank, BERTSum, etc.) to produce summaries for call center transcripts, focusing in particular on abstractive summaries for call transcripts. Extractive summarization of the call transcripts may be inappropriate due to a high rate of ASR transcription error and multiple topics scattered across utterances from multiple speakers in the transcripts. Stepanov et al. [1] describe an abstractive summarization technique where hand-written templates are filled with entities detected in the transcript using Named Entity Recognition (NER), PoS-tagging, chunking, and dependency parsing.

Our work uses pre-trained and fine-tuned LLMs to generate abstractive summaries of call transcripts directly. Because the call transcripts may cover a wide range of topics, fine-tuning LLMs will likely generate summaries that provide a wider conversational perspective. To evaluate the model-generated summaries, we compare them against ground truth or reference summaries. Ground truth or reference summaries can be derived manually by human readers, [13], by using the title or heading text, and topic descriptors [14]. Generating human summaries is expensive and non-scalable, while using topic descriptors as ground truth summaries can be vague and may lack details about the call.

Recently, GPT-3 has been used as a source of reference summaries [2, 15]. Asi et al. [2] used GPT-3 generated pseudo-labels per call segment, combined with human labels as summaries to fine-tune their model on conversational text. Similarly, Wang et al. [15] leveraged GPT-3 as a reference summary generator. In this vein, we use GPT-3 to generate short summaries for the call transcripts, which are used as the reference for fine-tuning and evaluation.

3 Methodology

3.1 Dataset

The data comes from two PII redacted sources: a financial service and a food ordering service. It consists of 5,452 call transcripts between callers and customer service agents. The separation between customer and agent text is removed in the transcripts, and the dialogue is combined to form a single, long-form paragraph per call. Among these call transcripts, 5,000 are used for training, 389 are set aside as validation datasets for hyper-parameter tuning, and the remaining 63 transcripts are used as the test set. The test set is kept small to facilitate the human evaluation of the model-generated summaries.

3.2 Experiment Details

Pre-Trained Models We use pre-trained BART [5], Pegasus [7], and T5 [6] models for summarization. Both BART and Pegasus models are trained on the Extreme Summarization (XSum) [16] dataset. The data used to pre-train the

Table 1. Pre-Trained Models

Models	Description	Parameters
bart.large.xsum	bart.large fine-tuned on Xsum	400 M
pegasus.xsum	pegasus fine-tuned on Xsum	568 M
T5-small	T5 pre-trained on C4	60 M

T5 model is known as C4 (Colossal Clean Crawled Corpus (700GB)) [17]. These models are summarized in Table 1.

Fine-Tuned Models We fine-tune both BART and T5 models for summarization and compare their summaries against GPT-3 generated summaries. To do this, GPT-3 is tasked with generating a short summary for each call. We use this summary as the ground truth. For each transcript, the following question-based prompt is concatenated with the transcript’s content: “what is a one-sentence Tl;dr of this call:”. GPT-3 parameters *temperature*, *top-p*, *frequency* and *presence* are adjusted to obtain optimal results. Huggingface transformers library [18], along with FastAI and Blurr packages, are used to fine-tune the models based on the call transcripts and corresponding reference summaries. The best results on the validation set are seen after fine-tuning three epochs for BART and eight epochs for T5. After training, the model inference is conducted on the test dataset.

Evaluation Most studies in text summarization use Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [19] score as the primary evaluation metric. ROUGE score is used to approximate the similarity of the model-generated summaries with the reference summaries. It consists of F1 scores for ROUGE-1, ROUGE-2, and ROUGE-L that measure the word overlap, bi-gram overlap, and longest common sequence between the ground truth and the generated summary, respectively. ROUGE score has limited capability of capturing semantic similarities such as paraphrasing, which is common in abstractive summarization. In our work, the ROUGE score and human evaluation are both used to investigate model performance. Human evaluation is important to evaluate the quality of the summaries as models are usually optimized for metrics like ROUGE. During the human evaluation, the model-generated summaries were manually verified for content and readability by seven different evaluators.

4 Results

In this section, BART and T5 stand for BART-large-xsum and T5-small models, respectively. Table 2 shows a sample call transcript along with summaries generated by various pre-trained and fine-tuned models, in each case demonstrating apparent ASR transcription errors. In this particular example, pre-trained Pegasus and BART models do not produce summaries that are relevant to the transcript, while both pre-trained and fine-tuned T5 models show tendencies of yielding extractive summaries. Among all models, GPT-3 generates the highest quality summary, as expected.

Table 2. Example Summaries Generated by the Fine-Tuned Models

Original transcript (abbreviated)	...if you would prefer to prepay please have your credit card ready at the end of the call...what can I get for you today. Can I get a kids low mane with chicken. No vegetables. And then a kids low Maine with steak no vegetables. Into Santa tie Donna or fudge brownie today...that bring your total to eleven fifty two and that'll be ready for pickup and about twelve minutes.
Pre-trained Pegasus	If you have any questions about this or any of our other shows please call us at 800-273-3217 800-273-3217 800-273-3217 800-273-3217 or email us at 800-273-3217 800-273-3217 800-273-3217 800-273-3217.
Pre-trained BART	For the first time in more than a decade we have an online ordering option for you to play the game Pokemon Go on your mobile phone or tablet.
Fine-tuned BART	The message is asking if the person would like to prepay for their order, and if so, to have their credit card ready. The total cost of the order is \$11.52.
Pre-trained T5	I have one order of the kids old man with chicken no vegetables and then a kids low Maine with steak no veggies, and another order the children bowl made from steak.
Fine-tuned T5	If you would like to prepay for prepay, please have your credit card ready at the end of the call.
GPT-3	The caller is ordering two kids meals, one with chicken and one with steak, and requests no vegetables. The total cost will be \$11.52 and the food will be ready for pickup.

4.1 Quantitative Results

Since GPT-3 output is used as the reference in this work, we evaluate the summaries generated by all other models by comparing them with the GPT-3 result for a given call transcript. The average ROUGE-1, ROUGE-2, and ROUGE-L F1-scores for the test set across four models, pre-trained and fine-tuned BART-large and T5-small, are presented in Table 3. Note we also experimented briefly with the pre-trained Pegasus summarization model, however, it does not gen-

Table 3. Rouge Scores Across the Models

Models	Rouge-1	Rouge-2	Rouge-L
Pre-trained BART	24.5	5.7	16.6
Fine-tuned BART	37.8	16.8	31.0
Pre-trained T5	21.9	5.1	15.2
Fine-tuned T5	30.6	10.3	24.1

eralize well to the call center dataset used in this study. Therefore, it is not fine-tuned nor included in subsequent analysis.

As seen in Table 3, for both BART and T5, a model fine-tuned with our domain-specific dataset shows significant improvement over the pre-trained model. This is not surprising because call transcripts have unique characteristics, such as ASR transcription errors and various topics scattered over short utterances by multiple speakers, which are not represented in typical LLM training data. Overall, fine-tuned BART-large-xsum model exhibits the highest ROUGE scores. It is worth noting that a high ROUGE score only indicates close resemblance to the reference text generated by GPT-3, which does not necessarily ensure a high-quality summary. For that purpose, human evaluation is needed.

4.2 Qualitative Results

To better estimate the model efficacy, we employed domain experts to conduct a human evaluation of the summary quality. Each generated summary, with the model name anonymized, is read by multiple reviewers and receives a score in the range of [1, 11] from each reviewer, where higher values reflect more satisfactory summaries.

Figures 1, 2 and 3 present the histograms of human evaluation scores for the test set samples for each model. Scores for GPT-3 outputs are predominantly in the range of [9, 11]. In contrast, most of the scores produced by pre-trained BART and T5 are found at the lower end of the range. After fine-tuning using domain-specific data, both BART and T5 exhibit notable improvements. In particular, fine-tuned BART in Fig. 2 demonstrates a score distribution similar to that of GPT-3 in Fig. 1, which is further supported by Table 4. These results indicate that, per human perception, the quality of call transcript summaries generated by this fine-tuned BART model approaches that of GPT-3. Thus, it is a candidate suitable for on-premise deployment in call center applications.

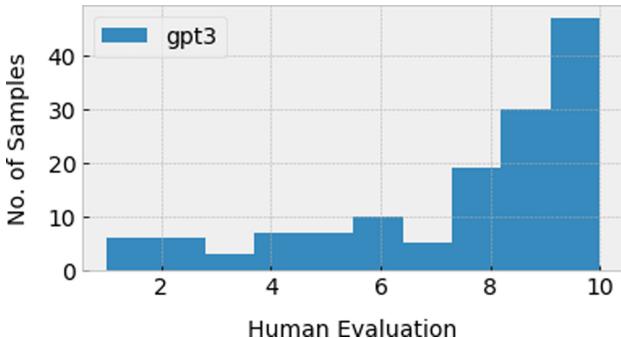


Fig. 1. Human Evaluation Scores for Summaries Generated by GPT-3

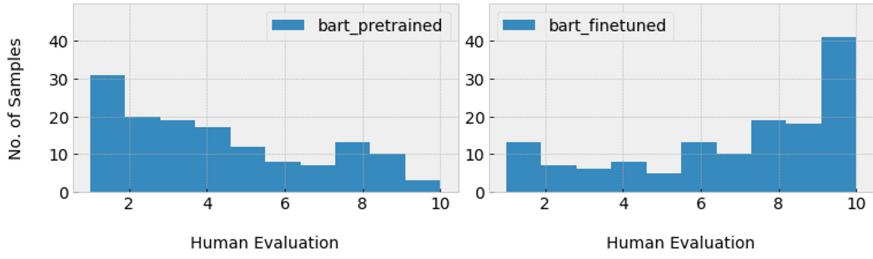


Fig. 2. Human Evaluation Scores for Pre-Trained and Fine-Tuned BART

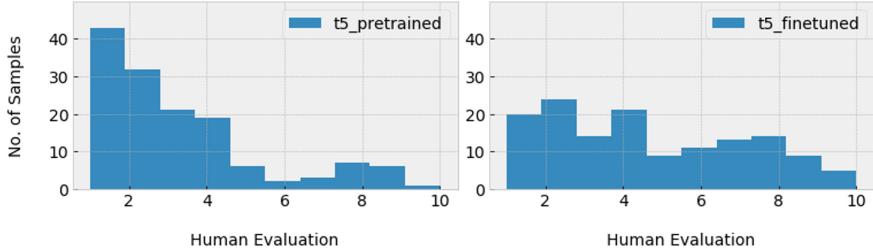


Fig. 3. Human Evaluation Scores for Pre-Trained and Fine-Tuned T-5

Table 4. Mean Human Evaluation Scores Across the Models

Models	Avg score
GPT-3	7.69
Pre-trained BART	4.12
Fine-tuned BART	6.95
Pre-trained T5	3.06
Fine-tuned T5	4.56

4.3 Comparison of ROUGE Score and Human Evaluation Score

As discussed earlier, in this study, ROUGE score is not a direct measure of the summary quality, rather, it assesses how much the model-generated text matches the GPT-3 output. With recent adoptions of leveraging the output from a state-of-the-art LLM such as GPT-3 as ground truth to fine-tune or domain-adapt a smaller model for application deployment [2, 15], it is worthwhile to investigate whether an evaluation metric such as ROUGE based on the model-generated ground truth summaries can still reflect the benchmark it is intended to measure.

Figure 4 displays ROUGE-2 (ROUGE-1/L omitted to reduce clutter) and the human evaluation scores of all models, using the data from Tables 3 and 4. It shows qualitative agreement between ROUGE and human evaluation, namely, the order remains the same when ranking the models by ROUGE score and by human evaluation score.

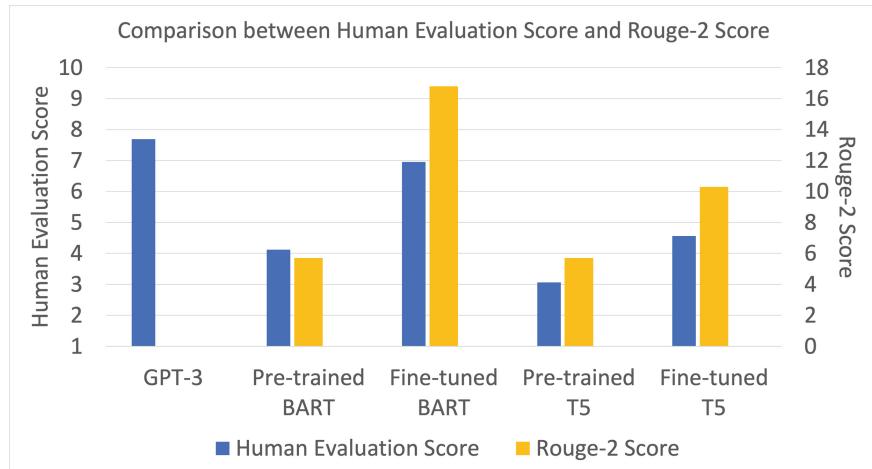


Fig. 4. ROUGE Score and Human Evaluation Score Across the Models

For quantitative comparison, the correlation coefficient between ROUGE scores and human evaluation scores of all samples in the test set is computed for each model, with one consideration: for a specific test sample, if the reference summary generated by GPT-3 is deficient, then there is no reason to expect a close match (i.e., a high ROUGE score) from the summary of another model regardless of its actual quality. Therefore, to understand how ROUGE scores correlate with human evaluation scores, a more meaningful result can be obtained by filtering out test samples with low-quality reference summaries from GPT-3.

Table 5. Correlation between ROUGE Scores and Human Evaluation Scores

Models	Pre-trained BART	Fine-tuned BART	Pre-trained T5	Fine-tuned T5
ROUGE-1	0.48	0.58	0.19	0.72
ROUGE-2	0.35	0.43	0.12	0.69
ROUGE-L	0.50	0.50	0.26	0.59

Table 5 presents the correlation coefficient for samples in the test set for which GPT-3 produces summaries with human score >8.0 . Fine-tuned models result in higher correlation between ROUGE and human scores than pre-trained models. Interestingly, the highest correlation is observed for the fine-tuned T5 model, even though its efficacy is lower than that of fine-tuned BART (Fig. 4). This is a consequence of T5 being more extractive, i.e. key phrases in the original text are selected as summary directly, which impacts both the ROUGE score and human evaluation score. On the other hand, the correlation is weaker for a more abstractive model such as BART, where a concise summary yielding a high human evaluation score doesn't necessarily contain the exact phrases found

in GPT-3 output. Therefore, the ROUGE score is better suited to evaluate extractive summaries than abstractive summaries. However, when a model is ineffective, the ROUGE score computation is dominated by the random matches between irrelevant phrases in its output and the reference summary, resulting in a low correlation between ROUGE and human evaluation, as observed in Table 5 for the pre-trained T5 model.

5 Conclusion

This work aims to develop a text summarization model for call transcripts that can be deployed on-premise at call centers, and demonstrate that LLMs can generate satisfactory summaries for deficient transcription text via fine-tuning. To overcome the unique challenges presented by call transcripts (e.g., high ASR transcription errors and multiple topics scattered across utterances from multiple speakers), pre-trained text summarization language models are fine-tuned using call center transcripts, assisted with GPT-3-generated reference summaries. This approach suggests that LLMs have the potential to generate a reasonably good summary of such imperfect texts, however, it does require fine-tuning. Fine-tuned BART-large-xsum model is found to output summaries with high ROUGE scores as well as satisfactory human evaluation results approaching that of GPT-3. In addition, we examine how ROUGE scores based on reference text generated by GPT-3 compare with human evaluations of the quality of text summaries and find qualitative agreement in model rankings using these two evaluation metrics. Moreover, their correlation exhibits a larger variation with model efficacy when the model summary is more extractive than abstractive. Additional evaluation metrics, such as relevance and factuality will be examined in a future study.

References

- Stepanov, E., Favre, B., Alam, F., Chowdhury, S., Singla, K., Trione, J., Bechet, F., Riccardi, G.: Automatic summarization of call-center conversations. In: Conference: IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2015) (2015)
- Asi, A., Wang, S., Eisenstadt, R., Geckt, D., Kuper, Y., Mao, Y., Ronen, R.: An End-to-End Dialogue Summarization System for Sales Calls (2022). [arXiv:2204.12951](https://arxiv.org/abs/2204.12951)
- Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: Librispeech: an asr corpus based on public domain audio books. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5206–5210. IEEE (2015)
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Amodei, D.: Language models are few-shot learners. *Adv. Neural. Inf. Process. Syst.* **33**, 1877–1901 (2020)
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Zettlemoyer, L.: Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension (2019). [arXiv:1910.13461](https://arxiv.org/abs/1910.13461)

6. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer (2019)
7. Zhang, J., Zhao, Y., Saleh, M., Liu, P.: Pegasus: pre-training with extracted gap-sentences for abstractive summarization. In: International Conference on Machine Learning. PMLR (2020)
8. Mihalcea, R., Tarau, P.: Textrank: bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 404–411 (2004)
9. Steinberger, J., Jezek, K.: Using latent semantic analysis in text summarization and summary evaluation. Proc. ISIM **4**(8), 93–100 (2004)
10. El-Kassas, W.S., Salama, C.R., Rafea, A.A., Mohamed, H.K.: Automatic text summarization: a comprehensive survey. Expert Syst. Appl. **165**, 113679 (2021)
11. Chandramouli, A., Shukla, S., Nair, N., Purohit, S., Pandey, S., Dandu, M.M.K.: Unsupervised paradigm for information extraction from transcripts using BERT (2021). [arXiv:2110.00949](https://arxiv.org/abs/2110.00949)
12. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding (2018). [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)
13. Biswas, P.K., Iakubovich, A.: Extractive summarization of call transcripts (2021). [arXiv:2103.10599](https://arxiv.org/abs/2103.10599)
14. Goo, C.W., Chen, Y.N.: Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts. In: 2018 IEEE Spoken Language Technology Workshop (SLT), pp. 735–742. IEEE (2018)
15. Wang, S., Liu, Y., Xu, Y., Zhu, C., Zeng, M.: Want to reduce labeling cost? GPT-3 can help (2021). [arXiv:2108.13487](https://arxiv.org/abs/2108.13487)
16. Narayan, S., Cohen, S., Lapata, M.: Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In: 2018 Conference on Empirical Methods in Natural Language Processing, pp. 1797–1807. Association for Computational Linguistics (2018)
17. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**, 1–67 (2020)
18. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Rush, A.M.: Huggingface's transformers: state-of-the-art natural language processing (2019). [arXiv:1910.03771](https://arxiv.org/abs/1910.03771)
19. Lin, C.Y.: Rouge: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004)
20. Uma, A.N., Sityaev, D.: Comparing Methods for Extractive Summarization of Call Centre Dialogue (2022). [arXiv:2209.02472](https://arxiv.org/abs/2209.02472)



Targeted Image Reconstruction by Sampling Pre-trained Diffusion Model

Jiageng Zheng^(✉)

Department of Computer Science, Lehigh University, Bethlehem, USA
jiz322@lehigh.edu

Abstract. A trained neural network model contains information on the training data. Given such a model, malicious parties can leverage the “knowledge” in this model and design ways to print out any usable information. Therefore, it is valuable to explore the ways to conduct such an attack and demonstrate its severity. In this work, we proposed ways to generate a data point of the target class without prior knowledge of the exact target distribution by using a pre-trained diffusion model. The result shows that the attacker can generate images that are similar to the attacking target by leveraging a pre-trained diffusion model.

Keywords: Model Inversion Attack · Latent Variable Generative Model · Diffusion Model

1 Introduction

Any machine learning model has learned something about the training data, so the leakage of a trained model may cause privacy concerns. If the training data contains private information, adversaries can leverage the visible information and capture the private data they want. Several studies have explored this possibility. For example, the membership inference attack (MIA) [5] compromises privacy by predicting whether a data point is a part of the model’s training data. Another study proposed a reconstruction attack [1] which aims to reconstruct an exact data point in the training set. However, it requires the attacker to have more knowledge beyond the trained model, such as all the data points, neural networks’ weight initialization, and learning rate schedule.

This work focuses on the reconstruction attack as in [1], but it requires less prior knowledge. More specifically, given a trained classification model and a distribution of the attack target, our adversary can generate the portrait of a previously invisible class (as shown in Fig. 1). This type of reconstruction attack can also threaten the privacy of the training data. For example, imagine we have a face recognition model which classifies the image input as a specific person. If a malicious party retrieves this model, the party can generate a sample image for each person.

A human knows what his/her best friend looks like, but if this human is asked to draw his/her best friend a picture, this human may fail due to poor drawing



Fig. 1. The Images in the Left Column are from the Training Set. The Images in the Right Column are Generated by the Adversarial using the Provided Classification Model and Part of the Visible Training Set. Notice that the Visible Training Set does not Contain the Images on the Left Column nor Other Images of the Same Person, so the Images in the Left Column are Supposed to be Private

skills. This phenomenon also applies to the trained neural network models. A trained classification model has learned how to process an image and do the classification. Suppose we query this model for a sample image of a specific class by fine-tuning the model's input. In that case, the tuned input will be a noise (Fig. 2), regardless it grants 100% classification confidence on the attack target. This happens because the classification model is not perfectly robust. Therefore, we use the distribution of attack targets to guide image generation.

There are two requirements to evaluate a reconstructed image. First, this image should maximize the classification confidence of the attack target. Second, it should belong to the distribution of the attack target. Since the distribution of the attack target is invisible, we can approximate this using other distributions

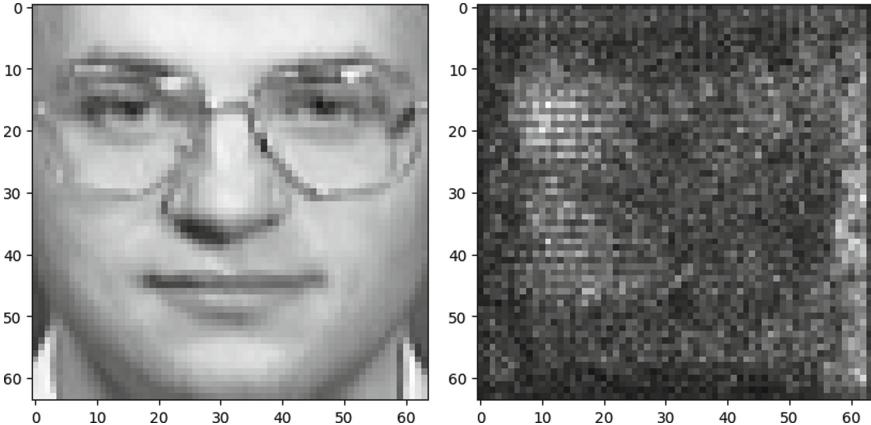


Fig. 2. This Image on the Right is Generated by Maximizing the Classification Possibility on an Attack Target (Image on the Left). The Classifier Classifies the Adversarial Noise to be 99%, the Attack Target. However, the Generation of an Adversarial Noise does not Satisfy our Goal

(for example, for a face recognition model, any distribution of face images can approximate the distribution of the attack target).

To satisfy the requirement of the generated distribution, we will use latent variable generative models (LVGM) to learn from a distribution. There are various types of LVGMs, and the options include the Generative Adversarial Network (GAN) [2], the Variational Autoencoder (VAE) [7], and the diffusion related models [4, 10, 13]. For different LVGMs, the methodologies for this reconstruction are also different, and the diffusion model is the focus of this work.

Notice that the goal of this work is different than [1], where they aimed to reconstruct the exact image in the training set. However, here we want to reconstruct an image with the same class label and a similar distribution to the attack target.

Moreover, the difference between this task and conditional image generation is that conditional image generation provides the entire data set, giving full knowledge of all classes. In contrast, for this work, a trained neural network model provides the knowledge of target classes in an implicit manner.

2 Background

2.1 Conditional Generative Adversarial Networks

In a conditional generative adversarial network [6], we train a generator to achieve some goal and deceive the discriminator so that the discriminator cannot distinguish the target domain from the domain of the generated results. The objective of the generator of conditional GAN can be expressed as

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G). \quad (1)$$

The first part of the expression showcases how the generator is trained to deceive the discriminator, and the second part represents the generator’s goal. For our work, the generator’s goal refers explicitly to maximizing the classification confidence on the attack targets.

The conditional GAN learns from a distribution by evolving a generator and a discriminator. The generator is sufficiently trained when the generation result is sufficiently good or when the discriminator cannot be optimized.

2.2 Variational Autoencoders

The neural networks used by the variational autoencoders typically include two modules: an encoder which maps the input into deep latent space, and a decoder to generate result from latent features. An example of VAE encoder [7] is expressed as below. Given the input x , the distribution of latent features is a normal distribution according to the mean and variance predicted by the encoder neural network.

$$(\boldsymbol{\mu}, \log \boldsymbol{\sigma}) = EncoderNeuralNet(\mathbf{x}) \quad (2)$$

$$q_{\phi}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma})) \quad (3)$$

The decoder neural network maps the distribution of latent features to the distribution of the inputs. The training of this decoder uses self-supervised learning (SSL). For example, the most straightforward training objective is whether the decoder can recover the original input image.

2.3 Diffusion Models

We train the diffusion model using the expression below [4]. The $\mathbf{x}^{(\alpha_i)}$ represents the noised input at the α_i step, and the neural network aims to predict the noise added.

$$\ell_{\text{diff}}(\mathbf{x}; w, \theta) := \sum_{i=1}^T w(\alpha_i) \mathbf{E}_{\epsilon \sim \mathcal{N}(0, I)} [\|\epsilon - \epsilon_{\theta}(\mathbf{x}^{(\alpha_i)}, \alpha_i)\|_2^2], \quad \mathbf{x}^{(\alpha_i)} := \sqrt{\alpha_i} \mathbf{x} + \sqrt{1 - \alpha_i} \epsilon \quad (4)$$

Given an image of Gaussian noises, the diffusion model samples a denoised result by applying steps of the denoise process (Fig. 3). Denoising is done by having a neural network predict the noise added on each step. A denoising step can be expressed below (algorithm 2 from [4]).

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_{\theta}(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z} \quad (5)$$

In the expression (5), the x_{t-1} represent the result after a step of denoising. In the most straightforward setup, the α and the σ are fixed constants computed using the variance schedule. The z is a noise in Gaussian distribution. The neural network takes a tuple of x_t and t and predicts the noise on step $t - 1$.

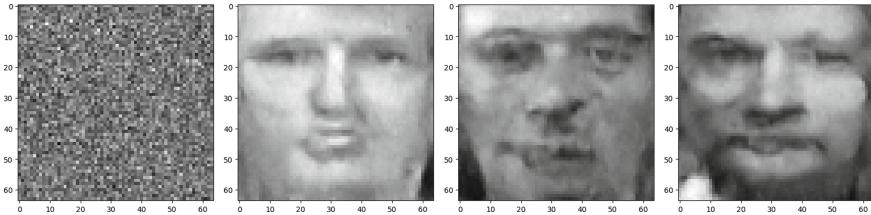


Fig. 3. The Leftmost Image is a Fixed Noise Input into the Sampling Process of a Trained Diffusion Model, and Other Images are the Result of Sampling. Even though we Fixed the First Input Noise, the Random Gaussian Noises are Involved in each Step of the Denoising, so the Sampling Results are Different

3 Methodologies

3.1 GAN Method

We designed a method based on the conditional generative adversarial network [6]. The generator G , takes a Gaussian noise $\mathcal{N}(0, 1)$ with the shape of the target image and outputs the same shape. The neural network architecture for the generator can be the U-net [8]. The discriminator D takes an image as input and outputs the prediction of whether this image is fake (generated by the generator) or real (belonging to the target distribution).

We designed the loss function to train the generator for this reconstruction attack as below.

$$L_g = L_{CrossEntropy}(C(G(\mathcal{N}(0, 1))), y_{target}) + \alpha L_{CrossEntropy}(D(G(\mathcal{N}(0, 1))), 1) \quad (6)$$

The C represents the classification model visible to adversaries; the G is the generator; the D is the discriminator. In the first part of this loss function, the generator's objective is to create an image, input this image to the classifier, and maximize the confidence that this generated image to be classified as the attack target. The image generated is input to the discriminator as in the second part of this loss function. The generator wants the discriminator to classify this image as “real” (or belongs to the target distribution). Notice that this objective function is similar to (1), while the only difference is that the generated image is specified to maximize the classification result of the attack target.

To train this generator, we first generate a temporary data set for the discriminator, which includes both generated images and the images from the target distribution. Then, we train the discriminator, evolve the generator, and generate a new training set for the discriminator to start the new iteration until the generator is good enough.

3.2 Sampling Pre-trained Diffusion Model

For the image interpolation tasks, the VAE-based methods handle them by gradually adjusting the latent features of the decoder's input. In contrast, diffusion

models handle them by gradually adjusting the intermediate noises. The well-trained VAE generator and diffusion model have learned how to generate an image to the target domain, so when interpolating two images, the generated interpolations will also belong to the target domain. For this reconstruction attack, the insight can be interpreted from the perspective of image interpolation: would there be an interpolation of images that happens to be our attack target?

For a well-trained diffusion model, we want to find a specific Gaussian noise from the distribution $\mathcal{N}(0, 1)$. We sample the diffusion model using this noise and input the denoised image into the classifier, and we want this specific noise to maximize the prediction of the attack target. The expression of the loss function is as below. Notice that we want to optimize the specific noise x_t .

$$L = L_{CrossEntropy}(C(denoise(x_t)), y_{target}) \quad (7)$$

The denoising process is the same as the formula (5). We fix the Gaussian noise z for each denoising step so that we are only optimizing one path of denoising. The denoising process includes t steps. For each step, the previous noise x_t contributes to a part of x_{t-1} 's computation, which is a suitable property since this is similar to the residual connections in the ResNet [3], aiming to alleviate vanishing gradient issues.

Notice that if it is large, the computational graph for the gradient computation will also be significant. However, the modern computer with more than 256GB memory can still afford this computation.

4 Experimental Setup

4.1 Data Set and Preparations

The data set we used is the Olivetti Faces. It contains facial images of 40 different people. There are ten images for each person, so it has 400 images in total. For classification models, the task is to classify the input image to the specific person. We have seven images for training and three for each person for validation. We trained a simple 2-layer CNN model C_2 , which gives 96.7% top-1 validation accuracy for classifying 40 classes. We also trained a VGG11 model [9] c_{vgg} , intentionally made this VGG model overfit, but it still has top-1 validation accuracy for 90%.

We set the attack targets for the reconstruction attack to range from person 1 to person 20. We assumed the adversarial obtains an estimation of target distribution by owning the rest of the training set (200 images from person 21 to person 40).

For our GAN method, as discussed in 3.1, the architecture of the generator is U-net [8], and the discriminator is a simple 2-layer CNN model. In each iteration, the training data for the discriminator contains the images of person 21 to person 40 (200 images) and another 200 images generated by the current generator. We train the generator until the progress of generated result becomes unobservable.

We also trained a diffusion model on the images of the last 20 people. The neural network architecture to predict the noise is also U-net. We set the number of steps for adding or removing noise to 600, and the variance schedule ranges from 10^{-4} to 0.02.

Moreover, we built a variational autoencoder to make more comparisons. We conduct the image reconstruction for the VAE method by fine-tuning the features in latent space.

4.2 Evaluation Metrics

Different from the reconstruction attack in [1], we cannot directly compute the difference between the generated result and the training set because we are not reconstructing any specific image but a class. One of the most straightforward ways is to conduct a human evaluation. For example, we can purchase the Amazon Mechanical Turk service (we did not do it).

A better way to evaluate our result is using similar evaluation metrics as conditional image generation. The evaluation metrics for conditional image generation on Cifar benchmarks use a pre-trained Inception v3 model [12]. One common evaluation metric is Frechet Inception Distance, where the Inception model extracts latent representations of the generated image. If we get a short distance between the extracted latent representation and that of the ground truth images, the generation is sufficiently good. However, we cannot use the FID metrics since the Inception model is not for facial images. To compromise, we use classification model C_2 to evaluate, which is a more robust model than the classifier C_{vgg} we used to attack. For simplicity, we directly input the generated image into the classifier C_2 and collect the prediction confidence on the attack target.

5 Experiment Results

5.1 Comparison of Different Ways of Image Generation

From Fig. 4, we can get the qualitative result that the diffusion-based method can generate better human faces. Notice that all the methods used the C_{vgg} classifier. To quantitatively analyze the generation of the target class, we input the generated images into the C_2 classifier and collect the confidence that the generated images are classified as the attack target. We show the results of the quantitative analysis in Table 1. Using the Diffusion method to attack target 2 gives the only satisfiable result. On the one hand, this result is reasonable since the C_{vgg} is a classification model, it does not have complete knowledge of all the classes, and one successful case indicates the threat of this type of attack is indeed present. On the other hand, there is room to improve the attack methodologies.

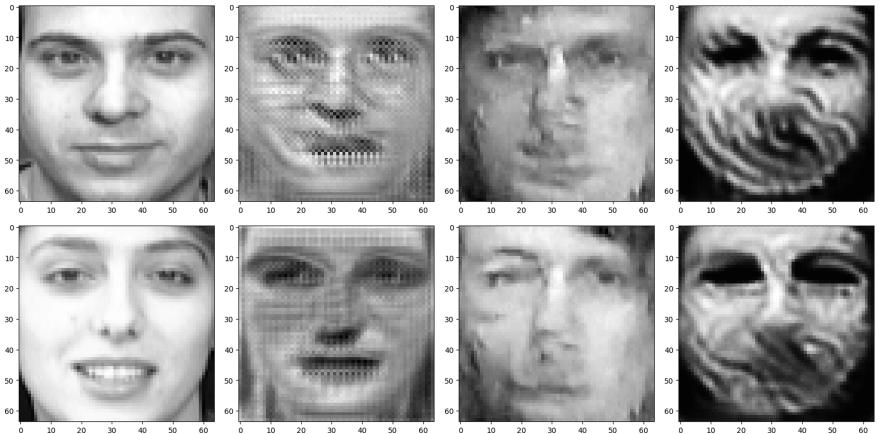


Fig. 4. From the Left-Most Column to the Right-Most Column, the Images Represent Examples of Attack Targets, Reconstruction Results of GAN, Reconstruction Results of Diffusion, and Reconstruction Results of VAE. The First Row is the Attack's First Target (the Eighth Person in the Data Set). The Second Row is the Second Attack Target (the Seventh Person in the Data Set)

Table 1. Quantitative Analysis of Generation Results

Confidence of being predicted as attack target	GAN\VAE\diffusion
Target1	0.0003 \ 0.0002 \ 0.0000
Target2	0.0030 \ 0.1069 \ 0.9077

5.2 Effect of Different Learning Rates for Fine-Tuning x_t

When fine-tuning the noise, the loss value will always converge no matter how significant the learning rate is (we tried a range from 1e-5 to 1e-10). However, when selecting different learning rates, the result of this diffusion method will be different (Fig. 5). To explain this observation, we need to trace the gradient update during the back propagation and do more experiments.

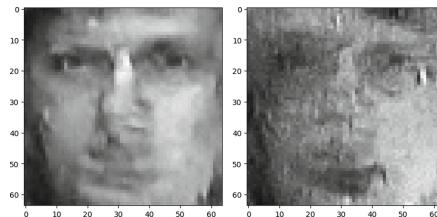


Fig. 5. Both Generated Image has a Loss Value of around 1.3, but for the Left Image, we Set the Learning Rate to 1 and Optimized for 16 Iterations. In Contrast, we Set the Learning Rate for the Right Image to 10 and Optimized it for 9 Iterations

6 Conclusions and Future Works

In this work, we explored the potential approaches to generate an image for a target class when provided with a trained classification model and an approximate distribution of the attack target. The method of sampling a pre-trained diffusion model has great potential and deserves further research investment. Limited by time and our computational power, more thorough analysis and experiments should be done on the method we proposed in 3.2. Moreover, some observations need to be better explained, such as the effect of the learning rate discussed in 5.2. It is also a research direction to explore the relationship between the generation quality and the provided classification models.

There are new variants of the Diffusion model’s sampling methods, for example, the denoising diffusion implicit models (DDIMs) [11], which can give high-quality samples much faster. The diffusion model’s sampling method improvements can significantly reduce this work’s memory consumption since the gradient computation graphs dominate our memory consumption for the massive number of denoising steps. Therefore, it is also valuable to update the methodologies proposed in this work along with the quick evolution of the existing diffusion model’s methodologies.

References

1. Balle, B., Cherubin, G., Hayes, J.: Reconstructing training data with informed adversaries (2022)
2. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Courville, A., Bengio, Y., Ozair, S.: Generative adversarial networks (2014)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015)
4. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models (2020)
5. Hongsheng, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P.S., Zhang, X.: A survey, membership inference attacks on machine learning (2021)
6. Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks (2016)
7. Kingma, D.P., Welling, M.: An introduction to variational autoencoders. Found. Trends Mach. Learn. **12**(4), 307–392 (2019)
8. Ronneberger, O., Fischer, P., Brox, T.: Convolutional networks for biomedical image segmentation, U-net (2015)
9. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014)
10. Sinha, A., Song, J., Meng, C., Ermon, S.: D2c: diffusion-denoising models for few-shot conditional generation (2021)
11. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models (2020)
12. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision (2015)
13. Zhu, J., Ma, H., Chen, J., Yuan, J.: Few-shot image generation with diffusion models (2022)



Hulk: Graph Neural Networks for Optimizing Regionally Distributed Computing Systems

Zhengqing Yuan¹, Huiwen Xue², Chao Zhang¹, and Yongming Liu^{1(✉)}

¹ School of Artificial Intelligence, Anhui Polytechnic University, Wuhu 241009, China
zhengqingyuan@ieee.org, liuyongming1015@163.com

² School of Optoelectronic Science and Engineering, Soochow University,
Suzhou 215031, China

Abstract. Large deep learning models have shown great potential for delivering exceptional results in various applications. However, the training process can be incredibly challenging due to the models' vast parameter sizes, often consisting of hundreds of billions of parameters. Common distributed training methods, such as data parallelism, tensor parallelism, and pipeline parallelism, demand significant data communication throughout the process, leading to prolonged wait times for some machines in physically distant distributed systems. To address this issue, we propose a novel solution called **Hulk**, which utilizes a modified graph neural network to optimize distributed computing systems. Hulk not only optimizes data communication efficiency between different countries or even different regions within the same city, but also provides optimal distributed deployment of models in parallel. For example, it can place certain layers on a machine in a specific region or pass specific parameters of a model to a machine in a particular location. By using Hulk in experiments, we were able to improve the time efficiency of training large deep learning models on distributed systems by more than 20%. Our open source collection of unlabeled data: <https://github.com/DLYuanGod/Hulk>.

Keywords: Optimize Communication Efficiency · Distributed Training · Parallel Deployment · Time Efficiency

1 Introduction

In recent years, there has been a trend of scaling up deep learning models, resulting in a more robust performance in specific domains. For instance, in the field of natural language processing, large-scale text data has been used to train deep learning models such as GPT-3 (175B) [2], T5 (11B) [19], and Megatron-LM (8.3B) [22], which have demonstrated impressive performance. However, training these models can be quite challenging. To solve the challenges posed by large-scale deep learning models, optimization of distributed computing is crucial.

Model parallelism (MP) is a technique used to solve the problem of a model being too large to fit into the memory of a single GPU or TPU by distributing the model across multiple GPUs or TPUs. However, this approach may introduce communication challenges between GPUs or TPUs during training. On the other hand, data parallelism (DP) can improve time utilization by addressing the batch size issue during training, but it cannot resolve the problem of a model being too large for a single GPU or TPU's memory capacity.

While DP and model MP have been effective in mitigating communication volume issues in recent years, such as large minibatch SGD [9], Megatron-LM [22], Gpipe [12], and Pathway [1] the challenge of scheduling distributed training across machines in different regions remains unsolved. If a model like GPT-3 with hundreds of billions of parameters exceeds the memory capacity of GPUs in the current region during training, it becomes necessary to schedule machines from other regions to complete the training. This will pose several challenges:

- Communication latency can be very high when training is distributed across machines in different regions.
- How can tasks be effectively allocated to different machines, such as assigning specific machines to maintain certain layers of the model's parameters (e.g., Machine 0 is responsible for Layer X) or designating machines to process specific data (e.g., Machine 2 handles Data Set Y)?
- How can we address the issue of disaster recovery in training, such as handling scenarios where a machine fails during the process?
- If you need to train not only a single task but also multiple tasks simultaneously, such as training both a GPT-3 and a GPT-2 model, how can you provide for these tasks?

Table 1. We Measured the Time it Takes for our Machines in Three Different Regions to Send and Receive 10 Words, using Eight Servers, and Calculated the Average

Regions	Communication time to send 64 bytes (ms)							
	California	Tokyo	Berlin	London	New Delhi	Paris	Rome	Brasilia
Beijing, China	89.1	74.3	250.5	229.8	341.9	–	296.0	341.8
Nanjing, China	97.9	173.8	213.7	176.7	236.3	265.1	741.3	351.3
California, USA	1	118.8	144.8	132.3	197.0	133.9	158.6	158.6

To elaborate on the first point, we collected all communication logs between the three machines and the eight servers over a three-month period. Our statistics reveal the communication time for every 64 bytes, as presented in Table 1. As observed in the table, the communication latency between certain nodes is high or even unfeasible. Here, the problem of communication time is difficult to solve in a distributed system without optimization.

1.1 Contributions

Graph data structures have been widely adopted since their introduction, as they can effectively represent interconnected structures such as social networks and knowledge graphs. Considering the tremendous success of graph neural networks [7, 14, 26] in recent years, we aim to leverage this powerful capability in real-world industrial systems. With the powerful representational capability of graphs, it becomes easier to model the relevant optimization problems described in our paper. Our design choices were influenced by the types of workloads observed in actual systems. Hulk has the following features:

Efficient Inter-Node Communication Our system minimizes the impact of communication latency between machines, ensuring that each machine is assigned the appropriate task.

Global Optimality Our model is built upon graph convolutional neural networks (GCNs) [14, 25] to extract features from the entire graph, enabling the selection of a globally optimal solution.

Disaster Recovery Since GCNs are utilized to assign tasks to different machines in the system, it becomes evident which tasks each machine is responsible for. Furthermore, in the event of a machine failure, the system can quickly recover the entire computation.

Scalability If a particular machine or machines are no longer needed, you can simply remove the corresponding edge information from the graph structure.

The novelty of the proposed system lies in the utilization of graph neural networks for optimizing machine learning systems. By relying on the neural network's output values and some algorithms, the scheduling problem of the entire system can be efficiently solved.

1.2 Engineering Challenges

Although graph neural networks are capable of addressing tasks such as node classification [14, 23, 24], link prediction [15, 21, 29], and graph classification [14, 28], there is currently no suitable task that can be directly applied to our system. How to construct a suitable loss function is a crucial problem that cannot be overlooked. Regarding the representation of optimization features, such as computation time and communication time, in the graph data structure, there are also challenges that need to be addressed.

2 Background

This section provides a brief introduction to machine learning systems and graph neural networks.

2.1 Machine Learning Systems

This subsection provides a brief overview of the evolution of machine learning systems.

Data Parallelism DP [5] is a commonly used technique in distributed training for deep neural networks, where the data is split into multiple copies and distributed to different machines for computation. Each machine calculates the loss and gradient of its assigned data and aggregates these gradients into a parameter server, which updates the model parameters. This method enables multiple machines to process large data sets in parallel, resulting in faster training speeds.

Parameter Server The parameter server is a distributed deep learning training method proposed by Mu Li et al. [16] that addresses the communication bottleneck problem in training large-scale deep learning models. It achieves this by placing the gradient aggregation and parameter updating process on the server side, and the computational nodes only need to send the locally computed gradient information to the server. This approach reduces communication overhead and improves training efficiency.

Megatron-LM Megatron-LM [22] combines model parallelism and data parallelism by dividing the model parameters into multiple parts, each trained on a different GPU. This allows for larger models to be used as each GPU only needs to focus on computing a part of the model using model parallelism. Data parallelism is used to assign different batches to different GPUs for processing, which improves training efficiency.

The training objective of Megatron-LM is to minimize the negative log-likelihood of the target sequence given the input sequence, which is expressed as:

$$L(\theta) = - \sum_{t=1}^T \log P(y_t | y_{<t}, x; \theta)$$

where T is the length of the sequence, y_t is the target token at time step t , $y_{<t}$ are the tokens before time step t , x is the input sequence, and θ represents the model parameters.

Gpipe In Gpipe [12], the model is split into sub-models, each assigned to a different GPU. DP concatenates Micro-batches along the pipeline to pass data and gradients between GPUs, enabling pipeline parallelism [4]. The training process in Gpipe can be expressed as the following equation:

$$\Delta W_{i,j} = \eta \sum_{k=1}^K (\nabla_{W_{i,j}} L \left(f^{i,j}(x_k^{i,j}), y_k^{i,j} \right) + \sum_{l=j+1}^M \nabla_{W_{i,l}} L \left(f^{i,l}(x_k^{i,l}), y_k^{i,l} \right))$$

where $W_{i,j}$ denotes the weight parameter of the j th layer of the i th submodel, $\Delta W_{i,j}$ denotes the corresponding parameter update, η denotes the learning rate, K denotes the number of Micro-batches, $f^{i,j}$ denotes the forward propagation function of the j th layer of the i th submodel, $x_k^{i,j}$ denotes the k th Micro-batch of the j th layer in the i th sub-model, $y_k^{i,j}$ denotes the label of the k th Micro-batch.

2.2 Graph Neural Networks

Graph Neural Networks (GNNs) [3, 11, 20, 30, 31] are a type of neural network designed to work on graph-structured data, where nodes represent entities and edges represent relationships between them. They have become popular in recent years due to their ability to capture complex relationships and patterns in data, making them useful for tasks such as node classification, link prediction, and graph classification.

2.3 Graph Convolutional Networks

Graph Convolutional Networks (GCNs) [14] are a type of deep learning model designed to work on graph-structured data. They use convolutional operations to aggregate information from neighboring nodes and update node representations. The key formulas for GCNs include the graph convolution operation, which calculates the node representation updates, and the graph pooling operation, which aggregates information across multiple nodes.

$$\mathbf{v}^{(l+1)} = \sigma \left(\sum_{u \in \mathcal{N}(v)} \frac{1}{c_{u,v}} W^{(l)} \mathbf{u}^{(l)} \right) \quad (1)$$

where $\mathbf{v}^{(l)}$ represents the feature representation of node v at layer l , $\mathcal{N}(v)$ denotes the set of neighbors of node v , $W^{(l)}$ is the weight matrix at layer l , σ is the activation function, and $c_{u,v}$ is a normalization factor that depends on the number of neighbors of node u and v . This formula is used to iteratively compute the feature representations of nodes in a graph using neighborhood information.

3 Data Representation

To better address the issues raised in Sect. 1, it is important to select an appropriate data structure to represent the system parameters. We adopt a graph-based data structure to represent our system parameters, with each node (denoted as v) representing a machine in a different region. Each node has unique features that include its geographic location, computational capacity, and GPU memory. The edges (denoted as e) between nodes denote the possibility of communication between the two connected machines, with the weight of each edge representing the time in milliseconds required to transmit each 64-byte message.

As depicted in Fig. 1, we randomly selected eight machines to construct a graph, where the edge weight represents the communication time, and the node features are embedded in the corresponding vector space.

For example,¹ node 0 can be represented as $v_0 = \{'Beijing', 8.6, 152\}$. Then we embed the node information using the following formula:

$$\mathbf{v}^{(0)} = \mathbf{x}_v \quad (2)$$

¹ <https://developer.nvidia.com/cuda-gpus>.

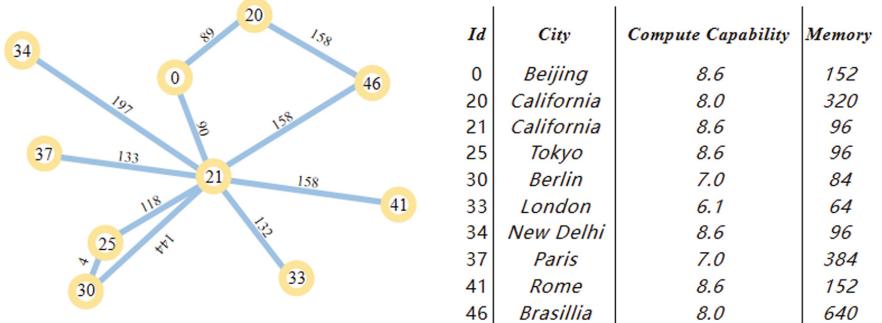


Fig. 1. In this Figure, the Graph Topology is Visualized on the Left, while the Characteristics of Each Node are Indicated on the Right. Where Computing Power is Determined based on Nvidia’s Official Website, and Memory Refers to the Total Memory Across All GPUs on each Machine

where $\mathbf{v}^{(0)}$ denotes the initial feature vector of node v and \mathbf{x}_v denotes the input feature vector of node v .

The node-to-node edges we represent by the adjacency matrix. The weight of an edge in the adjacency matrix is equal to the communication time between two corresponding nodes. The values for unconnected edges are set to 0, and the diagonal values in this matrix are all 0. Similarly, we then perform the edge information embedding with the following equation:

$$e_{vu} = g(\mathbf{e}_{vu}, \mathbf{u}, \mathbf{v}, \Theta_e) \quad (3)$$

where e_{vu} denotes the edge feature between node v and node u , \mathbf{e}_{vu} is the feature vector of edge vu , \mathbf{u} and \mathbf{v} are the feature vectors of node u and node v , respectively, g is a learnable function and Θ_e is its argument. We then sparsely label this subgraph to enable the neural network to learn the contents of the graph in a supervised manner.

4 Methods

The typical tasks of graph neural networks, such as node classification, do not utilize edge information and only leverage the graph topology. In real-world cases, the information carried by edges is often crucial, such as edge weights and directed edges. To incorporate edge information into nodes, we aim to perform edge pooling, which involves aggregating or pooling edges of neighboring nodes at each node to create a unified node representation that contains edge information. This is expressed in the following equation:

$$\mathbf{v}^{(l+1)} = \sigma \left(\sum_{u \in \mathcal{N}(v)} f(\mathbf{v}^{(l)}, \mathbf{u}^{(l)}, e_{vu}) \right) \quad (4)$$

where $\mathbf{v}^{(l+1)}$ represents the feature vector of node v in layer $l + 1$, σ is the activation function, $\mathcal{N}(v)$ denotes the set of neighboring nodes of node v , $\mathbf{u}^{(l)}$ represents the feature vector of node u in layer l , and f is a learnable function used to merge features of nodes and edges into new features of node v .

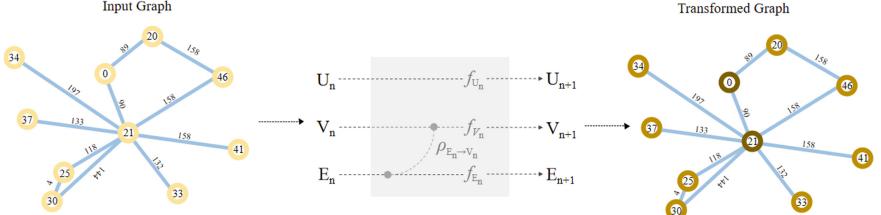


Fig. 2. The Edge Pooling Operation of the Above Fig. 1. Where U Represents the Information of the Whole Graph and f is the Respective Linear Layer

As depicted in Fig. 2, this is the first layer of the constructed network structure($l = 0$) that enables nodes to encode edge information.

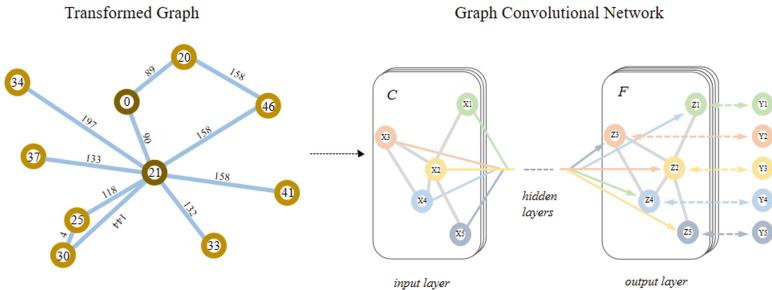


Fig. 3. The Transformed Graph Data are Entered into GCNs for Forward Propagation

After the edge features are embedded into node features, we can use the resulting transformed graph as input for a standard node classification task and train it using a graph convolutional neural network or graph attention network. As shown in Eq. 1. If we want to build N-layer GCNs with our $l = 2, 3, 4 \dots N+1$.

As shown in Fig. 3, Y represents the category of the classification, i.e., what tasks are appropriate.

Then we calculate its loss using the cross-entropy loss function [8]:

$$\mathcal{L} = - \sum_{i=1}^{|Y|} Y_i \log \hat{Y}_i \quad (5)$$

Here, \mathcal{Y} denotes the set of all labels, Y_i denotes the true label of node i , and \hat{Y}_i denotes the predicted label of node i . Then back propagation is performed to update the network parameters.

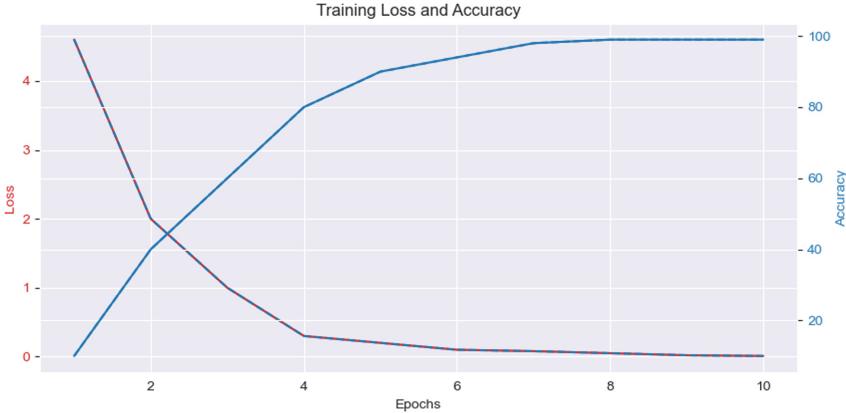


Fig. 4. Loss Rate and Accuracy Line Charts for 10 Steps of Training on this Data. The Parameters of GCNs are 188k and the Learning Rate is 0.01

As depicted in Fig. 4, we observed that the accuracy peaked at 99% during the sixth training step.

5 Structure

In this section, we build our system based on the GCNs trained in the previous Sect. 4 and solve the problem presented in Sect. 1.

5.1 Efficiency

We now have two tasks to perform. The first involves training the BERT-large model [6], while the second involves training the GPT-2 model [18]. As the largest GPT-2 model (1.5B parameters) is significantly larger than BERT-large (340M parameters), it is important to carefully allocate tasks to each machine in a sensible manner. The ratio of the number of parameters in GPT-2's largest model (1.5B) to BERT-large (340M) is approximately 4.4:1. Based on this information, we instruct the graph neural network to classify the classes according to this scale and optimize the communication time within each class. Also, we need to consider the memory and computing power characteristics of each machine.

Algorithm 1 Task Assignments

Require: Graph Data G_1 , Trained Graph Neural Network F , Number of Tasks N , Minimum Memory Threshold M_n for Each Task

Ensure: Task Assignments for Each Graph Data

- 1: $C \leftarrow 0$
- 2: **if** G_1 does not meet the requirements of all tasks **then**
- 3: Jump out of the algorithm and report an error.
- 4: **end if**
- 5: **for** i in range(1, N) **do**
- 6: $G_i, G_{i+1} \leftarrow F(G_i)$
- 7: Assign the smaller graph G_i to a task with the appropriate minimum memory threshold M_n
- 8: **if** G_i does not meet the requirements of the all task **then**
- 9: $C \leftarrow i$ and Continue
- 10: **if** $C >= 1$ **then**
- 11: $G_i \leftarrow G_i + G_C$
- 12: Assign the smaller graph G_i to a task with the appropriate minimum memory threshold M_n
- 13: $C \leftarrow 0$
- 14: **end if**
- 15: **end if**
- 16: **if** G_{i+1} does not meet the requirements of the all task **then**
- 17: Break and Provide a prompt and wait for other tasks to complete before proceeding with training.
- 18: **end if**
- 19: **end for**

We use Algorithm 1 to schedule multiple tasks, but it can also be used to determine superiority if there is only one task. Based on the computational power, memory and communication efficiency features, as well as the integration into node information, we only need to determine whether it is appropriate.

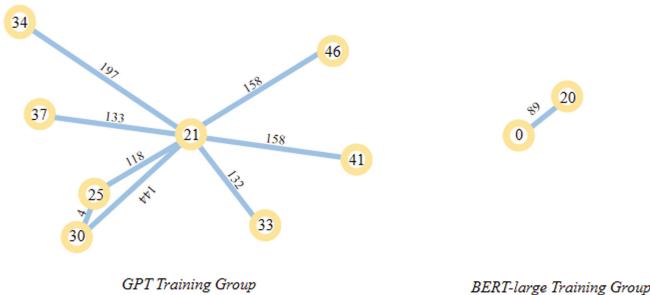


Fig. 5. The Data in Fig. 1 are Grouped using Algorithm 1. The Left Panel is the Training Group of GPT-2 and the Right Panel is the Bert-Large Training Group

Figure 5 demonstrates that the basic graph neural network is capable of carrying out classification tasks effectively and emulating human thought processes.

5.2 Scalability

If we need to add one or more machines to this system, we can simply define their $\{City, ComputeCapability, Memory\}$ and connect them to the existing nodes that can communicate with them using weights.

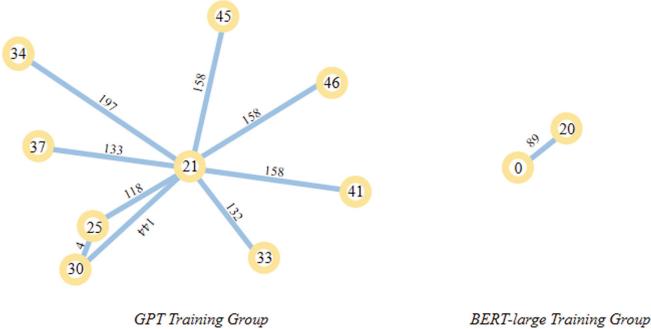


Fig. 6. Join the Machine with id 45 and Make Assignments

As shown in Fig. 6, the machine with id 45{*Rome, 7, 384*} in the dataset was added to the Hulk system and still works fine.

6 Experimentation and Evaluation

In this section, we test the Hulk system using multiple deep learning tasks in real industries with 46 high-performance GPU servers.

6.1 Experimental Setting

We have a total of 46 servers distributed across different countries and regions, with a combined total of 368 GPUs of various models such as NVIDIA A100, NVIDIA A40, NVIDIA V100, RTX A5000, GeForce GTX 1080Ti, GeForce RTX 3090, and NVIDIA TITAN Xp. And, we calculated the average of 10 communications between these machines over a 3-month period. Due to network policy restrictions in different countries, there are certain machines that are unable to communicate with each other. We adopt the parameter settings provided in the original paper for the training process.

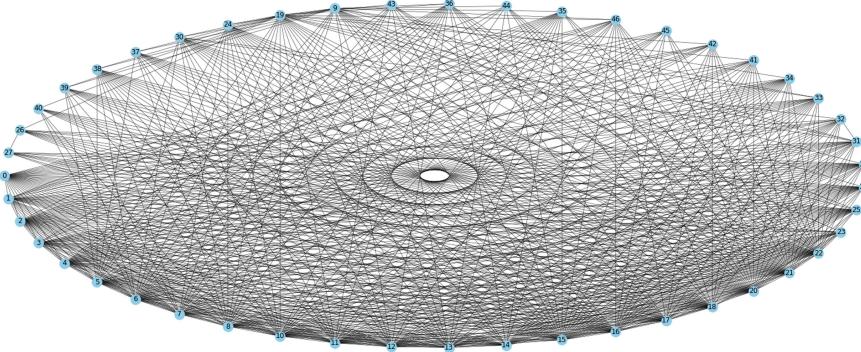


Fig. 7. 46 Servers Construct the Graph Structure Data

6.2 Data Building

We use network [10] library to build our graph structure data and visualize it as shown in Fig. 7. Additionally, we need to read the adjacency matrix of this data and consider the corresponding feature embedding representation.

6.3 Task Assignment

The four tasks we aim to train in this system are OPT (175B) [13], T5 (11B), GPT-2 (1.5B), and BERT-large (350M).

We need to classify all nodes into four distinct classes based on their characteristics and then deploy distributed algorithms tailored to each class.

Table 2. Model Node Allocation

Model	Nodes
OPT (175B)	0, 1, 2, 3, 4, 20, 21, 22, 23, 24, 27, 28, 29, 30, 31
T5	5, 6, 7, 8, 9, 10, 11, 12, 13, 14
GPT-2	15, 16, 17, 18, 19, 25, 26, 32, 33, 34
BERT-large	35, 36, 37, 38

As presented in Table 2, we feed the graph data into the graph neural network, which was trained in Sect. 4 and employs Algorithm 1, to derive node classification information. To handle the nodes in each class with different computational performance and memory, we utilize Gpipe to train the model in parallel. Depending on the computational power and memory of each node, we determine which part of the model it will handle.

6.4 Evaluation

To validate the performance of the Hulk system, we have chosen three commonly used distributed computing algorithms for evaluation.

System A It utilizes all available machines for training while discarding any machine that does not have sufficient memory to accommodate the entire model. It utilizes data parallelism to distribute the batch size across multiple machines, thereby enabling simultaneous training of the model on each machine.

System B It utilizes Gpipe for parallelism, assigning a certain layer of the model to a particular machine until the entire model is distributed across all machines.

System C It employs tensor parallelism with Megatron-LM across the entire system, requiring all machines to be utilized for model training.

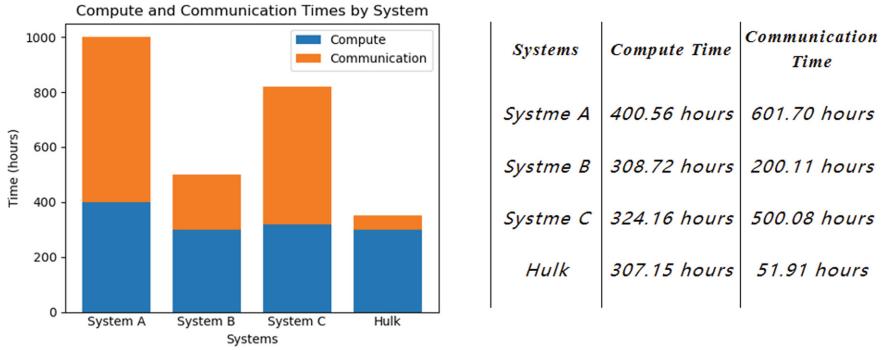


Fig. 8. Communication Time and Calculation Time on Four Models of the Four the 4 Systems

Result As shown in Fig. 8, the Hulk system can greatly reduce communication time and thus the overall training time. This illustrates that Hulk is effective in dividing the nodes into a specific model for training.

If we need to train 6 models, the parameters of each model are shown in Fig. 9. Among them, the parameters of RoBERTa [17] are 355M and the parameters of XLNet [27] are 340M.

Result As illustrated in Fig. 10, when the system needs to handle multiple tasks, the gap in communication time becomes more apparent. Our Hulk system is able to effectively reduce communication time (Because the GPT-3 (175B) model is not open source, we use the OPT (175B) with equivalent parameters instead).

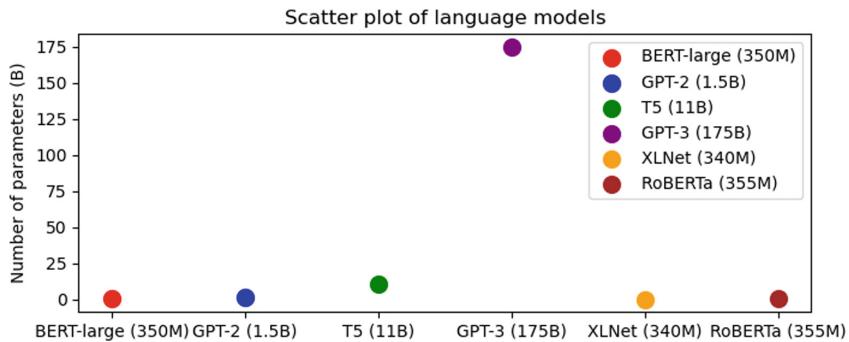


Fig. 9. Language Model Parameters

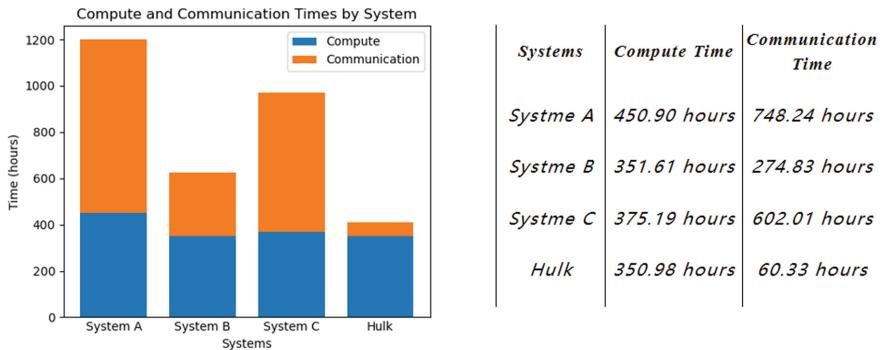


Fig. 10. Communication Time and Calculation Time on Six Models of the Four Systems

7 Conclusion

In this article, we introduce our novel solution, Hulk, which optimizes regionally distributed computer systems by tackling the challenges of scheduling distributed training across machines in different regions. Our real-world industrial solution, Hulk, utilizes graph neural networks with powerful representation capabilities to enhance communication efficiency between GPUs or TPUs across different countries or regions during training. With its efficient communication, global availability, fast recovery, and excellent scalability, Hulk stands out as a powerful tool for optimizing regionally distributed computer systems. The results demonstrate a significant increase in the efficiency of distributed training, crucial for the success of large-scale deep learning models. Overall, the use of Hulk can streamline the model deployment process and benefit researchers and practitioners seeking to optimize communication efficiency.

Acknowledgment. The authors gratefully acknowledge the support of the AIMTEEL 202201 Open Fund for Intelligent Mining Technology and Equipment Engineering

Laboratory in Anhui Province and the Anhui Provincial Department of Education Scientific Research Key Project (Grant No. 2022AH050995). The financial assistance provided by these projects was instrumental in carrying out the research presented in this paper. We would like to thank all the members of the laboratory for their valuable support and assistance. Without their help, this research would not have been possible. Finally, we would like to express our gratitude to the Anhui Polytechnic University for providing the necessary facilities and resources for this study.

References

1. Barham, P., Chowdhery, A., Dean, J., Ghemawat, S., Hand, S., Hurt, D., Isard, M., Lim, H., Pang, R., Roy, S. and Saeta, B., Schuh, P., Sepassi, R., Shafey, L., Thekkath, C., Wu, Y.: Pathways: asynchronous distributed dataflow for ml. In: Marculescu, D., Chi, Y., Wu, C. (eds.) *Proceedings of Machine Learning and Systems*, vol. 4, pp. 430–449 (2022)
2. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: *Advances in Neural Information Processing Systems*, vol. 33 (2020)
3. Bui, H.H., Luu, K., Nguyen, Q.H.: Structural analysis and role identification for financial networks using graph embeddings. In: *2021 IEEE 7th International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 207–214. IEEE (2021)
4. Dally, W.J.: Pipeline parallelism revisited. *Commun. ACM* **39**(11), 102–108 (1996)
5. Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Senior, A., Tucker, P., Yang, K., Le, Q.V., et al.: Large scale distributed deep networks. *Adv. Neural. Inf. Process. Syst.* **25**, 1232–1240 (2012)
6. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1 (Long and Short Papers), pp. 4171–4186 (2019)
7. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: *International Conference on Machine Learning (ICML)*, pp. 1263–1272 (2017)
8. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning (2016)
9. Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, S., Jia, Y., He, K.: Accurate, large minibatch sgd: training imagenet in 1 hour (2017). [arXiv:1706.02677](https://arxiv.org/abs/1706.02677)
10. Hagberg, A., Swart, P., Chult, D.S.: Exploring network structure, dynamics, and function using networkx. In: *Proceedings of the 7th Python in Science Conference*, vol. 11, pp. 11–15 (2008)
11. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: *Advances in Neural Information Processing Systems*, pp. 1024–1034 (2017)

12. Huang, Y., Wang, Y., Huang, W., Liu, T., Wang, D., Liu, P., Weinberger, K.Q.: Gpipe: efficient training of giant neural networks using pipeline parallelism. In: Advances in Neural Information Processing Systems, pp. 103–112 (2019)
13. Iyer, S., Lin, X.V., Pasunuru, R., Mihaylov, T., Simig, D., Yu, P., Shuster, K., Wang, T., Liu, Q., Koura, P.S., Li, X., O’Horo, B., Pereyra, G., Wang, J., Dewan, C., Celikyilmaz, A., Zettlemoyer, L., Stoyanov, V.: Scaling language model instruction meta learning through the lens of generalization, Opt-iml (2023)
14. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR) (2017)
15. Li, J., He, Y., Zhu, H.: Link prediction via subspace clustering based on graph convolutional networks. Neurocomputing **311**, 206–214 (2018)
16. Li, M., Andersen, D.G., Park, J.W., Smola, A.J., Ahmed, A.: Scaling distributed machine learning with the parameter server. In: Proceedings of the 11th USENIX conference on Operating Systems Design and Implementation (OSDI) (2014)
17. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: a robustly optimized bert pretraining approach. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (2019)
18. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. In: International Conference on Learning Representations (2019)
19. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. In: Advances in Neural Information Processing Systems (2019)
20. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Trans. Neural Netw. **20**, 61–80 (2008)
21. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European Semantic Web Conference, pp. 593–607. Springer (2018)
22. Shoeybi, M., Patwary, M., Puri, R., Kumar, J., Ganguli, S.: Megatron-lm: training multi-billion parameter language models using model parallelism (2019). [arXiv:1909.08053](https://arxiv.org/abs/1909.08053)
23. Velivckovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)
24. Wang, S., Wang, X., Liu, J., Xu, Y., Tang, J.: Heterogeneous graph attention network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 793–803 (2019)
25. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. **32**(1), 4–24 (2019)
26. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. **32**(1), 4–24 (2020)
27. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V.: Xlnet: generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems (2019)
28. Ying, R., You, J., Morris, C., Ren, X., Hamilton, W.L., Leskovec, J.: Hierarchical graph representation learning with differentiable pooling. In: Advances in Neural Information Processing Systems, pp. 4805–4815 (2018)

29. Zhang, M., Chen, Y., Tang, Y.: Link prediction based on graph neural networks. In: Advances in Neural Information Processing Systems, pp. 5165–5175 (2018)
30. Zhang, Y., Chen, Z., Song, S.: Scalable graph learning for anti-money laundering: a first look. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2268–2277. ACM (2018)
31. Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Sun, M.: Graph neural networks: a review of methods and applications (2018). [arXiv:1812.08434](https://arxiv.org/abs/1812.08434)



The Power of Words: Predicting Stock Market Returns with Fine-Grained Sentiment Analysis and XGBoost

Farshid Balaneji¹(✉), Dietmar Maringer¹, and Irena Spasic²

¹ University of Basel, Faculty of Business and Economics, Peter Merian-Weg 6,
4002 Basel, Switzerland

farshid.balaneji@unibas.ch, dietmar.maringer@unibas.ch

² Cardiff University, School of Computer Science and Informatics, Abacws,
Senghennydd Road, Cardiff, UK
spasic@cardiff.ac.uk

Abstract. This study investigates the relationship between news sentiment and the stock market's return. The sentiment was automatically analyzed using four methods, including lexicon-based and deep learning-based approaches, at three levels of granularity, i.e., sentence, paragraph, and full text. The sentiment was combined with features from the calendar year, lagged returns, and news publishers, which were fed into the XGBoost algorithm trained to classify the direction of market return for the following business day. The performance was maximized using Bayesian hyperparameter optimization and evaluated using nested cross-validation. The proof of concept was demonstrated using ten companies in the Dow Jones Index, which were grouped into five sectors. The findings indicate an asymmetric power of sentiment measures in different sectors, with the petroleum industry being the most responsive to the sentiment expressed in the news. The study highlights the significance of targeted sentiment measures in making informed decisions about the market direction, particularly for the petroleum industry.

Keywords: Sentiment Analysis · XGBoost Algorithm · Hyperparameter Optimization · Nested Cross-validation · Stock Markets

1 Introduction

Financial markets attract significant media attention, spurring many news articles and social media content. The availability of such data has provided new opportunities for market participants to gain insights and develop trading strategies. However, keeping abreast of the volume and velocity of financial news can be overwhelming for market participants, which may delay financial decisions. Traditional techniques relying on human expertise fall short when faced with rapidly changing markets and the corresponding data. Thus, there is a need for

scalable algorithms that can keep up with the ever-evolving market and increasing data volume. In this context, our research question is: How can we improve the predictions based on sentiment analysis in finance by addressing the limitations of existing approaches?

The financial market forecasting domain has witnessed the emergence of Artificial Intelligence applications, attracting significant attention from industry and academia for their potential to enhance human intelligence in finance. Specifically, Natural Language Processing (NLP) designs intelligent systems to extract meaningful information from text data. In particular, sentiment analysis can be helpful in finance for assessing opinions, sentiments, moods, and emotions across the market. Lexicon-based and machine learning-based algorithms are widely used for sentiment analysis. While the former tags words as positive, negative, or neutral based on a predefined dictionary, the latter requires manually labeled examples for training. Deep learning's recent advances have shown improved performance in sentiment analysis in the financial domain (see [1]).

However, sentiment analysis in finance has two main limitations. Firstly, many current methods focus on the coarse-grained analysis of sentiment in news documents (see [2,3]), which categorizes the sentiment of the whole document based on positive, negative, and neutral phrases, regardless of different sections of a news article that actually discusses a specific company and is more relevant to market participants. Secondly, assigning a single sentiment label, such as neutral, to the entire text based on the overall sentiment of the document overlooks valuable information contained within negative phrases.

Addressing these gaps in sentiment analysis is crucial for providing more accurate predictions and enabling market participants to make more informed decisions. In this research, we tackle the first gap by utilizing Targeted Sentiment Analysis (TSA), which analyzes different target entities and classifies the sentiment associated with relevant entities in each document [4]. To address the second gap, this study examines the distribution of sentiments across sentences and paragraphs in news articles of companies each business day. It applies different aggregation methods, such as quartiles of sentiment scores and frequency of positive, negative, and neutral sentiments, to include distribution information in the feature space.

The paper is organized as follows: the subsequent section reviews the relevant literature. Section 3 describes the data sources used in the study. Section 4 elaborates on the news processing steps. Section 5 explains the sentiment analysis methods and how the sentiment scores were aggregated across news articles. Section 6 discusses the feature construction and XGBoost training steps. Section 7 presents and discusses the classification results and implications. Finally, Sect. 8 concludes the paper and provides suggestions for future work.

2 Literature Review

Lexicon-based sentiment analysis methods use a lexicon that assigns an explicit sentiment score to each lexical entry (i.e., a word or phrase). Tetlock's work (see

[5]) was one of the first studies to exercise this approach in the financial domain. They used the General Inquirer (GI) categories from the Harvard psychosocial dictionary to understand the relationship between media sentiment and daily returns on the Dow Jones Industrial Average (DJIA). They found that high media pessimism predicted downward pressure on market prices, and conversely, low market returns led to high media pessimism. However, one limitation of this approach is that general-purpose lexicons may not accurately capture the domain-specific sentiment in financial texts.

According to Loughran and McDonald in [6], many of the negative words in the GI dictionary, such as “tax,” “cost,” and “capital,” do not necessarily convey a negative sentiment when utilized in a financial document. Therefore, the authors developed one of the first dictionaries specifically tailored for sentiment analysis within a business domain, which has since been widely employed in many financial applications. While this approach addresses some limitations of general-purpose lexicons, it still lacks the ability to adapt to the evolving language in financial news. It may not effectively capture nuanced sentiment expressions [7].

In their study, Li, Wu, and Wang aimed to integrate stock prices and financial news to create a stock price prediction model (see [8]). They combined different sentiment scores, including VADER [9], to calculate the sentiment vectors for the news articles and concluded that the Loughran-McDonald Financial Dictionary models the news sentiments better. Our paper extends their work as we analyzed sentiments on sentence, paragraph, and full-text levels, while they focus only on full-text sentiment analysis. However, this approach still struggles with the challenge of identifying and prioritizing relevant information within news articles [10].

In summary, we identified two crucial gaps in the literature. First, most studies focus on coarse-grained sentiment analysis and overlook the relevance of different sections of news articles to specific entities. Modeling the relationship between target entities and their contextual surroundings is challenging. Liu et al. proposed a bidirectional LSTM model [11] to address this issue, which we applied in our study. Our work is related to Wan et al. [12], which also used the proposed model and incorporated news sentiment analysis and network analysis of financial markets. However, we utilized targeted sentiment scores and combined lexicon-based and fine-tuned deep-learning models for sentiment calculation.

Secondly, existing studies in financial market forecasting often neglect the distribution of sentiments within individual news articles and across multiple articles on the same day. These studies tend to label documents as positive based on higher probabilities, disregarding the impact of negative and neutral sentences. This limitation may lead to biased or incomplete representations of news sentiment, potentially affecting the accuracy of prediction models. To address this gap, our research takes into account both the frequency of categorical sentiment scores and the distribution of continuous sentiment scores.

By considering the sentiment frequency of targeted sentiment analysis, lexicon-based, and deep-learning-based sentiment scores, our results show that

the targeted sentiment of news articles in the petroleum industry is more clustered in positive, negative, and neutral categories. Compared to four other sectors, we observed higher predictive power for sentiment analysis in forecasting the daily return direction for the following business day in the petroleum industry. This finding demonstrates the asymmetric influence of sentiment analysis on different industries.

3 Data Sources

To demonstrate a proof of concept, we selected ten companies from various industries listed on the DJIA, including Petroleum, Information Technology, Financial Services, and Semiconductors. Table 1 provides a full list of these companies, which were grouped into pairs based on their sectors. In addition, the pairs were chosen to have an equal number of observations between January 1st, 2017, and February 28th, 2020.

News data were collected from 28 major financial news publishers using Event Registry,¹ a news intelligence platform that retrieves news and press releases from various publishers. Event Registry applies Wikipedia's URLs to assign an entity a unique Uniform Resource Identifier (URI), which helps manage their synonyms within one or across different languages. For example, despite Exxon-Mobil Corporation being commonly shortened to Exxon, both names will be linked to the same Wikipedia page and thus treated as the same entity.

Table 1. A curated list of ten companies

Company name	Trading symbol	Sector
Exxon Mobil Corp	XOM	
Chevron Corp	CVX	Petroleum Industry
JPMorgan Chase & Co.	JPM	
Goldman Sachs Group, Inc.	GS	Financial Services
Microsoft Corp.	MSFT	
Apple Inc.	AAPL	Information Technology
Intel Corp.	INTC	
The International Business Machines Corp.	IBM	Semiconductor/Information Technology
Cisco Systems, Inc.	CSCO	
Visa Inc.	V	Information Technology/Financial Services

In addition to the news data, we collected market data for the companies and obtained historical adjusted closing prices from Historical Options Prices,² an online provider of end-of-day prices of equity options and stocks.

Labeled data that assign sentiment to each sentence is necessary to train and optimize deep-learning-based models for sentiment analysis. The Financial

¹ <https://eventregistry.org/>.

² For more details on the available data catalog, refer to <https://www.historicaloptiondata.com>.

Phrase Bank dataset created by Malo et al. (refer to [13]) is a publicly available dataset that can be used for this purpose. The authors created a corpus of English news from all listed companies on the Helsinki Stock Exchange to generate the database. They randomly selected a subset of 5000 sentences to represent the overall news database. Then a team of sixteen annotators with backgrounds in business and finance was instructed to consider the sentences from an investor’s viewpoint, i.e., whether the news may have a positive, negative, or neutral influence on the stock price. Sentences deemed irrelevant from an economic or financial perspective were labeled as neutral.

The annotation process involved five to eight annotators per sentence to ensure a high level of inter-annotator agreement. Although not all annotators agreed on the sentiment of the sentences, a high level of agreement was observed between neutral and negative sentences. The final dataset consisted of 2259 sentences labeled positive, negative, or neutral, with an agreed level of 100%.

The Financial Phrase Bank is accessible via the Dataset API of the Hugging Face library.³

4 Processing News Data

The text data require pre-processing prior to sentiment analysis. This section details the steps to group the publication dates, clean the text, and filter the relevant documents.

4.1 Aggregating Dates

The Event Registry provides date and time information for each piece of news to indicate when it was published and when it was parsed. Missing publication times were filled using the parsing times. Further, whenever the news was published after the New York Stock Exchange (NYSE) closing time, which is 4:00 PM Eastern Time (ET) Zone, or over a weekend or a bank holiday, we aggregated it with other news published on the following business day.

4.2 Filtering News

Some organizations may have multiple name formats (e.g., “Exxon Mobil,” “XOM,” and “ExxonMobil” all refer to the same organization). To address this, we mapped different name formats of an organization to a unique named entity and normalized the detected companies. Through this normalization process, “Exxon Mobil,” “XOM,” and “ExxonMobil” were standardized to “Exxon.” Then, we determined the frequency of mapped organizations in each document, e.g., Exxon, and retained only those that mention the company entity at least three times. We defined news with the same title as duplicate and only included unique titles in our analysis. After filtering the news documents based on the frequency of mentioning the company name, the number of documents reduces by 75% in the case of Goldman Sachs (GS).

³ https://huggingface.co/datasets/financial_phrasebank.

4.3 Cleaning Texts

Extracting information from unstructured text data requires a thorough cleaning procedure. We utilized the Stanford Stanza library (see [14]) to identify sentence boundaries and segment the text accordingly, and additionally, we relied on whitespaces to identify paragraphs.

Then, to clean the text using an open-source library textacy (see [15]), which provides a range of functionalities for cleaning and normalizing raw text. We normalized the text’s bullet point symbols and Unicode characters and removed currency symbols, numbers, hashtags, emails, punctuation marks, URLs, and phone numbers. Segments with less than 30 characters, which may be artifacts of text processing and do not convey meaningful information, are removed.

The cleaned paragraphs were then aggregated to create a full body of cleaned news items. This process resulted in the creation of three levels of granularity for each news article: sentence level, paragraph level, and full-body level.

4.4 Detecting Target Companies

The targeted sentiment analysis framework requires identifying sentences within news articles that contain the target company name and subsequently dividing those sentences into three parts: the text preceding the target company name, the target company name itself, and the text following the target company name. For example, the sentence “These are small numbers in relation to Exxon’s 240 billion in annual revenues and 360 billion market cap.” would be segmented into:

$$\left\{ \begin{array}{l} \text{left text: “These are small numbers in relation to,”} \\ \text{target: “Exxon”} \\ \text{right text: “240 billion in annual revenues and 360 billion market cap”.} \end{array} \right.$$

This process is repeated for all paragraphs that contain the targeted company name, which enabled us to extract all relevant sentences by standardizing the company name format.

5 Sentiment Analysis Methodologies

As discussed in Sect. 2, multiple techniques are available for determining text sentiment in news articles. This study employs four algorithms, including two lexicon-based methods (Vader and Textblob), one Transformer-based method (DistilBERT), and targeted sentiment analysis (NewsSentiment). The following sections will overview how these algorithms work and how sentiment measures across news articles are aggregated.

5.1 Lexicon-Based Methods

The Valence Aware Dictionary for sEntiment Reasoning (VADER) introduced in [9] is an open source⁴ rule-based and lexicon-based sentiment analysis tool that takes into account the context of words and calculates the sentiment of a sentence by aggregating the sentiments of its tokens. The sentiment of a document is calculated by summing up the sentiment scores of each word present in the lexicon, adjusted according to pre-defined rules, and then normalized to the interval $[-1, 1]$.

Textblob (see [16]) is another sentiment analysis tool that uses a sentiment lexicon and the *pattern.en* sentiment analysis engine in [17]. The engine is based on WordNet,⁵ an electronic lexical database, and it evaluates a text's sentiment by analyzing its adjectives. Textblob returns the sentiment as a tuple comprising a polarity score represented by a float number in the range of $[-1, 1]$ and the subjectivity of the text.

We applied the VADER and Textblob sentiment analysis techniques to i th news, which includes s sentences and p paragraphs. The process involves assigning sentiment scores to each sentence, paragraph, and clean body of the news. Following the step, i th news with k sentences and m paragraphs is mapped onto a list of sentiment scores for VADER and Textblob.

$$n_i = [\{s_{1_i}, \dots, s_{k_i}\}, \{p_{1_i}, \dots, p_{m_i}\}, \{b_i\}]$$

To create a comprehensive sentiment profile for a company, we aggregated the sentiment scores obtained from applying VADER and Textblob to the sentences and paragraphs and cleaned the body of multiple news articles pertaining to the company.

Finally, we aggregated the sentiments of news articles on each date bracket, and the daily sentiment profile gives us a distribution of sentiments across different levels of granularity. Various statistical measures were utilized to incorporate this distribution into the feature space, such as the first, second, and third quartiles of sentiment in sentences and paragraphs and the minimum, maximum, mean, and median of sentiments in the full body. This results in the sentiment profile, which includes multiple statistics to capture the sentiment distribution.

As a result, there are ten sentiment scores for each daily sentiment profile for VADER and the same amount for Textblob measures.

5.2 Transformers

The transformer architecture utilizes a self-attention mechanism (refer to [18]) to extract features for each token and determine the significance of surrounding tokens concerning the target word.

Over the past few years, the advancement of Transfer Learning has become an essential part of the NLP toolkit for understanding human language and solving

⁴ <https://github.com/cjhutto/vaderSentiment>.

⁵ <https://mitpress.mit.edu/books/wordnet>.

challenging tasks like sentiment analysis. Devlin et al. in [19] trained BERT (Bidirectional Encoder Representations from Transformers) on a large corpus of data to solve two main tasks: Masked Language Model and Next Sentence Prediction. Since its initial creation, several versions and modifications of BERT have emerged in recent years.

The RoBERTa model, proposed by Liu et al. in [20], was retrained on a ten times larger dataset used for training the BERT model. The DistilBERT model, introduced by Sanh et al. (see [21]), employed the knowledge distillation technique to create a smaller and more efficient pre-trained version of BERT while retaining 97% of its language understanding capabilities.

To select the appropriate transformer model for this study, we employed the Transformers library created by Wolf et al. in [22], which provides open-source and pre-trained models of various transformers, including RoBERTa and DistilBERT. We fine-tuned both models using the Financial Phrase Bank dataset by dividing the data into training and development sets (20% and 80%, respectively). Table 2 reports partial results of the fine-tuned models in terms of F1 score⁶ and accuracy (refer to 7.1). The fine-tuned RoBERTa and DistilBERT models are publicly available on the Hugging Face Hub.⁷

Table 2. The results of fine-tuned Roberta and Distilbert models using Hugging Face Library.

Roberta	Distilbert	Roberta	Distilbert	Roberta	Distilbert	Roberta	Distilbert
Epoch		Validation Loss		Accuracy		F1	
1	6	0.277	0.112	0.951	0.969	0.951	0.969
2	7	0.151	0.098	0.977	0.977	0.977	0.978
3	8	0.136	0.100	0.982	0.977	0.982	0.978
4	9	0.113	0.089	0.977	0.977	0.977	0.978
5	10	0.141	0.093	0.973	0.977	0.973	0.978

In line with other transformer models, DistilBERT uses the self-attention mechanism to extract contextual information from the entire sequence. However, self-attention's memory and computational requirements grow quadratically with sequence length, making it infeasible to process long sequences (see [23]). As a result, the DistilBERT model has applied at the sentence and paragraph granularity level and not the full text.

In contrast to VADER and Textblob, DistilBERT's results are in the form of categorical sentiments. To aggregate sentiments of multiple news articles for

⁶ The F1 is the harmonic mean of the precision and recall. The precision is the ratio of correctly predicted positive classes to all items predicted to be positive. And the Recall is the ratio of correctly predicted positive classes to all items that are actually positive.

⁷ <https://huggingface.co/Farshid/roberta-large-financial-phrasebank-allagree>,
https://huggingface.co/Farshid/distilbert-base-uncased_allagree3.

a specific company on a given date, we counted the number of positive, negative, and neutral sentences and paragraphs in each date bracket, created three attributes per granularity level, and normalized it in the range of [0, 1]. That leads to six features for the DistillBERT sentiment score.

5.3 Targeted Sentiment Analysis

To apply TSA and find the entity-level sentiment for specific companies in each input document, we practiced the publicly available⁸ pre-trained model GRU-TSC (Gated Recurrent Units (GRU) based Target-dependent Sentiment Classification (TSC)) proposed by Hamborg and Donnay in [24]. First, the authors constructed the Multi TSC in News Articles (NewsMTSC) dataset, comprising more than 11K labeled sentences sampled from news articles from online US news outlets. And then built the model on top of bi-directional (GRUs) architecture and showed its superior performances on NewsMTSC, which is competitive on prior TSA datasets. They reported that the model with the RoBERTa yielded the best results compared to using BERT, and thus we applied the GRU-TSC model with the Roberta base as the language model.

For example, consider the following sentence from news about Exxon Mobil on January 24th, 2017.

“Short sellers have been active in closing their short positions in Exxon since the start of the year at a time when the stock is down more than per share since January.”

First, the sentence is split into three segments:

```
{"left_text": "Short sellers have been active in closing their short
positions in",
"target": "Exxon",
"right_text": "since the start of the year at a time when the stock is
down more than per share since January"}
```

and then the GRU-TSC model is applied with the Roberta-base language model, which gives the following result:

```
[{"class_id": 2, "class_label": "positive", "class_prob": 0.357},
 {"class_id": 1, "class_label": "neutral", "class_prob": 0.322},
 {"class_id": 0, "class_label": "negative", "class_prob": 0.320}]
```

The output is a list of key-value pairs that shows the probability, “class prob”, of each sentiment class, “class label”, assigned to the above sentence.

To aggregate the targeted sentiment classes across news of a company on a specific date, we calculated statistics such as the mean, median, minimum, and maximum of class probabilities for each class label and granularity level. This approach captures sentiment information about the target companies rather than simply labeling a sentence or paragraph based on the maximum class probability, which results in 12 sentiment features for sentences and 12 for paragraphs.

⁸ <https://github.com/fhamborg/NewsMTSC>.

6 Classification with XGBoost Algorithm

6.1 Constructing Features

Our study aims to forecast the direction of the daily return in the next business day. The daily stock return on date t is defined by Eq. (1), where P_t and P_{t-1} represent the adjusted closing prices on date t and $t - 1$, respectively.

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}}. \quad (1)$$

For each given date t , three sets of features are generated from the daily stock returns. These include:

- Lagged returns r_{t-1} , r_{t-2} and r_{t-3} .
- The differences between consecutive returns, i.e., $\delta r_t = r_t - r_{t-1}$.
- The moving average over the last five-day window for daily returns.

The dependent variable on date t , y_t is the difference between the return on $t + 1$ and t mapped to $\{0, 1\}$,

$$y_t = \begin{cases} 1, & \text{if } \delta r_{t+1} \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Our analysis includes a set of features related to the trading calendar to account for seasonality effects. These features include the year, month, day of the year, day of the month, and week of the year.

In addition, if a news article is published by one of the 28 major publishers, the corresponding publisher will be assigned a value of 1, while all other publishers will be assigned a value of 0.

6.2 Return Classification

This study used the Extreme Gradient Boosting (XGBoost) algorithm, introduced by Chen and Guestrin in [25], to determine the return direction for the following business day.

The Gradient Boosting algorithms are based on the principle of building a strong learner by iteratively running weak learners, which are algorithms that perform better than random guessing. XGBoost utilizes a tree ensemble model, a collection of Classification and Regression Trees (CARTs). Each CART comprises a root node, a layer of leaf nodes, and a group of internal nodes connecting them. When the algorithm begins, all the data are located in the root node, and the binary condition on each internal node defines the data path to the next level. As individual trees may not produce accurate results, multiple CARTs are combined. Gradient-boosted decision trees train an ensemble of weak learner trees iteratively, with each iteration using the previous model's error residuals to fit the next model. The final prediction is the weighted average of all tree predictions.

The XGBoost algorithm has a large number of hyperparameters, and their selection can significantly impact the performance of the XGBoost model. Due to the time-consuming model evaluation process and many hyperparameters, we automated hyperparameter optimization using Optuna (see [26]), an open-source Python library that supports a wide range of hyperparameter optimization algorithms and features efficient pruning strategies. Optuna utilizes advanced algorithms for sampling hyperparameters and reducing ineffective trials.

6.3 Cross-Validation

Cross-validation (CV) is a widely-used evaluation method. There are different extensions of cross-validation to incorporate feature selection and parameter tuning. In this study, we implemented the nested CV proposed by Varma and Simon in [27] as an unbiased estimate of the true error.

Nested CV effectively incorporates feature selection and hyperparameter tuning to train an optimal classification model. It involves dividing the data into k outer folds and creating inner folds within each outer training set for feature selection, parameter tuning, and model training. Our implementation of nested CV includes the following steps (Fig. 1).

1. The data were partitioned into an original training dataset (80% of the data) and an original test dataset (20% of the data).
2. The original training dataset was divided into three outer folds, each consisting of training and test data.
3. For each outer training fold, the following steps were performed:
 - (a) The outer training fold was divided into three inner folds, and for each inner fold, the following steps were executed:
 - (i) The optimal hyperparameters for XGBoost were determined using Optuna with 250 trials, considering the full set of features.
 - (ii) The optimal model was used to rank XGBoost's feature importance, and the set of features with the highest threshold was selected.
 - (iii) The inner train set was transformed by selecting the features from the previous step and an XGBoost model was fitted with the optimum hyperparameters. The accuracy of the model on the inner test set was then evaluated.
 - (b) The inner model and set of features with the highest accuracy were selected, and the model was fitted to the combined inner trains.
 - (c) The model's accuracy was computed on the corresponding outer test.
4. The model with the highest accuracy among all outer tests and associated features and hyperparameters was selected.
5. The final model was fitted to the original training data, and its accuracy was reported on the original test dataset (from step 1).

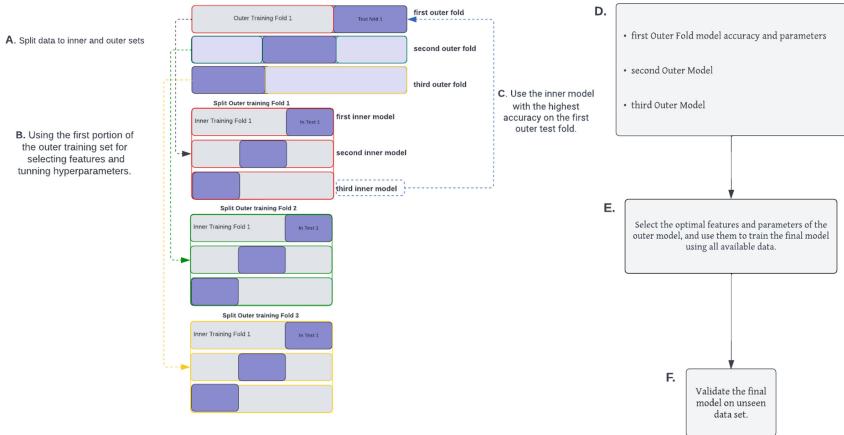


Fig. 1. A. The data were split into 80% training and 20% test sets. The training data was further divided into three folds for training and testing. B. Optimal hyperparameters and best features for XGBoost were selected through multiple trials and inner folds. E. The best model was then trained on the original training data, and F. was evaluated on the original test data.

7 Results and Discussion

7.1 Data Flow

Figure 2 depicts the data flow architecture. The pipeline commences with acquiring data from web sources, including Event Registry for news content and Historical Options Data for daily closing prices.

Subsequently, as described in Sect. 4, the text documents undergo preprocessing, which entails the identification of target companies and the substitution of their names with a standardized version, followed by sentence and paragraph segmentation and cleaning of the resulting splits.

In the sentiment analysis phase, the VADER and TextBlob algorithms quantify sentiment at the sentence, paragraph, and article levels. Furthermore, targeted sentiment analysis and DistilBERT are applied to sentences and paragraphs to yield transformed sentiment scores. The features are generated based on the dates, sentiment scores, returns, and news publishers, with returns calculated and the difference between two consecutive business days determined.

The data were divided into the original training and test sets, with 80% of the data utilized in the nested cross-validation phase to determine the optimal hyperparameter values and select the most informative features via the XGBoost importance measure. Finally, the remaining 20% of the data is employed to evaluate the performance of the final model, which incorporates the optimal combination of hyperparameters and features.

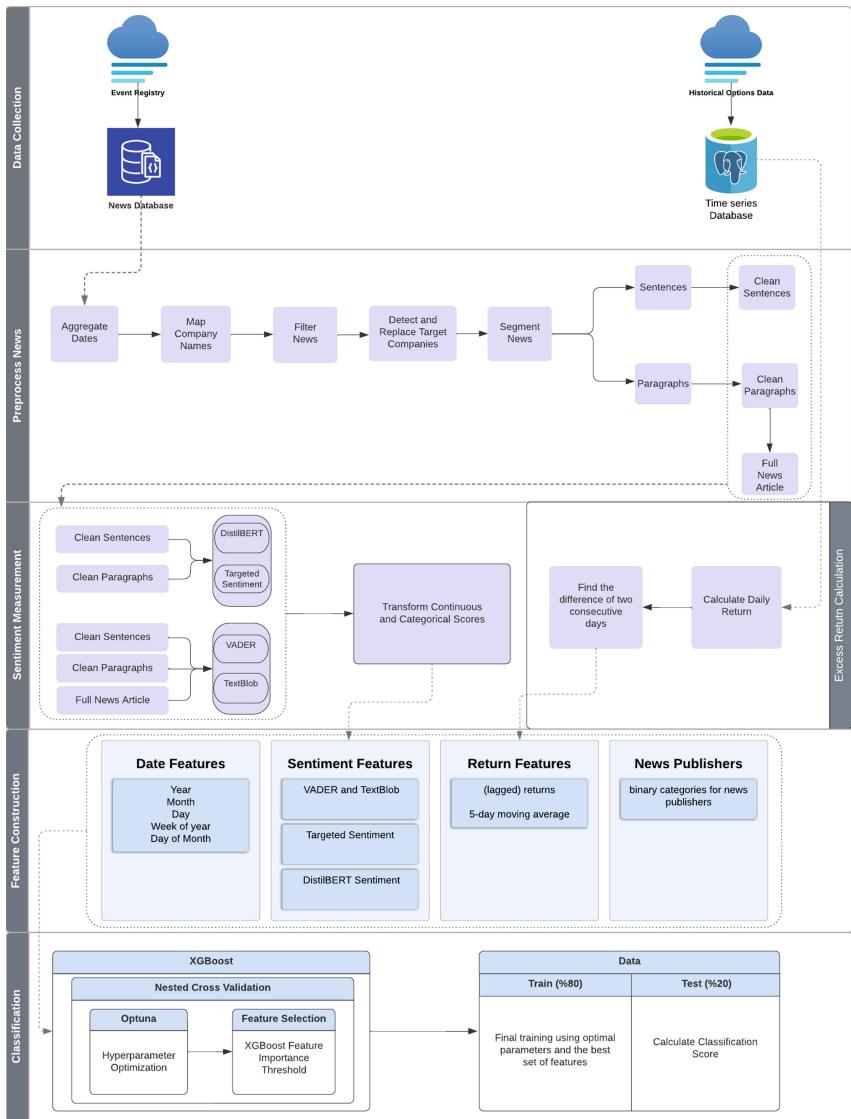


Fig. 2. Data Flow Diagram: From Web Source to Final Model (1) Data collection from Event Registry (news) and Historical Options Data (stock prices) (2) Preprocessing: target company identification and standardization, sentence and paragraph segmentation, and cleaning (3) Sentiment Analysis: VADER, TextBlob, DistilBERT, and targeted sentiment measures (4) Features Constructed: date, sentiment score, returns, and news publisher (5) Data divided: 80% for hyperparameter optimization and best feature selection, 20% for final model evaluation.

We opted for accuracy as the selection criteria to choose the best models within the nested CV process. In classification tasks, accuracy is a prevalent performance metric that calculates the rate of correct classifications out of the total number of predictions made. This can be expressed mathematically using the following equation:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

where TP (true positives) represents the number of correctly classified up movements, TN (true negatives) represents the number of correctly classified down movements, FP (false positives) represents the number of down movements incorrectly classified as up movements, and FN (false negatives) represent the number of up movements incorrectly classified as down movements.

To make the results robust, we repeated each sector's XGBoost classification process 20 times and reported the average accuracy across all runs.

7.2 Results

Table 3 provides the results of the nested CV experiments to evaluate the accuracy of the XGBoost model. The columns “Min Accuracy”, “Mean Accuracy” and “Max Accuracy” displays the minimum, average and maximum accuracy of 20 experiments on the original test set. The “A priori UP” column denotes the a priori probability for upside changes, i.e., the ratio of up movements of returns to the total number of observations in the original test set. In conclusion, the results of the nested CV experiment indicate the highest mean accuracy (67.3%) in the petroleum industry, which includes two companies, ExxonMobil (XOM) and Chevron (CVX). In contrast, the mean accuracy in other sectors is between 50% and 51.8%, which is very close to the a priori UP probabilities.

Table 3. The XGBoost model's accuracy in nested cross-validation experiments

Sector	Min accuracy	Mean accuracy	Max accuracy	A priori “Up”
IBM-INTC	0.440	0.518	0.576	0.486
JPM-GS	0.467	0.512	0.559	0.494
AAPL-MSFT	0.457	0.500	0.569	0.493
V-CSCO	0.448	0.501	0.563	0.476
XOM-CVX	0.582	0.673	0.743	0.497

The performance of the XGBoost classifier can be visualized using a confusion matrix, as presented in Fig. 3. The labels zero and one in the matrix correspond to the “Down” and “Up” directions of the return in the following business day, respectively. Each sub-figure in Fig. 3 represents the percentage of TN predictions in the up-left square, FN classifications in the bottom left square, FP predictions

in the top-right square, and TP predictions in the right-bottom square. Each square's integer value represents the corresponding labels' rounded mean across 20 experiments. The diagonal elements of the confusion matrix indicate the number of instances where the predicted label matches the true label. In contrast, the off-diagonal elements correspond to instances misclassified by the model. The petroleum industry, consisting of ExxonMobil (XOM) and Chevron (CVX), shows a better model performance with higher diagonal values in the confusion matrix. However, in other sectors, the model can identify downward movements of the return in the following business day but lacks accuracy in predicting upward returns.

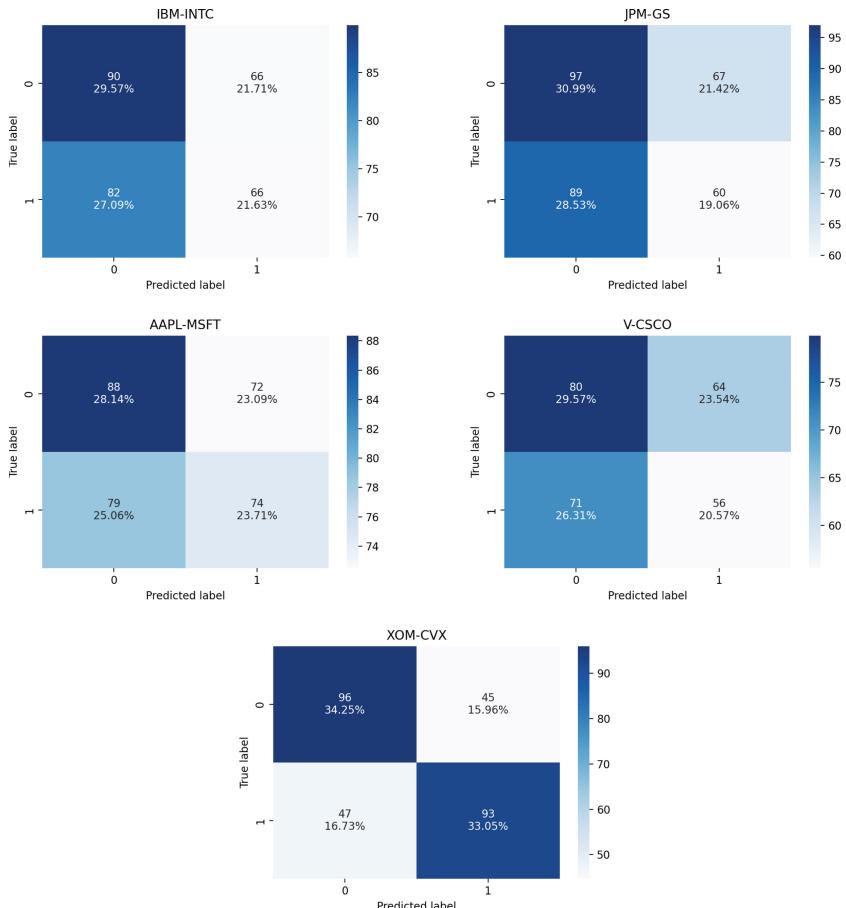


Fig. 3. The confusion matrix for five sectors.

Several factors explain this observation. On one side, ExxonMobil and Chevron are among the largest oil and gas companies [28], and they are exposed

to the oil market. Upon further analysis, a positive correlation was discovered between the prices of CVX, XOM, and CL.⁹ Specifically, the correlation coefficients between the daily price series of CVX and XOM, CVX and CL, and XOM and CL between January 1st, 2017, and February 28th, 2020, were 0.60, 0.80, and 0.51, respectively. This indicates that the change in the price of CVX and XOM is highly associated with oil prices.

This positive correlation can be explained by the fact that the oil and gas industry is highly interdependent, with many factors affecting the prices of these assets similarly. For example, changes in the global demand and supply of crude oil can affect the prices of CVX and XOM, as well as the price of CL, which is a benchmark for global crude oil prices. Similarly, geopolitical events, natural disasters, and government policy changes can all impact these assets' prices, leading to a positive correlation between them.

However, the stock prices of companies such as J.P. Morgan (JPM), Microsoft (MSFT), and Apple (AAPL) are influenced by a range of factors, including financial performance, market share, product development, and other business-specific operations. Furthermore, macroeconomic factors such as interest rates, inflation, and exchange rates can also affect the stock prices of these companies and generate news with varying sentiments that lead to lower accuracy in daily return forecasts. This results in more mixed signals in the news that could offset its effects.

While investigating the sentiment scores, we observed an interesting pattern related to targeted sentiment. Figure 4 displays the distribution of average targeted sentiment scores of sentences for the petroleum industry represented by (XOM-CVX), the financial sector represented by (JPM-GS), and the information technology sector represented by (AAPL-MSFT). The *x*-axis represents the average probability of the sentiment score on a particular date, while the *y*-axis shows the number of days with such probability. According to the targeted sentiment distribution plots in Fig. 4, the probability of negative sentiments in the petroleum industry is more tightly distributed than in the financial sector, with a clear demarcation between negative, neutral, and positive sentiments. In contrast, the sentiment distribution is more evenly spread across positive, negative, and neutral sentences in the case of the two other sectors. We did not detect similar patterns when comparing the sentiment distributions of Vader, Textblob, and Distilbert scores. The targeted sentiment scores of the petroleum industry revealed new characteristics of the aggregated sentiment scores, indicating that the model achieves higher classification accuracy when the probability distribution of sentiment categories is clustered.

8 Conclusion and Future Work

In this study, we investigated the impact of financial news on ten companies across five sectors, employing four different sentiment scores at both fine-

⁹ The ticker symbol represents crude oil futures contracts traded on the New York Mercantile Exchange (NYMEX) and the Chicago Mercantile Exchange (CME).

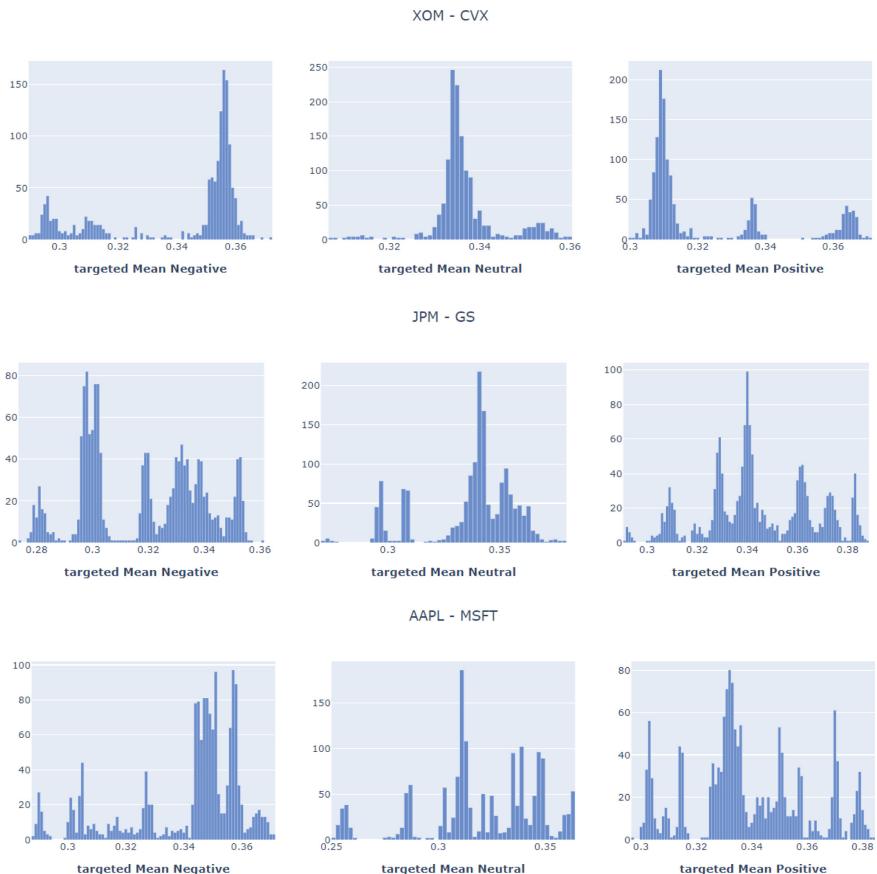


Fig. 4. The distribution of positive, negative, and neutral targeted sentiments across sentences for the petroleum industry (top), financial services (middle), and information technology (bottom).

grained and coarse-grained levels to forecast the next day's return direction. Our findings demonstrate that sentiment scores influence different sectors asymmetrically, with the petroleum industry exhibiting greater predictability using sentiment analysis than other sectors. This result is consistent with the research of Wan et al. in [12], who also utilized targeted sentiment scores and observed a more significant influence of sentiment scores on the financial industry. The results highlight the effectiveness of targeted sentiment analysis in identifying asymmetries that are not apparent at the full-text level.

Our study has several limitations, including the daily analysis of news, which overlooks intraday price fluctuations, the limited integration of technical indicators, and the need to study more sectors within a broader window to validate our findings.

For future research, the following three avenues could be pursued:

1. Investigate the impact of financial news and sentiment analysis on intraday price fluctuations. By analyzing news and sentiment more frequently throughout the trading day, we can gain a more detailed understanding of the immediate effects of news on market dynamics.
2. Incorporate a more comprehensive set of technical indicators in conjunction with targeted sentiment analysis to enhance prediction results. Combining the fundamental and technical analysis insights can create more accurate and robust forecasting models.
3. Expand the scope of the study to include more sectors and a wider time window. This will help validate our findings that the distribution of targeted sentiment scores is an essential indicator when applying sentiment analysis to financial markets and may reveal additional insights into sector-specific behavior.

In conclusion, our study contributes to understanding how sentiment analysis can be used to forecast the direction of the next day's return in different sectors. The findings have important implications for investors, policymakers, and researchers interested in using sentiment analysis to inform financial market decisions. By addressing the limitations of our study and exploring the suggested future research directions, we can further refine our understanding of the complex relationship between news sentiment and financial market dynamics.

References

1. Carosia, A.E.O., Coelho, G.P., Silva, A.E.A.: Analyzing the Brazilian financial market through Portuguese sentiment analysis in social media. *Appl. Artif. Intell.* **34**, 1–19 (2020). <https://doi.org/10.1080/08839514.2019.1673037>
2. Jing, N., Wu, Z., Wang, H.: A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction. *Expert Syst. Appl.* **178**, 115019 (2021). <https://doi.org/10.1016/j.eswa.2021.115019>
3. Johnman, M., Vanstone, B.J., Gepp, A.: Predicting FTSE 100 returns and volatility using sentiment analysis. *Account. Financ.; Wiley Online Library* **58**, 253–274 (2018). <https://doi.org/10.1111/acfi.12373>
4. Yadav, A., Vishwakarma, D.K: Sentiment analysis using deep learning architectures: a review. In: *Artificial Intelligence Review*, vol. 53, pp. 4335–4385. Springer (2020). <https://doi.org/10.1007/s10462-019-09794-5>
5. Tetlock, P.C: Giving content to investor sentiment: the role of media in the stock market. *J. Financ.* **62**, 1139–1168 (2007). <https://doi.org/10.1111/j.1540-6261.2007.01232.x>
6. Loughran, T., McDonald, B.: When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *J. Financ.* **66**, 35–65 (2011). <https://doi.org/10.1111/j.1540-6261.2010.01625.x>
7. Nardo, M., Petracco-Giudici, M., Naltsidis, M.: Walking down Wall Street with a tablet: a survey of stock market predictions using the Web. *J. Econ. Surv.* **30**, 356–369 (2016). <https://doi.org/10.1111/joes.12102>

8. Li, X., Wu, P., Wang, W.: Incorporating stock prices and news sentiments for stock market prediction: a case of Hong Kong. *Inf. Process. Manag.* **57**, 102212 (2020). <https://doi.org/10.1016/j.ipm.2020.102212>
9. Hutto, C., Gilbert, E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 8, pp. 216–225. (2014). <https://doi.org/10.1609/icwsm.v8i1.14550>
10. Feuerriegel, S., Gordon, J.: News-based forecasts of macroeconomic indicators: a semantic path model for interpretable predictions. *Eur. J. Oper. Res.* **272**, 162–175 (2019). <https://doi.org/10.1016/j.ejor.2018.05.068>
11. Liu, J., Chen, Y., Liu, K., Zhao, J.: Attention-based event relevance model for stock price movement prediction. In: China Conference on Knowledge Graph and Semantic Computing, pp. 37–49. Springer (2017). https://doi.org/10.1007/978-3-319-69627-9_4
12. Wan, X., Yang, J., Marinov, S., Calliess, J.P., Zohren, S., Dong, X.: Sentiment correlation in financial news networks and associated market movements. *Sci. Rep.* **11**, 1–12 (2021). <https://doi.org/10.1038/s41598-021-82338-6>
13. Malo, P., Sinha, A., Korhonen, P., Wallenius, J., Takala, P.: Good debt or bad debt: detecting semantic orientations in economic texts. *J. Assoc. Inf. Sci. Technol.*; Wiley Online Library **65**, 782–796 (2014). <https://doi.org/10.1002/asi.23062>
14. Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C. D.: Stanza: A Python natural language processing toolkit for many human languages. In: Association for Computational Linguistics (ACL) System Demonstrations (2020). www.nlp.stanford.edu/pubs/qi2020stanza.pdf
15. De Wilde, B.: Textacy: NLP, before and after spaCy (2022). www.pypi.org/project/textacy/
16. Loria, S.: Textblob: simplified text processing. Release 0.16 (2021). www.textblob.readthedocs.io/
17. Sohangir, S., Petty, N., Wang, D.: Financial sentiment lexicon analysis. In: 2018 IEEE 12th International Conference on Semantic Computing (ICSC), pp. 286–289 (2018). <https://doi.org/10.1109/ICSC.2018.00052>
18. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30, pp. 5998–6008. (2017). www.arxiv.org/abs/1706.03762
19. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186. Association for Computational Linguistics (2019). <https://doi.org/10.18653/v1/n19-1423>
20. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: a robustly optimized Bert pretraining approach (2019). <https://doi.org/10.1145/3340531.3412026>
21. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter (2019). www.arxiv.org/abs/1910.01108
22. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J.: Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45 (2020). www.aclweb.org/anthology/2020.emnlp-demos.6
23. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: the long-document transformer (2020). <https://doi.org/10.48550/arXiv.2004.05150>

24. Hamborg, F., Donnay, K.: NewsMTSC: A dataset for (multi-)target-dependent sentiment classification in political news articles. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 1663–1675. Association for Computational Linguistics (2021). <https://doi.org/10.18653/v1/2021.eacl-main.142>
25. 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 (2016). <https://doi.org/10.1145/2939672.2939785>
26. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2623–2631 (2019)
27. Varma, S., Simon, R.: Bias in error estimation when using cross-validation for model selection. BMC Bioinform. **7**, 1–8 (2006). <https://doi.org/10.1186/1471-2105-7-91>
28. Reiff, N.: The World’s Top 10 Oil Companies (2023). www.investopedia.com/articles/personal-finance/010715/worlds-top-10-oil-companies.asp



Improving Neural Network Using Jaya Algorithm with Opposite Learning for Air Quality Prediction

Iyad Abu Doush^{1(✉)}, Khalid Sultan², Ahmad Alsaber³, Dhari Alkandari⁴, and Afsah Abdullah⁵

¹ Department of Computing, College of Engineering and Applied Sciences, American University of Kuwait, Salmiya, Kuwait

idoush@auk.edu.kw

² Department of Computer Engineering, College of Engineering and Applied Sciences, American University of Kuwait, Salmiya, Kuwait

ksultan@auk.edu.kw

³ College of Business and Economics, Department of Management, American University of Kuwait, 15 Salem Al Mubarak St, Salmiya P.O.Box 3323,, Safat 13034, Kuwait

aalsaber@auk.edu.kw

⁴ Department of Earth and Environmental Sciences, Kuwait University, P.O. Box 25944, Safat 1320, Kuwait

dhary.alkandary@ku.edu.kw

⁵ The Office of Research and Grants, American University of Kuwait, Salmiya, Kuwait

[aabduallah1@auk.edu.kw](mailto:aabdullah1@auk.edu.kw)

Abstract. The Multi-Layer Perceptron Neural Network (MLP) is the commonly used Feedforward Neural Network (FNN) for tackling classification and prediction problems. The efficiency of MLP relies on the appropriate selection of its weights and biases. Usually, a gradient-based technique is used for tuning the selection of these parameters during the learning process. This technique suffers from its slow convergence and being stuck in local optima. Predicting urban air quality is vital to prevent urban air pollution and improve the life of residents. The air quality index (AQI) is a quantitative air quality tool. In this paper, an enhanced Jaya optimization algorithm is used to improve the MLP outcome (called EOL-Jaya-MLP). The opposite-learning method is used to improve the algorithm search space exploration. A three-year dataset from air quality monitoring stations is used in this study. The proposed technique is compared against the original Jaya and six machine learning techniques. Interestingly, the EOL-Jaya-MLP outperforms other techniques when predicting the AQI.

Keywords: Jaya Optimization Algorithm · Optimization · Feedforward Neural Networks · Multi-Layer Perceptron · Swarm Intelligence

1 Introduction

Modern human activities inevitably involve energy usage and its consequences. Kerosene, coal, and straw burning are only a few humans-caused bases of air pollution, along with emissions from businesses, vehicles, and aerosol cans. Day by day, a variety of detrimental pollutants, including NO_2 , CO , NH_3 , PM , CO_2 , O_3 , Pb , SO_2 , etc., are discharged into the atmosphere. The tendency toward increased global air pollution has grown clearer with the spread of high-intensity human engineering activities [17, 19, 32]. Extreme weather is a frequent occurrence that has contributed to numerous disasters, and air pollution poses a serious threat to human health [10, 31].

Machine learning (ML)-based and statistical AQI forecasting are the two primary categories. To map the association between the goal data and the time-series historical data, statistical forecasting techniques create data-driven mathematical models. These techniques can offer precise and timely predictions with a straightforward mathematical basis. Then again, once combined with environmental applications, ML algorithms, with their demonstrated dominance and effectiveness in many predicting issues, might be quite alluring to academics. Many models, counting statistical, physical, deterministic, and ML, have been employed to predict AQI in the literature. The conventional methods based on statistics and probability are exceedingly intricate and ineffective. It has been verified that the ML-based AQI prediction models are more dependable and consistent. Data collecting was made simple and accurate by modern technologies and sensors. Only ML algorithms are capable of handling the rigorous analysis needed to make accurate and trustworthy predictions from such vast environmental data. The significance of supervised ML algorithms for practical environment protection challenges was extensively covered [5].

Wang et al. [25] used a radial basis neural network (NN) to estimate SO_2 levels and came to the conclusion that the outcomes could be useful for AQI forecasting in the future. In a similar vein, [8] demonstrated that in terms of predicting various pollutant concentrations, a feedforward NN outperformed multilinear regression. For predicting the pollutant levels necessary to compute the AQI per hour in California, the work proposed in [9] used the support vector machine (SVM), another reliable ML approach. The classification accuracy of the models was 94.1% for the projected air quality. A novel method of forecasting the AQI directly using ensemble learning was suggested in [15]. In this method, the projected AQI amounts from five separate regression and ML models were additionally processed and then introduced to ensemble models to improve forecasting accuracy. Huang et al. [20] present a solution for optimizing the back-propagation (BP) neural network based on an improved particle swarm optimization (PSO) algorithm to predict the air quality index (AQI). The improved PSO algorithm is used to optimize not only the various strategies of the inertia weight but also the learning factor while ensuring its high global search capability from the very beginning and later enabling its fast convergence to the optimal solution.

Neural Networks come in different shapes including Feedforward Neural Network (FNN), backpropagation Neural Network (BNN), convolutional neural net-

work, etc. [1, 2, 4, 18]. The most commonly used one is FNN which takes the input data and trains the model by connecting neurons in one direction using different layers [16]. The Multi-Layer Perceptron (MLP) is a type of FNN that is applied when tackling non-linear problems as each layer contains more than one perceptron.

The MLP learns from experience by minimizing the cost function which is usually the Mean Square Error (MSE). The MSE evaluates the neuron weights and biases by comparing the obtained output (predicted) with the actual output [13]. The neuron weights are optimized to minimize the MSE in MLP using a gradient-based algorithm represented by the back-propagation (BP) algorithm [13]. However, the gradient-based algorithm has two shortcomings: slow search time and being stuck in local optima [13]. So, heuristic-based techniques are utilized to overcome these issues.

The JAYA algorithm, derived from the well-known principle of “survival of the fittest,” is highly efficient. It focuses on finding the best solution by concentrating the solutions in the population. One notable advantage of JAYA is its simplicity of implementation, as it does not require algorithm-specific parameters [24, 30]. Consequently, researchers have utilized this algorithm to tackle various optimization problems [7, 12, 26–28, 30]. Nonetheless, JAYA does have a few drawbacks, such as the issue of premature convergence, wherein it may become trapped in local optima once it reaches the equilibrium state [14]. To address these limitations, several modified versions of the JAYA algorithm have been proposed in the literature, including self-adaptive [22], multi-population [21], chaotic [14], and performance-guided approaches [30].

Elite opposition-based learning (EOL) is a special type of learning proposed by Zhou et al. [23]. The technique insight is coming from developing a new set of solutions that is the opposite of elite (best) solutions with the hope to come closer to the global optimal. EOL is used to enhance the learning process of different optimization algorithms such as harris hawks [23] and grasshopper optimization [29].

In this paper, the Jaya algorithm exploration power is improved by using EOL. Such a change in the algorithm can enhance its effectiveness in finding more favorable regions. This paper develops a machine-learning model based on MLP-NN enhanced using EOL-Jaya (EOL-Jaya-MLP) for AQI prediction. First, the data is preprocessed to be cleaned and to fit the model. Second, the NN model using EOL-Jaya-MLP is utilized for AQI prediction. Lastly, the prediction results are evaluated using the mean square error (MSE) performance measure. The performance of the proposed EOL-Jaya-MLP model is compared against the original Jaya algorithm (JAYA-MLP) and six classical machine learning techniques (i.e., linear regression (LR), linear discriminant analysis, KNeighbours Classifier (KNN), Decision Tree Classifier, Gaussian NB, and Support Vector Machine (SVM)). Interestingly, the proposed EOL-Jaya-MLP is able to produce high-quality AQI prediction results.

2 Jaya Optimization Algorithm with Elite Opposition-Based Learning for MLP

The operational procedure of the suggested model for predicting AQI is as follows: Initially, the necessary raw data is gathered, followed by the selection of the most significant features. Next, the MLP network is constructed, and the EOL-Jaya-MLP approach is employed to train the MLP network, aiming to discover the optimal combination of weights and biases that yield precise predictions. Lastly, the resulting AQI predictions are assessed for their effectiveness.

The MLP is used to predict the output for each instance in the dataset, and the prediction accuracy is measured using mean square error (MSE). The MSE is a common metric that is the difference between the MLP output (prediction) and the actual data. The equation for MSE is demonstrated in Eq. (1). Note that y is the actual value, \hat{y} is the predicted value, and k is the number of training samples.

$$MSE = \frac{1}{k} \sum_{i=1}^k (y - \hat{y})^2 \quad (1)$$

To predict \hat{y} , the MLP is fed with the current weights and biases, allowing it to determine the accurate output corresponding to each data input. The quality of the MLP (i.e., the used weights and biases) is identified by comparing the MLP output against the actual data output.

Elite opposition-based learning (EOL) is used to improve the Jaya algorithm. To elaborate, let $\mathbf{X} = < x_1, x_2, \dots, x_m >$ be the elite candidate solution that has m decision variables. EOL can be applied to the candidate solutions by calculating the elite opposite-based solution \mathbf{X}° using Eq. 2

$$\mathbf{X}^\circ = < x_1^\circ, x_2^\circ, \dots, x_m^\circ >, \text{ where } x_i^\circ = \delta(da_i + db_i) - x_i \quad (2)$$

Note that $\delta \in (0, 1)$ and it controls the amount of opposition, da_i and db_i are the dynamic boundaries which are computed as follows:

$$da_i = \min(x_i), \quad db_i = \max(x_i) \quad (3)$$

A correction mechanism is used if the opposite decision variable x_i° is outside $[LB_i, UB_i]$ using the equation:

$$x_i^\circ = rand(LB_i, UB_i), \text{ if } x_i^\circ < LB_i \text{ or } x_i^\circ > UB_i \quad (4)$$

Note that $rand(LB_i, UB_i)$ is a random number between LB_i and UB_i . This technique is used to improve the set of generated weights and biases in the population. The proposed EOL-Jaya-MLP is presented in Algorithm 1.

Algorithm 1 EOL-Jaya-MLP Pseudo-code

```

Initialize the parameters of Jaya ( $MaxItr$ ).
randomly initialize P solutions within the problem bounds ( $LB, UB$ )
Evaluate the initial  $P$  solutions
Set  $i = 0$ 
while  $i \leq MaxItr$  do
    Select the best individual as the elite individual  $X_e$  from  $X_i$ 
    Update the dynamic interval boundaries  $[da_i, da_j]$  in  $X_i$  according to Eq. 4
    Generate the opposite population
    Select the fittest P individuals from opposite populations and the population
    find the best  $X^{best}$  and worst  $X^{worst}$  solution in the current population
    for  $j = 1$  to  $P$  do
        for  $i = 1$  to  $n + m$  do Calculate  $X_{i,j}^{new}$ 
             $\triangleright$  Make sure that  $X_{i,j}^{new}$  within the problem bounds
            if  $(X_{i,j}^{new} \leq LB_j)$  then  $X_{i,j}^{new} = LB_j$ 
            else if  $(X_{i,j}^{new} \geq UB_j)$  then  $X_{i,j}^{new} = UB_j$ 
            end if  $\triangleright$  Replace the current solution in case of improvement
        end for
        if  $f(X_i^{new}) < f(X_i)$  then  $X_i \leftarrow X_i^{new}$ 
        end if
    end for
     $i = i + 1$ 
end while  $\triangleright$  End of Runs
Return the best solution

```

3 Experimental Results

The effectiveness of the proposed EOL-Jaya-MLP trainer to accurately predict AQI is investigated using a dataset on pollutants gathered from Kuwait's Environmental Public Authority (K-EPA). Four environmental monitoring stations provided the data between January 1, 2018, and December 31, 2021. Namely, Ali-Sabah, Fahheel, Mansouria, and Rumaithiya. Each station's 24-hour aggregate for PM_{10} , SO_2 , and NO_2 as well as its 8 h aggregation for O_3 and CO were calculated using the daily data. The micrograms per cubic meter ($\mu\text{g}/\text{m}^3$) units were used to measure all contaminants. The average is deemed missing if less than 75% of the data is provided (that is., less than 6 h), according to US Environmental Protection [3]. We made use of the [6] recommended Air Quality Index (AQI).

The correlation coefficient matrix is used to select the most promising features for AQI prediction. Clearly, Fig. 1 demonstrates that the main influencing features for AQI are $PM_{2.5}$, $PM10$, CO , NO_2 , O_3 , and SO_2 .

The experiments were conducted using the Python programming language and executed on a system equipped with an Intel(R) Core(TM) i7-7700HQ CPU running at a frequency of 2.80GHz, along with 16 GB of RAM.

The configuration parameters for EOL-Jaya-MLP and Jaya-MLP are defined as follows: a population size of 25, a maximum of 100 iterations, and 30 runs.

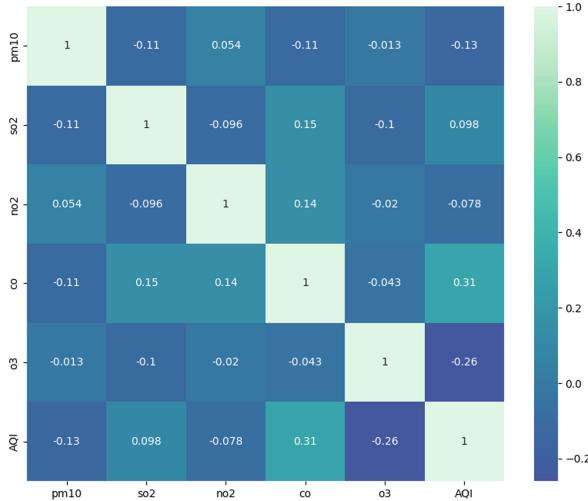


Fig. 1. Colored correlation block diagram for each Index

The weight and bias values are randomly assigned within the range of -1 to 1 . The feedforward neural network consists of a hidden layer with seven neurons. It is important to note that the dataset is divided into a 90

As shown in Fig. 2 and Table 1 the proposed EOL-Jaya-MLP outperforms Jaya-MLP in terms of best MSE when predicting AQI on the dataset from the four stations, and it outperforms Jaya-MLP in terms of mean MSE on three out of four datasets.

Table 1. The mean MSE and best MSE when using the techniques EOL-Jaya-MLP and Jaya-MLP when predicting AQI for each one of the four stations

	Mean MSE		Best MSE	
	EOL-Jaya-MLP	Jaya-MLP	EOL-Jaya-MLP	Jaya-MLP
Ali-sabah	0.083169656	0.098549169	0.037637432	0.050442
Fahaheel	0.105735534	0.119378516	0.011107124	0.025936
Mansouria	0.090753734	0.102748294	0.031313891	0.041229
Rumaithiya	0.059473693	0.054699359	0.012034273	0.01973

For further validation, the proposed EOL-Jaya-MLP is compared against six classical machine learning techniques in terms of mean MSE using Fahaheel station data. Table 2 shows that the proposed algorithm obtains the best results.

To investigate potential disparities in Mean Squared Error (MSE) between the outcomes of EOL-Jaya-MLP and Jaya-MLP, a lower-tailed Mann-Whitney two-sample rank-sum test was conducted. This test serves as an alternative to the independent samples t-test but operates under distinct assumptions [11]. As illustrated in Fig. 3, the results of the lower-tailed Mann-Whitney U test were

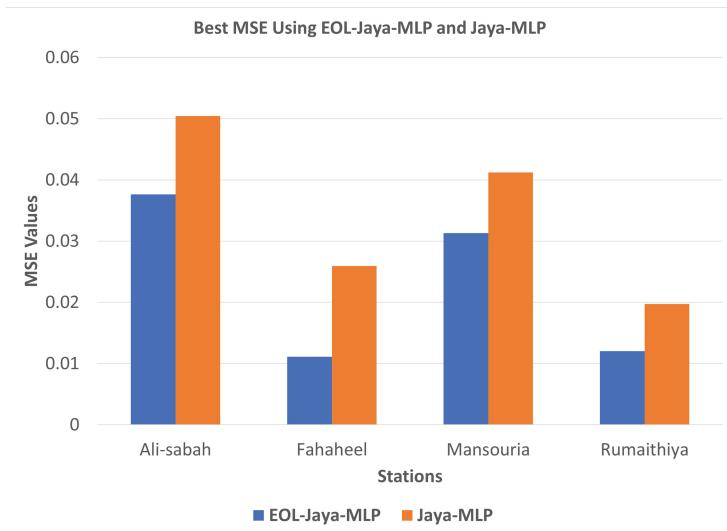


Fig. 2. The best MSE when using the techniques EOL-Jaya-MLP and Jaya-MLP

Table 2. Comparing the proposed algorithm with other machine-learning techniques in terms of MSE

Fahheel station		
	MSE mean	MSE std
LogisticRegression	0.831794872	0.252803349
LinearDiscriminantAnalysis	0.852820513	0.290401341
KNeighborsClassifier	0.913846154	0.349992392
DecisionTreeClassifier	0.936923077	0.322374491
GaussianNB	1.116410256	0.21772598
SVM	0.812307692	0.226033458
EOL-Jaya-MLP	0.105735534	0.072358783
Jaya-MLP	0.119378516	0.059941818

statistically significant at a significance level of 0.05, with $z = -1.69$ and $p = 0.045$. Additionally, in the boxplot depicted in Fig. 4, the mean rank for the Enhanced Jaya group was 112.91, while the mean rank for the Jaya group was 128.09. These findings indicate that the MSE distribution of the Enhanced Jaya group significantly differed from that of the Jaya group.

Variable	Enhanced Jaya		Jaya		U	z	p
	Mean Rank	n	Mean Rank	n			
MSE	112.91	120	128.09	120	6,289.50	-1.69	.045

Fig. 3. Lower-tailed Mann-Whitney U test

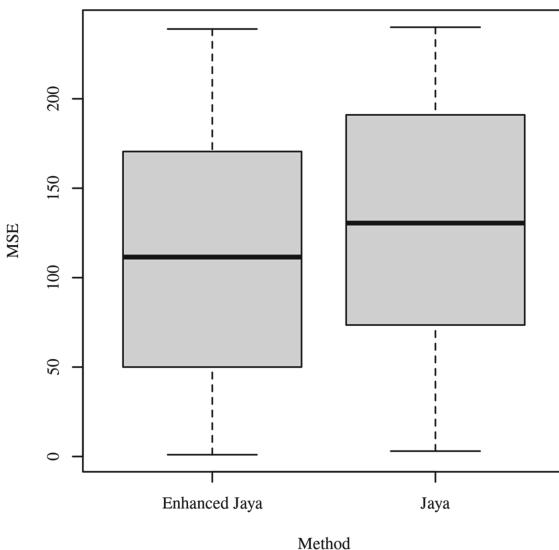


Fig. 4. The boxplot of the ranks of MSE

4 Conclusion and Future Work

In this paper, a new framework is proposed to predict the AQI using an improved Jaya optimization algorithm with elite opposition-based learning to train a multilayer perceptron feedforward neural network (EOL-Jaya-MLP). The proposed framework looks for promising values for weights and biases to have accurate AQI predictions. Such a framework can help in helping stakeholders to indicate future air pollution and take decisions for improving the air quality for urban residents. The evaluation of the framework involves four datasets obtained from different stations that collect pollutant data. Each dataset is divided into 90% for training and 10% for testing purposes. The performance of the proposed method is compared with the original Jaya algorithm (Jaya-MLP) and six other machine learning techniques. The results demonstrate the competitiveness of the proposed algorithm in terms of Mean Square Error (MSE), indicating its effectiveness in accurately predicting AQI.

In future work, we intend to investigate different strategies for parameter tuning within the EOL-Jaya-MLP framework. Also, expanding the research to real-time AQI prediction is an area of considerable importance to enable timely and accurate predictions, facilitating proactive measures to mitigate air pollution and protect public health. Furthermore, exploring various feature selection techniques can enhance the accuracy and efficiency of the proposed framework is another future direction. Finally, investigating the potential for hybridization of EOL-Jaya-MLP with other machine learning techniques could be explored.

Acknowledgment. This research project is supported by the American University of Kuwait internal grant fund.

References

1. Abu-Doush, I., Ahmed, B., Awadallah, M.A., Al-Betar, M.A., Rababaah, A.R.: Enhancing multilayer perceptron neural network using archive-based harris hawks optimizer to predict gold prices. *J. King Saud Univ.-Comput. Inf. Sci.* **101557** (2023)
2. Abu Doush, I., Awadallah, M.A., Al-Betar, M.A., Alomari, O.A., Makhadmeh, S.N., Abasi, A.K. and Alyasseri, Z.A.A.: Archive-based coronavirus herd immunity algorithm for optimizing weights in neural networks. *Neural Comput. Appl.* **1–19** (2023)
3. US Environmental Protection Agency: National ambient air quality standards for ozone; final rule. *Fed. Regist.* **80**(206), 65–292 (2015)
4. Al-Betar, M.A., Awadallah, M.A., Doush, I.A., Alomari, O.A., Abasi, A.K., Makhadmeh, S.N., Alyasseri, Z.A.A.: Boosting the training of neural networks through hybrid metaheuristics. *Clust. Comput.* **1–23** (2022)
5. Al-Jamimi, H.A., Al-Azani, S., Saleh, T.A.: Supervised machine learning techniques in the desulfurization of oil products for environmental protection: a review. *Process. Saf. Environ. Prot.* **120**, 57–71 (2018)
6. Al-Shayji, K., Lababidi, H., Al-Rushoud, D., Al-Adwani, H.: Development of a fuzzy air quality performance indicator. *Kuwait J. Sci. Eng.* **35**, 101–126 (2008)
7. Awadallah, M.A., Al-Betar, M.A., Hammouri, A.I., Alomari, O.A.: Binary JAYA algorithm with adaptive mutation for feature selection. *Arab. J. Sci. Eng.* **1–16** (2020)
8. Cai, M., Yin, Y., Xie, M.: Prediction of hourly air pollutant concentrations near urban arterials using artificial neural network approach. *Transp. Res. Part D: Transp. Environ.* **14**(1), 32–41 (2009)
9. Castelli, M., Clemente, F.M., Popović, A., Silva, S., Vanneschi, L.: A machine learning approach to predict air quality in California. *Complexity* (2020)
10. Chen, X., Yin, L., Fan, Y., Song, L., Ji, T., Liu, Y., Tian, J., Zheng, W.: Temporal evolution characteristics of PM2. 5 concentration based on continuous wavelet transform. *Sci. Total Environ.* **699**, 134244 (2020)
11. Conover, W.J., Iman, R.L.: Rank transformations as a bridge between parametric and nonparametric statistics. *Am. Stat.* **35**(3), 124–129 (1981)
12. Degertekin, S.O., Lamberti, L., Ugur, I.B.: Sizing, layout and topology design optimization of truss structures using the JAYA algorithm. *Appl. Soft Comput.* **70**, 903–928 (2018)
13. Iyad Abu Doush and Ayah Sawalha: Automatic music composition using genetic algorithm and artificial neural networks. *Malays. J. Comput. Sci.* **33**(1), 35–51 (2020)
14. Farah, A., Belazi, A.: A novel chaotic JAYA algorithm for unconstrained numerical optimization. *Nonlinear Dyn.* **93**(3), 1451–1480 (2018)
15. Sankar Ganesh, S., Arulmozhivarman, P., Tatavarti, R.: Forecasting air quality index using an ensemble of artificial neural networks and regression models. *J. Intell. Syst.* **28**(5), 893–903 (2019)
16. Hecht-Nielsen, R.: Theory of the backpropagation neural network. In: *Neural Networks for Perception*, pp. 65–93. Elsevier (1992)

17. Kumar, P., Druckman, A., Gallagher, J., Gatersleben, B., Allison, S., Eisenman, T.S., Hoang, U., Hama, S., Tiwari, A., Sharma, et al.: The nexus between air pollution, green infrastructure and human health. *Environ. Int.* **133**, 105181 (2019)
18. Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D.: Face recognition: a convolutional neural-network approach. *IEEE Trans. Neural Netw.* **8**(1), 98–113 (1997)
19. Li, X., Zheng, W., Yin, L., Yin, Z., Song, L., Tian, X.: Influence of social-economic activities on air pollutants in Beijing, China. *Open Geosci.* **9**(1), 314–321 (2017)
20. Chao, L., Gao, L., Yi, J.: Grey wolf optimizer with cellular topological structure. *Expert Syst. Appl.* **107**, 89–114 (2018)
21. Rao, R.V., Saroj, A.: A self-adaptive multi-population based Jaya algorithm for engineering optimization. *Swarm Evol. Comput.* **37**, 1–26 (2017)
22. Rao, R.V., More, K.C.: Design optimization and analysis of selected thermal devices using self-adaptive Jaya algorithm. *Energy Convers. Manage.* **140**, 24–35 (2017)
23. Sihwail, R., Omar, K., Ariffin, K.A.Z., Tubishat, M.: Improved harris hawks optimization using elite opposition-based learning and novel search mechanism for feature selection. *IEEE Access* **8**, 121127–121145 (2020)
24. Singh, S.P., Prakash, T., Singh, V.P., Babu, M.G.: Analytic hierarchy process based automatic generation control of multi-area interconnected power system using Jaya algorithm. *Eng. Appl. Artif. Intell.* **60**, 35–44 (2017)
25. Wang, C.Y., Zhang, W.Y., Wang, J.J., Zhao, W.F.: The prediction of SO₂ pollutant concentration using a RBF neural network. In: Applied Mechanics and Materials, vol. 55, pp. 1392–1396. Trans Tech Publications (2011)
26. Wang, L., Huang, C.: A novel elite opposition-based Jaya algorithm for parameter estimation of photovoltaic cell models. *Optik* **155**, 351–356 (2018)
27. Wang, S.-H., Phillips, P., Dong, Z.-C., Zhang, Y.-D.: Intelligent facial emotion recognition based on stationary wavelet entropy and Jaya algorithm. *Neurocomputing* **272**, 668–676 (2018)
28. Congcong, W., He, Y.: Solving the set-union knapsack problem by a novel hybrid Jaya algorithm. *Soft. Comput.* **24**(3), 1883–1902 (2020)
29. Yildiz, B.S., Pholdee, N., Bureerat, S., Yildiz, A.R., Sait, S.M.: Enhanced grasshopper optimization algorithm using elite opposition-based learning for solving real-world engineering problems. *Eng. Comput.* **38**(5), 4207–4219 (2022)
30. Yu, K., Qu, B., Yue, C., Ge, S., Chen, X., Liang, J.: A performance-guided Jaya algorithm for parameters identification of photovoltaic cell and module. *Appl. Energy* **237**, 241–257 (2019)
31. Zanocco, C., Flora, J., Boudet, H.: Disparities in self-reported extreme weather impacts by race, ethnicity, and income in the united states. *PLOS Climate* **1**(6), e0000026 (2022)
32. Zheng, W., Li, X., Yin, L., Wang, Y.: Spatiotemporal heterogeneity of urban air pollution in china based on spatial analysis. *Rend. Lincei* **27**(2), 351–356 (2016)



Enhancing Early-Stage XAI Projects Through Designer-Led Visual Ideation of AI Concepts

Helen Sheridan^(✉) , Dymphna O'Sullivan , and Emma Murphy

School of Computer Science, TU Dublin, Grangegorman, Dublin, Ireland
`{helen.sheridan, dymphna.osullivan, emma.x.murphy}@tudublin.ie`

Abstract. The pervasive use of artificial intelligence (AI) in processing users' data is well documented with the use of AI believed to profoundly change users' way of life in the near future. However, there still exists a sense of mistrust among users who engage with AI systems some of this stemming from lack of transparency, including users failing to understand what AI is, what it can do and its impact on society. From this, the emerging discipline of explainable artificial intelligence (XAI) has emerged, a method of designing and developing AI where a systems decisions, processes and outputs are explained and understood by the end user. It has been argued that designing for AI systems especially for XAI poses a unique set of challenges as AI systems are often considered complex, opaque and difficult to visualise and interpret especially for those unfamiliar with their inner workings. For this reason, visual interpretations which match users' mental models of their understanding of AI are a necessary step in the development of XAI solutions. Our research examines the inclusion of designers in an early-stage analysis of an AI recruitment system taking a design thinking approach in the form of 3 workshops. We discovered that workshops with designers included yielded more visual interpretations of big ideas related to AI systems, and the inclusion of designers encouraged more visual interpretations from non-designers and those not typically used to employing drawing as a method to express mental models.

Keywords: Explainable Artificial Intelligence · Artificial Intelligence · Design Thinking

1 Purpose

It has been argued that those that develop AI systems may not be in the most optimum position to explain the inner workings and outputs of those systems especially when we consider the diverse range of users who engage with AI driven systems [1]. AI systems experts cannot unknow what they know which can prove problematic when trying to explain complex systems sometimes referred to as a "blackbox" [2, 3]. In more recent years teachings and processes from human computer interaction (HCI) including user experience (UX) design has shown promising results [4]. Including AI experts and computer scientists in the XAI—HCI development loop has been suggested but research has gone further and suggested that cross-disciplinary teams which might include those with a design background and including those stakeholders which have a vested interest in the AI domain under investigation might yield more useful results [5].

However, the practical implementation of engaging cross-disciplinary teams including designers and non-designers to yield meaningful evaluations, especially evaluations which might consider a cross section of user types, still requires further research and consideration [1]. The use of visuals in the ideation of AI systems has been considered fundamental in the exploration of mental models for AI systems but encouraging drawing, illustration and visual modes of expression from those not used to expressing ideas in this way can be challenging [6]. As such, we deem a gulf of evaluation exists around investigating users' mental models and understanding of AI as a stage in the development of XAI which we will explore by:

- Investigating design thinking as a method to facilitate visual ideation of AI with designers and non-designers.
- Identifying pain points and subsequently developing big ideas as a stage in the development of XAI solutions. See Fig. 1.
- Exploring the visual ideation results from the inclusion of designers in workshops versus workshops with no designers.



Fig. 1. Where pain point definition and big ideas development intersect ideation of XAI solutions can be facilitated.

Section 2 describes the background to the problem discussing AI as a design material and the importance of visuals in ideation. Section 3 outlines the methods used including describing our participants, workshop protocols, data collection and personas utilised during each workshop. Section 4 describes our results comparing quantity of visual ideas developed during each workshop and Sect. 5 concludes our discussion.

2 Background

2.1 AI as a Design Material

Design for AI has not received the same level of development as design for other more traditional information technology systems as AI systems are often considered complex and difficult to work with and design for [3]. UX designers have conveyed difficulties ideating and designing for AI since AI systems are often opaque and their outputs difficult to explain. As such they often find it "difficult to articulate what AI can/cannot do" [7] which can lead to designers missing the design potential of working AI systems [7].

Frustrations have been expressed when UX and design experts are included in the AI development lifecycle late in the process which may be exacerbated by the difficulties faced in facilitating collaboration between UX/designers and developers [7].

2.2 Importance of Visuals in Ideation

Ideation in design thinking is considered a fundamental mode or stage in the process of developing solutions to users' problems. Using drawing or visual interpretations to elicit solutions also gives some insight into participants' mental models of the problem being examined. One category used to examine mental models is through the use of drawings [8] amongst other techniques which have been further simplified into graphical and language-based approaches [9] with drawing categorised as a graphical approach. Drawing and visualisation have been described as allowing for expression of ideas facilitating bringing something from our consciousness into being and "as an externalisation of a concept or idea" [10]. For this reason, the use of drawing during the design thinking process is strongly encouraged as it allows for expression of complex internal ideas and language in a simple form allowing for self-expression and for the sharing of ideas amongst sometimes divergent participants. Previous research regarding AI and XAI focuses primarily on surveys, interviews, and observations, three well tested and researched techniques often employing think aloud methods, user testing of prototypes or of existing AI driven systems [14–16]. These methods have yielded valuable results in the understanding of system usability and model accuracy but what they may miss out on is exploration of users' cognitive interaction with AI [14], their mental models and visual interpretation of systems which are difficult to articulate and visualise and as such are difficult to explain.

3 Methods

We conducted three design thinking workshops with varying participant numbers and demographics with each workshop facilitated by experts in UX and design thinking. By implementing the empathise, define, and ideate modes of design thinking [10] we facilitated participants in the exploration of pain points leading to big ideas development as a precursor for further research which will involve the prototype and test modes of design thinking [11].

Workshop 1 Participants: 4 Male Computer Science undergraduates having recently faced CV rejection. 1 group empathising with Andrew Wilson Persona and facilitated by 1 expert in UX and HCI.

Workshop 2 Participants: 16 cross-disciplinary participants considered experts in philosophy, psychology, computer science, AI systems development. 3 groups, 1 group empathising with Andrew Wilson persona and 2 groups empathising with Maria Atkins persona and facilitated by 2 experts in UX and HCI.

Workshop 3 Participants: 11 computer science and 9 design undergraduates. 4 groups, 2 empathising with Andrew Wilson persona and 2 empathising with Maria Atkins Persona and facilitated by 2 experts in UX and HCI.

Each workshop group was introduced to two personas with corresponding scenarios, designed following interviews with a recruitment specialist from a large multinational and with recent graduates on the hunt for employment. Maria Atkins, a talent acquisition specialist who is losing trust in the results of her company's CV filtering system and Andrew Wilson, a recent atypical graduate struggling to find employment. See Fig. 2. Empathising took the form of empathy mapping, considering what each persona might say, think, feel or do when considering the AI system and As-Is-Scenario design which digs further into empathising by breaking down each persona's engagement with the AI system into steps and considering what they might do, think and feel during each step.



Fig. 2. Version of personas and scenario referenced during workshops.

In the As-Is-Scenario design, participants vote using five sticky dots on the areas of most pain for their persona, which leads to the grouping of similar pain points and the election of four main pain points. Proceeding pain point definition big ideas development begins where participants are asked to design 3 big ideas and 1 absurd big idea to solve each pain point. Groups played back or presented their findings at key moments after pain point definition and after big ideas development. See Fig. 3 for sample workshop sheets.

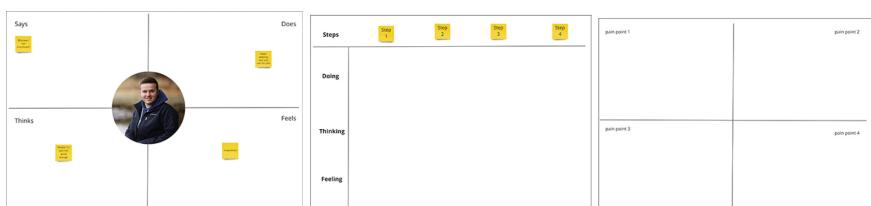


Fig. 3. Sample worksheets, empathy map, as-is-scenario and big ideas grid used in design thinking workshops.

A graphical representation of group 1 from workshop 3 having empathised with Andrew Wilson, using empathy mapping and as-is-scenario mapping followed by pain point voting and 4 pain points identified can be seen in Fig. 4.

3.1 Data Collection

Data was collected using surveys (via Microsoft forms to establish participant demographics), digital photography (of worksheets during and after workshops), audio recordings transcribed after each workshop (of participant playbacks and post workshop interviews). Ethical approval was sought prior to proceeding.

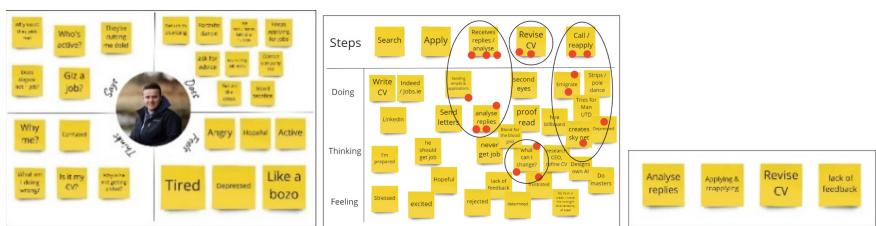


Fig. 4. Graphical representation of empathy map, as-is-scenario and pain points: workshop 3 group 1.

4 Results

Results analysed participant's outputs concentrating on pain points with corresponding big ideas, sub divided into visual big idea or non-visual big idea. Criteria for categorisation as a visual big idea included any drawn or illustrated elements including drawn big ideas which included some written elements for clarification. Examples of those big ideas categorised as visual and non-visual can be seen in Fig. 5.



Fig. 5. Example big ideas (01) Visual big idea, (02) Visual big idea, (03) Non-visual big idea.

Visual big ideas were not assessed in any form regarding quality of drawing or visualisation as our research is not concerned with the standard of drawing presented but

rather than participants engage in drawing as a means of articulating their understanding of complex AI systems. Our research found that workshop 1, with 4 computer science participants facilitated by 1 UX participant produced 12.5% visual big ideas the lowest percentage expressed across all 3 workshops. See Fig. 6.

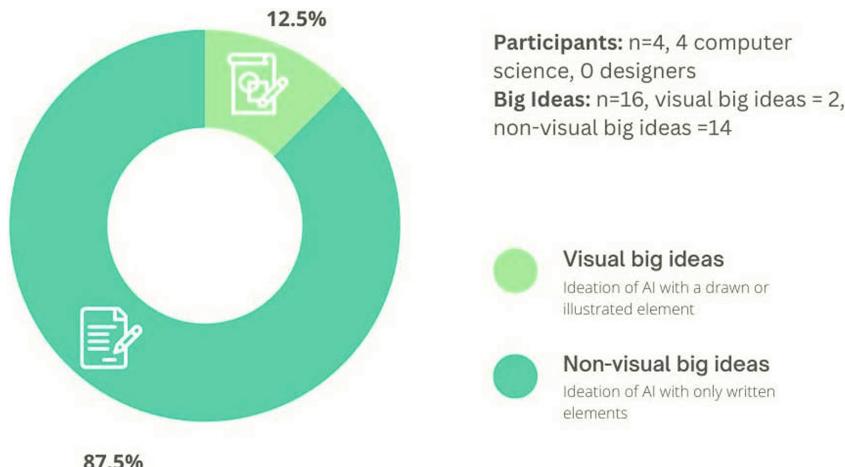


Fig. 6. Workshop 1: Computer science participants breakdown of visual big ideas vs non-visual big ideas.

Workshop 2, which comprised 14 participants with cross-disciplinary expertise (philosophy, psychology, computer science and AI systems development) facilitated by 2 UX participants produced 21% visual big ideas. See Fig. 7.

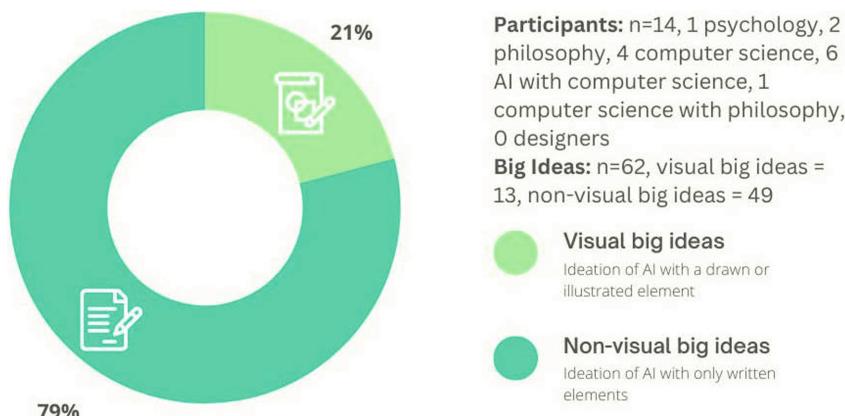


Fig. 7. Workshop 2: Cross-disciplinary participants breakdown of visual big ideas vs non-visual big ideas.

Workshop 3 which included 20 participants with design and computer science backgrounds facilitated by 2 UX participants produced 40.4% visual big ideas. See Fig. 8.

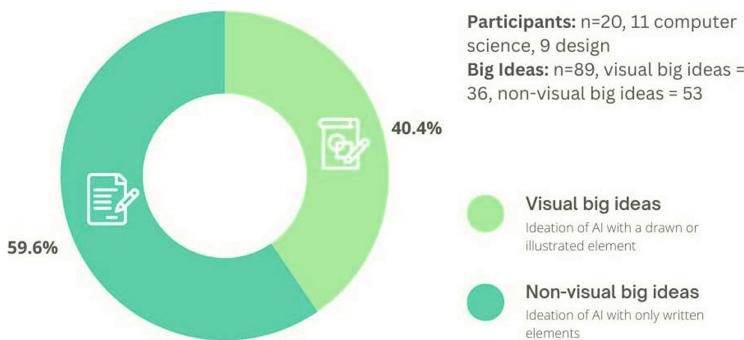


Fig. 8. Workshop 3: Computer science and design participants breakdown of visual big ideas vs non-visual big ideas.

These results, at first examination may not appear unforeseen. It might be anticipated that a design thinking workshop with designers as participants would produce a higher percentage of drawn or visual interpretations of big ideas. However, our observations during and after each workshop supported the theory that those not typically used to drawing to express ideas were encouraged to do so when designers were included alongside non-designers. One reason for this was expressed by a computer science participant P1 in workshop 3, *“Thinking of the imaginative side and then thinking technically worked well. The creative and technical sides were kind of meshed together. When one person is talking about one side and another person is talking about the other then you’re really talking about both. They kind of meld together”*. In this instance those who might have been accustomed to expressing their ideas in more technical terms were encouraged to think more creatively by engaging with those familiar with drawing as a means of expression. Therefor this may have aided big ideas ideation in terms considered more creative such as through drawing.

Those with a design background described similar thoughts post workshop 3. P2 described *“I think with the two, creative and technical mindsets, everyone had that over-the-top thinking, but having the creative and technical to exaggerate and then bring it back down to reality, it definitely melded brilliantly”* and P3 stated *“You put the wild on the top and the realistic at the bottom and work your way down and then you figure out, well actually this could be a good idea if we work together with the computer skills and the design aspects”*. Whilst not necessarily describing the process of drawing as a means of ideation participants did describe the merging of two discipline types, computer science and design as allowing for the amalgamation of the technical and creative. We posit that it is this amalgamation which may have encouraged those more accustomed to technical expressions of big ideas to embrace drawing and conversely for those participants accustomed to drawing as a means of expressing big ideas to consider how complex, technical ideas might be expressed visually.

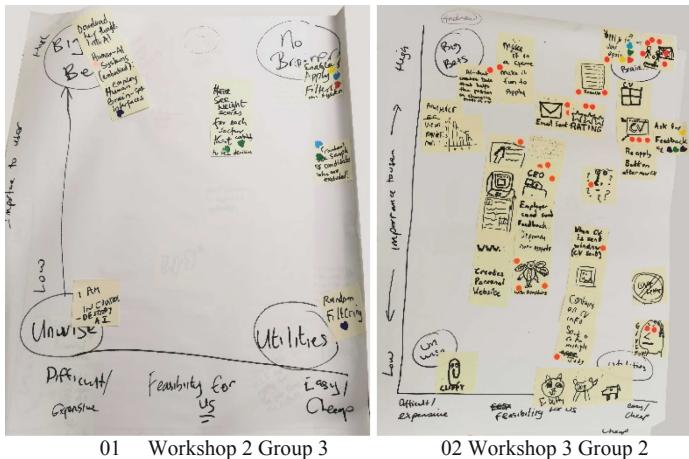


Fig. 9. Example big ideas works sheets with no visual big ideas 01 and the largest quantity of visual big ideas 02.

This may be due to social learning norms such as imitation and emulation [12] where humans have evolved to mimic others in groups as a means of creating social rapport, solidarity and the desire for others to like and trust us, all steps in humans' evolution survival mechanisms [13]. Two contrasting examples of big ideas work sheets, one with no visual big ideas compared to that with the largest quantity of big ideas expressed using visual means can be viewed in Fig. 9(01) and (02).

5 Conclusion

We conducted three design thinking workshops with computer science participants, cross-disciplinary participants and computer science and design participants, examining scenarios which described persona's engagement and frustrations with an AI driven recruitment system. Concentrating on the empathise, define and ideate modes of the design thinking framework gave all participants the methods and tools to collaborate successfully and converge on areas of opacity in the form of pain points and develop meaningful XAI solutions in the form of big ideas. This resulted in participants exploring mental models of systems which are usually confusing, non-transparent, abstract and difficult to articulate and visualise. Our study concluded that the inclusion of designers as participants in design thinking workshops alongside participants with expertise in other domains encouraged significantly more visual interpretations of big ideas than workshops with no designers present. Interestingly, those participants not usually used to conveying ideas using drawing were encouraged to draw and more likely to do so when designers were present. This supports the theory that as a stage in the ideation and exploration of mental models of complex systems, such as AI driven systems, including designers in conjunction with other users or stakeholders is recommended if visual interpretations are required from all participants. Limitations of this research centre around participants and domain explored. For future workshops we envisage including

a broader range of stakeholders such as those employed in human resources or those searching for employment. Also, the domain of recruitment and examination of those filtering CVs with the aid of AI would lend itself to presentation through typical user interfaces. How might participants visualise the inner workings of AI systems with limited or no user interface such as smart watches or smart heating systems? Future research will examine a greater range of big ideas which could be further developed in the prototype and test modes of the design thinking approach to develop more useful and meaningful XAI.

References

1. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. *Artif. Intell.* **267**, 1–3 (2019)
2. Miller, T., Howe, P., Sonenberg, L.: Explainable AI: beware of inmates running the asylum or: how I learnt to stop worrying and love the social and behavioural sciences (2017). [arXiv: 1712.00547](https://arxiv.org/abs/1712.00547)
3. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **1**(5), 206–215 (2019)
4. Dove, G., Halskov, K., Forlizzi, J., Zimmerman, J.: UX design innovation: challenges for working with machine learning as a design material. In: Proceedings of the 2017 Chi Conference on Human Factors in Computing Systems, pp. 278–288 (2017)
5. Jensen, M.B., Lozano, F., Steinert, M.: The origins of design thinking and the relevance in software innovations. In: Product-Focused Software Process Improvement: 17th International Conference, PROFES 2016, Trondheim, Norway, November 22–24, 2016, Proceedings, vol. 17, pp. 675–678. Springer International Publishing (2016)
6. Peukert, D., Lam, D.P., Horcea-Milcu, A.I., Lang, D.J.: Facilitating collaborative processes in transdisciplinary research using design prototyping. *J. Des. Res.* **18**(5–6), 294–326 (2020)
7. Yang, Q., Steinfeld, A., Rosé, C., Zimmerman, J.: Re-examining whether, why, and how human-AI interaction is uniquely difficult to design. In: Proceedings of the 2020 Chi Conference on Human Factors in Computing Systems, pp. 1–13 (2020)
8. Jonassen, D., Cho, Y.H.: Externalizing mental models with mindtools. In: Understanding Models for Learning and Instruction, pp. 145–159 (2008)
9. Al-Diban, S., Ifenthaler, D.: Comparison of two analysis approaches for measuring externalized mental models. *J. Educ. Technol. Soc.* **14**(2), 16–30 (2011)
10. Brooks, M.: Drawing, visualisation and young children’s exploration of “big ideas.” *Int. J. Sci. Educ.* **31**(3), 319–341 (2009)
11. Luchs, M.G., Swan, S., Griffin, A.: Design Thinking: New Product Development Essentials from the PDMA. Wiley (2015)
12. Whiten, A., McGuigan, N., Marshall-Pescini, S., Hopper, L.M.: Emulation, imitation, over-imitation and the scope of culture for child and chimpanzee. *Philos. Trans. R. Soc. B: Biol. Sci.* **364**(1528), 2417–2428 (2009)
13. Wood, C.: Why Imitation Is at the Heart of Being Human. https://greatergood.berkeley.edu/article/item/why_imitation_is_at_the_heart_of_being_human. Accessed 01 Mar 2023
14. Sperrle, F., El-Assady, M., Guo, G., Borgo, R., Chau, D.H., Endert, A., Keim, D.: A survey of human-centered evaluations in human-centered machine learning. *Comput. Graph. Forum* **40**(3), 543–568 (2021)
15. Oh, C., Song, J., Choi, J., Kim, S., Lee, S., Suh, B.: I lead, you help but only with enough details: understanding user experience of co-creation with artificial intelligence. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–13 (2018)

16. Ngo, T., Kunkel, J., Ziegler, J.: Exploring mental models for transparent and controllable recommender systems: a qualitative study. In: Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization, pp. 183–191 (2020)



Using AutoML to Analyze the Effect of Attendance and Seat Location on University Student Grades

Ac Hýbl and Germán H. Alférez^(✉)

School of Computing, Southern Adventist University, PO Box 370, Collegedale, TN 37315-0370, USA
matoush@southern.edu, harveya@southern.edu

Abstract. A common claim is that class attendance and sitting at the front of a classroom may improve student grades. This study employs Automated Machine Learning (AutoML) to analyze this claim. The data used in this study came from an attendance-tracking system from a private university in Tennessee, USA. The correlation analysis in Microsoft Azure's Machine Learning workspace was performed by training regression models. No correlation was found between student attendance and seat choice and final course grades. The K-means clustering algorithm was used to train clustering models in Microsoft Azure. At $k = 2$ clusters, a cluster with perfect attendance shows a higher average grade than a cluster with a late attendance average. Seat choice within the classroom does not prove important to the clustering models.

Keywords: Microsoft Azure · Automated Machine Learning · Regression · K-Means Clustering · Class Attendance · Seat Selection

1 Introduction

Although recent studies explore the effects of student seating location on social interaction or course engagement [7, 15], there is not yet consensus on the claim that seating location directly affects academic performance [2, 8, 9, 14]. Moreover, although there are several research projects that have tried to analyze the relation between seating location and student academic performance, they are focused on small datasets [3, 8, 9, 13].

Our contribution is to apply two machine learning (ML) techniques, namely regression and clustering, to the attendance and seating data from Southern Adventist University (SAU), a private institution located in Tennessee, USA, to analyze the common claim that class attendance and sitting at the front of a classroom may improve student grades. The goal is not only to test whether these algorithms would arrive at conclusions that would support the aforementioned claims, but also to introduce machine learning as a method of analysis for student attendance and seating location. In our study, 221,600 attendance records from $n = 2,067$ students were analyzed from data collected in 2021 and 2022.

To perform machine learning experiments, we made use of the Automated Machine Learning (AutoML) functionality of Microsoft Azure. AutoML was used because of its automated training and evaluation of machine learning models without the need for extensive coding.

This research work answers the following questions:

1. Do class attendance and punctuality foreshadow higher course grades?
2. Do students that sit in the front of class receive higher marks than those that choose to sit near the back?

This paper is organized as follows. Section 2 presents the state of the art. Section 3 presents the methodology. Section 4 presents the results. Section 5 presents the conclusions and future work.

2 State-of-the-Art

Regular attendance is widely believed to have numerous benefits for students, including improved learning opportunities and better chances for success. The National Center for Education Statistics, for instance, explains that students who are frequently absent during their schooling years, starting from kindergarten to high school, miss out on crucial learning opportunities, thus hampering their chances for success [12]. Not only does this impede a teacher’s ability to facilitate students’ success, but research has also shown that students who exhibit frequent absenteeism tend to display a “history of negative behaviors” even after they leave school.

Numerous studies have explored the relationship between student attendance and academic performance. In their 2015 study of first-year psychology courses, Alexander and Hicks analyzed whether class attendance was linked to increased student performance in modern classrooms with online lectures [1]. Their results featured significant ($p < 0.001$ and $p < 0.05$) correlations between student attendance and performance on assignments.

Furthermore, several studies have been done on seat location and student grades. For instance, in a 1973 issue of *Sociometry*, Becker et al. demonstrate that students sitting nearer to their instructor not only received higher grades than those further away, but also liked their professor more ($p < 0.01, n = 282$) [3].

Consistent findings have emerged in a 2017 study by Shernoff et al. [14]. This study considered whether students ($n = 407$) with particular personality traits chose their seating location preferentially, whether seating location affected subjective experiences such as engagement and attention, and whether these factors affected student performance. To differentiate “causal mechanism[s] from self-selection,” the Experience Sampling Method (ESM) method was employed. ESM consists of analyzing a participant’s own experience of their surroundings over the course of the study to eliminate the effects of self-selection. Results showed that students in the front rows were more engaged in class and received higher grades than those in the back rows.

Another recent study by Lyu, Jiang, and Wu examined, among several other research questions, the effects of seating on academic achievement in a college setting using $n = 306$ students [9]. This study employed seating networks and clusters and identified the top and bottom performers in each class. Results showed that students with high academic achievement preferred to sit in the front of the classroom while the opposite was also true. One very interesting finding was that while the lowest three performers in each class moved around the classroom over the course of a semester, the top three performers generally remained in the same seats located in the front and middle of a classroom's center columns. In fact, the top of the class *never* moved.

Similarly, Bergtold, Yeager, and Griffin investigated the role of seating location in student performance using a variety of variables [4]. These included gender, grades, math level, GPA, class rank, major, spacial locale, endogenous peer effects, and the effects a student's classmates. The relevant results of this study again supported sitting in the back of the classroom decreased a student's grades. Various statistical analyses have similarly concluded that seating location can impact student academic performance [11].

However, not all studies agree with this body of research. While Chan et al. have generally found that seat location can affect student performance, they differentiate the strength of this correlation by discipline [5]. They found that while students in soft fields such as psychology and sociology tend to perform better when seated in the front of the classroom, seating location has little effect on students in hard fields such as engineering and math.

Two 2007 studies found similar results that student seating did not affect performance [2,8]. In the first study, Kalinowski et al. conducted a randomized blind research in a sophomore biology classroom ($n = 43$) using exam scores to assess whether placing students in the front, center, or back of the classroom would affect their grades [8]. They found no evidence that grades or student attitudes were affected by seat location. Moreover, when given the choice, students with high GPAs were found to choose seats in the front of the classroom, suggesting that self-selection may play a role in some of these studies. In the second study, Armstrong et al. included $n = 5814$ students [2]. Some classrooms used randomly-assigned seating, while others allowed students to choose their own seats and self-report them. Out of 20 classrooms in the study, only seven showed a significant correlation between seat location and student performance. One classroom found that students in the back of the classroom performed better than those in the front. The remaining six allowed for student-chosen seating, thus allowing for self-selection effects. Moreover, the correlations for these classrooms were negligible ($r < 0.16$).

More recently (2016), Pichierri and Guido Pichierri and Guido analyzed the effect of seat location on student academic performance and how shyness may be a moderating factor [13]. Data was collected over five years for a total of $n = 232$ regularly-attending students. In general students in the front of the classroom performed better than those in the back. However, this effect decreased as shyness increased.

Despite the above research in this area, we were unable to find studies that incorporated student seat location and machine learning. The most related idea found incorporated simulating student-teacher proximity using an agent-based modeling approach [6]. Moreover, the studies presented in this section tend to be performed on a small scale. Using, few classes, few students, or only one semester of data. Our research uses a larger dataset.

3 Methodology

The first step to performing successful experiments in data science is obtaining good data. Therefore, this section first describes where data was obtained, how it was organized, and what precautions were practiced to avoid data-related problems. Next the tools used are introduced along with the procedures and experiments performed.

3.1 Data Collection

The data used in the experiments came from an attendance system that records attendance and seating at SAU. Like many other institutions, SAU implemented strict quarantines and digital contact tracing (DCT) in 2020 to enable students to attend classes in person during the COVID-19 pandemic [10]. Thousands of students at SAU were required to use quick response (QR) codes to record their class attendance. The web interface also required students to select their seat within their classroom allowing for digital contact tracing (DCT) whenever a new case of COVID-19 was identified.

As campus activity returned to normal and concern over COVID-19 subsided over the next few semesters, the new attendance system remained. Many professors simply found the system much more convenient than manually taking note of absent or late students. Not only did the new system allow professors to discuss course material sooner, but it continued collecting valuable attendance and seating data which proved to be useful for more than just contact tracing.

Because the driving questions of this research concern a typical classroom setting, the data collected needed to reflect only this setting. Some courses at SAU only had a small number of students per semester. Thus, only large classrooms were considered. Additionally, because stickers on the desks in each classroom indicated the seat row and column (as shown in Fig. 1), only classrooms with bolted desks were selected. This ensured that rearranged classrooms did not introduce flawed data into the study.

Some faculty raised further concerns that the attendance system had changed as the university relaxed its COVID-19 restrictions. For instance, during its first semester of use, the attendance tracking system (ATS) only permitted students to sit in every other seat, thus restricting student seat choice. However, in following semesters, ATS allowed students to sit in any seat in a classroom. In favor of consistency, only the latter of these systems was used, resulting in two semesters of data (Fall 2021 and Winter 2022).

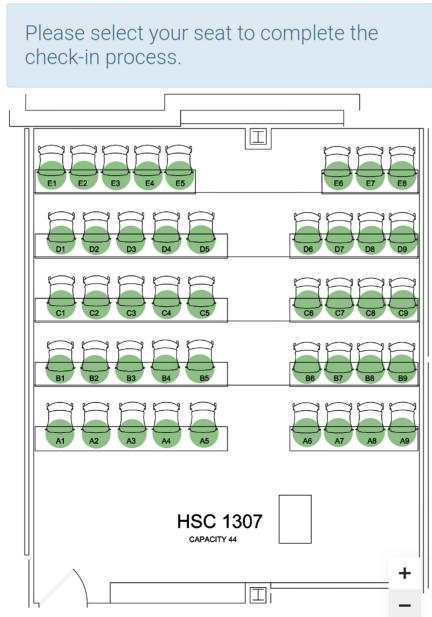


Fig. 1. The ATS Interface in One of the Many Classrooms on Campus

After selecting locations and times for data collection, all other entities were determined as shown in Fig. 2. 159 course sections with twenty or more students enrolled were found using the seventeen chosen classrooms over the two semesters. 2,067 students were enrolled in one or more of these sections, which were taught by sixty-three professors representing thirteen departments. The final attendance dataset not only represented most of the diversity at the university, but also consisted of nearly a quarter million (221,600) rows. Therefore, this data-collection method improves upon previous studies with fewer data points [3,9,14].

Each section had associated events that represented one class period. The ATS stored data for each student that was present, but it did not always specify if a student was absent from a class. Thus, using several structured query language (SQL) scripts, this data was imputed. If a student was enrolled in a section but had no record of attending any one of its events, they were assumed absent.

3.2 Data Organization and Tools

Dividing classes, students, and attendance into separate relations was the most natural way of storing and organizing the data. Because the data was pulled from a data warehouse partially external to the university, it was not separated

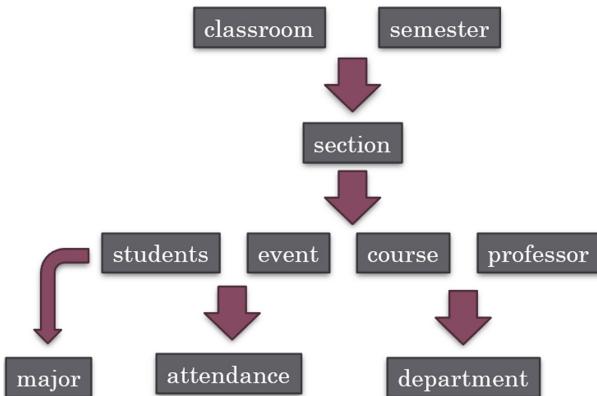


Fig. 2. The Natural Organization of Entities in Data Collection. Choosing Classrooms and Semesters Resulted in Specific Sections. These Sections had Specific Professors, Students, and Events. Attendance is a Many-to-Many Relationship between Students and Events

this way. To structure the data, several segregation and cleaning steps were performed in Tableau Prep Builder.¹

Because Microsoft Azure was to be used for the data science experiments, it was also chosen to store and serve the data as well. Using an Azure SQL Database on an Azure SQL Server, we constructed a relational database from the outputs of Tableau Prep Builder. After resolving all the bugs encountered during data collection, the cleaned comma-separated values (CSV) files provided by Tableau Prep Builder were simply imported as tables into the database using Microsoft SQL Server Management Studio (SSMS). Primary and foreign keys were also configured in SSMS. The resulting schema for this database can be seen in Fig. 3. All entities are transitively related using primary/foreign keys.

The initial reasons for using Microsoft Azure were its advertised ease of use and AutoML workspace. AutoML is an emerging technology that offers automatic training of various machine learning models without the need for coding. In a typical solution, data is first supplied to a model training component. The component trains a prediction model, and may even automatically select the most appropriate algorithm for solving a given classification, regression, clustering, or forecasting problem. It may also tune the model's hyperparameters.

All experiments were run on a Microsoft Azure's cloud computing platform. The specific machine configuration selected, Standard D2 v2, had the following specifications: Cores: 2; RAM: 7 GB; disk: 100 GB; temporal storage (SSD): 100 GiB; NICs: 2; network bandwidth: 1500 Mbps; throughput IOPS: 8 × 500; maximum data disks: 8; and maximum temporal storage throughput: IOPS/Read MBps/Write MBps: 6000/93/46.

¹ <https://www.tableau.com/products/prep>.

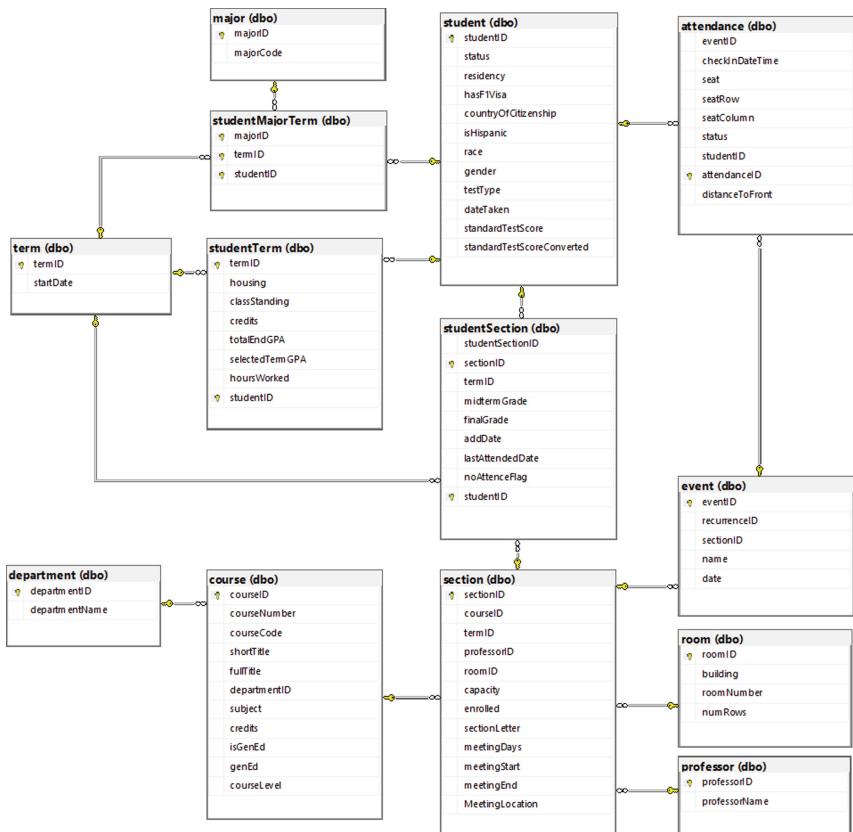


Fig. 3. The Database Schema as Visualized by Microsoft SQL Server Management Studio

3.3 Experiment Plan

The general plan for regression and clustering was to connect the Microsoft Azure AutoML workspace to the Microsoft Azure SQL Database so that the AutoML components could automatically extract the latest version of the data from the database. Using these two platforms, we created automatic pipelines that built machine learning models for grade forecasting. Finally, various subsets of attributes were provided to the AutoML pipeline to run experiments on.

Once a model was created, Microsoft Azure supplied various metrics associated with that model. For regression, these included error scores and correlation coefficients, allowing for a simple correlation analysis of any subset of attributes. For clustering, metrics included cluster densities and diameters as well as each record's cluster assignment. Using this information, the model could provide the average value of each attribute for each cluster, which could give insight into how the algorithm naturally organized the data.

To perform logistic regression and clustering, class data first needed to be transformed into a numerical format. The twelve grade categories “A-F” were converted to the numbers 1–12, respectively. Also, “I” (incomplete) and “IP” (incomplete passing) were assigned values of 13 and 14.² Other categorical variables were converted in a similar manner. For example, there were five categories for attendance status. The labels “Present,” “Online,” “Late,” “Excused,” and “Absent” were assigned the values 0–4 respectively.

Furthermore, as shown in the ATS interface in Fig. 1, students selected their seat using a numerical column and a *row letter*. Most training models would perform better with a *row number* rather than a letter. Moreover, the number of rows and spacing between those rows in each classroom varied, rendering any categorical row data inconsistent. To provide the most useful data to the algorithms that would train the models, the row letters were extracted, aggregated for each classroom, and converted to a normalized distance from the front of the classroom. This new attribute, called *distanceToFront*, measured how far a student’s chosen seat was from the front of the classroom. Values closer to “0” indicate seats closer to the front row of a classroom while those closer to “1” represent seats at the back of a classroom.

The final query fetches the attributes of interest for this project (shown in Listing 1.1).³ These attributes included student demographic information, credit load, hours worked during the semester at the university, distance to the front of the classroom for every attendance record, attendance status, and final course grade. This query was run against the Azure SQL database and the resulting data was used as a starting point for Azure’s AutoML experiments.

```
select s.isFemale, s.isHispanic, s.race, st.housing,
       st.gradeLevel, st.credits, st.hoursWorked,
       a.distanceToFront, a.col, a.statusCode, sn.finalGradeCode
  from attendance a join student s on a.studentID = s.studentID
  join studentTerm st on s.studentID = st.studentID
  join studentSection sn on s.studentID = sn.studentID
    and sn.termID = st.termID
  join section n on sn.sectionID = n.sectionID
  join event e on n.sectionID = e.sectionID
    and e.eventID = a.eventID
  join course c on n.courseID = c.courseID
 where c.department != 'NRSG' and c.department != 'PEAC';
```

Listing 1.1. Query to Fetch the Attributes of Interest

Correlation Analysis Regression experiments were configured as “jobs” and started in Microsoft Azure’s Machine Learning workspace. To analyze correlation

² This assumes that not completing a class is a less favorable outcome than failing it. Also “Incomplete Passing” is marked as lower than “Incomplete,” but it is not a cause for concern as this represents less than 0.1% of the data.

³ Notice that courses in the Nursing (NRSG) and Physical Education (PEAC) departments were filtered out. Nursing courses were removed because students were often assigned seats in these courses, thus removing the student’s ability to choose their seat. Other classrooms allowed students to choose their own seat, and we preferred not to mix these two techniques in this study. Physical education courses, on the other hand, were not considered a “normal classroom setting.”

across different groups of attributes, the experiment was run multiple times with different subsets of the columns shown in the query shown in Listing 1.1. The platform automatically selected the best-performing algorithms based on grade-prediction power.

Clustering Rather than telling the algorithm how the data should be fit, unsupervised learning groups data based on all attributes without prior direction. Clusters formed by minimizing a cost function can be assessed for conclusions.

Microsoft Azure's AutoML platform can perform popular clustering algorithms such as K-means. The best way to use clustering in Microsoft Azure is to create a pipeline. In a pipeline, several components can be “wired together” to create a single process. This process is capable of gathering data, processing it, training machine learning models, testing those models, and generating result data in automatic succession. The machine learning pipeline shown in Fig. 4 was used to perform clustering for this project. Note that to make this image fit better on paper, a data cleaning step and a column selection step were removed from the pipeline. The cleaning step simply imputed missing values using column means.

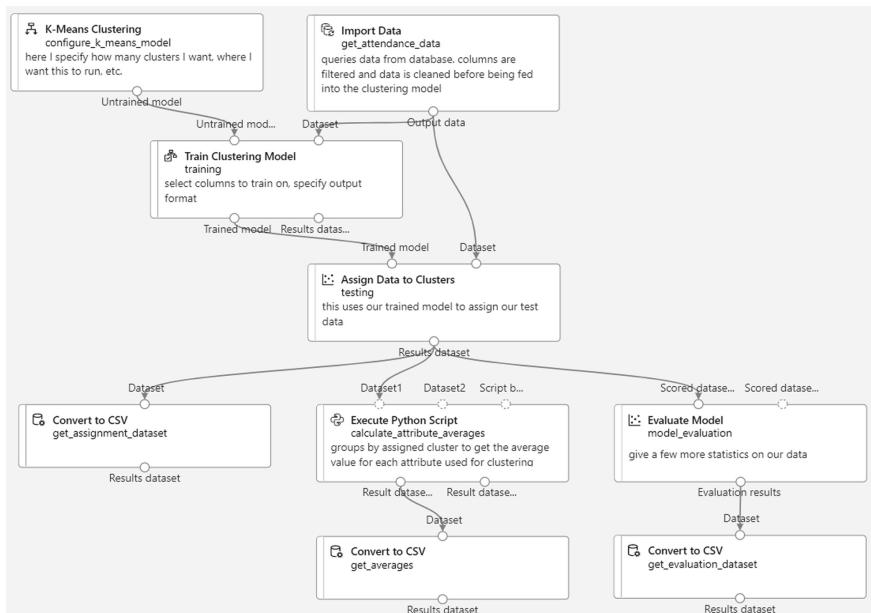


Fig. 4. The Clustering Pipeline Created in this Project

The **Import Data** component fetches dynamic data from the Azure SQL Database. Different attributes can be selected within or after the query that this component performs. Initially, numerical, textual, and categorical data was

being returned. However, after several experiments and database adjustments, only numerical data was fetched using the query shown in Listing 1.1.

On the top left of Fig. 4, the **K-Means Clustering** component is used to initialize and configure an untrained clustering model. Several parameters can be set here: the number of clusters desired in the output, the feature normalization option, a model weight initialization algorithm, and a multi-dimensional distance metric.⁴ The **Train Clustering Model** component then trains this model using the imported data.

The **Assign Data to Clusters** component takes all the provided data and assigns it to a cluster using the trained model. Using this data, the model can be evaluated in the **Evaluate Model** component, which measures average distances between all clusters.

Microsoft Azure provides a mechanism for executing Python code in a pipeline. A custom Python script only needs to contain an `azureml_main()` function that receives and returns up to two dataframes. To generate the necessary data, the **Execute Python Script** component receives all the records along with their cluster assignments and calculates the average value of every feature for each cluster (see Listing 1.2).

```
import pandas as pd
def azureml_main(dataset, optional_data = None):
    return dataset.groupby("Assignments").mean()
```

Listing 1.2. The Data was Grouped by Cluster and then Aggregated into a Mean

Finally, all the data generated by these components is converted into CSV format as described in the following section.

4 Results

This section presents the results of the correlation and clustering analyses.

4.1 Correlation Analysis

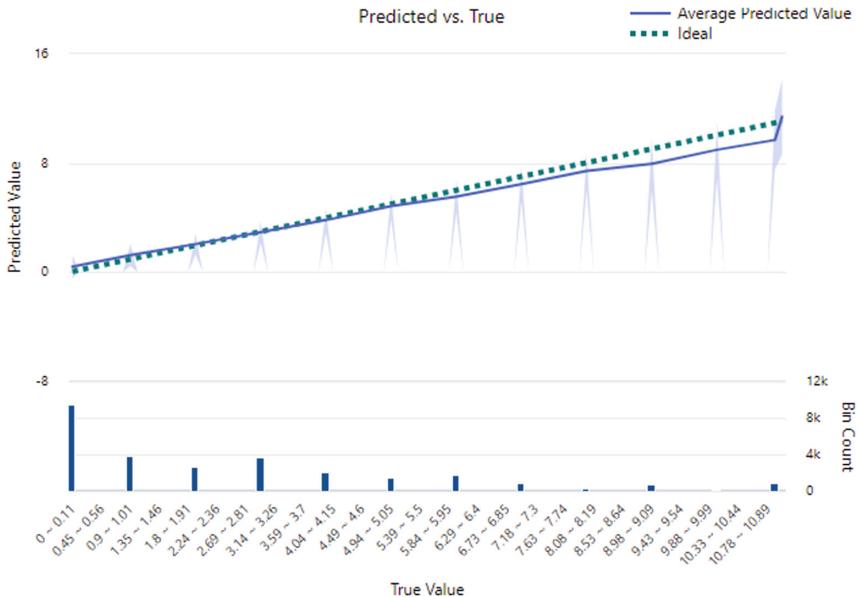
In the first regression experiment, all of the attributes of interest mentioned in the previous section were fed into the regression model.⁵ The Root Mean Squared Error (RMSE), Explained Variance (EV), Spearman Correlation Coefficient, and R^2 Score are presented in Table 1. This configuration provided a fairly accurate regression model. The Spearman correlation coefficient was over 0.9, suggesting a high correlation between all the input and target attributes. As a visual representation of its accuracy, the regression graph is shown in Fig. 5.

⁴ In our experiments, the number of clusters varied from two to fourteen, normalization was enabled, weights were initialized with the “K-Means++” algorithm, and a Euclidean distance metric was used.

⁵ This table summarizes the findings of four regression experiments. The first included demographics, course and work amounts, attendance information, and *finalGradeCode*, which is the target variable. The next three experiments only included distance to the front of the classroom or attendance status, or both.

Table 1. Regression Performance Metrics

Attributes Used	RMSE	EV	Spearman	R^2 Score
All shown in SQL query	1.110	0.859	0.906	0.859
statusCode, distanceToFront	2.314	0.043	0.186	0.043
statusCode	2.923	0.020	0.107	0.020
distanceToFront	2.922	0.020	0.140	0.020

**Fig. 5.** The Predicted Values Approximate the True Values very Well

Microsoft Azure also provides insight into which attributes were most important to the accuracy of the model. As shown in Fig. 6, the model relied heavily on class standing, term information, and demographic information. The distance to the front of the classroom falls in fifth place and attendance status in seventh.

As evidenced, the predictive capacity of a student's ultimate grade can be derived with a noteworthy level of precision by examining their grade level, workload, and demographics. Given the model's underlying objective of optimizing accuracy, repeated training iterations would fail to encourage greater weighting of the (*distanceToFront*) and (*statusCode*) features. Consequently, it was necessary to limit the model's scope to only these two attributes. The following experiment used only attendance status and distance to the front of the classroom. Microsoft Azure took forty-eight minutes to converge on an optimal model. This model offered a mean absolute error that was nearly a fifth of the entire range. Very little of this error was explained by the variance in attendance

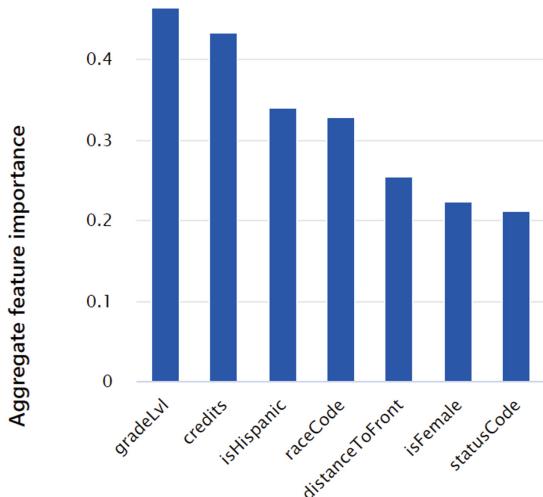


Fig. 6. The Top Seven Attributes Used by the First Regression Model, Ranked in Order of Predictive Importance

data. Additionally, both correlation metrics suggested no correlation between the attendance data and student grades. In fact, the R^2 score suggests that there is a 95.7% chance of getting this correlation from unrelated attributes.

Isolating the attendance status and seat row attributes even further did not improve the results. Both prediction models produced separately by these two attributes yielded even lower correlation values.

Overall, the results of the correlation analysis showed that attendance and seat choice could not be used to accurately or precisely predict student grades in the data obtained from SAU.

4.2 Clustering

Before other hyperparameters were tuned, the number of clusters, k , had to be decided. To this end, we made use of the Elbow Method. As shown in Fig. 7, the inflection point is slightly unclear. Thus, experiments were first performed with $k = 5$ clusters. The first experiment used all the queried columns. However, the clustering algorithm grouped data primarily using demographic information. For example, four of the five clusters would be entirely based on a student's gender and Hispanic status.

After limiting the number of attributes available, K-Means grouped the data primarily based on attendance status and distance to the front of the classroom. However, five clusters proved too many for this set of attributes so the count was reduced to $k = 3$.



Fig. 7. The Average Distance to a Cluster Compared Across Various Number of Clusters

The averages in Table 2 were obtained using three clusters and two input attributes.⁶ Data placed in Cluster 0 represents a mostly “Present” attendance status and a seat location roughly halfway from the front of the classroom. Nearly all of the attendance records in this cluster have perfect scores of “A”. Cluster 1 has similar averages, but with grades much closer to “C+” and “C”. The cluster with chronically late attendance and a seating preference slightly beyond the halfway point has an average grade around a “B+” or “B”. Although Cluster 2 does show a group of students that is often late and performs worse than average, the three clusters together are inconclusive.

Table 2. Average Attribute Values in Three Clusters

	Status	distanceToFront	Grade	Radius	Point count
Cluster 0	0.020	0.575	1.004	0.404	104,833
Cluster 1	0.030	0.613	6.384	0.569	42,212
Cluster 2	2.922	0.610	3.618	1.000	18,143
Total	0.341	0.589	2.666	0.512	165,188

Finally, the experiments were repeated with only two clusters. The results are displayed in Table 3. As seen before, distance to the front of the classroom is nearly the same for both clusters. However, Cluster 1 specifically represents records where a student was, on average, late to class. This cluster has an average

⁶ The radius feature was originally named “Average Distance to Cluster Center” and has been scaled so that the largest value is 1.

grade between “B+” and “B”. Cluster 0, on the other hand, displays much better attendance and grades on average between “A-” and “B+”.

Table 3. Average Attribute Values in Two Clusters

	Status	distanceToFront	Grade	Radius	Point count
Cluster 0	0.022	0.586	2.548	0.689	147,033
Cluster 1	2.921	0.610	3.624	1.000	18,155
Total	0.341	0.589	2.666	0.651	165,188

Though interesting, these clusters are still far from ideal. One major issue is that they are unbalanced. One includes 89% of the attendance data and the other only 11%. A more balanced dataset is desirable.

To further validate Microsoft Azure’s automated clustering algorithm, clustering was also performed in Weka 3.⁷ The results from this experiment were identical to those obtained from Microsoft Azure’s K-Means clustering.

4.3 Discussion

The correlation analysis did not support any correlation between a student’s attendance and their performance in class ($R^2 = 0.020$). Clustering provided more insight. A cluster of data was identified that represented tardiness or absence along with lower grades. No experiments showed that sitting nearer to the front of the classroom positively impacted a student’s grades. Instead, this attribute seemed generally unimportant.

5 Conclusion and Future Work

In this study, after applying regression with AutoML, no correlation was found between student attendance and seat choice and final course grades. Also, the K-means analysis shows two clusters, one of them with perfect attendance and a higher average grade than the other with a late attendance average. This reveals that unsupervised learning through clustering does support the claim that attendance has a positive effect on student performance based on the data used in this study. However, seat choice within the classroom did not prove important even in the unsupervised learning analysis.

As future work, we will explore other technologies for machine learning because Microsoft Azure’s AutoML proved difficult to navigate and configure. Also, we plan the run more experiments with balanced data to include the similar amounts of high and low scores. Specifically, more data with final grades below “B” should be collected and combined with the current dataset. Also, a

⁷ cs.waikato.ac.nz/ml/weka.

better balance is needed between the different attendance categories. Currently, “Late,” “Absent,” and “Excused” attendance records represent only 11.6% of attendance data.

Also, we plan to evaluate additional variables in the study. For instance, the relative distance of students to the front of smaller and larger classrooms should be considered. Also, data about the horizontal placement of a student within a classroom will be included as well as teaching styles, tardiness thresholds, and class time.

References

1. Alexander, V., Hicks, R.E.: Does class attendance predict academic performance in first year psychology tutorials? *Int. J. Psychol. Stud.* **8**(1) (2016)
2. Armstrong, N., Chang, S.M.: Location, location, location. *J. College Sci. Teach.* **37**(2) (2007)
3. Becker, F.D., Sommer, R., Bee, J., Oxley, B.: College classroom ecology. *Sociometry* 514–525 (1973)
4. Bergtold, J.S., Yeager, E.A., Griffin, T.W.: Spatial dynamics in the classroom: does seating choice matter? *PLOS ONE* **14**(12), 1–16, 12 (2020)
5. Chan, K.L., Chin, D.C.W., Wong, M.S., Kam, R., Chan, B.S.B., Liu, C.-H., Wong, F.K.K., Suen, L.K.P., Yang, L., Lam, S.C., et al.: Academic discipline as a moderating variable between seating location and academic performance: implications for teaching. *High. Educ. Res. Dev.* **41**(5), 1436–1450 (2022)
6. Dong, Z., Liu, H., Zheng, X.: The influence of teacher-student proximity, teacher feedback, and near-seated peer groups on classroom engagement: an agent-based modeling approach. *PLoS ONE* **16**(1), e0244935 (2021)
7. Gao, N., Rahaman, M.S., Shao, W., Ji, K., Salim, F.D.: Individual and group-wise classroom seating experience: effects on student engagement in different courses. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **6**(3) (2022)
8. Kalinowski, S., Toper, M.L.: The effect of seat location on exam grades and student perceptions in an introductory biology class. *J. College Sci. Teach.* **36**(4) (2007)
9. Lyu, Q., Jiang, Y., Wu, J.: Relations between university students’ academic achievement and their seating positions in classrooms. In: 2021 7th International Conference on Education and Training Technologies, pp. 36–43 (2021)
10. Malloy, M.L., Hartung, L., Wangen, S., Banerjee, S.: Network-side digital contact tracing on a large university campus. In: Proceedings of the 28th Annual International Conference on Mobile Computing And Networking, MobiCom’22, pp. 367–380. Association for Computing Machinery, New York (2022)
11. Minchen, B.J.: The effects of classroom seating on students’ performance in a high school science setting. Ph.D. thesis (2007)
12. NCES. Every school day counts: the forum guide to collecting and using attendance data (2009)
13. Pichierri, M., Guido, G.: When the row predicts the grade: Differences in marketing students’ performance as a function of seating location. *Learn. Individ. Differ.* **49**, 437–441 (2016)

14. Shernoff, D.J., Sannella, A.J., Schorr, R.Y., Sanchez-Wall, L., Ruzek, E.A., Sinha, S., Bressler, D.M.: Separate worlds: the influence of seating location on student engagement, classroom experience, and performance in the large university lecture hall. *J. Environ. Psychol.* **49**, 55–64 (2017)
15. Wang, J., Xie, K., Liu, Q., Long, T., Lu, G.: Examining the effect of seat location on students' real-time social interactions in a smart classroom using experience sampling method. *J. Comput. Educ.* 1–19 (2022)



The Role of the User in Meaningful Production with AI

Iro Laskari^(✉)

Assistant Professor, School of Applied Arts and Sustainable Design, Hellenic Open University,
Patras, Greece

ilaskar@gmail.com, laskari@eap.gr

Abstract. This poster documents the basic points of a research project that investigates the role of the user of AI systems in the production of meaning and content. Whether the user is a student, a researcher or an artist, the output of the human-machine cooperation should be original. As a generative assembly of textual or visual data, the outgoing content looks like the rephrasing of things that have already been said. But how AI generated texts and images surprise us? How is their production been triggered? Beginning from the assumption that the user is a compositor of thoughts and connotations, which, in cooperation with the AI system could lead to the formation of an out of the box thinking, we notice that her practice has similarities to the practices of the gardener, of the interviewer and of the interrogator. But what do these similarities concern? What do a gardener, an interviewer and an interrogator have in common with the user of AI? We'll attempt to answer these questions with the help AI itself, placing ourselves in the position of the user in question, thus creating a simulation. This is what we like to call a tautological method, since the compositing procedure replicates itself, permitting a first person observation. Whether this approach will be fruitful or not, is something that we are going to discover at the end of this poster.

Keywords: AI user · Generative narratives · Meaning production

1 Purpose

Content creation with the use of Artificial Intelligence (AI) has been a very hot topic lately. Chat GPT for text writing and DALL-E or Midjourney for the creation of original images, are some of the tools that gain more and more interest both to ordinary users as well as to researchers. They seem to be more than tools as they're able to compose complete tasks with only a few word prompts. But where exactly does the human user stand at this procedure? What is her role as a content compositor?

2 Background/Significance

As an artist and a researcher I've always been interested in the language of non-linear media, their grammar and syntax for the creation of lingual and audiovisual narratives. AI seems to impose its own writing methods represented through text and images. The

various potential forms of an AI narrative are infinite. In this new landscape, unanticipated correlations amongst the protagonists, their surroundings, the story plot and time, are being established. But where does the human user stand at this procedure?

This poster presents the basic parts of a broader research on the role or the user in meaning production and content production with the use of AI.

3 Method

In order to investigate the role of the user in the composition of meaning with the use of AI, we'll put ourselves in the user's shoes, through simulation. Furthermore, using the notion of tautology [1], which defines the concepts through the same concepts ("Drama is drama"), we'll ask ourselves to compose this specific poster with the help of AI.

According to some directions given by the University of Houston on the Basic Steps of a Simulation Study [5], the application of simulation involves specific steps in order to be successful. Regardless of the type of problem and the objective of the study, the process by which the simulation is performed remains constant. The following briefly describes the basic steps in the simulation process:

1. *Problem Definition*: We examine the role of the AI user in the production of meaning.
2. *Project Planning*: Assuming that her role has common elements with the roles of the gardener, the interviewer and the interrogator, we break the subject in parts, in order to examine it both partly and in a combination. The length of the simulation is the same as the poster's.
3. *System Definition*: The system under research is the same as the one used for the research, since we don't examine the internal mechanisms of AI, but the "correct-effective-creative" act of the user. In order to achieve that, we create a simulation, which limited to the length of the poster and to the duration of its submission deadline, which tries to answer to a specific question (What do a gardener, an interviewer and an interrogator have in common with the user of AI?). The notion of tautology is being applied, since we'll try to answer the question on the creative use of tools via the use of the same tools, and by placing the writer of the poster at the position under question.
4. *Model Formulation*: We've created a flowchart in the AI app "Mindmeister", in order to visualize the main idea of the research.
5. *Verification & Validation*: Since we have a 1:1 analogy between the system under question and the simulation, no significant difference exists between the model and the real system, which means that the model reflects reality. Validation is planned to be achieved with the help of AI, through the process of elimination.

4 Synergies with AI

Brian Massumi, the translator of Deleuze's and Guattari's "A thousand plateaus, capitalism and schizophrenia", at his forward writes [3]: "Deleuze and Guattari argue that there is no distinction between the individual and the collective. Traditionally the individual has always been associated with desire and the collective with the law. Deleuze and

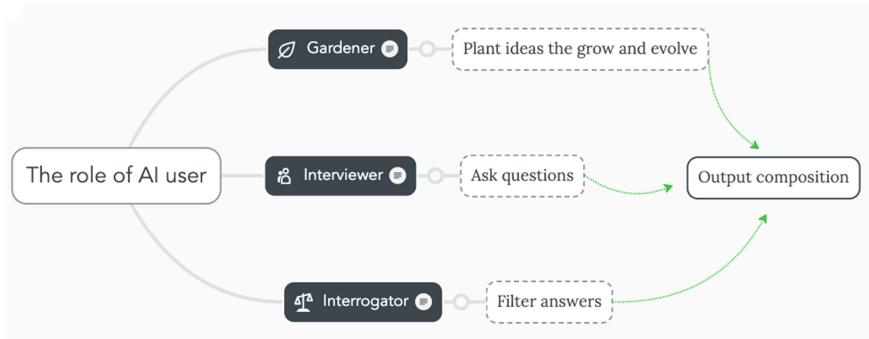


Fig. 1. Mindmap created in mindmeister.

Guattari instead propose a “social desire.” This suggests that desire is always in movement, always constituted by different elements depending upon the situation. This, they suggest, is machine-like rather than a drama (of Oedipal representation). Desire is not lack, which suggests negativity. It is affirmative in its state of movement and change”.

In our case, when a user collaborates with AI in order to create content, the collective comprises human and computer partners or contributors, who work complementary. The human attempts to create a prompt in a language that the AI partner understands well, such as English and AI transforms this demand into larger text, image, sound and so on. With the help of neural networks and complicate computation, AI seems to be able to make connotations and logical thoughts, as it creates compositions that are both aesthetically and conceptually interesting. But how logical are these outputs? How close are they to reality? Are they misleading? At this point the human partner of this collectivity has to take action again, in order to filter the outputs, and this is how this collaboration is affirmative in its state of movement and change, as Deleuze and Guattari argue.

4.1 Common Points Between the AI User and a Gardener

Lev Manovich [4] suggests that AI media can be created by automatically “translating” content between media types. He puts the word “translation” in quotes, because this is not a literal one-to-one translation: “input from one medium instructs a neural network to predict the appropriate output from another. Such input can also be said to be “mapped” to some outputs in other media. Text is mapped into new styles of text, images, animation, video, 3D models, and music. The video is converted into 3D models or animation. Images are “translated” into text, and so on. Text-to-image method translation is currently more advanced than others, but they will catch up eventually”.

We’ve asked Open GPT to point out the common points between AI and a garden, and it answered as follow:

AI and a garden may seem like very different things, but there are some common elements between the two:

1. Both require nurturing: Just as a garden requires care and attention to grow, AI systems require ongoing monitoring and maintenance to ensure they continue to perform well.

2. Both involve complex systems: A garden is a complex ecosystem, with many different elements interacting in intricate ways. Similarly, AI systems involve complex algorithms, data structures, and neural networks that interact in complicated ways.
3. Both involve learning: In a garden, plants learn to adapt to their environment and grow stronger. In AI systems, algorithms learn from data and adjust their behavior accordingly.
4. Both can produce something useful: A garden can produce food, flowers, and other useful resources. AI systems can produce insights, predictions, and other valuable outputs.
5. Both can be unpredictable: In a garden, weather, pests, and other factors can make it difficult to predict outcomes. Similarly, in AI systems, unexpected inputs or errors can lead to unpredictable results.

Overall, while AI and a garden may seem like very different things, there are some commonalities between the two in terms of their complexity, learning, and ability to produce useful outputs.

We asked Chat GPT if the user of AI have anything in common with a regular gardener and here's what it answered.

Yes, the user of AI has some things in common with a regular gardener:

1. Both need to have a clear goal in mind: A gardener typically has a vision for what they want their garden to look like, and chooses the right plants and techniques to achieve that vision. Similarly, the user of AI needs to have a clear goal in mind for what they want to achieve with the AI system, and must choose the right algorithms and approaches to accomplish that goal.
2. Both need to monitor and adjust their approach: A gardener needs to regularly monitor their garden, making adjustments to watering, fertilizing, and pest control as needed. Similarly, the user of AI needs to monitor the performance of the system and make adjustments to the algorithms and data inputs as needed to improve its performance.
3. Both need to be patient: A garden takes time to grow and flourish, and a gardener needs to be patient in order to see the results of their efforts. Similarly, the user of AI needs to be patient and give the system time to learn and improve in order to achieve the desired results.
4. Both need to be knowledgeable: A gardener needs to have knowledge of the plants they are growing, including their growth habits, water and sunlight requirements, and how to control pests and diseases. Similarly, the user of AI needs to have knowledge of the algorithms and data inputs they are using, in order to choose the right approach and make informed decisions about how to improve the system's performance.

Overall, while the tasks involved in using AI and gardening may be very different, the user of AI and the gardener both need to have clear goals, monitor and adjust their approach, be patient, and have knowledge of the tools they are using.

But why compare the AI user to the gardener after all? Because of the similarity among the garden and an AI system, given that they are both bottom up systems. In a bottom-up approach, we start with individual elements and gradually build up to the system as a whole. In the case of a garden, the individual elements such as the soil, seeds or plants, water, sunlight, and nutrients, interact with each other to create a complex

and dynamic system. Similarly, in an AI system, we start with individual components, such as algorithms, data, models, and computing resources, and build up to the overall system. Small scale interactions produce emergent outputs, that are greater than the sum of their parts.

4.2 Common Points Between the AI User and an Interviewer

Interviewing Chat GPT on the common points between the AI user and an interviewer, we got the following result:

They both require effective communication skills and the ability to ask the right questions to obtain the information they need. An interviewer must be able to put their subjects at ease in order to get the information they need, and similarly, the end user of AI must be able to communicate effectively with the AI systems they are working with. This requires a thorough understanding of the data being analyzed and the ability to ask the right questions in order to obtain the most useful insights.

Additionally, both an interviewer and an end user of AI must be able to interpret and analyze the information they receive. Just as an interviewer must be able to identify patterns and anomalies in the information they gather, the end user of AI must also be able to extract meaningful insights from complex data sets.

4.3 Common Points Between the AI User and an Interrogator

Interrogation is a complex process that requires the interrogator to carefully observe their subject and gather information through careful questioning and analysis. Similarly, the end user of AI must be able to identify patterns and anomalies in the data being analyzed. They must also be able to ask the right questions in order to obtain the information they need. Just as an interrogator must be able to remain calm and focused under pressure, the end user of AI must also be able to remain focused and objective even when dealing with large amounts of complex data.

The representation of the above concepts is depicted by the AI image generators Midjourney and DALLE (Figs. 1, 2, 3 and 4). With AI, the same text prompt is turning to larger texts and images. According to Lev Manovich, “translations (or mappings) between one media and another were done manually throughout human history, often with artistic intent. Novels have been adapted into plays and films, and comic books have been adapted into television series. A fictional or non-fictional text is illustrated with images. Each of these translations was a deliberate cultural act requiring professional skills and knowledge of the appropriate media [4].

Some of these translations can now be performed automatically on a massive scale thanks to AI, becoming a new means of communication and culture creation. What was once a skilled artistic act is now a technological capability available to everyone. We can be sad about everything that will be lost as a result of the automation and democratization of this critical cultural operation—skills, originality, “deep creativity,” and so on. However, any such loss may be only temporary if culture AI development improves its abilities to be original and understand context” [4].



Fig. 2. Created in midjourney “the user of ai as a gardener, an interviewer and an investigator, high fidelity”.

4.4 Paradoxon

After all this dialogical work, we phrased a paradox question, just to check if Chat GPT makes random but believable connections between irrelevant things, because this would make our position collapse. The question was “Does the user of AI have anything in common with a cow?” The answer was negative:

The user of AI does not have much in common with a cow, as cows are living animals and the user of AI is a human utilizing a technological tool. However, there are a few very general similarities that could be drawn between the two: [...] However, these similarities are very general and should not be taken too far, as the user of AI and a cow are very different entities with different needs and characteristics.



Fig. 3. Created in midjourney “t a human gardener who is also an investigator, interviewing an ai system, cinematic, photo realistic”.

5 Conclusions

In a system like Chat GPT, the user shares some characteristics with the gardener, the interviewer and the investigator, as she treats a bottom-up structure and her way to make the system flourish is through written questions-commands, widely known as *prompts*. But the most important characteristic of the human user is her interpreting ability while reading: the fact that she attributes causality to the outcomes of the AI system. The cause-result relationship occurs from the juxtaposition of the fragments of meaning produced by the AI system.

Chat GPT produces fragments of meaning, that the user is invited to assemble and interpret. That's why the conduction of a dialogue is needed in order to take the meaning production a bit further: The user asks questions and the system “responds” via AI generated text, and the overall meaning occurs due to the interpretational ability of the human user. Meaning is richer than the sum of its parts. It emerges from the succession



Fig. 4. Created in DALLE “the user of AI as a gardener, an interviewer and an investigator, realistic”.

of fragmented texts, just as the succession of shots in a movie, through its editing, leads to complex forms of narratives, due to the phenomenon of *semantic montage*.

Massumi describes the work of Deleuze and Guattari as malleable [3]: «You can take a concept that is particularly to your liking and jump with it to its next appearance. They tend to cycle back. Some might call that repetitious. Deleuze and Guattari call it a refrain. Most of all, the reader is invited to lift a dynamism out of the book entirely, and incarnate it in a foreign medium, whether it be painting or politics. The authors steal from other disciplines with glee, but they are more than happy to return the favor. Deleuze’s own image for a concept is not a brick, but a “tool box”. He calls his kind of philosophy “pragmatics” because its goal is the invention of concepts that do not add up to a system of belief or an architecture of propositions that you either enter or you don’t, but instead pack a potential in the way a crowbar in a willing hand envelops an energy of prying». Although AI content-creation engines are becoming popular at the 2020s, their theoretic background has already been developed since the second half of the 20th

century. Barthes's "death of the author" [2] is now incarnated, since in such systems the author asks questions, makes prompts and seeds ideas to exhort the AI system flourish.

References

1. Barthes, R.: *Mythologies*, p. 152. Editions du Seuil, Paris (1957)
2. Barthes, R.: *Image-music-text*. Macmillan, Paris (1977)
3. Deleuze, G., Guattari, F.: *A thousand plateaus, capitalism and schizophrenia*, p. xv. University of Minnesota Press, Minneapolis (1987)
4. Manovich, L.: <https://www.instagram.com/p/CqLJYnpPPX-/>. Last accessed 03 April 2023
5. University of Houston.: <https://uh.edu/~lcr3600/simulation/steps.html>. Last accessed 28 Feb 2023



Personalizing Text-to-Image Diffusion Models by Fine-Tuning Classification for AI Applications

Rafael Hidalgo, Nesreen Salah, Rajiv Chandra Jetty, Anupama Jetty,
and Aparna S. Varde^(✉)

School of Computing, Montclair State University, Montclair, NJ, USA
`{hidalgor2, salahn1, jettyr1, jettyr1, vardea}@montclair.edu`

Abstract. Stable Diffusion is a captivating text-to-image model that generates images based on text input. However, a major challenge is that it is pretrained on a specific dataset, limiting its ability to generate images outside of the given data. In this paper, we propose to harness two models based on neural networks, Hypernetworks and DreamBooth, to allow the introduction of any image into Stable Diffusion, addressing versatility with minimal additional training data. This work targets AI applications such as augmenting next-generation multipurpose robots, enhancing human-robot collaboration, feeding intelligent tutoring systems, training autonomous cars, injecting subjects for photo personalization, producing high quality movie animations etc. It can contribute to AI in smart cities: facets such as smart living and smart mobility.

Keywords: ANN · Data mining · Image processing · Movie animations · Photo personalization · Stable diffusion · Text-to-image creation

1 Introduction

In this paper, we address data mining in text-to-image generation via the paradigm of *Stable Diffusion* with fine-tuning using architectures based on artificial neural networks (ANN). It adds real-life perspectives to the images created (see Fig. 1) and can be useful in various AI applications such as movie animations.

From credit card fraud detection to TikTok, artificial neural networks are being used all around us. As is well-known today, an ANN is a machine learning paradigm modeled after the human brain. It is composed of many interconnected processing nodes, called neurons, which work together to process input data and make predictions or decisions based on that data. One of the main areas for neural networks usage is classification, i.e. the process of analyzing input data to estimate a target output as being in one of several predefined classes or categories. ANNs are particularly well-suited for tasks involving images.



Fig. 1. Images created by stable diffusion after fine tuning via hypernetworks (3 Left) and dreamBooth (3 Right); prompts to generate each image are below it

A very interesting technology using ANNs for image classification and creation is the diffusion model. Specifically, we refer to *Stable Diffusion*, developed by Stability AI, CompVis LMU et al. [3] used for text-to-image creation. While it is a popular technology [10, 12], it incurs challenges [7] including the need for: exhaustive training data to generate images outside pretrained datasets; exploration of real-life applications especially with good social impact; and more diversity and inclusion (D&I) as per country, ethnicity etc.

Given this motivation and challenges, we address the problem in this paper, on “thinking outside the box” more specifically defined as follows.

- Encompass numerous real-life perspectives into Stable Diffusion
- Create new images with minimal training data, yet addressing versatility
- Incorporate more diversity and inclusion for global mass appeal.

We propose a solution to this problem by exploring two advances, namely, Hypernetworks and DreamBooth in conjunction with Stable Diffusion that can be adapted to work with low volumes of training data while still producing a robust set of images in various contexts. We add real-life angles to this work by considering numerous facets for generating the images, and outlining several targeted applications. Furthermore, we take into account D&I from a global perspective, considering multiple countries, religions, subject names and ethnicity, aiming to stay away from stereotypes solely based on Western cultures (oft found in image searches). Our work yields high levels of user satisfaction, as evident from the experimentation.

While there is much literature in the area [10, 12, 29], to the best of our knowledge, our paper is early work on exploring Stable Diffusion with Hypernetworks and DreamBooth, contextualizing them with multiple real-life perspectives, using relatively less new training data, and leveraging D&I. This constitutes the novelty of our paper, contributing modestly to neural models and image processing, helping applications such as photo personalization and movie animations, and broadly making impacts on AI in smart cities, vis-a-vis smart living and other facets. We present the details of our work in the forthcoming sections.

The rest of this paper is organized as follows. Section 2 explains the models and methods harnessed in our work, namely, Stable Diffusion, Hypernetworks and DreamBooth. Section 3 describes the implementations of our proposed

approaches along with algorithms. Section 4 synopsizes our experimental evaluation, presenting a discussion as well. Section 5 overviews related work in the area, placing our own work in context, and emphasizing its novelty. Finally, Sect. 6 states the conclusions and outlines prospective avenues for future research.

2 Models and Methods

In this paper, we focus on the paradigm of Stable Diffusion. We aim to add more real-life perspectives to the images in text-to-image creation via minimal additional data. This is explored via recent advances: Hypernetworks and DreamBooth. We discuss these main models, and the methods we apply on them.

2.1 Stable Diffusion

Considering the fundamental concept of diffusion models, we begin the process through a forward diffusion process using Markov chains (Fig. 2). Through Markov chains, we take an image and add some Gaussian noise to it. This step can be repeated an infinite number of times, but is usually terminated when enough noise is added to the picture such that it is not recognizable any more. Each image in the Markov chain from start to end can then be used to train a convolutional neural network (CNN) called U-Net to denoise the image [17, 28].

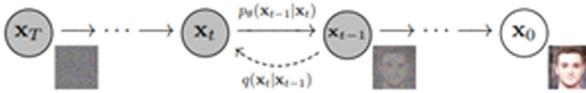


Fig. 2. Denoising image via Markov chains [17]

Specifically U-Net is trained by receiving a noised image along with the number of iterations of noise applied to the noised image. Thereafter U-NET is instructed to calculate the total noise of the image so as to revert the image back to its original form. Depending on how well U-NET performs, it receives a negative or positive reward (in line with reinforcement learning). Eventually U-NET is able to calculate noise removal in all iterations of the Markov chain noising process and is trained to do this in a single step [17, 28].

However, many errors can occur if the network is allowed to remove noise in a single step, so the neural network instead is made to calculate all of the noise it believes is polluting an image, but only removes a fraction of that noise, just enough to undo the noising process by one iteration. The neural network then repeats this for an image until it reaches iteration 0 of the noising process, after which it yields the original image [12, 17, 28].

A GPT (Generative Pre-trained Transformer) is integrated to the model, enabling it to take text prompts to produce the desired image. After receiving

text, the model receives a random noise image and the number of noise iterations the image underwent. The ANN denoises the image as per the process above [17, 28]. In order to ensure that our image is being well denoised, a technique called classifier free guidance (CFG) is applied. In CFG, the image is fed into the neural network twice. The first copy of the image is run with the text embedding, and the second copy without it. Noise is calculated from both the images, and the difference between them is obtained. This is then used to guide the neural network into refining its denoising process to obtain an image as described by the text prompt [17, 28].

2.2 Hypernetwork Model

Hypernetworks are neural networks used to predict the weights of primary networks. The Hypernetwork consists of a lightweight feature extractor and a set of refinement blocks. Each refinement block is tasked with predicting the weights of a primary network. By training Hypernetworks over a large collection of data, the weights of the primary network are adjusted with specific inputs yielding more expressive models [4, 16]. Figure 3 depicts a Hypernetwork.

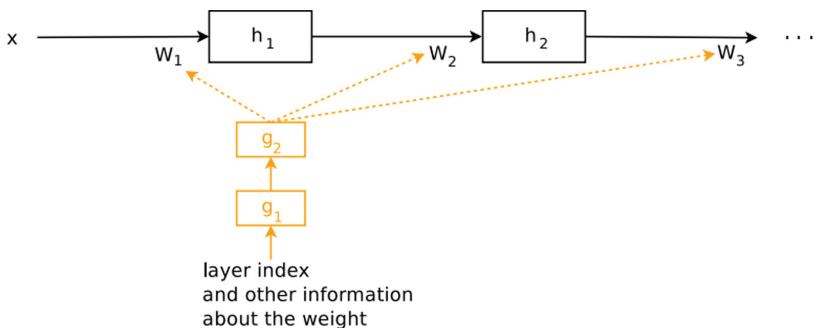


Fig. 3. Diagram of a hypernetwork [16]

A major advantage of Hypernetworks is that they are tiny, mostly of the order of Megabytes (80 Mb per Hypernetwork). Another advantage is that they are modular and can be attached and detached from the main network as needed. A disadvantage is that the Hypernetworks change all the weights of the main networks. Hypernetworks have been deployed to applications such as 3D modeling, semantic segmentation, neural architecture search, continual weights and so on [10]. We therefore use them in conjunction with Stable Diffusion. The details are described in the next section on implementation.

2.3 DreamBooth Model

DreamBooth is a deep learning text-to-image diffusion model meant to fine-tune larger models. Prior to this, there existed large text-to-image models but they

lacked the ability to generate realistic pictures of the subjects in the reference set, whereas in DreamBooth we find a new approach for “personalization” of text-to-image diffusion models. Using just a few pictures of the subject given as inputs, it fine tunes the pretrained text-to-image models (see Fig. 4) [14,30].

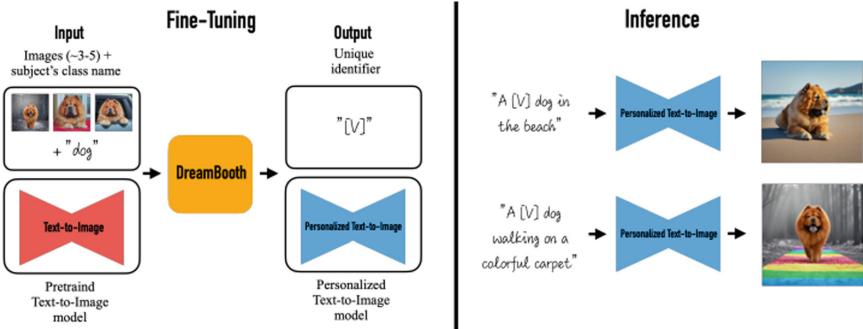


Fig. 4. Illustration of dreamBooth [30]

It operates in two steps. In the first step, it generates a low-resolution image from text-to-image models with input images and text-prompts containing unique identifiers, followed by the class name of the subject, to prevent overfitting and language drift (that causes the model to associate with the class name). In the following step, it fine-tunes the super-resolution component using the input images, and generates low-resolution as well as high-resolution versions of the input images [14,30].

This allows the model to maintain high fidelity to small, but important, details of the subject. DreamBooth binds a unique identifier to the specific subject. Once the subject is embedded in the output domain of the model, it uses a unique identifier to generate fully novel photo-realistic images of the subject contextualized in different scenarios such as various poses, views, and lighting conditions that do not appear in the reference images. We can apply this technique to perform different tasks such as subject re-contextualization, text-guided view synthesis, appearance modification, and artistic rendering, while maintaining the key features of the subject [14,30]. We now present the implementation of our approaches on DreamBooth and Hypernetworks along with Stable Diffusion.

3 Implementation of Approaches

In order to discover knowledge from data during text-to-image creation, as well as to explore a wide range of contexts for applications, we seek to incorporate our own personal images into Stable Diffusion. We use them for a variety of text-to-image generation scenarios. This is achieved by two approaches: adapting Hypernetworks, and deploying DreamBooth, as follows.

3.1 Hypernetwork Implementation

In our Hypernetwork adaptation, 24 photos of a given subject are taken. In the work shown in this paper, the subject is “Rafael Hidalgo” a Latino male student with a family background spanning multiple countries. The photos are cropped to produce 1×1 pixel images, and converted to 512×512 -pixel images [8] as required by the model. Figure 5 depicts a sample of the images used here.



Fig. 5. Pictures for hypernetwork model training

An open-source Stable Diffusion Web UI (User Interface) developed by a GitHub user named “automatic1111” is installed here [5]. In this Web UI, the user “automatic1111” employs a Gradio library, Python, and PyTorch to use Stable Diffusion models for text-to-image generation. Besides utilizing this Web UI, for text-to-image generation, it is used to train the Hypernetwork. Stability AI, one of the main companies responsible for generating the Stable Diffusion model, has saved their copy of the model in the Hugging Face website, which contains a library of transformers, datasets, and demos of machine learning projects. We download a copy of this Stable Diffusion model onto a personal computer to use with the automatic1111 web UI. [3, 29]. With the Web UI and the model installed, we train the Hypernetwork using 24 images of the subject. First, we preprocess the pictures by running the subject’s images through a Bootstrapping Language-Image Pretraining (BLIP) model. It generates relevant captions for the image, so it can associate pictures with the respective image captions.

Once the captions are generated for all images of the subject, the training is processed using a personal computer and 8GB of V-RAM. The Hypernetwork is trained for 10,000 steps. For instance, sample pictures are generated every 100 steps by the prompt “A portrait of a man, trending on artstation, greg rutkowski, 4k”. After 10,000 steps, the Hypernetwork is tested via the use of different prompts. More about this appears in the section on experimental results with discussion. Algorithm 1 has the pseudocode on our Hypernetworks execution, and synopsizes its overall processing.

Algorithm 1 Training the Hypernetwork along with Stable Diffusion

-
- 1: Load pre-trained ANN model α
 - 2: Initialize Hypernetwork HN
 - 3: Preprocess data δ for subject σ
 - 4: Use BLIP to generate captions γ for σ
 - 5: Load modified subject data δ_m
 - 6: Split δ_m into training set τ , validation set v
 - 7: **for** each iteration i **do**
 - 8: Have HN generate weights ω to influence α
 - 9: Train α using ω on τ
 - 10: Get results ρ from α similar to σ via associated class χ
 - 11: Update HN via backpropogation based on ρ
 - 12: **end for**
 - 13: Save the trained HN
 - 14: Attach HN to α
 - 15: Generate new subject data δ_n
 - 16: Output δ_n for σ w.r.t. context
-

3.2 DreamBooth Implementation

Our deployment of DreamBooth is on similar lines as Hypernetwork, with a few variations. Multiple photos of our subject are taken. The subject in this task is “Nesreen Salah”, a female student of Egyptian descent and American upbringing. The photos are adjusted to form 1×1 scale pixel images, and converted to 512 \times 512-pixel images as befits the model. Figure 6 has a snapshot of the images [8].

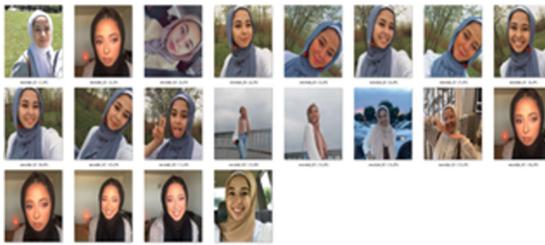


Fig. 6. Images for dreamBooth model training

Analogous to the Hypernetwork, our DreamBooth model is trained on 20 pictures of our subject. The only caveat however, is that the process to train DreamBooth is resource-heavy, and it is recommended that at least 24 GB of V-RAM be available to train DreamBooth [10]. Therefore, we harness *Google Colab* (Google’s Research Colaboratory) to procure a computer with the necessary amount of V-RAM. We adapt a DreamBooth implementation from a GitHub user with the username “ShivamShrirao” to train our modified model [31].

We take a copy of Stable Diffusion from the Hugging Face website for this task as well, however, this time we incorporate it into the DreamBooth implementation on Google Colab. Thereafter, we upload our dataset of pictures to the DreamBooth implementation. We then correlate the pictures with an instance-name of our choosing and with a class-prompt that already exists in the Vanilla Stable Diffusion Model. For our instance-name, we need to ensure that it does not exist in Stable Diffusion, so as to avoid confusion with another instance. It is best to stay away from conventional names, as yet another aspect of diversity. Hence, for our subject, we choose the name “nesrelah_001”; and we associate our subject with the class “person” [29].

Once the instance name and class name are established, we train our model using 4,040 steps. In order to generate these pictures, we use the prompt “nesrelah_001 person”. After 4,040 steps, we download the new trained model.ckpt file from Google Colab onto a personal computer, to use it with the “automatic1111” Web UI as described earlier. The DreamBooth-modified Stable Diffusion model is tested via various prompts. More on this appears in our experimental results section [5]. Algorithm 2 has our pseudocode for the DreamBooth execution.

Algorithm 2 Training via DreamBooth in Conjunction with Stable Diffusion

- 1: Load pre-trained ANN model β
 - 2: Initialize DreamBooth DB
 - 3: Preprocess data Δ for subject κ
 - 4: Associate κ with class ζ related to it
 - 5: Associate κ with unique identifier ι
 - 6: Load modified subject data Δ_M
 - 7: Split Δ_M into training set θ , validation set ψ
 - 8: **for** each iteration j **do**
 - 9: Send input I to β using ι of κ
 - 10: Get predicted results ϕ from β similar to κ via ζ
 - 11: Calculate loss $\lambda = |\phi - \mu|$ where μ = actual results
 - 12: Use λ to update β via backpropagation
 - 13: Update DB using $ArgMin(\lambda)$
 - 14: **end for**
 - 15: Save the trained DB
 - 16: Attach DB to β
 - 17: Generate new subject data Δ_N
 - 18: Output Δ_N for κ w.r.t. context
-

4 Experimental Results with Discussion

In order to test our models, we mostly use AI art search engines to help us decide the prompts to use. It is exciting to note that with a relatively small dataset, and two fine-tuning implementations, we can significantly harness the power of Stable Diffusion to create prompts with any subject of our choosing.

Hence, this facilitates “thinking outside the box”. As seen here, we can put both the subjects on horses, render paintings of the subjects in various art styles, alter the subjects to look like Chibi figurines, and make a subject into a Jade statue. Accordingly, we evaluate our work (which at this point is a pilot study). We assess and compare the two fine-tuning models.

The main difference between these two fine-tuning technologies seems to be the following. In Hypernetworks, the class associated with the subject becomes affected by that very same subject. As can be seen in the prompts used for Hypernetworks, no specific name is selected for the subject. Instead, the subject is just referred to as “man”. This essentially enables Stable Diffusion to create images of the subject, but at the cost of making all instances of “man” look like the subject. On the other hand, as can be seen with the prompts generated via DreamBooth, the subject is kept distinct from the class. Yet, the caveat with DreamBooth (as mentioned earlier) is that more V-RAM is needed to process the training, and the model itself is 2 GB large, versus the size of Hypernetwork, which is around 80 MB [10].

Based on our experimentation with Stable Diffusion alone, as well as using it in conjunction with Hypernetworks and DreamBooth, a summary of our results is presented here. The prompts in these experiments are chosen (guided by AI art search engines) as per the operations we aim to execute using Stable Diffusion with Hypernetwork/DreamBooth, anticipating various tasks that targeted users may perform with them.

4.1 Hypernetwork Experiments

The prompts for the Hypernetwork include:

1. “Man Riding a Horse, Facing Camera” (Fig. 7).
2. “Portrait of a man, trending on artstation, greg rutkowski, 4k” (Fig. 1).
3. “Man made of fire, intricate heat distortion designs, elegant, highly detailed, sharp focus, art by Artgerm and Greg Rutkowski and WLOP” (Fig. 1).
4. “Chibi Man Figurine, Modern Disney Style” (Fig. 8).
5. “Man with a majestic beard, closeup, D&D, fantasy, intricate, elegant, highly detailed, digital painting, artstation, concept art, matte, sharp focus, illustration, art by Artgerm, Greg Rutkowski, Alphonse Mucha” (Fig. 1).

4.2 DreamBooth Experiments

The prompts for DreamBooth include:

1. “nesrelah_001 on a horse” (Fig. 9).
2. “Highly detailed marble and jade sculpture of nesrelah_001, volumetric fog, Hyperrealism, breathtaking, ultra realistic, unreal engine, ultra detailed, cyber background, Hyperrealism, cinematic lighting, highly detailed, breathtaking ,photography, stunning environment, wide-angle [cgi, 3d, doll, octane, render, bad anatomy, blurry, fuzzy, extra arms, extra fingers, poorly drawn hands, disfigured, tiling, deformed, mutated]” (Fig. 1).



Fig. 7. Prompt: man riding a horse, facing camera



Fig. 8. Prompt: chibi man figurine, modern disney style

3. “Portrait of nesrelah_001, dramatic lighting, illustration by greg rutkowski, yoji shinkawa, 4k, digital art, concept art, trending on artstation” (Fig. 1).
4. “An epic fantastic realism comic book style portrait painting of nesrelah_001 robot with kanji tattoos and decals, apex legends, octane render, intricate detail, 4 k hd, unreal engine 5, ex machina, irobot, gerald brom” (Fig. 1).
5. “Chibi nesrelah_001 figurine, Disney style” (Fig. 10)



Fig. 9. Prompt: Nesrelah_001 on a horse



Fig. 10. Prompt: Chibi Nesrelah_001 Figurine, modern disney style

4.3 Overall Assessment with Comparison

Considering these experiments, Table 1 synopsizes our evaluation, the base case being that of Stable Diffusion without fine-tuning; the others being its fine-tuning with Hypernetwork and DreamBooth respectively. The evaluation is fairly simplistic at this point because this is a pilot study. Users evaluating this work are asked to mention whether they are satisfied with the created images (comments optional). We have a small group of 20 student users. It is evident that Hypernetworks and DreamBooth both yield higher levels of user satisfaction than Stable Diffusion alone. Yet the training data used for both these models is substantially low, compared to the original Stable Diffusion dataset, indicating that we can achieve good results in text-to-image generation in various scenarios, “thinking outside the box”, without much additional data.

Table 1. Summary of assessment outcomes in pilot study

Approach	Avg. accuracy	Synopsis of users' general comments
<i>Base Case</i>	Approx. 80%	Users not satisfied with ~20% of images
<i>Hypernetwork</i>	Approx. 90%	Users partly satisfied with ~10% of images
<i>DreamBooth</i>	Approx. 95%	Users find ~5% of images slightly below expectation

Table 2. Comparative Observations of Fine-Tuning Models in Pilot Study

Model	V-RAM	Main source	Train-time	Other observations
<i>Hypernetwork</i>	8 GB	Personal Comp.	10000 steps	Class affected by subject
<i>DreamBooth</i>	24 GB	Google Colab	4040 steps	Class distinct from subject

As per quantitative and qualitative observations, the time needed to train these models is of the order of minutes (not hours) while the resources used are

mainly a personal computer and Google Colab. More specifically, the Hypernetwork needs 8GB of V-RAM and is trained on a PC, the training requiring 10,000 steps; while DreamBooth needs 24 GB of V-RAM and is trained mainly using Google Colab, the training occurring in 4,040 steps. This is summarized in Table 2. Thus, the Hypernetwork is less resource-consuming and is trained using more steps while DreamBooth is more resource-intensive and is trained with relatively fewer steps. Also, the Hypernetwork makes the class get directly affected by the subject associated with it, while DreamBooth keeps the subject and class distinct from each other (as explained at the beginning of this section). Both the models are effective for fine-tuning classification in Stable Diffusion, requiring reasonable training time and resources, as evident from the Tables.

4.4 Targeted Applications

The potential applications of these technologies are quite substantial. One can certainly leverage such advancements to be able to create art, despite one's skills. An interesting application that could be considered is the generation of new pictures of a dearly departed family member, friend, or famous personality, e.g. movie actor. Another application can be to generate different styles of clothing to figure out what looks best on a given person. One can even save substantial amounts of money on a photo-shoot by tweaking several parameters. These applications entail subject injection in photographs to encompass reality.

On the whole, the real-life image curation can have many benefits due to infusing more personalization, potentially useful in contexts such as:

- Photo adjustments good for clothes comparison, formal photo-shoots etc.
- Autonomous vehicles for automated driving across various regions.
- Human-robot collaboration/interaction with enhanced image classification.
- Intelligent tutoring systems and mobile apps with personalized icons for user interaction to achieve worldwide outreach in a variety of scenarios.
- Movie animations with high quality images and videos entailing mass appeal.
- Next-generation multipurpose robots with improved object detection and versatile behavior

Some of these applications can make positive impacts on various facets of AI in smart cities in line with earlier work in our research lab, e.g. improved autonomous vehicles for smart mobility [25], enhanced human-robot collaboration for smart manufacturing [11] etc. Much of the work in this paper can be used for future studies in our lab across various projects spanning visualization, textual data, mobile app development, smart city applications and so forth [6, 20, 21, 26, 38]. Hence, our work in this paper is applied research with implementation and experimentation, contributing to neural networks and image processing, making positive impacts on numerous targeted applications.

5 Related Work

There is much work on Stable Diffusion and fine-tuning, as elaborated in the literature [10, 12, 29]. There is also a myriad of research on image mining in general [18, 22, 34, 35, 37]. Our work in this paper is orthogonal to such research. Yet, one must be cognizant of the technological and ethical limitations. The creators of the Stable Diffusion paradigm have stipulated the following. “While the capabilities of image generation models are impressive, they can also reinforce or exacerbate social biases. Stable Diffusion v1 was trained on subsets of LAION-2B(en), which consists of images that are primarily limited to English descriptions. Texts and images from communities and cultures that use other languages are likely to be insufficiently documented. This affects the overall output of the model, as Caucasian and Western cultures are often set as the default. Further, the ability of the model to generate content with non-English prompts is significantly worse than with English-language prompts [29].”

Other researchers have shown that there are true societal biases. Bianchi et al. [7] compiled images on specific prompts to see if they catered to a specific race, ethnicity, sexual orientation, etc. For instance, in “An American man and his car” versus “An African man and his car”, the picture of the American man is often portrayed as affluent, while the African man is usually shown as impoverished. Perhaps future implementations of Stable Diffusion can address this problem, or perhaps fine-tuning technologies such as Hypernetworks and DreamBooth can help remove societal biases inherent in current models [7]. In connection with this, issues such as public opinion can often be important; just as people voice their concerns on environmental matters [13], they can also express their reactions to artistic pursuits, thereby offering the scope for multidisciplinary research spanning data science, analogous to a few other works [32, 36].

Furthermore, some limitations of the overall technology include using an artist’s name to generate an image in that artist’s style. We used artist-names such as Greg Rutkowski and Alphonse Mucha to generate our chosen art in styles reminiscent to those they produce. By using an artist’s style, the artistic quality of the image increases. However, if someone were to profit from the image, it then begs the question: Should the artist be credited, and even compensated for the given image? Also, would future work by the artist be devalued if AI is able to generate the same art-style for free? These and other limitations indicate that Stable Diffusion has to be taken with a grain of salt. Much of this presents the potential for future work.

Other related work entails infusing commonsense knowledge (CSK) into image generation [15], object detection [9] and more aspects [2, 19]. There is a plethora of work on CSK as outlined in tutorials [27, 33] some of which can be relevant for image creation. Cultural commonsense knowledge is being studied [1, 23] and can be explored further to enhance text-to-image and image-to-text (automated captions) generation. This is another aspect of D&I. Our paper does not address CSK, however, it addresses societal context and global diversity to some extent. Future work in this area can consider facets of CSK as well.

On a final note, we can mention that recent advances such as ChatGPT [24] might possibly be correlated with some of our work in this paper. Much of ChatGPT thrives on reinforcement learning. Hence, the text and images used in this study can be used to train ChatGPT and other such systems in order to make them more versatile and globally-oriented, propelling additional work on the lines of thinking outside the box. We make just a modest contribution here, helping to augment text-to-image creation with fine-tuning classification, thereby being advantageous to many AI applications.

6 Conclusions

In this paper, we aim to generate novel photo-realistic images from text prompts, using a given reference set, laying much emphasis on subject personalization. This is achieved through Stable Diffusion, DreamBooth, and Hypernetworks. Comparing the results, there definitely is a better degree of control with DreamBooth versus Hypernetworks. Also, DreamBooth seems to generate higher quality images. However, the Hypernetwork model is less resource-heavy and can be implemented completely on a typical PC [16, 30].

Main Contributions: Briefly, our contributions are highlighted as follows.

1. Investigating Stable Diffusion with Hypernetworks and DreamBooth for text-to-image generation.
2. Getting high user satisfaction for image creation with low training data.
3. Addressing novel and versatile contexts, producing good quality images, and outlining various targeted applications.
4. Leveraging diversity and inclusion as per various real-life perspectives.
5. Making broader impacts on AI in smart cities in a modest manner.

Limitations and Future Work: Various limitations of Stable Diffusion seen in the literature and corroborated by our experiments, present the scope for future work in the area. The notion of diversity and inclusion for image creation from text needs more attention. Though we have focused on some of it, there are still open avenues, e.g. as noticed in the related work. Racial, ethnic and other types of diversity in text-to-image generation calls for further research. Exploring these aspects while also addressing privacy-preserving issues and confidentiality concerns, can pose more challenges. These can open up avenues for future work.

Furthermore, qualitative performance of the models used in this work can be judged on a wide variety of tuning datasets. In this paper, two diverse datasets have been used to tune each of the two models, while in the future more heterogeneous data can be considered. Likewise, quantitative performance in terms of “user preference” on images obtained by the two models trained with the same training data can be judged as well, e.g. users should choose the “best image” between any pair of images generated by the two models. Such detailed experiments can be carried out with larger datasets as well as bigger user study groups such as those on AMT (Amazon Mechanical Turk). Permissions for such work need to be obtained from the respective IRB (Institutional Review Board)

since there are human subjects involved in the study, and this can be rather time-consuming. Hence, a small scale study has been conducted in this paper with informal evaluations, which puts forth the scope for further work.

Additionally as future research, we can explore details from an application-standpoint, e.g. human-robot collaboration, autonomous vehicles, and intelligent tutoring systems, blending that with relevant projects in our labs. It is also important to address ethical issues such as artist compensation which might be beyond the scope of our own work but can be addressed by other researchers working across the concerned applications. Finally, we aim to investigate more specific roles that commonsense knowledge can play with respect to the overall theme of this research. Our work in this paper modestly contributes to ANN and image processing, making potential impacts on various AI applications.

Acknowledgments. Acknowledgments and Disclaimer Dr. Aparna Varde acknowledges NSF grants 2018575 “MRI: Acquisition of a High-Performance GPU Cluster for Research & Education”, and 2117308 “MRI: Acquisition of a Multimodal Collaborative Robot System (MCROS) to Support Cross-Disciplinary Human-Centered Research & Education at Montclair State University”. She is a visiting researcher at Max Planck Institute for Informatics, Germany (ongoing from sabbatical). She is an Associate Director of the School of Computing, and an Associate Director of the CESAC: Clean Energy & Sustainability Analytics Center, Montclair State University. We make a disclaimer that the opinions presented here are extracted from online sources; and the content of this paper including the images is not meant to offend/hurt any national, ethnic, cultural, racial, religious and other groups. The images produced here are taken with the consent of the respective subjects. Any resemblance to anyone else is coincidental. This is a pilot study.

References

1. Acharya, A., Talamadupula, K., Finlayson, M.A.: An atlas of cultural commonsense for machine reasoning. In: AAAI Conference on Artificial Intelligence (2021)
2. Aditya, S., Yang, Y., Baral, C., Fermuller, C., Aloimonos, Y.: From images to sentences through scene description graphs using commonsense reasoning and knowledge (2015). [arXiv:1511.03292](https://arxiv.org/abs/1511.03292)
3. Stability A.I. Stable diffusion public release
4. Alaluf, Y., Tov, O., Mokady, R., Gal, R., Bermano, A.H.: Hyperstyle: Stylegan inversion with hypernetworks for real image editing (2022). [arXiv:2111.15666](https://arxiv.org/abs/2111.15666) [cs]
5. AUTOMATIC1111. Stable Diffusion Web UI (2022)
6. Basavaraju, P., Varde, A.S.: Supervised learning techniques in mobile device apps for androids. ACM SIGKDD Explor. **18**(2), 18–29 (2017)
7. Bianchi, F., Kalluri, P., Durmus, E., Ladzhak, F., Cheng, M., Nozza, D., Hashimoto, T., Jurafsky, D., Zou, J., Caliskan, A.: Easily accessible text-to-image generation amplifies demographic stereotypes at large scale (2022). arxiv.org
8. Birme. Birme - bulk image resizing made easy 2.0 (online & free)
9. Chernyavsky, I., Varde, A.S., Razniewski, S.: CSK-Detector: Commonsense in object detection. In: IEEE International Conference on Big Data, pp. 6609–6612 (2022)
10. Cheung, B.: Stable diffusion training for personal embedding (2022)

11. Conti, C.J., Varde, A.S., Wang, W.: Human-robot collaboration with commonsense reasoning in smart manufacturing contexts. *IEEE Trans. Autom. Sci. Eng.* **19**(3), 1784–1797 (2022)
12. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis (2021). [arXiv:2105.05233](https://arxiv.org/abs/2105.05233) [cs, stat]
13. Du, X., Kowalski, M., Varde, A.S., de Melo, G., Taylor, R.W.: Public opinion matters: Mining social media text for environmental management. *ACM SIGWEB Autumn* 1–15 (2020)
14. Hugging Face. Dreambooth fine-tuning example
15. Garg, A., Tandon, N., Varde, A.S.: I am guessing you can't recognize this: Generating adversarial images for object detection using spatial commonsense. In: AAAI Conference on Artificial Intelligence, vol. 34, pp. 13789–13790 (2020)
16. Ha, D., Dai, A., Le, Q.V.: Hypernetworks (2016). [arXiv:1609.09106](https://arxiv.org/abs/1609.09106) [cs]
17. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models (2020). [arXiv:2006.11239](https://arxiv.org/abs/2006.11239) [cs, stat]
18. Hsu, W., Lee, M.L., Zhang, J.: Image mining: Trends and developments. *J. Intell. Inf. Syst.* **19**(1), 7–23 (2002)
19. Ilievski, F., Szekely, P., Cheng, J., Zhang, F., Qasemi, E.: Consolidating common-sense knowledge (2020). [arXiv:2006.06114](https://arxiv.org/abs/2006.06114)
20. Kaluarachchi, A., Roychoudhury, D., Varde, A.S., Weikum, G.: SITAC: Discovering semantically identical temporally altering concepts in text archives. In: International Conference on Extending Database Technology (EDBT), pp. 566–569. ACM (2011)
21. Karthikeyan, D., Shah, S., Varde, A.S., Alo, C.: Interactive visualization and app development for precipitation data in sub-saharan africa. In: IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), pp. 1–7. IEEE (2020)
22. Karthikeyan, D., Varde, A.S., Wang, W.: Transfer learning for decision support in covid-19 detection from a few images in big data. In: IEEE International Conference on Big Data, pp. 4873–4881 (2020)
23. Nguyen, T.-P., Razniewski, S., Varde, A., Weikum, G.: Extracting cultural commonsense knowledge at scale. In: WWW, the ACM Web Conference (2023)
24. OpenAI. Introducing ChatGPT (2022). <https://openai.com/blog/chatgpt>
25. Persaud, P., Varde, A.S., Robila, S.: Enhancing autonomous vehicles with common-sense: Smart mobility in smart cities. In: 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 1008–1012. IEEE (2017)
26. Puri, M., Varde, A., Du, X., De Melo, G.: Smart governance through opinion mining of public reactions on ordinances. In: 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 838–845. IEEE (2018)
27. Razniewski, S., Tandon, N., Varde, A.S.: Information to wisdom: Commonsense knowledge extraction and compilation. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp. 1143–1146 (2021)
28. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models (2022). [arXiv:2112.10752](https://arxiv.org/abs/2112.10752) [cs]
29. Rombach, R., Esser, P.: Compvis/stable-diffusion-v-1-4-original · hugging face (2022)
30. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation (2022). [arXiv:2208.12242](https://arxiv.org/abs/2208.12242) [cs]
31. Shirao, S.: Diffusers/examples/dreambooth at main · shivamshrirao/diffusers (2022)

32. Suchanek, F.M., Varde, A.S., Nayak, R., Senellart, P.: The hidden Web, XML and the semantic Web: Scientific data management perspectives. In: International Conference on Extending Database Technology (EDBT), pp. 534–537. ACM (2011)
33. Tandon, N., Varde, A.S., de Melo, G.: Commonsense knowledge in machine intelligence. ACM SIGMOD Record **46**(4), 49–52 (2017)
34. Theisen, W., Cedre, D.G., Carmichael, Z., Moreira, D., Weninger, T., Scheirer, W.: Motif mining: Finding and summarizing remixed image content. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1319–1328 (2023)
35. Varde, A., Rundensteiner, E., Javidi, G., Sheybani, E., Liang, J.: Learning the relative importance of features in image data. In: *IEEE ICDE (International Conference on Data Engineering), workshops*, pp. 237–244 (2007)
36. Varde, A.S.: Challenging research issues in data mining, databases and information retrieval. ACM SIGKDD Explor. **11**(1), 49–52 (2009)
37. Varde, A.S., Rundensteiner, E.A., Ruiz, C., Maniruzzaman, M., Sisson, R.D., Jr.: Learning semantics-preserving distance metrics for clustering graphical data. In: Proceedings of the 6th International Workshop on Multimedia Data Mining: Mining Integrated Media and Complex Data, pp. 107–112 (2005)
38. Varghese, C., Pathak, D., Varde, A.S.: SeVa: A food donation app for smart living. In: 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0408–0413. IEEE (2021)



ShapTime: A General XAI Approach for Explainable Time Series Forecasting

Yuyi Zhang, Qiushi Sun, Dongfang Qi, Jing Liu, Ruimin Ma,
and Ovanes Petrosian^(✉)

Saint-Petersburg State University, 198504 St. Petersburg, Russia
st088518@student.spbu.ru, petrosian.ovanes@yandex.ru

Abstract. The application of Explainable AI (XAI) in time series forecasting has gradually attracted attention, given the widespread implementation of machine learning and deep learning. ShapTime - A general XAI approach based on Shapley Value specially developed for explainable time series forecasting, which can explore more plentiful information in the temporal dimension, instead of only roughly applying traditional XAI approaches to time series forecasting as in previous works. Its novel components include: (1) It provides the relatively stable explanation in the temporal dimension, that is, the explanation result can reflect the importance of time itself, which is more suitable for time series forecasting than traditional XAI approaches; (2) It builds the practical application scenario of XAI - improving forecasting performance guided by explanation results. This is distinctly different from previous works, which only present the results of XAI as the demonstration of innovation. Eventually, in five real-world datasets, ShapTime's average performance improvements for Boosting, RNN-based and Bi-RNN-based reached 18, 20 and 35%, respectively.

Keywords: Time-series forecasting · Explainable AI · Shapley value

1 Introduction

Numerous time series forecasting competitions including M4 [1] and M5 [2] have shown that ML and DL perform significantly better than traditional statistical methods, especially for more complex tasks. This has led to research on the application of Explainable AI (XAI) in time series forecasting. Explainable time series forecasting aims to improve the trustworthiness of ML and DL in fields such as Finance, Energy and Meteorology. There are two main approaches to apply XAI in time series forecasting models: (1) directly using the existing model-agnostic method with high generality; (2) developing a model-specific method specifically for the model. These two approaches directly caused two key problems.

Problem 1. *In time series forecasting, the existing model-agnostic method is roughly applied, resulting in insufficient explanation.*

The essential reason for the insufficient explanation is that most of the existing model-agnostic methods are feature attribution methods, given that XAI was originally developed based on regression and classification tasks, such as SHAP [3], LIME [4], etc. In contrast to time-series data, there is no temporal relationship among data instances for regression and classification, so the corresponding model-agnostic method pays more attention to feature importance (or contribution). However, this is not sufficient for time series forecasting. In time series forecasting models $y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \omega_t$; $\omega_t = b + k_1 x_1 + k_2 x_2 + \dots + k_n x_n$. y_{t-i} is the historical data of the target variable, and ω_t is composed of features x_i and intercept terms b . These traditional model-agnostic methods can only output feature importance k_i , but not the importance of time itself ϕ_i . Actually, due to human cognitive inertia, it is common to use feature importance as the explanation result even in model-specific methods.

Problem 2. *The specially developed model-specific method needs to be embedded in the model, resulting in low generality and high application cost.*

Several works have noticed Problem 1 developed some explainable time series forecasting models that can show periodicity as well as trend, which acts on the temporal dimension. However, these methods are still not able to output ϕ_i , and the degree of explanation is limited. In addition to the above problems, there is also a common problem in the field of XAI, which is listed as Problem 3 in this work.

Problem 3. *Lack of application scenarios.*

In works involving XAI, usually, explanation results are only presented as an innovation. In a large number of previous works, the application scenarios of XAI are only mentioned in the introduction part, including helping users trust the model and developers debugging the model, but these scenarios are not implemented.

Overall, on the one hand, the current model-agnostic method with high generality cannot fully explain the time series forecasting task, that is, it cannot realize the explanation of the temporal dimension. On the other hand, the model-specific method that can achieve temporal dimension explanation to a certain extent also has limitations, and its low generality also increases the cost of use. Therefore, a general XAI approach to explainable time series forecasting is needed, given the increasing importance of ML and DL in time series forecasting.

ShapTime¹ realizes the attribution of time by calculating the Shapley Value [5] on the temporal dimension, and finally outputs the importance of time itself ϕ_i . Therefore, ShapTime can realize the explanation in the temporal dimension, and it belongs to the model-agnostic method, which means that it can be deployed on any forecasting model at a lower cost. The foundation of ShapTime is the Shapley Value, which is the basis for numerous attribution methods including SHAP. Shapley Value comes from cooperative game theory, which studies

¹ See <https://github.com/Zhangyuyi-0825/ShapTime>.

how to reasonably distribute the benefits to the players in the alliance, and it has been proved to have some good properties. Therefore, in recent years, the development of XAI methods around Shapley Value is trying to become a stable path [6], and our ShapTime is exploring explainable time series forecasting as a branch on this path. Its contributions include:

- It realizes time attribution in the temporal dimension, that is, the importance of time itself ϕ_i can be obtained
- As a highly general model-agnostic method, it can be deployed in any forecasting model
- Its explanation results can be used as a guide to improve the forecasting performance of the model.

Subsequently, in Sect. 2, related work is summarized. In Sect. 3, the details of ShapTime are described. In Sect. 4, the experimental results are presented, including explanation results in the temporal dimension, forecasting performance improvements, and evaluation metrics. Section 5 is the conclusion.

2 Related Works

Previous approaches to explainable time series forecasting can be divided into three categories: Model-Agnostic methods, Model-Specific methods and Hybrid methods.

Model-Agnostic methods. It is realized by perturbing the input data set and inducing the change of the output, and finally attributing this change to the input features, so as to realize the explanation of the model. This is one approach that is widely used compared to others because of its generality(e.g. [7–11]). However, this type of XAI method was originally developed based on classification and regression tasks, so they often output feature importance instead of the importance of time itself, that is, they cannot output ϕ_i .

Model-Specific methods. This is a specific method for the development of time series forecasting models, that is, to embed the interpretation function into the model (e.g. [12,14]), in order to achieve better interpretation effect for time series forecasting. Although some works pay attention to the explanation of the temporal dimension (e.g. ([13,15])), they are still based on features, that is, the XAI approach outputs the feature importance at each moment and stitches them together to achieve the explanation of the temporal dimension. Strictly speaking, such an explanation also fails to output ϕ_i , and seriously lacks generality.

Hybrid methods. This is a approach to achieve explanation effects by hybridizing modules with a certain degree of interpretation function into time series forecasting model. The most representative approach is to hybridize the Attention Mechanism into the time series forecasting model (e.g. [16–18], and achieve the explanation through the explainability of the Attention Mechanism. Some works have discussed the explainability of the Attention Mechanism. Even though there are some controversies (e.g. [19]), researchers still hold a positive attitude towards its explainability (e.g. [20,21]. The Hybrid methods suffers from

the same problem as above approaches, that is, even if there is an explanation in the temporal dimension, this explanation is dependent on features. On the other hand, it also requires the development of new models and thus lacks generality.

3 Proposed Framework

3.1 Shapley Value for XAI

Shapley Value is one of the classic theories in cooperative games, which aims to distribute the benefits fairly to the players in the alliance. There is the correspondence between Shapley Value and model explanation: the features used for training correspond to “players”, and the model’s predictions correspond to “revenues”. Therefore, the assignment achieved by Shapley Value can attribute the prediction result to the features, that is, the contribution (importance) of each feature to the prediction result. The assignment is achieved by the following formula:

$$k_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} \times (v(S \cup \{i\}) - v(S)). \quad (1)$$

where N is the set of all players (features) (1,2,3...i...n), which is the complete set; S is a subset of N , in which removes the explained feature i , with a total of 2^N ; v is the gain function ($v(S) = E_{\hat{D}} [f(x) | x_S]$, where \hat{D} is the empirical distribution of the training data and f is the black-model).

The reason why Shapley Value has become the basis of many XAI approaches is that it has some desirable properties, including: Efficiency, Symmetry, Linearity and Null player, and among many attribution methods, it is the only mapping ($v : 2^N \rightarrow \mathbb{R}$) that can satisfy the above properties.

Efficiency: The sum of all feature attributions is equal to the predicted result of the model (the total revenue of the alliance), which is a prerequisite for the establishment of the attribution method.

$$\sum_{i \in N} k_i = v(N) \quad (2)$$

Symmetry: If feature i and feature j can be substituted for each other, then their attribution results are also equal. This is the natural property that should be possessed as the attribution method.

$$v(S \cup \{i\}) = v(S \cup \{j\}) \quad (3)$$

Linearity: If there is the combination of two predictive models, then the contribution of the features within the combination should be equal to the sum of the contributions from the respective models. This property opens up possibilities for Shapley Value applications within hybrid frameworks.

$$k_i(v + w) = k_i(v) + k_i(w) \quad (4)$$

$$k_i(av) = ak_i(v) \quad (5)$$

Null player: If the value of a feature is meaningless (including null values or all equal values), then the feature will not have any impact on the attribution process.

$$v(S \cup \{i\}) = v(S) \quad (6)$$

Shapley Value has two key elements: “player” and “gain function”. When both are determined, Shapley Value has the possibility to be calculated theoretically. Accordingly, Shapley Value is the attribution of “player”.

3.2 ShapTime

In order to achieve attribution on the temporal dimension, time points should be considered as “players”. Corresponding to the time series forecasting model, y_{t-i} is regarded as a “player”, and after defining the gain function, the attribution of y_{t-i} , namely ϕ_i , can be realized through Shapley Value. The modified formula (ShapTime) is:

$$\begin{aligned} \phi_{t_i} = & \sum_{S' \subseteq T \setminus t_i} \frac{|S'|!(|T| - |S'| - 1)}{|T|!} \\ & \times (v'(S' \cup t_i) - v'(S')). \end{aligned} \quad (7)$$

where t_i is super-time (players), T is the set of all super-time. S' is a subset of T , in which removes the explained super-time t_i , with a total of 2^T . v' is the gain function of ShapTime.

Super-Time The so-called temporal attribution is to attribute the forecasting results to time, which means treating the time points as “players”, which will inevitably cause the dimension explosion and the system crash. A similar problem is encountered in the XAI approach for image recognition—a large number of pixels makes the computational cost increase exponentially. In order to solve this problem, [4] proposed the concept of super-pixel, which is to cluster pixels with high similarity into a whole, and then participate in the calculation of pixel contribution, thereby greatly reducing the calculation cost.

In view of the similarity of the problem, we refer to his method to construct super-time in the temporal dimension, that is, the data set is divided into n super-times t_i according to the temporal dimension.

$$t_i = \{y_{t-i}, y_{t-i-1}, y_{t-i-2}, \dots\} \quad (8)$$

In ShapTime, super-time is equivalent to “player”. Crucially, although super-time controls the computational cost within an acceptable range, considering the complexity of $O(2^n)$ in the ShapTime, we recommend that n not exceed 10 or 11. Algorithm 1 shows the construction process of Super-time.

Algorithm 1 Super-Time

Input: Original data set: X
Parameter: The number of super-time: n
Output: All the super-time t_i

- 1: Let $L = \text{int}(\text{length}(X)/n)$.
- 2: Let $\text{start} = \text{length}(X) - L \times n$.
- 3: Let $X_{\text{used}} = X[\text{start} :, :]$.
- 4: **for** i in range (n) **do**
- 5: $t_i = X[i \times L : (i + 1) \times L]$.
- 6: **end for**
- 7: **return** all the t_i

Algorithm 2 Gain Function of ShapTime

Input: Explained model: f ; Super-time: t_i
Output: All the gain value: v'

- 1: Let $S' \subseteq T \setminus t_i$.
- 2: Let $T = \{t_0, t_1, \dots, t_n\}$
- 3: **for** S' in T **do**
- 4: **if** $\text{length}(S') == 1$ **then**
- 5: $S' = \{t_i\}$.
- 6: $v' = \text{sum}(f(t_i))/\text{length}(t_i)$.
- 7: **else**
- 8: $S' = \{t_i, t_j, \dots\}$.
- 9: $S_c = \text{concat}(t_i, t_j, \dots)$
- 10: $v'(S_c) = \text{sum}(f(S_c))/\text{length}(S_c)$.
- 11: **end if**
- 12: **end for**
- 13: **return** 2^T gain values: v'

Gain Function In ShapTime, the attributed object is super-time, which is the collection of time points within a period of time, that is, the “player” is no longer a time point at this time, but a time period. Correspondingly, our forecasting target y_t is also the collection of time points. However, the calculation of Shapley Value requires that for each “player” combination, the corresponding gain function outputs a value. Therefore, here, the averaged model f forecasting results are taken as the gain function v' of ShapTime (lines 6 and 10 in Algorithm 2). The formula is:

$$v'(S') = \text{avg}(f_{S'}(x_{S'})) \quad (9)$$

Eventually, the gain value v' of all combinations of super-time will be obtained, a total of $2^{|T|}$. In this way, according to Eq. (7), the forecasting result can be attributed to each super-time t_i , that is, ϕ_{t_i} , so as to realize the explanation in the temporal dimension.

Performance Improvement Even though research on explainability has received increasing attention in recent years, however, in general, explanation

results are simply presented as samples. This phenomenon does not only occur within the domain of explainable time series forecasting, but is the pervasive problem for the entire domain of Explainable AI (Problem 3). To explore the application scenarios of the XAI approach, we try to use ShapTime's explanation results as a guideline, aiming to improve the performance of time series forecasting.

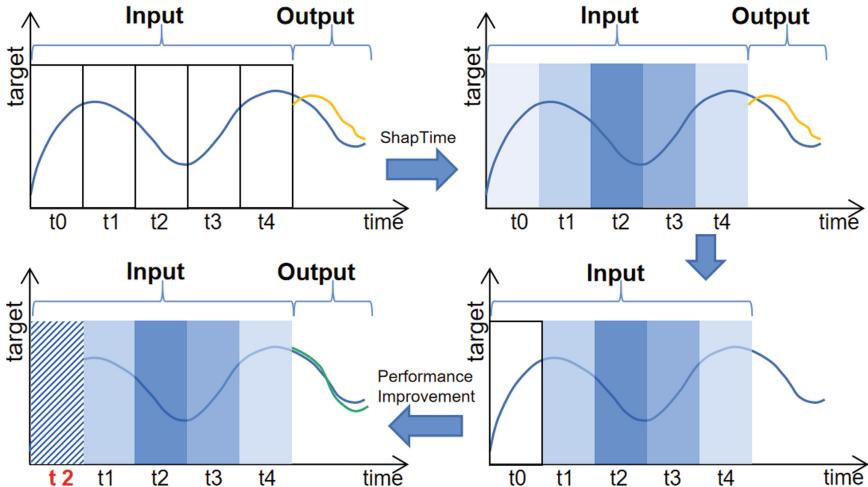


Fig. 1. ShapTime-based performance improvement process. The Blue line represents the original data set, and the orange line represents the forecasting results of the model. The blue background represents the explanation result of shaptime. The darker the color, the more important the t_i is, and vice versa. After obtaining the importance in the temporal dimension, the t_i with the lowest value is replaced by highest one. The green line represents the improved forecasting results

First of all, the input data set is divided equally into several super-times, and then, the importance of each super-time is calculated using ShapTime and visualized by the heat map, where the darker color means more important and the lighter color is less important. The way to improve performance is to replace low-importance t_i with high-importance t_i , and use this newly constructed data set to retrain the model. The reason for constructing such the improvement scheme (Fig. 1) is that ShapTime based on Shapley Value is the model-agnostic method, so its operation logic is based on the data set. In detail, the disturbance of the input data will change the forecasting result, and this type of method realizes attribution through this change. In fact, $(v'(S' \cup t_i) - v'(S'))$ represents this change. Therefore, dataset-specific refinements are naturally adopted when utilizing the explanation results output by this method.

Evaluation Criteria Currently, there is the lack of recognized correct labels in the XAI research field, which poses challenges for the evaluation of XAI methods. On the other hand, our ShapTime is the model-agnostic method for the explanation of the temporal dimension. In the field, this form of explanation is scarce, resulting in the lack of objects for comparison, so evaluation by contrast is difficult to achieve. Therefore, we propose some reasonable evaluation criteria to evaluate ShapTime effectively to a certain extent. It also provides a basic benchmark for subsequent research.

Property 1. *Under the premise of using the same data set, the explanation results of the XAI approach to similar types of models should be consistent to a certain extent.*

Property 2. *The XAI approach should be subject to sensitivity analysis, that is, when vital “players” are perturbed, there is the significant drop in forecasting performance.*

XAI’s evaluation criteria are critical for users to build trust in machines. The most important of these is that the explanation results of XAI should maintain a certain degree of stability (Property 1), which is the basis for building trust. On this basis, the XAI method also guarantees validity (Property 2), that is, when important “players” are changed, the performance of the model will decline significantly. This can prove to a certain extent that the explanation results of this XAI approach are valid.

4 Experiments

In order to verify the practical application ability of ShapTime, we selected 5 real data sets, including: Climate Data, Energy Consumption, Solar Power, Gold Price, Tesla Stock. Among them, the first three are periodic data, and the last two are trend data, so as to test the performance of ShapTime under different data types.

On the other hand, there are two types of black box models involved in training: Boosting model and RNN-based model. The former consists of XGBoost and LightGBM; the latter includes RNN, LSTM, GRU (RNN-based) and Bi-RNN, Bi-LSTM, Bi-GRU(Bi-RNN-based). These forecasting models basically include the current mainstream methods in competition and practice. Their forecasting performance measures are shown in Table 1.

4.1 Explanation

We applied ShapTime on the above eight models and generated the explanation results (Fig. 2: For the sake of brevity, we only show the explanation results of ShapTime for XGBoost and LSTM in five data sets, and the complete explanation results are in the Github).

Table 1. Metrics to forecasting performance. forecasting models are divided into three categories: boosting; RNN-based; Bi-RNN-based. r^2 and MSE are used as forecasting performance metrics, for each model, the first row is r^2 and the second row is MSE . Within each class of models, the best model is marked (in Bold); among all models, the best model is additionally marked (in Brackets). As can be seen from the table, Bi-RNN-based performs best in trending data (Gold Price and Tesla Stock), and boosting performs best in periodic data (Daily Climate, Energy Consumption, and Solar Generation)

Data	Climate	Energy	Gold	Solar	Tesla
XGB	0.7756	0.7570	0.7988	0.9307	0.7632
	(0.0081)	0.0095	0.0006	0.0055	0.0029
LGB	(0.7814)	(0.8290)	0.7779	0.9322	0.7777
	0.0082	(0.0071)	0.0006	0.0053	0.0023
RNN	0.7170	0.6574	0.7018	0.9587	0.8133
	0.0104	0.0128	0.0011	0.0051	0.0022
LSTM	0.7507	0.6249	0.8182	0.9544	0.8586
	0.0081	0.0153	0.0005	0.0057	0.0017
GRU	0.6719	0.7130	0.7096	(0.9661)	0.8176
	0.0117	0.0119	0.0009	(0.0035)	0.0025
Bi-R	0.7257	0.6804	0.8048	0.9312	0.7990
	0.0102	0.0120	0.0008	0.0099	0.0029
Bi-L	0.6903	0.7122	(0.8791)	0.9326	(0.8689)
	0.0103	0.0119	(0.0005)	0.0082	(0.0016)
Bi-G	0.7136	0.5927	0.8664	0.9481	0.7055
	0.0109	0.0154	0.0006	0.0056	0.0045

In explanation results, the Daily Climate dataset (Fig. 2a–f) is taken as the example. In this example, the number n of super-time is set to 8, then the corresponding theoretical modified time series model is:

$$y_t = \phi_{t_0} t_0 + \phi_{t_1} t_1 + \cdots + \phi_{t_7} t_7 + \omega_t \quad (10)$$

After the training is completed, we use ShapTime to explain XGBoost and LSTM, so as to attribute the forecasting results of the model to each super-time, and then get each ϕ_{t_i} value, and display them visually through the heat map. We can clearly see that both models capture the most recent super-time t_7 as the most important input. In the Energy Consumption dataset, the two models regard t_7 as the most important super-time; in Gold Price it is t_9 ; in Tesla Stock it is t_{10} . Different situations appear in Solar Generation. The super-time captured by the two models is different. XGBoost captures t_0 as the most important, while LSTM captures t_5 .

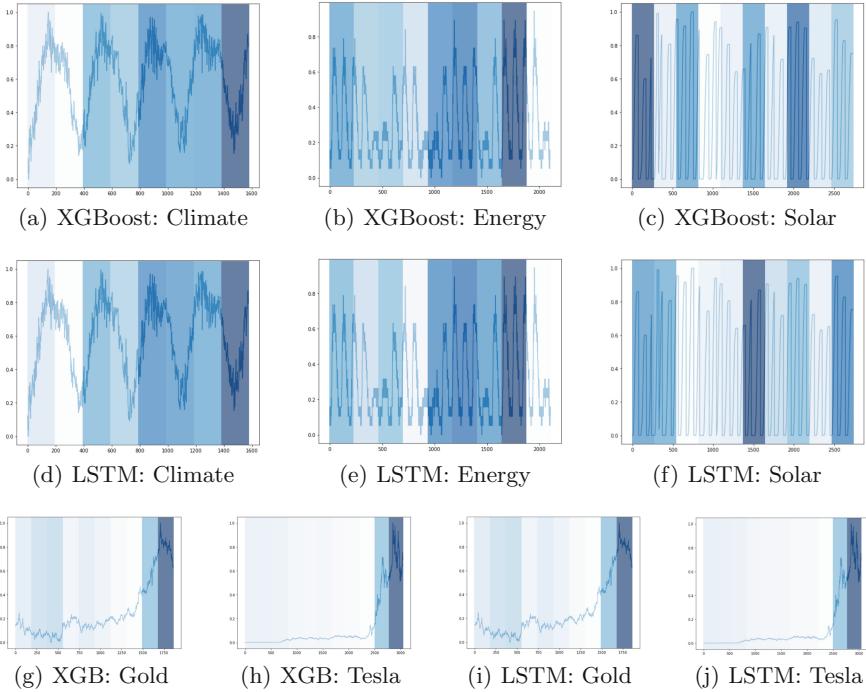


Fig. 2. Example of explanation results from shapetime. This is shapetime’s explanation of XGBoost and LSTM. The heat map is used to visualize the explanation results of the input data. The darker the color, the more important the super-time. This means that the model pays more attention to this super-time during the training process. It can be seen from figures that the explanation of trend data by LSTM and XGBoost is basically consistent, and there is a certain degree of difference in the explanation of periodic data, especially in the solar generation data set

Looking at all the 5×8 explanation results, the most important super-time captured by all models is the last one in the trend data, but in the periodic data, there is no obvious rule. However, if we analyze the forecasting performance of the model (Table 1), we can still find some potential rules, that is, the Boosting model is more suitable for periodic data, while the RNN-based model is more suitable for periodic data when the explanation results are generally consistent.

4.2 Evaluation

According to the above evaluation criteria, we evaluate the explanation results of ShapTime, and the evaluation examples are shown in Fig. 3. Similarly, for simplicity, we only display the evaluation results on the Boosting model. Figure 3a-d represent the explanations of ShapTime for XGBoost and LightGBM, respectively. It can be intuitively seen that the explanation of ShapTime can be maintained all the time for the same type of models, which is a common pattern

in all the explanation results. This is key to the trust of users from all industries. Imagine if the explanation results change frequently during use, which will directly lead to user distrust or even abandonment.

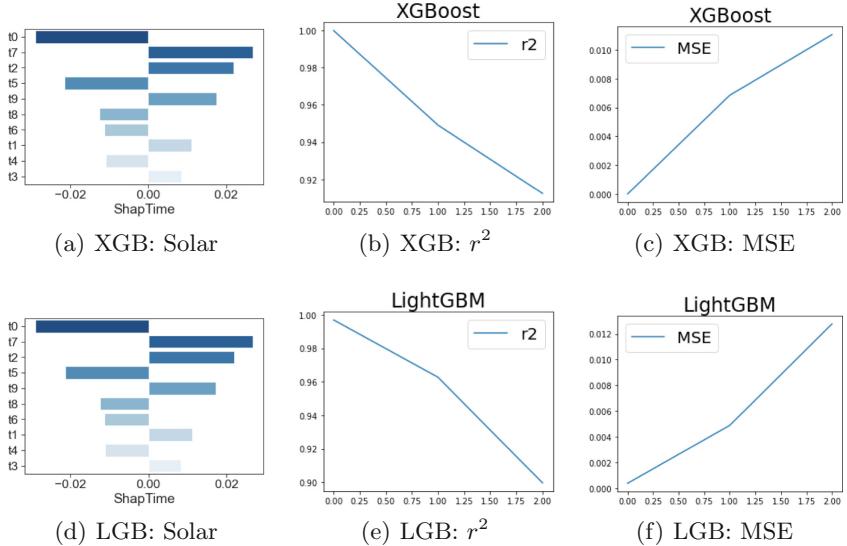


Fig. 3. Evaluation Example of ShapTime Explanation Results. **a, b** Satisfy Property 1, **b, c, e, f** Satisfy Property 2

This is exactly the purpose for which Property 1 was defined. However, in case of different types of models, in principle, differences in the explanation results need to be allowed, since models of varying architectures are not equally sensitive to the nature of the distribution of the data. As in Fig. 2c–h, the explanation results present differences in the various models.

On the other hand, theoretically, if we perturb the training data according to the explanation results, i.e., the important super-time is replaced with the least important, then the forecasting performance of the model will be significantly decreased. In this example, the most important t_0 is replaced with the least contributing t_3 , as well as t_7 is replaced with t_4 . Correspondingly, the r^2 and MSE of XGBoost and LightGBM exhibit significant and gradual performance degradation (Fig. 3b, c, e, f). This evaluation schema is the relatively classical evaluation method in the field of XAI, and it is also known as sensitivity analysis. In this work, it is summarized in Property 2.

4.3 Improvement

To create valuable XAI application scenarios, we use the explanation results of ShapTime as our guide for achieving improved forecasting performance, and the

improvement process is shown in Fig. 1. Table 2 shows the improved performance metrics (compared to Table 1), and the results show that Bi-RNN-based and Boosting still maintain their original advantages in trending and periodic data, respectively.

Table 2. ShapTime guided forecasting performance improvement. r^2 and MSE are used as forecasting performance metrics, for each model, the first row is r^2 and the second row is MSE

Data	Climate	Energy	Gold	Solar	Tesla
XGB	(0.7898)	0.7975	0.8880	0.9521	0.8198
	0.0075	0.0093	(0.0004)	0.0038	0.0021
LGB	0.7847	(0.8445)	0.8715	0.9573	0.8038
	0.0082	(0.0069)	0.0004	0.0033	0.0022
RNN	0.7255	0.7572	0.8535	0.9701	0.8385
	0.0096	0.0116	0.0006	0.0024	0.0020
LSTM	0.7751	0.6412	0.8287	0.9597	0.8812
	(0.0070)	0.0142	0.0005	0.0047	0.0016
GRU	0.7470	0.7376	0.7825	0.9740	0.8498
	0.0092	0.0101	0.0006	0.0026	0.0020
Bi-R	0.7428	0.7425	0.8980	0.9693	0.8932
	0.0082	0.0101	0.0004	0.0036	0.0016
Bi-L	0.7532	0.7604	0.9002	0.9742	(0.9080)
	0.0084	0.0100	0.0004	0.0028	(0.0013)
Bi-G	0.7412	0.7092	(0.9043)	(0.9840)	0.8505
	0.0100	0.0112	0.0004	(0.0014)	0.0021

The measure of the degree of improvement is presented in Fig. 4a, which shows the average degree of improvement for each type of model in each data set. The percentages are calculated based on the degree of improvement in Table 2 compared to Table 1. It can be intuitively seen that ShapTime shows the maximum improvement for Bi-RNN-based, and the most significant improvement for Solar Generation. Overall, the effect of ShapTime on Boosting is less significant than that of other types of models. Specifically, ShapTime has the most significant improvement on Bi-GRU. In the original forecasting performance (Table 1), Bi-GRU does not possess the best performance in all datasets, while after the improvement (Table 2), the best forecasting performance of Gold Price and Solar Generation appears in Bi-GRU.

The partial visualization of the improvement effect is shown in Fig. 4b-e. From the forecasting effect, the forecasting results after ShapTime improvement still maintain approximately the same pattern as the original forecasting results, however, they can be much closer to the original data, thus achieving the performance improvement.

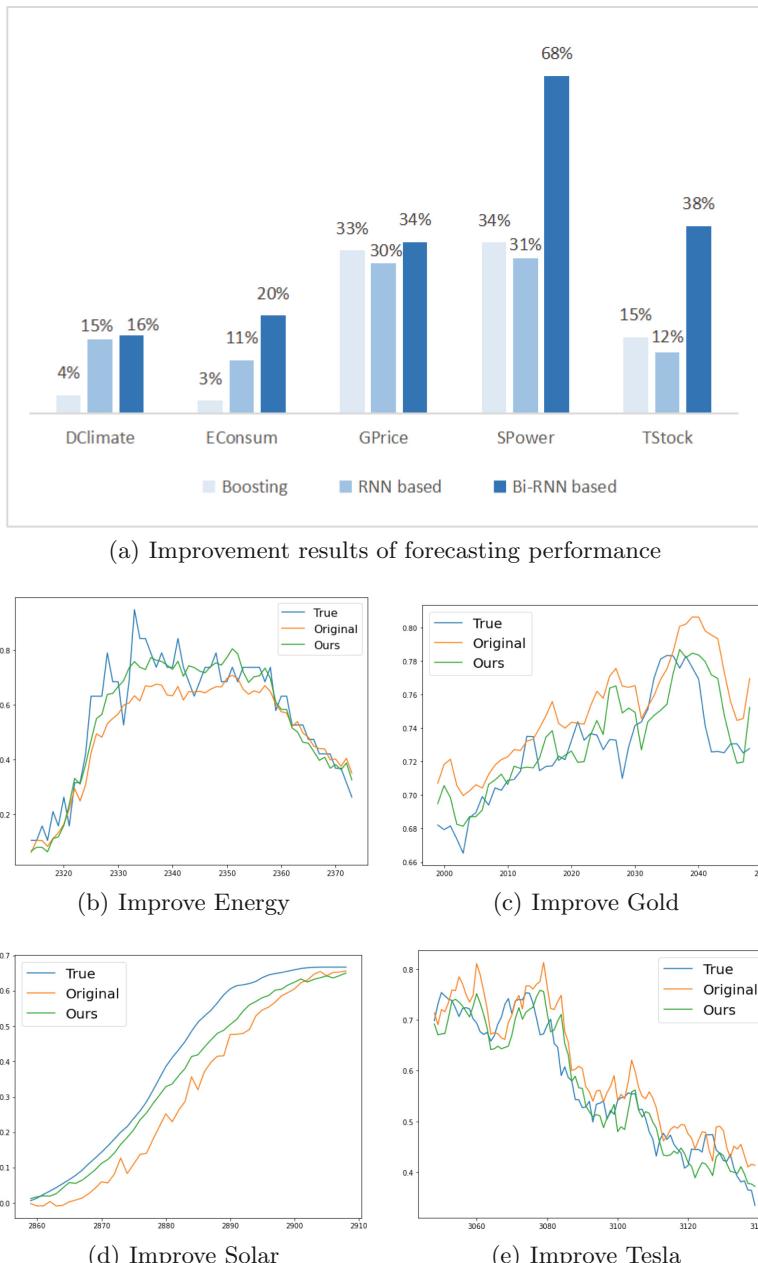


Fig. 4. Visualization of performance improvement. “Original” is the output of the original forecasting model, and “Ours” is the output of the shapetime-guided forecasting model

5 Conclusion

In this work, the XAI approach specifically oriented to time series forecasting is developed, and we name it ShapTime since its computation is based on Shapley Value. It enables attribution in the temporal dimension, thus explaining the importance of time itself, which differs from previous works.

With the explanation of ShapTime, we are able to understand the forecasting model to some extent. In trending time series data, all models focus on the most recent data as the most important learning object, while in periodic time series data, this pattern does not obviously exist and different models do not necessarily focus on the same time period for learning.

On the other hand, with the help of ShapTime explanation, we have been able to achieve the improved performance in time series forecasting. By replacing data in times of low contribution with high ones, performance improvements can be achieved to some extent. The improved performance metrics show that the Boosting model and the Bi-RNN-based model are still able to maintain their original advantages in periodic data and in trending data, respectively. In summary, ShapTime showed the most significant improvement for the Bi-RNN-based model, with the average improvement of 35%. In particular, ShapTime showed the most significant improvement for the Bi-GRU model, with the 73.87% improvement in the Solar Generation dataset.

Acknowledgment. This work was supported by Saint-Petersburg State University, project ID: 94062114.

References

1. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: The M4 competition: results, findings, conclusion and way forward. *Int. J. Forecast.* **34**(4), 802–808 (2018)
2. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: The M5 competition: background, organization, and implementation. *Int. J. Forecast.* **38**(4), 1325–1336 (2022)
3. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. *Adv. Neural Inf. Process. Syst.* **30** (2017)
4. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should i trust you? Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144 (2016)
5. Shapley, L.S.: A value for n-person games. *Classicsn Game Theory* **69** (1997)
6. Sundararajan, M., Najmi, A.: The many Shapley values for model explanation. In: International Conference on Machine Learning, pp. 9269–9278. PMLR (2020)
7. Pan, Q., Hu, W., Chen, N.: Two birds with one stone: series saliency for accurate and interpretable multivariate time series forecasting. In: IJCAI, pp. 2884–2891 (2021)
8. Ozyegen, O., Ilic, I., Cevik, M.: Evaluation of local explanation methods for multivariate time series forecasting (2020). [arXiv:2009.09092](https://arxiv.org/abs/2009.09092)

9. Zhang, Y., Petrosian, O., Liu, J., et al.: FI-SHAP: explanation of time series forecasting and improvement of feature engineering based on boosting algorithm. In: Intelligent Systems and Applications: Proceedings of the 2022 Intelligent Systems Conference (IntelliSys), vol. 3, pp. 745–758. Springer International Publishing, Cham (2022)
10. Jabeur, S.B., Mefteh-Wali, S., Viviani, J.L.: Forecasting gold price with the XGBoost algorithm and SHAP interaction values. *Ann. Oper. Res.* 1–21 (2021)
11. Zhang, Y., Ma, R., Liu, J., et al.: Comparison and explanation of forecasting algorithms for energy time series. *Mathematics* **9**(21), 2794 (2021)
12. Oreshkin, B.N., Carpov, D., Chapados, N., et al.: N-BEATS: Neural basis expansion analysis for interpretable time series forecasting (2019). [arXiv:1905.10437](https://arxiv.org/abs/1905.10437)
13. Wang, J., Wang, Z., Li, J., et al.: Multilevel wavelet decomposition network for interpretable time series analysis. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2437–2446 (2018)
14. Shen, Q., Wu, Y., Jiang, Y., et al.: Visual interpretation of recurrent neural network on multi-dimensional time-series forecast. In: 2020 IEEE Pacific Visualization Symposium (PacificVis), pp. 61–70. IEEE (2020)
15. Guo, T., Lin, T., Antulov-Fantulin, N.: Exploring interpretable lstm neural networks over multi-variable data. In: International Conference on Machine Learning, pp. 2494–2504. PMLR (2019)
16. Lim, B., Arik, S.Ö., Loeff, N., et al.: Temporal fusion transformers for interpretable multi-horizon time series forecasting. *Int. J. Forecast.* **37**(4), 1748–1764 (2021)
17. Ding, Y., Zhu, Y., Feng, J., et al.: Interpretable spatio-temporal attention LSTM model for flood forecasting. *Neurocomputing* **403**, 348–359 (2020)
18. Zhou, B., Yang, G., Shi, Z., et al.: Interpretable temporal attention network for COVID-19 forecasting. *Appl. Soft Comput.* **120**, 108691 (2022)
19. Jain, S., Wallace, B.C.: Attention is not explanation (2019). [arXiv:1902.10186](https://arxiv.org/abs/1902.10186)
20. Serrano, S., Smith, N.A.: Is attention interpretable? (2019). [arXiv:1906.03731](https://arxiv.org/abs/1906.03731)
21. Wiegreffe, S., Pinter, Y.: Attention is not not explanation (2019). [arXiv:1908.04626](https://arxiv.org/abs/1908.04626)



Command Line Interface Risk Modeling

Anthony Faulds^(✉)

Microsoft Corporation, Redmond, WA 98052, USA
tonyfaulds@microsoft.com

Abstract. Protecting sensitive data is an essential part of security in cloud computing. However, only specific privileged individuals are permitted access to view or interact with this data; therefore, it is unscalable to depend on these individuals also to maintain the software. A solution is to allow non-privileged individuals access to maintain these systems but mask sensitive information from egressing. To this end, we have created a machine-learning model to predict and redact fields with sensitive data. This work concentrates on Azure PowerShell as the conduit connecting to a secure cloud environment from outside. However, it shows how it can be applied to other command-line interfaces and APIs. Using the F5-score as a weighted metric, we demonstrate through empirical, quantitative measures that different transformation techniques map this problem from unstructured data to the well-researched area of natural language processing. These transformations are compared using different models to demonstrate the robustness of the techniques. These findings are generally helpful across administrators implementing security systems and can be used in various systems to reduce information leaving a defined boundary. Specifically, this research demonstrates a solution that allows systems to scale human code development and maintenance without oversight having to be scaled proportionally.

Keywords: Command-line interface · Sensitive information · Secure · Machine learning · Bag of words · Term frequency-inverse document frequency · Word embedding

1 Introduction

Sovereign Cloud is a unique offering by Microsoft that allows customers access to the features of Azure Cloud but with isolated physical resource infrastructure. Some Sovereign Clouds have varying levels of internet access depending on the customer's risk aversion. Each customer defines the requirements of a user to gain access to the system. These requirements can range from receiving certification to having government clearance. Users who fulfill the appropriate requirements and have administrator-level access are called operators. Maintaining the Azure Cloud requires directly responsible individuals (DRI) who can support and debug issues related to different Azure services.

Unfortunately, the number of individuals that are both a DRI, have Azure expertise, and an operator, have the necessary credentials is a small group of people. This group of people is difficult to hire, train and grow because clearances can take months, and learning Azure enough to support a systems infrastructure can take months. One way to solve this is to separate the operator and DRI roles. Figure 1 shows the split of DRI and operator and a gatekeeper that filters egress data so that only non-sensitive information can leave the Sovereign Cloud. As a result, DRIs can increase as Microsoft hires more developers in the Azure infrastructure. In addition, DRIs can interact directly with the Sovereign Cloud, but sensitive information must be redacted from their interactions. To solve the issue of redacting sensitive information, the Command And Query (CNQ) team developed a command-line interface (CLI) to allow DRIs to send commands to the Sovereign Cloud while using a combination of ML and operator feedback to redact sensitive information returned in CLI responses. This solution enables the number of DRIs to scale without requiring the number of operators to increase at the same rate or requiring that DRIs also be operators.

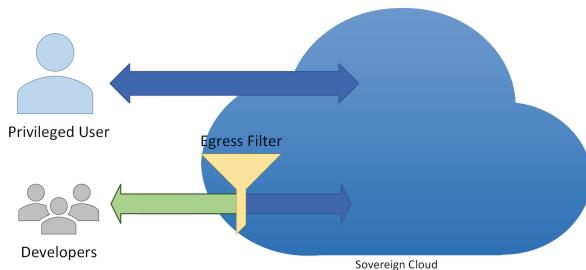


Fig. 1. Sovereign secure access

PowerShell is a CLI that enables scripting and automation. Azure Cloud has created PowerShell commands that allocate resources, query resources, connect to resources, deletes resources, and more.

Sensitive information is data in the Sovereign Cloud or how to connect to the data, connection strings, and passwords. Each Sovereign Cloud customer has its definition of sensitive information, but some pieces of data are universally sensitive across all customers.

We show how we can map this problem to more widely researched NLP problems. The result is that each feature becomes a document with words inside that document. CLI risk prediction is a unique problem, and this work explores different ways to map the CLI command and its response to machine learning problems. Although we show how to map CLI data to documents, similar to the data sets in the NLP modeling space, CLI is different because it does not hold to traditional grammar rules and requires some brevity. For example, variable names and commands are long enough to be self-describing but shorter than it is context compared to a typical sentence.

Although this technique is specific to Powershell, the same process can apply to other CLIs and APIs. For example, changing from binary class to multiclass classifier can map this problem to determine various types of Personally Identifiable Information (PII) within command responses. In addition, we can adjust from predicting the probability of sensitive data to predicting the likelihood that a command can cause a system outage.

2 Problem Definition

In security, it is crucial to constrain the access or use of sensitive information. This information should have limited transmission or visibility except for those who are allowed to see this data. Masking this data from the view of non-qualified individuals is one approach to security. It will enable non-qualified individuals or systems to maintain secure systems but limits the information seen.

For Sovereign Cloud, most clients have similar definitions of sensitive information. For example, Key Vault information is considered sensitive information. Everything from passwords, connection strings, and certificates are examples of the many things a Key Vault may store. PowerShell has commands that can create resources that respond with connection string information and also have commands to fetch these fields specifically.

As Sovereign Cloud customers evolve, things like the server names, geographical locations, or PII included in the system will fall under sensitive information in the future.

For this work, we want to map the PowerShell command-line redaction to a well-known problem. Since there are no numbers, it seems best to map this work to an NLP problem. Furthermore, ML models predict sensitivity per field to allow fine-grain resolution for redacting fields. Therefore a single PowerShell command is mapped to multiple predictions, one per response.

3 Related Work

There are many approaches to NLP problems. Even early approaches converted words into numbers, or vectors for numerical analysis [4, 6, 14]. For this work, we compare several of those to show the best combination.

The bag of words (BOW) method is a simplified text representation [3, 16]. It ignores word order and grammar and stores the count of words as a feature. The word counts can use all vocabulary in the training data set or a specified dictionary. Next, BOW converts a string of words into an integer vector with counts of the word in the document.

Term frequency-inverse document frequency (TF-IDF) extends BOW from word count to include information about how many words are in a document and how frequently the words are found across documents [5, 10, 12, 15]. Beel, Gipp, Langer, and Breitinger have documented the popularity and usefulness of this algorithm [1]. TF-IDF is a combination of two components, TF and IDF. For TF, this is the ratio of the count of a single word in a document over the number

of words in the document. Closer to 1.0 indicates how prevalent the word is in a document compared to other terms in that document. IDF is the inverse of how dominant a word is in all documents. For example, a word like “the” being in all documents would give it a unary IDF of 1, whereas “contemplate” being used in 1 of 1000 documents would give it a unary IDF of 1000. Multiplying these two numbers together results in TD-IDF, a combination of word frequency relative to document length and inverse popularity of the word. This combination results in large TD-IDF values only when a word is rare or used a lot in a single document. TF-IDF is used as inputs or features to different techniques, from clustering to visual search of videos [9, 11, 13].

Another approach that helps aid in understanding similar words is word embedding. Mikolov, Chen, Corrado, and Dean developed a method of using neural networks to project words into a vector space [7, 8]. This technique transforms words into a real-value vector in which words close in vector distance have similar meanings. For example, BOW and TF-IDF treat “secret” and “secrets” as two different words. For this work, these words are the same. Therefore, word embedding provides a unique approach to encoding the information better. For this problem, we also want the model to generalize well. Our training set is a small fraction of the commands we want to predict. The naming convention of commands spans many teams. So it is vital to capture the information without needing to train on every variation of the word “secret”

4 Tokenizing the CLI

PowerShell commands are combined with response names and types to map this problem to what is known as a document in the NLP space. All three values are in Pascal Case, so we can separate words by the first capital letter, removing non-alphanumeric values. For example, “New-AzKeyVault” becomes “New Az Key Vault.” Powershell contains several variables with multiple upper-case characters in a row. For this, we split the upper case into one separate word. For example, complicated names like “New-AzVMConfig” become “New Az VM Config.” The result is that each feature becomes a document with words inside that document.

The system for PowerShell is an azure command that has a response. This response is a hierachal set of variables where each variable has a type and a value Table 1. For this work, we want to redact or allow each variable individually. Therefore, the information we have for each variable includes (1) variable name, (2) variable type, (3) variable value, (4) variable parent name hierarchy, and parent variable type hierarchy. The variable name is the name of the variable as assigned by the system. Variable type is the object type, such as int, float, string, or custom object. Finally, the variable value is the actual value returned and set in that variable.

Finally, to turn it into words and sentences, we implement tokenization on non-alpha numerics and go from lowercase to uppercase. Table 1 shows the original PowerShell command. This single command is mapped to 4 items for machine learning with features and labels, as shown in Table 2. Finally, Table

[3](#) demonstrates concatenating and tokenization to get something that looks like documents used for NLP.

Table 1. Example powershell command

```
Command:
Get-AzLocation

Response:
Location: eastasia
DisplayName: East Asia
Providers: {Microsoft.Devices,
Microsoft.Cache, ...}
```

Table 2. Example list of machine learning features

#	Command	Module	Field Name	Field Type	Parent Name	Parent Type
1	Get-AzLocation	Resources		PSResource-ProviderLocation		
2	Get-AzLocation	Resources	Location	String	PSResource-ProviderLocation	
3	Get-AzLocation	Resources	DisplayName	String	PSResource-ProviderLocation	
4	Get-AzLocation	Resources	Providers	System.Collections.Generic.List<string>	PSResourceProviderLocation	

For this work, we omit variable values. There are three reasons we overlook it in model training and prediction. The first is that the variable value can be any type. Therefore it may not be a string or number. Casting all these values to strings may be cumbersome, or setting up a model to handle the native type is difficult. Second, we are predicting whether the variable value is sensitive and do not want this value to egress the Sovereign Cloud. If we include this value in our training set, we limit where and how we can train the model. The training data will have sensitive data, so human interaction with the training data is not allowed. It makes the data more challenging to handle. Finally, the most sensitive data are passwords, which are strings of random characters. It is unlikely that the model will pick that up as a signal to indicate a password as much as a field name of “Password” or “Passwd.”

We desire to preprocess the data set to produce a similar data set to NLP problems. All the features are text and, if combined, resemble a sentence. In NLP, words are in a spoken or written language. For this work, we use words to mean the combination of letters determined by our tokenization process. Although

Table 3. Conversion of Commands to Documents

#	Document
1	Get Az Location; Resources; PS Resource Provider Location
2	Get Az Location; Resources; Location; String; PS Resource Provider Location
3	Get Az Location; Resources; Display Name; String; PS Resource Provider Location
4	Get Az Location; Resources; Providers; System Collections Generic List string; PS Resource Provider Location

similar to a sentence, it does not contain correct English grammar. The coding standard for variable names and types in PowerShell is Pascal Case. Pascal Case is where a variable name or type is a combination of words that join into one word, and the first letter is in uppercase while the rest are in lowercase. For example, a Pascal case for the phrase virtual machine name is VirtualMachineName. The initial preprocessing tokenizes between each lowercase letter and an uppercase letter.

5 NLP Techniques

5.1 Bag of Words

For this work, we explore five techniques for converting text to numbers for use in machine learning models. This technique creates a dictionary based on all the words in the training set. Each vocabulary item has a positional value. The transformation turns each item from a sentence to an array where the array is the vocabulary length, and the word count found in the document is assigned to the positional value.

Equation 1 shows the term frequency (tf) of each term (t) per document (d).

$$tf(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t'd}} \quad (1)$$

The advantage is that this transformation is easy to implement. It also reinforces items that have words used frequently. One downside of BOW is that document lengths can significantly affect counts. A lengthy document can naturally have more of a single word, even though that may not reinforce its importance in predicting that document. However, this does not significantly affect this problem because all documents are close to the same length. BOW also does not consider the ineffectiveness of frequently used words or a word's relative position.

5.2 TF-IDF

TF-IDF resolves some of the BOW disadvantages. This transformation works similarly to BOW but includes a factor that decreases the value of a word-based if used more frequently across all documents. Equation 4 shows the IDF based on the number of document occurrences of the term (n_t) and the number of documents (N).

$$N = |D| \quad (2)$$

$$n_t = |\{d \in D : t \in d\}| \quad (3)$$

$$idf(t, D) = \log \frac{N}{n_t} \quad (4)$$

$$tfidf(t, d, D) = tf(t, d) * idf(t, D) \quad (5)$$

Less frequently used TF-IDF still has the disadvantage of not considering word order.

5.3 Word Embedding

Word embedding (WE) is a technique that converts each word to a unit vector with a specified number of dimensions. The mapping of word embeddings results in vectors that are close in Euclidean distance having similar meanings. Since each word in order is converted to a vector, concatenating all the vectors results in a feature set sensitive to word order.

WE creates unique vectors for each word. The size of the vector can be tuned to get the best results. WE algorithm is shown in Fig. 1. Mikolov et al. [7] developed the Word2Vec algorithm.

Using a pre-trained Word2Vec model converts words into similar vectors that are not similar. For example, the initial for Azure, “Az,” and the initial for the identifier “Id” can be misinterpreted as Arizona and Idaho, respectively. A pre-trained Word2Vec model sees these are similar items and therefore assigns them similar vectors. To mitigate this, we trained a Word2Vec using all of the Azure Powershell commands and responses. Since this is more akin to unsupervised learning, we do not need labels. This trained model performs better and learns relationships with this specialized vocabulary in Azure Powershell.

Table 4 shows a length-3 vector used for word embedding and the euclidean distance of each word from the WE of “password.” Using WE, the word “passwords” is closer to “password” than to “private.” This demonstrates an advantage of WE in that similar words have similar vector representations.

The disadvantage of word embeddings is not being able to handle homonyms. Words spelled the same but with different meanings have the same encoding. Acronyms in variable names and commands cause similar issues. For example, VM could mean “virtual machine” for one PowerShell command and “virtual

Algorithm 1 WE Algorithm

input max words m , vector length per word l
output float matrix V

```

Initialize:  $v_k = 0 \forall k \in 1, \dots, m * l$ ,  $i = 0$ 
 $W$  = vector of words from tokenization of document
 $J \leftarrow \|W\|$ 
for  $j = 1, 2, \dots, J$  do
     $w \leftarrow Word2Vec(W_j)$ 
    for  $k = 1, 2, \dots, l$  do
         $v_{i+k} = w_k$ 
    end for
     $i \leftarrow i + 1$ 
    if  $i > l * m$  then
        END LOOP
    end if
end for

```

Table 4. WE Example

Word	WE 3	Distance
Password	[0.1, 0.2, 0.5]	0.00
Passwords	[0.2, 0.1, 0.4]	0.17
Private	[0.9, -0.1, 0.1]	0.94

memory” for another. In NLP problems, the word order in a sentence can help a model determine the word’s meaning. Also, using modeling for parts of speech help in this problem. These are not as helpful in this domain because commands and variable names while being self-descriptive, are not complete sentences, and there is no requirement to use correct grammar. Also, a system like PowerShell, and future CLIs that will be supported, are built by multiple teams without overarching coding standards.

5.4 BOW Per Feature

Another approach explored is using BOW per feature (BOW-PF). For example, the word “Connection” in the field name or parent field name is more important than the word “Connection” found in the command’s name. This hybrid approach adds a type of word order sensitivity. For this transformation, we train the BOW transformation on the original data set to use the entire training vocabulary. From there, we use BOW to transform each feature separately. Instead of using the semicolon concatenated document, we use the features from Table 2. This data creates six times as many features, but now BOW counts are per feature.

The BOW transform trains on all features as tokenized sentences. Apply the transform to each of the features separately. This results in transformed features

of the size: original feature size * vocabulary size (v). The first feature will be the count of the first vocabulary word in the first feature. The second feature will be the count of the second vocabulary word in the first feature. The $1+v$ feature will be the count of the first vocabulary word in the second feature.

Algorithm 2 BOW-PF Algorithm

- 1: *Parameters:* m = max words, C = vector of features each containing a document,
n = number of features
- 2: *Initialize:* $v_{i,j} = 0 \forall i \in 1, \dots, n; j \in 1, \dots, m * l, i = 0$
- 3: VOCAB = unique list of words across all features in the data set
- 4: $J \leftarrow \|C\|$
- 5: **for** $j = 1, 2, \dots, J$ **do**
- 6: **for** $k = 1, 2, \dots, n$ **do**
- 7: $w \leftarrow BOW(c_{j,k}, VOCAB)$
- 8: **for** $l = 1, 2, \dots, \|w\|$ **do**
- 9: $v_{j+k} = w_l$
- 10: **end for**
- 11: **end for**
- 12: **end for**

This transformation has similar advantages to BOW. Where it differs is the added advantage of separating word count per feature so that the model can learn the nuances of words in command name vs. variable name. However, a disadvantage is that it creates more features and can take an ML model longer to learn. In future work, this can be resolved by adding a feature selection step that could reduce the number of features before training a final model.

5.5 TF-IDF Per Feature

Similar to the BOW-PF, the TF-IDF per feature (TF-IDF-PF) trains the transform using the TF-IDF on the documents shown in Table 3. Then applies that transform to each feature separately. This process also has six times as many features. The advantages include TD-IDF's ability to scale word importance to word prevalence across documents.

In the future, we would like to explore reducing the features used for BOW-PF and TF-IDF-PF. One idea is to use BOW applied to the features as a sentence. Train using a tree model and use feature importance to reduce the vocabulary. Then apply the reduced vocabulary BOW transform to each feature separately. Doing this will decrease training time. It should also help with the generalization of the model. This technique excludes rare features that can make the model overfit.

Algorithm 3 TF-IDF-PF Algorithm

```

1: Parameters: m = max words, C = vector of features each containing a document,
   n = number of features
2: Initialize:  $v_k = 0 \forall k \in 1, \dots, m * l$ ,  $i = 0$ 
3: VOCAB = unique list of words across all features in the data set
4:  $J \leftarrow \|C\|$ 
5: for  $j = 1, 2, \dots, J$  do
6:   for  $k = 1, 2, \dots, n$  do
7:      $w \leftarrow TF - IDF(c_{j,k}, VOCAB)$ 
8:     for  $l = 1, 2, \dots, \|w\|$  do
9:        $v_{j+k} = w_l$ 
10:      end for
11:    end for
12:  end for

```

6 Experiments

6.1 Machine Learning Models

To compare different preprocessing steps, we want to pick a couple of different machine learning models to compare results. Some model types work differently in handling hundreds of features, which causes underfitting or overfitting.

Logistic Regression (LR) is a prime candidate as a baseline for comparison. It trains quickly relative to other types of models. It is also easier to analyze the relationship between features and the label. A more advanced method is AdaBoost Trees (BT) [2]. This method trains multiple trees on the training data, with each subsequent group of trees trained on a weighting based on incorrectly classified examples. The last method used is Neural Networks (NN). Each of these methods can be refined to build optimal F5-Scores. This work aims not to find the best modeling technique but to determine which transformation technique works well in each model type. The same hyperparameters are used across all transformations for each model type to compare transformation methods accurately.

6.2 Evaluation Data Sets

We use a manually labeled data set containing over 60,000 entries derived from 1,420 commands, as shown in Table 5. Those commands make up 4% of all Azure PowerShell commands. In addition, we labeled items that allow for authentication, authorization, or specifically designated sensitive for this data set. This includes connection strings, certificates, passwords, and values stored in a Key Vault. This does not include IP, URLs, or ports. These items aid in the connection to a resource but do not provide authentication or authorization.

Some libraries have more redacted fields.

Table 5. Training Data Statistics

Name	Count
Azure command coverage	1,420
Training size	62,579
Redacted	2,155

6.3 Metrics

The model should attempt to redact all sensitive fields. It is understood by the customer and us that this is impossible, whether it be because of statistically based models that could miss some sensitive information or human mistakes. To that end, we work with the customer to provide the best solution and work to asymptotically approach 100% recall in partnership with the customer. False positives are not a major issue. If something is redacted that should not be, it could hinder the DRI's work. False negatives are a bigger issue. They allow DRIs to see sensitive information. Sensitive information is defined as passwords, connection strings, and other objects that are keys to information or keys to resources. For this work, we are currently assuming DRIs as good users, and the system is reducing information to just a need-to-know basis.

Therefore we need a metric that encapsulates the critical importance of recall. We must balance that will precision. If too many values are hidden, it is difficult for a person to do maintenance. Therefore we have to balance security with usefulness. A house without windows or doors is the most secure but also difficult to live within.

The best option is to use the F_β score shown in Eq. 6. We want a continuous function that takes into account a tradeoff between precision and recall. This is a practical solution but does not provide a single metric if we want to increase recall. Therefore one of the best single metrics for this work is the F_β score. This equation specifies a tradeoff relationship between recall and precision. For this work, we define that recall is five times more important than precision. For an f-score to be better, if we increase recall by one percent, that is worth 5% false positives (decrease in precision).

$$F_\beta = (1 + \beta^2) * \frac{precision * recall}{(\beta^2 * precision) + recall} \quad (6)$$

Each model has a range of $F\beta$ -scores determined by a score threshold. This demonstrates the differences between the models based on these curves.

Because models are statistically derived, often taking samples of training data to improve the loss, we run each model 20 times and take the average max f5-score. With this, we also run statistical significance.

For this work, we propose several metrics for comparison. One often-used metric is the area under the ROC curve (AUC). This is the area under the receiver operator curve (ROC). ML models output probabilities of an event.

The split of this probability defines one of two categories based on a threshold for a two-categorical system. Over that value, we select one. Under we select the other. As the threshold changes, the metrics change. AUC is a single metric across all the threshold's possible values. It tends to indicate whether a model has better overall thresholds than another. AUC is a valuable metric in some problem domains. However, sometimes, our data is highly skewed. As a result, AUC can yield misleading results.

For our work, we are most interested in precision and recall. Sensitive data should never leave the Sovereign Cloud. For that reason, recall is critical. For comparing models, we want to hold recall at 0.99 or higher and continue to increase precision. This number means that the model redacts sensitive information 99% of the time at the cost that it could block nonsensitive data, which may be helpful for a DRI to do their work. Therefore, we want to maintain a high standard of 0.99 or higher and continually improve at not blocking nonsensitive data.

6.4 Vocabulary Size

Each of the transformations discussed uses a vocabulary of words. In NLP, words are a combination of letters that make a word in a specified spoken or written language. We refer to a combination of letters as a word for this work, although acronyms and shortened words are frequently used in programming. Therefore our vocabulary differs a little from words found in the English language.

The vocabulary or number of unique words in the training set determines the number of features for BOW and TF-IDF. The features for BOW are the counts of each vocabulary word defined by Eq. 1. The features of TF-IDF are the same in the count but determined by Eq. 5. For WE, fixed-length vectors represent each word. The average number of words in the tokenized command data set is 78. In order to make a fair comparison to the other transformations, the size of the WE final vector is equal to the vocabulary size of BOW and TF-IDF. This experimentation uses a length of 20 float vectors per word. Table 6 demonstrates how original text features are expanded by BOW, TF-IDF, and WE.

6.5 Feature Importance

Tables 7 and 8 display the differences in the importance of different vocabulary words using BOW and TF-IDF transform. One crucial difference and something important to this problem is that “certificate” and “certificates” are treated as two separate words. For BOW, the word, certificate, is a rarer word, so it is not as crucial as “certificates.” On the other hand, TF-IDF emphasizes the importance of the IDF multiplier and makes “certificate” almost as important. WE algorithm handles this by making the vector encoding of the two words very close in Euclidean distance. As a result, each algorithm does a little better at emphasizing the importance or lack of difference between “certificate” and “certificates.”

Table 6. Feature size

Transform	Feature size
BOW	1,559
TF-IDF	1,559
WE	1,560
BOW-PF	9,354
TF-IDF-PF	9,354

Table 7. Feature importance for BOW

Word	Importance
Key	0.071
Certificates	0.041
Encryption	0.032
Secret	0.032
Network	0.029
Name	0.022
Vault	0.021
Compute	0.021
Management	0.020
String	0.019

Table 8. Feature importance for TF-IDF

Word	Importance
Certificates	0.049
Key	0.042
Certificate	0.041
Name	0.037
String	0.032
Osprofile	0.031
Az	0.030
Encryption	0.024
System	0.023
Id	0.022

BOW-PF introduces a partial word order. It weighs words used in the field name and parent name more strongly than the module name or the actual command. The feature importance of BOW-PF is shown in Table 9. Interestingly, the model weights the word key high in the field name.

Table 9. Feature importance for BOW-PF

Feature	Word	Importance
Field name	Key	0.0562
Parent type	Network	0.0343
Field name	Data	0.0253
Field name	Certificate	0.0239
Parent name	Network	0.0208
Field name	Name	0.0206
Parent type	Service	0.0188
Field name	String	0.0186
Field type	Encryption	0.0184
Parent type	Psmanaged	0.0184

TF-IDF-PF has a similar finding as TF-IDF, where the plural of a word has similarly high importance, shown in Table 10.

It is also worth noting that “string” is an essential word in all models. Almost all secrets come from strings or complex string types. Very little sensitive information is stored in integers, booleans, or enumerated types.

Table 10. Feature importance for TF-IDF-PF

Feature	Word	Importance
Field name	Key	0.0630
Field type	Collections	0.0308
Command	Az	0.0283
Field name	Data	0.0280
Parent type	Osprofile	0.0252
Field type	Azure	0.0212
Parent type	Models	0.0211
Field type	System	0.0203
Parent type	Cluster	0.0190
Field name	Keys	0.0189

The training data is built by manually labeling 1500 commands and each of the response attributes. The data is split into 80% training and 20% validation

data. The split is done by Azure Powershell command so that the validation set has commands that have not been used in training. This should yield a good estimate of the generalizability of the algorithm and how well it works on unseen commands.

Each model is run 20 times, with the results averaged. Displayed in Fig. 2a are the runs of each transform. Some depict using logistic regression, and others depict using boosted trees. First, each model trains using the training set. Next, calculating the maximum F5-score is done using the test set. Finally, using the trained model and the threshold for the max F5-score, the validation set is used to determine F5-score, as seen in new examples.

6.6 Metric Comparison

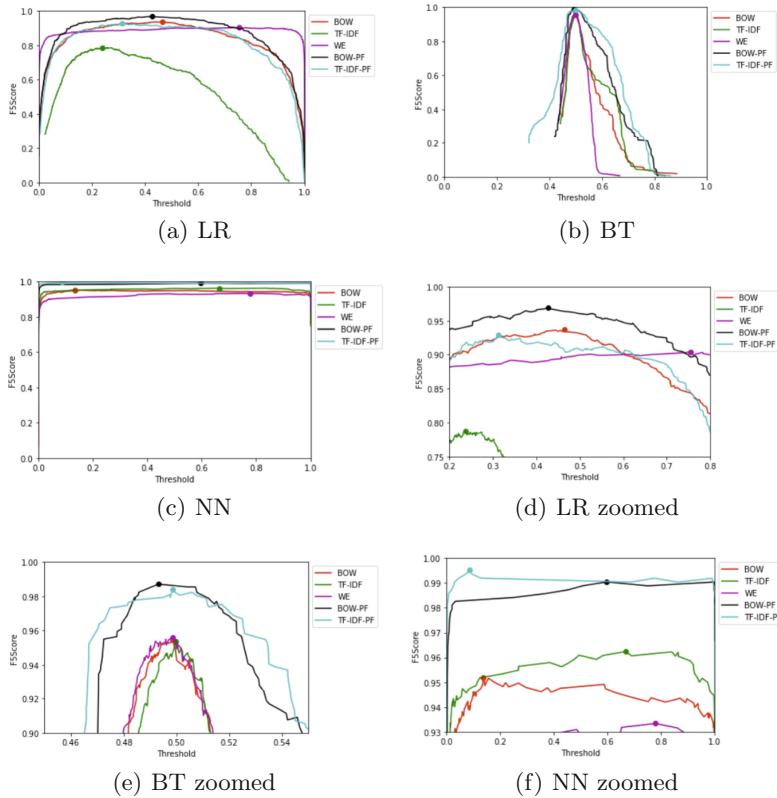


Fig. 2. F5-score for transforms and models

F_β -score changes with the threshold selected. This model must make a binary decision, so we choose the threshold to give us the max F5-Score. We use the

test data set to determine the threshold with the max F5-Score. We then use the validation set to test the model and threshold against unseen examples to determine how well the model will generalize and how well it will work in a production setting.

Each curve shows the F5-Score, as calculated from Eq. 6, for all thresholds. The point on the curve is the max F5-Score overall threshold value. It should be noted that a curve with a plateau top has a robust threshold. Slight changes in threshold do not change F5-Score much. It also implies that unseen examples are not sensitive to slight changes in threshold value. A sharp peak at the maximum F5-Score implies an unstable maximum. Using this model against this or other data sets could result in very different F5-Scores if anything is slightly perturbed.

For the logistic regression example, TF-IDF is the weakest. BOW is effective, while BOW-PF is the most effective. The PF is a form of word importance that weights the words. TF-IDF does not work as well in highly skewed data. The word frequencies help predict both the positive and negatively labeled items. Infrequent words help identify risky items, but most items are not risky, so words that help predict non-risky items are frequent, and therefore TF-IDF decreases the weights.

BT has more of a peak to most curves. Unlink LR BT is discontinuous. The inference of a BT model with perturbation of features causes different parts of the tree to be traversed. The ordering of worst-to-best algorithms stays relatively consistent across models. BT's ability to handle discontinuous functions allows it to fit the data better, resulting in a higher F5-Score. BOW-PF still performs the best as it uses word counts, does not discount frequently used words, and includes a type of word ordering captured in the Per-Feature part of the algorithm.

Table 11 demonstrates how BOW-PF and TF-IDF-PF can be used to apply a word order that is not in BOW or TF-IDF. It is recommended that if you use LR or BT, BOW-PF is the transformation that should be used. BOW-PF captures the use of words in each feature, for example, emphasizing the field name having the word “key” over the command having the word “key.” The inverse of the document frequency is not handled well in LR and BT. NN, on the other hand, picks up the nuances of the inverse of document frequency and handles TF-IDF and TF-IDF-PF better than the BOW counterparts.

Table 11. Max F5-scores for transforms

Transform	LR	BT	NN
BOW	0.937	0.955	0.952
TF-IDF	0.788	0.954	0.962
WE	0.904	0.956	0.933
BOW-PF	0.969	0.987	0.990
TF-IDF-PF	0.929	0.984	0.995

7 Conclusion and Future Work

This work demonstrates how to convert PowerShell commands to the well-researched NLP domain. From there, we compare popular techniques and find that BOW-PF performs best at creating features for different types of models. This work exploited the sentence-like structure of commands, variable names, and variable types. Using F5-Score as a metric for comparison allows us to summarize system goals into one value for optimizing across thresholds and in subsequent retraining.

Slight adaptions to this work yield a system that works across different CLIs. Systems other than PowerShell will be more complex because the output is not in a structured programming object. Also, the variables' names are Pascal-case for PowerShell, but other CLIs will require different transformations to convert commands and variables to words. This will require more work to convert as even if the responses are structured as objects and the variables' names are Pascal-case, the names are variables are still created by different development teams. It is similar to the NLP space of handling different languages. For example, one CLI might call a password variable "Password," whereas another CLI interface might call it "Psswd." The system will need to be trained to pick up new vocabulary.

This work can extend to APIs. Powershell, behind the scenes, makes API calls to pass commands and receive back the response. Powershell turns those JSON or XML responses into objects. This system is built on structured response objects. APIs have similar structures in which there is an endpoint, similar to a command, and a response, often a JSON object, analogous to the response object of PowerShell. Therefore with little work, the risk modeling can be moved to API space to redact or mask values returned in API calls.

References

1. Beel, J., Gipp, B., Langer, S., Breitinger, C.: Paper recommender systems: a literature survey. *Int. J. Digit. Lib.* **17**(4), 305–338 (2016)
2. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **55**(1), 119–139 (1997)
3. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: European Conference on Machine Learning, pp. 137–142. Springer, Berlin (1998)
4. Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. *J. Doc.* (2004)
5. Lavelli, A., Sebastiani, F., Zanoli, R.: Distributional term representations: an experimental comparison. In: Proceedings of the thirteenth ACM International Conference on Information and Knowledge Management, pp. 615–624 (2004)
6. Luhn, H.P.: A statistical approach to mechanized encoding and searching of literary information. *IBM J. Res. Dev.* **1**(4), 309–317 (1957)
7. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013). [arXiv:1301.3781](https://arxiv.org/abs/1301.3781)
8. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **26** (2013)

9. Moon, A., Raju, T.: A survey on document clustering with similarity measures. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **3**(11), 599–601 (2013)
10. Rajaraman, A., Ullman, J.D.: Mining of Massive Datasets. Cambridge University Press (2011)
11. Ramos, J., et al.: Using tf-idf to determine word relevance in document queries. In: Proceedings of the First Instructional Conference on Machine Learning, vol. 242, pp. 29–48. New Jersey, USA (2003)
12. Robertson, S.: Understanding inverse document frequency: on theoretical arguments for idf. *J. Doc.* (2004)
13. Sivic, J., Zisserman, A.: Efficient visual search of videos cast as text retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(4), 591–606 (2008)
14. Teller, V.: Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition (2000)
15. Vinokourov, A., Cristianini, N., Shawe-Taylor, J.: Inferring a semantic representation of text via cross-language correlation analysis. *Adv. Neural Inf. Process. Syst.* **15** (2002)
16. Zhang, Y., Jin, R., Zhou, Z.-H.: Understanding bag-of-words model: a statistical framework. *Int. J. Mach. Learn. Cybern.* **1**(1), 43–52 (2010)



An Intrinsic Framework of Information Retrieval Evaluation Measures

Fernando Giner^(✉)

E.T.S.I. Informática UNED, C/ Juan del Rosal, 16, 28040 Madrid, Spain
fginer3@alumno.uned.es

Abstract. Information retrieval (IR) evaluation measures are cornerstones for determining the suitability and task performance efficiency of retrieval systems. Their metric and scale properties enable to compare one system against another to establish differences or similarities. Based on the representational theory of measurement, this paper determines these properties by exploiting the information contained in a retrieval measure itself. It establishes the *intrinsic framework* of a retrieval measure, which is the common scenario when the domain set is not explicitly specified. A method to determine the metric and scale properties of any retrieval measure is provided, requiring knowledge of only some of its attained values. The method establishes three main categories of retrieval measures according to their intrinsic properties. Some common user-oriented and system-oriented evaluation measures are classified according to the presented taxonomy.

Keywords: Information retrieval · Evaluation · Metric · Scale

1 Introduction

Information Retrieval (IR) is a field dealing with the analysis, storage and searching of a user's information need [84]. To effectively compare and progressively develop better IR systems, assessment plays an important role. Even being evaluation a rich scenario that has contributed to the development of the IR field, a better comprehension of evaluation measures is needed. Specifically, their classification according to the scale types of Stevens [91], which helps to make explicit the assumptions behind measures, and to justify the validity of conclusions as consequence of the obtained results.

Due to the empirical nature of IR, this task is not exempt of the historic confusion and controversy about the scale types and the statistical methods that can be performed [44, 63, 96]. Recently, two approaches have discussed the role of IR evaluation metrics¹ as being interval scales. On the one hand, Moffat [66] states

¹ Here, the commonly used term “IR evaluation *metric*” collides with the mathematical term “*metric*”, which will be used later in this paper. To solve this issue, the rest of the paper will refer the term “IR evaluation *metrics*” as “IR evaluation *measures*”, keeping the term “*metric*” for its mathematical sense.

that retrieval measures can be considered interval scales, whenever they have a real-world basis (an external validity) and can be motivated as corresponding to the usefulness of system outputs. Advocated with rhetoric arguments, this viewpoint shows similarities to the *operational paradigm* [63, 64]. In contrast, Ferrante et al. [31, 32] investigate the implications of retrieval measures being interval scales, by grounding their arguments [33] on the *representational theory of measurement* (RTM) [59–61, 72]. Regarding this viewpoint, which could be termed as the *representational paradigm* [63, 64], interval scales are real mappings whose attained values are equispaced. This position seems to have a sound theoretical basis; however, the recent arguments provided [33] have a limited application [41].

The goal of this paper is not to challenge or agree with any of these two viewpoints, neither to find common points shared by both. In fact, they consider different measurement theories since the controversy lies in the assumption or not of the RTM, which is not mentioned in Moffat's work. The goal of this paper is to exemplify the fact of considering retrieval measures as equispaced mappings [31] by providing a classification of IR evaluation measures, and to give some insights about the arguments of this position by overcoming the concerns of [41].

Retrieval measures can be classified based on several factors, such as the domain, range, relation, or expression among others. Probably, the analytical expression is one of the main aspects that has been considered, since it determines the performance of retrieval systems [86] or differentiates user browsing models [17, 67]. Along with the analytical expression, the relationship between the items of the domain set and the range set also accounts for the type of a retrieval measure. For instance, the metric properties of the domain set detect whether every item to be measured can be distinguished [37], or the relationships on the range set determine the operations that can be performed, and the statistics that can be applied [91]. Based on the RTM, this paper shows that the attained values of a retrieval measure determine its metric and scale properties. It establishes the *intrinsic framework*, which exploits the information contained in the retrieval measure itself to provide a taxonomy of IR evaluation measures.

The intrinsic framework is not limited to the IR field. In the area of databases or data mining among others, many empirical studies use the same or similar measures presented here. In studies that are relied on the RTM, the scale characterization is central to make statistical inferences regarding an attribute to be measured. The metric characterization is also useful to determine whether the distance associated to the measure is similar to the Euclidean geometry, or it is a more elastic geometry, where the notion of “closeness” allows the existence of distinct points whose distance is zero. At theoretical level, the intrinsic framework provides a formal basis that every retrieval measure assumes when its empirical domain is not explicitly specified. This paper describes some consequences about retrieval measures under the assumption of the RTM, which can be useful for IR theory works that challenge or agree with the representational paradigm.

The rest of the paper is organised as follows: Sect. 2 reports some related work. In Sects. 3 and 4, the intrinsic framework is presented, and its metric and scale properties are characterised. Section 5 provides a taxonomy to determine the intrinsic category of a retrieval measure. In Sect. 6, some common IR evaluation measures are classified with the intrinsic framework. Finally, in Sect. 7, some conclusions are drawn.

2 Related Work

In the IR field, there is a great body of research work on evaluation [45, 74], which has led to consider it as a key area [1]. Large-scale campaigns and initiatives, such as TREC [98], NTCIR [83], CLEF [34], FIRE and INEX, have promoted improvements in academy and industry. The Cranfield 2 experiments [20], are considered the first attempt of evaluation in the IR field [24], and is the underlying framework of many modern experiments. IR systems are usually compared with a set of topics or search requests [16], where IR evaluation measures are computed [97, 99]. The attained values of IR evaluation measures are commonly supported with significance test results or confidence intervals [18, 49, 80, 87, 93]. Empirically quantifying and statistically assessing the performance of IR systems enable to establish differences and similarities [79].

The formal analysis of IR evaluation measures has contributed to a better understanding of their meaning. Some works have shown that retrieval measures correspond to different user browsing models [6, 17, 19, 67, 100, 101]. Others have characterised the effectiveness of retrieval measures with formal properties [2, 4, 48, 65, 88, 92], which help to know the appropriateness of retrieval measures on a specific scenario. The use of formal properties as a method to explore retrieval models and how best to improve them, in order to achieve higher retrieval effectiveness has been fostered by Fang et al. [27–30], and successfully applied to the study of basic models [26, 27], pseudo-relevance feedback methods [21, 22, 69], translation retrieval models [54, 71] and neural network retrieval models [76].

There have been several approaches, which explore retrieval measures from a measurement viewpoint, van Rijsbergen [9, 94] tackled the foundations of measurement in IR through a conjoint (additive) structure based on precision and recall, then he examined the properties of a measure on this prec-recall structure. Bollman et al. [11] defined a similar conjoint structure, but on the contingency table of the binary retrieval; then, they studied the properties of the proposed MZ-metric. Later, Bollman [10] shown that retrieval measures can be expressed as a linear combination of the number of relevant/nonrelevant retrieved documents, whenever they satisfy two proposed axioms. Flach [35] modelled the empirical domain through confusion matrices, then the relation between measurement theory and machine learning evaluation is sketched. However, these works do not address the scale properties of retrieval measures.

Some works consider the gold standard as measurement, they analyse the scale properties in ordinal classification [7, 39, 95], or the scale properties of the

ground truth, human annotation or predicted variables [43]. Other works consider the system outputs and the gold standard as independent measurements. It allows to introduce axioms over the similarity of assessors scales and system scales [14, 62]; and to provide a single and unified explanation for most classification, ranking, and clustering measures [5], or a methodology to determine the most appropriate task/metric formalization for a given data mining problem [3].

Recently, Fuhr [38] proposed some experimental protocols to measure the usefulness of IR systems, such as avoiding the use of MRR and ERR since they violate basic requirements for a metric. In contrast, Sakai [81] argues some of Fuhr's statements since they do not explain the experimental alignment between retrieval measures and user's perception of usefulness. As a consequence or parallel to this difference, there exists a currently active dialogue [31, 32, 66], which considers retrieval measures as real mappings that quantify the usefulness of retrieval systems. Based on the RTM, Ferrante et al. [31, 32] investigate the implications of IR measures being interval scales. This viewpoint claims that interval scales are real mappings whose attained values are equispaced. They propose intervalization as a feasible technique to obtain meaningfulness. In contrast, Moffat [66] considers that document rankings are categorical data, retrieval measures are numeric mappings defined by the context of the dataset, and they are bounded to a set of target values by some external reality. Moffat claims that retrieval measures can be considered interval scales, whenever they have a real-world basis, i.e., an external validity; for instance, a prize assignation to classes of rankings.

These two viewpoints determine the scale type of retrieval measures based on different assumptions. The former, which could be termed the *representational paradigm*, grounds its arguments on the work of Ferrante et al. [33]. They developed a framework for both set-based and rank-based IR evaluation measures as well as both binary and multi-graded relevance, determining whether retrieval measures are interval scales. However, the provided arguments have a limited application [41], which can be addressed by allowing the domain set to be specified by the retrieval measure itself. The intrinsic framework presented here follows this point of view.

3 Formalisation of the Intrinsic Framework

In *batch evaluation*, a topic or query is submitted to an IR system, which returns as output a *search engine result page* (SERP),² $\hat{\mathbf{r}}$. Then, evaluation measures quantify, in numeric terms, the *effectiveness* of the retrieval system, i.e., its ability to leave aside nonrelevant documents while retrieving relevant ones. Thus, an effectiveness measure can be seen as a mapping that relates a set of possible rankings, \mathbf{R} , with real numbers, i.e., an IR evaluation measure assigns numbers

² Typically a SERP includes content in a non homogeneous manner, such as images, query suggestions, knowledge panels, etc. However, here, we consider the classical ordered (or unordered) list of documents since it is the common structure considered when the evaluation of ranking models is studied.

(numerical range) that correspond to a set of rankings (empirical domain). Once a retrieval measure has been defined, it has consequences on both the empirical domain and the numerical range.

On the empirical domain, every retrieval measure, f , establishes an implicit ordering relationship, \preceq_f , which is defined by their attained values as follows:

$$\hat{\mathbf{r}}_1 \preceq_f \hat{\mathbf{r}}_2 \iff f(\hat{\mathbf{r}}_1) \leq f(\hat{\mathbf{r}}_2), \quad (1)$$

for all $\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2 \in \mathbf{R}$; therefore, every IR evaluation measure has a naturally associated ordering, which is inherently derived from the measure itself. Every pair of elements is comparable with this binary relationship, \preceq_f , and the transitivity is verified trivially; thus, it is a *weak order*³ [72]. The ordering structure, (\mathbf{R}, \preceq_f) , can be represented with a Hasse diagram, denoted by G_f , where nodes are labelled with the elements of \mathbf{R} . They are placed in different levels depending on the attained values of f . An edge indicates that the attained value of one element is greater than the other and there are no other rankings between them. For instance, the associated graph of a retrieval measure, f , such that $f(\hat{\mathbf{r}}_1) < f(\hat{\mathbf{r}}_2) = f(\hat{\mathbf{r}}_3) = f(\hat{\mathbf{r}}_4) < f(\hat{\mathbf{r}}_5) < f(\hat{\mathbf{r}}_6)$, is shown in Fig. 1.

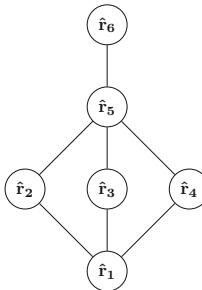


Fig. 1. Example of Hasse diagram, G_f , associated to a retrieval measure.

On the numerical range, the attained values of an IR evaluation measure enable to examine one system against another to establish relationships, differences or similarities. The effectiveness of a pair of retrieval systems is usually compared by considering the absolute/relative difference of the attained values or with testing validation based on this difference [86]. Thus, in order to perform comparisons, any retrieval measure has a naturally associated distance,

³ The associated weak order, \preceq_f , may be transformed into a total order by considering the following equivalence relation: $\hat{\mathbf{r}}_1 \sim_f \hat{\mathbf{r}}_2 \iff f(\hat{\mathbf{r}}_1) = f(\hat{\mathbf{r}}_2)$. Let \mathbf{R}^* be the set of equivalence classes, and let $\hat{\mathbf{r}}_1^*$ and $\hat{\mathbf{r}}_2^*$ be two elements of this set containing the individual system output rankings $\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2 \in \mathbf{R}$, respectively. It can be defined the following ordering on \mathbf{R}^* : $\hat{\mathbf{r}}_1^* \preceq_f^* \hat{\mathbf{r}}_2^* \iff \hat{\mathbf{r}}_1 \preceq_f \hat{\mathbf{r}}_2$. Then, $(\mathbf{R}^*, \preceq_f^*)$ is called the *reduction* or *quotient* of (\mathbf{R}, \preceq_f) , where \preceq_f^* is well-defined and $(\mathbf{R}^*, \preceq_f^*)$ is a totally ordered set [72].

$d_f : \mathbf{R} \times \mathbf{R} \longrightarrow \mathbb{R}$, which is the absolute value of the difference between two attained values or most commonly known as the *Euclidean* distance of them:

$$d_f(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2) = |f(\hat{\mathbf{r}}_1) - f(\hat{\mathbf{r}}_2)| \quad (2)$$

Remark 1. By weighting each edge, $\hat{\mathbf{r}}_1\hat{\mathbf{r}}_2$, of the associated Hasse diagram, G_f , with the value $|f(\hat{\mathbf{r}}_1) - f(\hat{\mathbf{r}}_2)|$, it trivially holds that the distance d_f between a pair of elements is the minimum length distance on this edge-weighted graph, i.e., d_f is the natural distance on the Hasse diagram, G_f .

Therefore, every IR evaluation measure, f , has a naturally associated context that is intrinsically derived from the measure itself. This context is composed of its associated ordering, \preceq_f , and distance, d_f . These two mathematical tools are inherently connected to the retrieval measure, since the ordering directly reflects the purpose for which it was designed, and the distance is the usual manner to perform comparisons on the attained values. They are intrinsic entities that only depend on the definition of the retrieval measure itself, and can be considered jointly to establish the following concept.

Definition 1. The *intrinsic framework* of an IR evaluation measure, f , is the set of possible rankings endowed with its associated ordering, \preceq_f , and its associated distance, d_f ; it is denoted by $(\mathbf{R}, \preceq_f, d_f)$.

The intrinsic framework is closely related to the mathematical concept of *intrinsic geometry* [25, 40] of curves and surfaces.⁴ Consider the reduction or quotient $(\mathbf{R}^*, \preceq_f^*)$ of the ordering associated with a retrieval measure. As \preceq_f^* is a total order, its associated Hasse diagram, G_f^* , is a chain that joins consecutive elements with straight lines. By weighting each edge as indicated in Remark 1, the Hasse diagram, G_f^* , can be considered a curve on its own, whose distance between any pair of contiguous elements is the associated weight. The intrinsic framework represents the intrinsic geometry of G_f^* , where the distance is computed with the minimum path length. In a strict sense, the intrinsic framework is not the intrinsic geometry of G_f^* since it is a discrete curve, which lacks differentiability. However, this intrinsic framework enables to measure distances on G_f^* with mathematical tools derived from the retrieval measure itself.

Comparisons on the intrinsic framework of a retrieval measure have sense since they are derived from the measure itself. If we consider a different retrieval measure, then the ordering of the intrinsic framework is changed. However, the criterion for measuring the minimum path length of the new measure will remain

⁴ Imagine hypothetical beings living on the surface of a two-dimensional Euclidean space, \mathbb{R}^2 , ignorant of the surrounding three-dimensional space (but with a sense of Euclidean distance). These beings are local observers, whose view reaches only a two coordinated environment. The geometrical elements of this surface capable of being observed or measured by these beings (essentially lengths) constitute what is called the *intrinsic geometry* of the surface. The intrinsic properties of the surface are those which depend exclusively on the surface itself.

coherent since it verifies the axiomatic properties of a distance. Another property of the intrinsic framework is that it considers rankings of any size since no assumption has been made about that. In addition, no assumption has been made about relevance grades, then the intrinsic framework can handle binary or multi-graded relevance labels.

4 Intrinsic Properties of an IR Evaluation Measure

In this section, the metric and scale properties of the intrinsic framework are characterised.

4.1 Metric Properties of an IR Evaluation Measure

Let us recall some basic notions of metric spaces [37, 47]. Consider an IR evaluation measure, f ; the associated distance, d_f , is a *pseudometric* if it verifies the following two properties: (i) *symmetry*, $d_f(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2) = d_f(\hat{\mathbf{r}}_2, \hat{\mathbf{r}}_1)$, and (ii) *triangular inequality*, $d_f(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_3) \leq d_f(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2) + d_f(\hat{\mathbf{r}}_2, \hat{\mathbf{r}}_3)$. If d_f also verifies (iii) the *identity of indiscernible*, $d_f(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2) = 0 \Leftrightarrow \hat{\mathbf{r}}_1 = \hat{\mathbf{r}}_2$, then d_f is a metric.

Strictly speaking it must be distinguished the evaluation measure, f , from the distance, d_f . According to the previous paragraph, the metric properties should be applied to the distance with statements such as “this *distance* is a pseudometric”. However, in the retrieval context, as the distance is derived from the IR evaluation measure, it is very common to assign the metric properties to the retrieval measure with statements such as “this *retrieval measure* is a metric”. The following result shows that every retrieval measure is a pseudometric.

Proposition 1. *Let $(\mathbf{R}, \preceq_f, d_f)$ be the intrinsic framework of an IR evaluation measure, f , then the associated distance, d_f , is a pseudometric.*

In the retrieval scenario, it has been coined the term “IR evaluation *metric*” to design any IR evaluation measure. However, the following result shows that not every retrieval measure is a metric.

Proposition 2. *Let $(\mathbf{R}, \preceq_f, d_f)$ be the intrinsic framework of an IR evaluation measure, if f is an injective or one-to-one function,⁵ then the associated distance, d_f , is a metric.*

Thus, every retrieval measure is a pseudometric, and only the retrieval measures that assign distinct values to every system output ranking are metrics.

⁵ In basic algebra [36, 50, 51], f is an injective function, if f maps distinct elements to distinct elements, formally: $f(\hat{\mathbf{r}}_1) = f(\hat{\mathbf{r}}_2)$ implies $\hat{\mathbf{r}}_1 = \hat{\mathbf{r}}_2$, $\forall \hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2 \in \mathbf{R}$.

4.2 Scale Properties of an IR Evaluation Measure

From the point of view of the RTM, a consistent assignment of real numbers to the empirical domain is a *scale* for the attribute to be measured [90]. Stevens [91] distinguished four main types of measurement scales: *nominal*, *ordinal*, *interval* and *ratio*. By considering an order relationship, \trianglelefteq , on the underlying empirical domain; the scales, φ , reflecting or preserving this ordering are called *ordinal scales*, formally: $a \trianglelefteq b \Leftrightarrow \varphi(a) \leq \varphi(b)$.

In the retrieval scenario, every IR evaluation measure, f , defined on the ordering structure, $(\mathbf{R}, \trianglelefteq_f)$, is an ordinal scale since the definition of the associated ordering (see Eq. 1) just verifies this property.

The intrinsic framework, $(\mathbf{R}, \trianglelefteq_f, d_f)$, of a retrieval measure allows to consider an interval, defined as the set of elements between two end-points, formally: $[\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2] = \{\hat{\mathbf{r}} \in \mathbf{R} : \hat{\mathbf{r}}_1 \trianglelefteq_f \hat{\mathbf{r}} \trianglelefteq_f \hat{\mathbf{r}}_2\}$. It can be quantified by its cardinality or *span of the interval*, denoted by $\Delta_{\hat{\mathbf{r}}_1 \hat{\mathbf{r}}_2}$, formally: $\Delta_{\hat{\mathbf{r}}_1 \hat{\mathbf{r}}_2} = |[\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2]|$. The span of an interval represents how closely spaced is a pair of rankings.

Then, an order relationship, defined on the set of possible intervals, can be introduced as follows: $[\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2] \trianglelefteq_{d_f} [\hat{\mathbf{r}}_3, \hat{\mathbf{r}}_4] \Leftrightarrow \Delta_{\hat{\mathbf{r}}_1 \hat{\mathbf{r}}_2} \leq \Delta_{\hat{\mathbf{r}}_3 \hat{\mathbf{r}}_4}$. This order relationship, \trianglelefteq_{d_f} , is a weak order since every pair of intervals is comparable and the transitivity is verified trivially.

Following the RTM [72], an IR evaluation measure, f , defined on the ordering structure $(\mathbf{R}, \trianglelefteq_f)$ is an *interval scale* if it preserves differences, i.e., if equally spaced intervals are assigned to equal differences, formally:

$$[\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2] \trianglelefteq_{d_f} [\hat{\mathbf{r}}_3, \hat{\mathbf{r}}_4] \iff f(\hat{\mathbf{r}}_2) - f(\hat{\mathbf{r}}_1) \leq f(\hat{\mathbf{r}}_4) - f(\hat{\mathbf{r}}_3).$$

The following result characterise the interval scales.

Proposition 3. *Consider an ordinal scale, f , which is a metric, then f is an interval scale if and only if the attained values are equally spaced.*

Thus, every retrieval measure is an ordinal scale on its associated ordering, and only the retrieval metrics whose attained values are equispaced are interval scales.

5 Intrinsic Taxonomy of IR Evaluation Measures

The characterisations of Sect. 4 enables the classification of retrieval measures into three main categories, which provide a ready-to-use rule to identify intrinsic properties in terms of the attained values:

1. Every IR evaluation measure, f , is an ordinal scale and a pseudometric on its intrinsic framework. By default, these are the properties of any retrieval measure. This category is denominated **ordinal/pseudometric**.
2. If the attained values of f are different for every system output ranking, i.e., the retrieval measure is a one-to-one function, then f is a metric (not necessarily an interval scale). These retrieval measures are **ordinal/metric**.

3. If the attained values of f are equally spaced, then the retrieval measure is an interval scale. This category is denominated **interval/metric**.

A quick glance at this taxonomy confirms that there are no ratio scales. The reason is that the intrinsic framework aims to deduce the metric and scale properties of an IR evaluation measure from the information contained in the retrieval measure itself, i.e., from its associated ordering and distance. Ratio scales need an additional operator among rankings [64, 72], which is not present in the definition of the retrieval measure. Therefore, IR evaluation measures can be ratio scales, but an *extrinsic* operation among system output rankings has to be previously specified.

6 Some Examples

In this section, some common retrieval measures are classified according to the taxonomy of Sect. 5. The empirical domain of each retrieval measure is its intrinsic framework.⁶

6.1 Set-Based Retrieval

In this case, an IR system returns a set of documents in response to a query; by denoting with 1 a relevant document and with 0 a nonrelevant document, some examples of system outputs are as follows: $\hat{\mathbf{r}}_1 = \{0, 0, 0, 0, 0\}$, $\hat{\mathbf{r}}_2 = \{1, 0, 0, 0, 0\}$ and $\hat{\mathbf{r}}_3 = \{1, 1, 0, 0, 0\}$. In general, the system output can be summarised with a contingency table of two binary variables: relevance and retrieval. Table 1 illustrates the frequency distribution, where the two factors are shown simultaneously.

Table 1. Contingency table of the set-based retrieval in the binary case.

	Relevant documents	Non relevant documents
Documents retrieved	tp true positive	fp false positive
Documents non retrieved	fn false negative	tn true negative

Some widely used evaluation measures are recall [9], precision [9], fallout [9], classification accuracy [8], miss rate, and error rate, defined as follows:

$$\begin{aligned} \text{recall} &= \frac{tp}{(tp + fn)} & \text{precision} &= \frac{tp}{(tp + fp)} \\ \text{fallout} &= \frac{fp}{(fp + tn)} & \text{classification accuracy} &= \frac{(tp + tn)}{(tp + fn + fp + tn)} \\ \text{miss rate} &= \frac{fn}{(tp + fn)} & \text{error rate} &= \frac{(fp + fn)}{(tp + fn + fp + tn)} \end{aligned}$$

⁶ As noted in Sect. 4, the intrinsic properties of a retrieval measure deduced with this framework are based on the RTM.

All of them are metrics since the numerator of their analytical expressions has at least one of the terms: tp , tn , fp or fn , which are distinct for every new relevant document present in the system output. In addition, they are interval scales since the denominator of their analytical expression is constant and the numerator increases one unit with every relevant retrieved document. Thus, their attained values are equispaced.

Other related evaluation measures are inverse recall, inverse precision, specificity, false discovery rate, false omission rate, and F-measure [9], defined as follows:

$$\begin{array}{ll} \text{inverse recall} = \frac{tn}{(fp + tn)} & \text{inverse precision} = \frac{tn}{(fn + tn)} \\ \text{specificity} = \frac{tn}{(tn + fp)} & \text{false discovery rate} = \frac{fp}{(fp + tp)} \\ \text{false omission rate} = \frac{fn}{(fn + tn)} & \text{F-measure} = \frac{2 \cdot \text{prec} \cdot \text{recall}}{(\text{prec} + \text{recall})} \end{array}$$

The F -measure is a metric since it attains different values (the harmonic mean is an increasing function). The rest of the measures are metrics and interval scales since they verify the same properties indicated in the previous paragraph. However, the F -measure is not an interval scale; for instance, consider a collection of 15 documents, where 5 are relevant to a topic, then $F(\hat{\mathbf{r}}_1) = 0.000$, $F(\hat{\mathbf{r}}_2) = 0.300$ and $F(\hat{\mathbf{r}}_3) = 0.509$.

The generality factor or prevalence $= (tp + fn)/(tp + fn + fp + tn)$ [9] is a pseudometric since all its values are the same for every system output.

The utility measure [85], $\text{utility} = \alpha \cdot tp + \beta \cdot fn + \gamma \cdot fp + \delta \cdot tn$, where α, β, γ and δ are the positive weights assigned by the user, present several possibilities varying the four parameters. In general, if the four parameters are different, then the utility measure is a metric, and some combinations of parameters could yield an interval scale.

Some user-oriented measures are based on the following four variables: (i) total number of relevant documents known to the user: U ; (ii) number of relevant documents known to the user, which were retrieved: R_k ; (iii) number of relevant documents unknown to the user, which were retrieved: R_u ; and (iv) the number of retrieved documents: A . Some examples of user-oriented measures are coverage ratio [58], retrieval recall [58], novelty ratio [58], and recall effort [58], defined as follows:

$$\begin{array}{ll} \text{coverage ratio} = \frac{R_k}{U} & \text{retrieval recall} = \frac{(R_k + R_u)}{U} \\ \text{novelty ratio} = \frac{R_u}{(R_u + R_k)} & \text{recall effort} = \frac{U}{A} \end{array}$$

The relevance recall and the novelty ratio are not metrics by considering the following two system outputs: (i) $\hat{\mathbf{r}}_5 = \{1 \text{ retrieved relevant document known to the user}\}$; and (ii) $\hat{\mathbf{r}}_6 = \{1 \text{ retrieved relevant document known to the user, } 1 \text{ non-relevant document retrieved}\}$. The relevance recall and the novelty ratio

attain the same value on these system outputs. The recall effort is neither a metric since it attains the same value on the following two system outputs: (i) $\hat{\mathbf{r}}_7 = \{1 \text{ retrieved relevant document unknown to the user}, 1 \text{ non-relevant document retrieved}\}$; and (ii) $\hat{\mathbf{r}}_8 = \{2 \text{ non-relevant documents retrieved}\}$. The coverage ratio is not a metric by considering two system outputs, which differ in one non-relevant retrieved document.

Table 2 provides a summary of the intrinsic properties of these retrieval measures.

Table 2. Intrinsic properties of some retrieval measures in the set-based retrieval.

	Ord/Pseudom	Ord/Metr	Interv/Metr
Recall [9]			✓
Precision [9]			✓
Fallout [9]			✓
Miss rate			✓
Classification accuracy [8]			✓
Error rate			✓
Inverse recall			✓
Inverse precision			✓
Specificity			✓
False discovery rate			✓
False Omission Rate			✓
<i>F</i> -measure [9]		✓	
Generality factor [9]	✓		
Coverage ratio [58]	✓		
Retrieval recall [58]	✓		
Novelty ratio [58]	✓		
Recall effort [58]	✓		

6.2 Rank-Based Retrieval

In this case, an IR system returns a ranked list of documents, $\hat{\mathbf{r}} = \langle d_1, d_2, \dots \rangle$, in response to a query. Once documents have been retrieved, in the multi-graded case, the *qrels* are used to determine the relevance of the retrieved documents to the topic. Denoting by r a rank position of $\hat{\mathbf{r}}$, each retrieved document, d_r , is assigned a relevance degree, a_r , for $r \geq 1$. For instance, in the binary case, denoting with a_1 a relevant document and with a_0 a nonrelevant document, some example rankings of length four are as follows: $\hat{\mathbf{r}}_1 = \langle a_1, a_0, a_0, a_0 \rangle$, $\hat{\mathbf{r}}_2 = \langle a_0, a_1, a_0, a_0 \rangle$ and $\hat{\mathbf{r}}_3 = \langle a_0, a_1, a_0, a_1 \rangle$. These relevance degrees can be categorical labels, to handle numerical values, a *gain function* is considered, $gain(\cdot)$,

by assigning a positive real number to each relevance degree. The gain at rank r will be denoted by $g(r) = \text{gain}(a_r)$, where $g(r) = 0$ for a non-relevant document. We assume that the gain function is the indicator function. For instance, in the binary case, if $g(a_1) = 1$ and $g(a_0) = 0$, then the previous ranking examples can be expressed as $\hat{\mathbf{r}}_1 = \langle 1, 0, 0, 0 \rangle$, $\hat{\mathbf{r}}_2 = \langle 0, 1, 0, 0 \rangle$ and $\hat{\mathbf{r}}_3 = \langle 0, 1, 0, 1 \rangle$. The cumulative gain at rank r is the total gain up to rank r , which is defined as $cg(r) = g(r) + g(r-1)$, for $r > 1$ and $cg(1) = g(1)$.

In this multi-graded context, the precision at rank r [9, 15, 33] can be expressed as follows:

$$\text{Prec}@r = \frac{cg(r)}{r} ,$$

This retrieval measure is a pseudometric, but not a metric since it holds that $\text{Prec}@4(\hat{\mathbf{r}}_1) = 0.250 = \text{Prec}@4(\hat{\mathbf{r}}_2)$. Similarly, the recall at rank r is a pseudometric since its analytical expression is also based on the cumulative gain at r .

In this scenario, an *ideal* ranking can be considered by listing first all documents with the higher relevance degree, then all documents with the contiguous relevance degree, and so on. Denoting by $ig(r)$ the gain at rank r for an ideal ranking, $cig(r)$ the cumulative gain at rank r for an ideal ranking, and by L the length of a ranking, then the sliding ratio [58, 70]:

$$\text{sr} = \frac{cg(L)}{cig(L)} ,$$

is a pseudometric since $\text{sr}(\hat{\mathbf{r}}_1) = 1.000 = \text{sr}(\hat{\mathbf{r}}_2)$. A modified version of the sliding ratio [77]:

$$\text{msr} = \frac{\sum_{r=1}^L \frac{1}{r} g(r)}{\sum_{r=1}^L \frac{1}{r} ig(r)} ,$$

is a metric since it is sensitive to the ranking order. However, it is not an interval scale since $\text{msr}(\hat{\mathbf{r}}_1) = 1$, $\text{msr}(\hat{\mathbf{r}}_2) = 0.5$ and $\text{msr}(\langle 0, 0, 1, 0 \rangle) = 0.333$.

To present the following retrieval measures, we need to introduce the indicator function $isrel(r)$, defined as $isrel(r) = 1$ if the document at rank r is relevant, and $isrel(r) = 0$ otherwise. Thus, the value $count(r) = \sum_{k=1}^r isrel(k)$ is the number of relevant documents within the top r documents of the ranking.

Let R be the total number of relevant retrieved documents, then the R-precision:

$$\text{R-precision} = \frac{count(R)}{R} ,$$

is a pseudometric; considering the ranking $\hat{\mathbf{r}}_4 = \langle 1, 0, 0, 1 \rangle$, it holds that $\text{R-precision}(\hat{\mathbf{r}}_3) = 0.500 = \text{R-precision}(\hat{\mathbf{r}}_4)$.

The normalized recall [75]:

$$\text{R}_{\text{norm}} = 1 - \frac{\sum_{r=1}^R isrel(r) \cdot r - \sum_{r=1}^R r}{R \cdot (L - R)} ,$$

is an interval scale in the binary case since every relevant document retrieved contributes one unity to the measure.

The normalized precision [75]:

$$P_{\text{norm}} = 1 - \frac{\sum_{r=1}^R isrel(r) \cdot \ln r - \sum_{r=1}^R \ln r}{\ln \frac{L!}{R! (L-R)!}},$$

is a metric, but not an interval scale. It verifies the same property than the normalized recall; however, the logarithm is not a constant increasing function.

The weighted R-precision [53]:

$$\text{R-WP} = \frac{cg(R)}{cig(R)},$$

is a pseudometric since it attains the same values than the R-precision in the binary case. The R-measure [78], defined as:

$$\text{R-measure} = \frac{cg(R) + count(R)}{cig(R) + R},$$

is a pseudometric since it attains the same values than the R-precision in the binary case. In the multi-graded case, the R-measure and R-WP are also pseudometrics since they attain the value 1, for every ranking that all the top R documents are (at least partially) relevant.

The average precision [13, 46]:

$$\text{AP} = \frac{1}{R} \cdot \sum_{r=1}^L isrel(r) \cdot \frac{count(r)}{r},$$

is not a metric since it holds that $\text{AP}(\hat{\mathbf{r}}_1) = 0.250 = \text{AP}(\hat{\mathbf{r}}_3)$. As AP is not an interval scale, then the mean average precision on a set of Q queries, $\text{MAP} = \frac{1}{Q} \sum_{i=1}^Q AP_i$, and the geometric mean average precision, $\text{GMAP} = \exp \frac{1}{Q} \sum_{i=1}^Q \log AP_i$, should not be considered according to the permissible operations on the scale types [91] since they are means of ordinal values. This result confirms the findings of [73]. The average weighted precision [53]:

$$\text{AWP} = \sum_{r=1}^L isrel(r) \cdot \frac{cg(r)}{cig(r)},$$

is not a metric. Considering the rankings $\hat{\mathbf{r}}_1$ and $\hat{\mathbf{r}}_3$, with only two relevant documents to the query, then $\text{AWP}(\hat{\mathbf{r}}_1) = 0.250 = \text{AWP}(\hat{\mathbf{r}}_3)$.

The Q-measure [78], defined as:

$$\text{Q-measure} = \frac{1}{R} \cdot \sum_{r=1}^L isrel(r) \cdot \frac{cg(r) + count(r)}{cig(r) + r},$$

is neither a metric since $\text{Q-measure}(\hat{\mathbf{r}}_1) = 0.250 = \text{Q-measure}(\hat{\mathbf{r}}_3)$ (in the binary case, it attains the same values as AP).

The reciprocal rank at rank r :

$$\text{RR}_r = \text{isrel}(r) \cdot \frac{1}{r},$$

is not a metric since $\text{RR}_4(\hat{\mathbf{r}}_2) = 0.500 = \text{RR}_4(\hat{\mathbf{r}}_3)$. Thus, the expected reciprocal rank [19, 89]:

$$\text{ERR} = \frac{1}{Q} \sum_{i=1}^Q \text{RR}_i,$$

should not be considered according to the permissible operations on the scale types [91] since it is a mean of ordinal values.

The discounted cumulative gain [52, 57]:

$$\text{DCG}_b = \sum_{r=1}^L \frac{g(r)}{\max\{1, \log_b r\}},$$

is not a metric since $\text{DCG}_2(\hat{\mathbf{r}}_1) = 1.000 = \text{DCG}_2(\hat{\mathbf{r}}_2)$.

The graded rank-biased precision [68, 82]:

$$\text{RBP}_p = \frac{1-p}{g(a_c)} \cdot \sum_{r=1}^L p^{i-1} \cdot g(r)$$

is not a metric, in general. For instance, consider the rankings $\langle 1, 0, 0 \rangle$ and $\langle 0, 1, 1 \rangle$, the corresponding scores are $(1+0+0)/(1-p)$ and $(0+p+p^2)/(1-p)$ respectively. Equating these expressions is obtained $1 = p + p^2$ with a real solution. Thus, for this parameter p , there are two rankings with the same score. Therefore, RBP_p is not an interval scale. However, there are particular cases where it is an interval scale since their attained values are equispaced; for instance, $\text{RBP}_{0.5}$ [31].

The binary preference evaluation measure [12]:

$$\text{bpref} = \frac{1}{R} \cdot \sum_{r=1}^L 1 - \frac{r - \text{count}(r)}{R},$$

is a pseudo metric, but not a metric, since it attains the same value for a ranking with one relevant document in the first ranking position and a ranking with two relevant documents in the first and second ranking position.

An extension of the cumulated gain (CG) is the family of retrieval measures XCG [55]. They consider the dependency of XML elements, such as overlap and near-misses. In this paper, an adapted version of these measures is considered, through the definition $xG[r] = g(r)$, for every ranking position, r . The user-oriented measure of normalised extended cumulated gain [56]:

$$\text{nxCG}[r] = \frac{xCG[i]}{xCI[i]} = \frac{cg(r)}{cig(i)},$$

is not a metric since $\text{nxCG}[4](\hat{\mathbf{r}}_1) = 1.000 = \text{nxCG}[4](\hat{\mathbf{r}}_2)$, when there is only one relevant document. Thus, the mean average nxCG at rank r [56]:

$$\text{MANxCG}[r] = \frac{\sum_{j=1}^r \text{nxCG}[j]}{r} = \frac{\sum_{j=1}^r \frac{cg(j)}{cig(j)}}{r}$$

should not be considered according to the permissible operations on the scale types [91]. In addition, considering it as a measure, it is not a metric since $\text{MANxCG}[4](\hat{\mathbf{r}}_1) = 0.250 = \text{MANxCG}[4](\hat{\mathbf{r}}_2)$.

The system-oriented effort-precision / gain-recall [56]:

$$\text{gr}[r] = \frac{\text{xCG}[r]}{\text{xCI}[L]} = \frac{cg(r)}{cig(n)}$$

is not a metric since $\text{gr}[4](\hat{\mathbf{r}}_1) = 1.000 = \text{gr}[4](\hat{\mathbf{r}}_2)$.

Finally, the expected search length [23], defined as:

$$\text{esl} = j + \frac{i \cdot s}{t + 1}$$

where j is the total number of non-relevant documents in all levels preceding the final level; t is the number of relevant documents in the final level; i is the number of non-relevant documents in the final level and s is the number of relevant documents required from the final level to satisfy the need according its type. The esl is not a metric in the Type 2 retrieval since it attains the same value for two rankings, which only differ in the order of the documents of a specific level.

In general, in the non-binary case of many retrieval measures, there are several ways in which different rankings can be awarded the same score. For instance, if the possible qrel values are: $\{a_0 = 0, a_1, a_2, a_3, a_4 = 1\}$, there are three variables to combine them in such a way that distinct rankings map to identical scores. Table 3 provides a summary of the intrinsic properties of these retrieval measures.

7 Conclusions

As indicated in Sect. 1, there are different approaches to determine the scale type of retrieval measures. The first step should be to theoretically ground their arguments, in order to make explicit the assumptions behind retrieval measures. Once these arguments have been correctly justified, then it is possible to assess whether one, both, or any other alternative are valid.

The results obtained here correspond to the representational paradigm, i.e., when the RTM is assumed. This paper has provided a theoretical basis of the metric and scale properties of a retrieval measure, when its empirical domain is not explicitly specified. These properties are deduced from the information contained in the retrieval measure itself, i.e., they are intrinsic properties. A taxonomy and a ready-to-use rule based on the attained values are introduced,

Table 3. Intrinsic properties of some retrieval measures in the rank-based retrieval.

	Ord/Pseudom	Ord/Metr	Interv/Metr
Prec@ r [9,15]	✓		
R-Precision	✓		
sliding ratio [58]	✓		
modified slid. ratio		✓	
R_{norm} [75]			✓ *
P_{norm} [75]		✓	
R-WP [53]	✓		
R-measure [78]	✓		
Avg. Prec. [13,46]	✓		
AWP [53]	✓		
Q-measure [78]	✓		
RR	✓		
DCG $_b$ [57]	✓		
RBP $_p$ [68,82]	✓		
bpref [12]	✓		
nxCG[r] [56]	✓		
MAnxCG[r] [56]	✓		
gr[r] [56]	✓		
esl [23]	✓		

(*)Only in the binary case

and some common user-oriented and system-oriented retrieval measures have been classified according their intrinsic properties. It has been found that the strength of the set-based (first generation of) retrieval measures are their formal properties, most of them are metrics and interval scales. Thus, operations involving order, addition or difference operations among their attained values can be computed, according to the permissible operations on the scale types of Stevens [91]. On the other hand, rank-based (modern) retrieval measures attempt to capture more accurate aspects of systems' usefulness, dropping their formal properties. In general, they are pseudometrics and ordinal scales; thus, only operations involving the order of their attained values should be performed, according to the permissible operations. Thus, retrieval measures face a compromise between satisfying formal properties and capturing the users perception of usefulness.

The intrinsic framework can be useful to determine the metric and scale properties of emergent or existing IR evaluation measures, when the RTM is assumed. In addition, it enables to study other properties that exclusively depend on the retrieval measure itself, which is an interesting subject to be explored in future work.

A Appendix

A.1 Formal Proofs

Proof. (*Proposition 1*) Symmetry is trivially verified since $d_f(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2) = |f(\hat{\mathbf{r}}_1) - f(\hat{\mathbf{r}}_2)| = |f(\hat{\mathbf{r}}_2) - f(\hat{\mathbf{r}}_1)| = d_f(\hat{\mathbf{r}}_2, \hat{\mathbf{r}}_1)$. Triangular inequality is also trivial, by considering the triangular inequality on the real numbers: $|f(\hat{\mathbf{r}}_1) - f(\hat{\mathbf{r}}_2)| \leq |f(\hat{\mathbf{r}}_1) - f(\hat{\mathbf{r}}_3)| + |f(\hat{\mathbf{r}}_3) - f(\hat{\mathbf{r}}_2)|$. \square

Proof. (*Proposition 2*) An interesting result about metric spaces [42] states the following: “*Let (\mathbf{R}_2, d_2) be a metric space and let $f : \mathbf{R}_1 \longrightarrow \mathbf{R}_2$ an an injective or one-to-one function, then (\mathbf{R}_1, d_1) is a metric space, where $d_1(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2) = d_2(f(\hat{\mathbf{r}}_1), f(\hat{\mathbf{r}}_2))$, $\forall \hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2 \in \mathbf{R}_1$.*

In the retrieval scenario, $(\mathbf{R}_2, d_2) = (\mathbb{R}, |\cdot|)$, which is the metric space of the real line endowed with the usual norm (the absolute value). Let f be a one-to-one IR evaluation measure; from the previous result, it follows that $(\mathbf{R}_1, d_1) = (\mathbf{R}, d_f)$ is a metric space, i.e., d_f verifies the three postulates of a metric. \square

Proof. (*Proposition 3*) It will be seen the implication from right to left. Consider a metric ordinal scale, f , where the attained values are equally spaced.

An interval is called *prime* if $[\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2] = \{\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2\}$. First, it will be seen that the function, $F(\mathbf{x}, \mathbf{y}) = |f(\mathbf{x}) - f(\mathbf{y})|$, attains its minimum value on any prime interval.

Let $[\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_3] = \{\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2, \hat{\mathbf{r}}_3\}$ be a non-prime interval, where $\hat{\mathbf{r}}_1 \preceq_f \hat{\mathbf{r}}_2 \preceq_f \hat{\mathbf{r}}_3$, then it holds that $f(\hat{\mathbf{r}}_1) \leq f(\hat{\mathbf{r}}_2) \leq f(\hat{\mathbf{r}}_3)$ since f is an ordinal scale. It implies that $|f(\hat{\mathbf{r}}_3) - f(\hat{\mathbf{r}}_1)| \leq |f(\hat{\mathbf{r}}_3) - f(\hat{\mathbf{r}}_2)| + |f(\hat{\mathbf{r}}_2) - f(\hat{\mathbf{r}}_1)|$, i.e., the minimum value of F is not attained at $[\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_3]$. In addition, it holds that the function F assign the same value for every prime interval. Given a prime interval, $[\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2]$, it can be considered one of its consecutive prime intervals, $[\hat{\mathbf{r}}_2, \hat{\mathbf{r}}_3]$, since \preceq_f is a weak order (every pair of elements is comparable). These two prime intervals verify that $f(\hat{\mathbf{r}}_1) < f(\hat{\mathbf{r}}_2) < f(\hat{\mathbf{r}}_3)$ since f is a metric, and the attained values of f are equally spaced. Thus, it can be assumed that $F(\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2) = k \in \mathbb{R}^+$ for any prime interval $[\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2]$.

Now, it will be seen that equally spaced intervals (not necessarily prime) are assigned equal differences. Consider any non-prime interval, $[\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_m] = \{\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2, \dots, \hat{\mathbf{r}}_m\}$. As f is a metric, then it attains different values for different elements. Thus, it can be assumed that $f(\hat{\mathbf{r}}_1) < f(\hat{\mathbf{r}}_2) < \dots < f(\hat{\mathbf{r}}_{m-1}) < f(\hat{\mathbf{r}}_m)$. Then, every interval $[\hat{\mathbf{r}}_i, \hat{\mathbf{r}}_{i+1}]$ are prime intervals for $i = 1, \dots, m-1$ since F attain the minimum at these intervals. As $f(\hat{\mathbf{r}}_m) - f(\hat{\mathbf{r}}_1) = f(\hat{\mathbf{r}}_m) - f(\hat{\mathbf{r}}_{m-1}) + f(\hat{\mathbf{r}}_{m-1}) - \dots - f(\hat{\mathbf{r}}_2) + f(\hat{\mathbf{r}}_2) - f(\hat{\mathbf{r}}_1)$ and $f(\hat{\mathbf{r}}_{i+1}) - f(\hat{\mathbf{r}}_i) = k$ for $1 \leq i \leq m-1$, then $f(\hat{\mathbf{r}}_1) - f(\hat{\mathbf{r}}_m) = k \cdot m$, which only depends on the span of the interval, m , not on the considered elements. Therefore, equally spaced intervals are assigned equal differences, i.e., f is an interval scale.

Finally, it will be seen the other implication. Consider any prime interval, $[\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2]$, of \mathbf{R} , as f is an interval scale, then equally spaced intervals are assigned to equal differences, i.e., the value $|f(\hat{\mathbf{r}}_2) - f(\hat{\mathbf{r}}_1)|$ is constant for every prime interval of \mathbf{R} . In addition, it should be an strictly positive value. To see that the

attained values are equally spaced, it is sufficient to check that different elements of \mathbf{R} are assigned different values of f , which is hold since f is a metric. \square

References

1. Allan, J., Aslam, J., Belkin, N., Buckley, C., Callan, J., Croft, B., Dumais, S., Fuhr, N., Harman, D., Harper, D.J., et al.: Challenges in information retrieval and language modeling: report of a workshop held at the center for intelligent information retrieval, university of massachusetts amherst, september 2002. In: ACM SIGIR Forum. 1, pp. 31–47. ACM New York, NY, USA (2003)
2. Amigó, E., Gonzalo, J., Artiles, J., Verdejo, F.: A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retrieval* **12**(4), 461–486 (2009)
3. Amigo, E., Gonzalo, J., Mizzaro, S.: What is my problem identifying formal tasks and metrics in data mining on the basis of measurement theory. *IEEE Trans. Knowl. Data Eng.* (2021)
4. Amigó, E., Gonzalo, J., Verdejo, F.: A general evaluation measure for document organization tasks. In: Proceedings of the 36th international ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 643–652 (2013)
5. Amigó, E., Mizzaro, S.: On the nature of information access evaluation metrics: a unifying framework. *Inf. Retr. J.* **23**(3), 318–386 (2020)
6. Azzopardi, L., Thomas, P., Craswell, N.: Measuring the utility of search engine result pages: an information foraging based measure. In: The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 605–614 (2018)
7. Baccianella, S., Esuli, A., Sebastiani, F.: Evaluation measures for ordinal regression. In: 2009 Ninth International Conference on Intelligent Systems Design and Applications, pp. 283–287. IEEE (2009)
8. Belew, R.K.: Finding Out About: A Cognitive Perspective on Search Engine Technology and the WWW. Cambridge University Press (2000)
9. Blair, D.C.: Information retrieval, 2nd ed. C.J. van rijsbergen. London: Butterworths. *JASIS* **30**(6), 374–375 (1979). <https://doi.org/10.1002/asi.4630300621>
10. Bollmann, P.: Two axioms for evaluation measures in information retrieval. In: SIGIR, vol. 84, pp. 233–245. Citeseer (1984)
11. Bollmann, P., Cherniavsky, V.S.: Measurement-theoretical investigation of the mz-metric. In: Proceedings of the 3rd Annual ACM Conference on Research and Development in Information Retrieval, pp. 256–267. Citeseer (1980)
12. Buckley, C., Voorhees, E.M.: Retrieval evaluation with incomplete information. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 25–32 (2004)
13. Buckley, C., Voorhees, E.M.: Evaluating evaluation measure stability. In: ACM SIGIR Forum. 2, pp. 235–242. ACM New York, NY, USA (2017)
14. Busin, L., Mizzaro, S.: Axiometrics: An axiomatic approach to information retrieval effectiveness metrics. In: Proceedings of the 2013 Conference on the Theory of Information Retrieval, pp. 22–29 (2013)
15. Büttcher, S., Clarke, C.L., Yeung, P.C., Soboroff, I.: Reliable information retrieval evaluation with incomplete and biased judgements. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 63–70 (2007)

16. Carmel, D., Yom-Tov, E.: Estimating the query difficulty for information retrieval. *Synth. Lect. Inf. Concepts, Retr., Serv.* **2**(1), 1–89 (2010)
17. Carterette, B.: System effectiveness, user models, and user utility: a conceptual framework for investigation. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 903–912 (2011)
18. Carterette, B.A.: Multiple testing in statistical analysis of systems-based information retrieval experiments. *ACM Trans. Inf. Syst. (TOIS)* **30**(1), 1–34 (2012)
19. Chapelle, O., Metlzer, D., Zhang, Y., Grinspan, P.: Expected reciprocal rank for graded relevance. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 621–630 (2009)
20. Cleverdon, C.W.: The significance of the cranfield tests on index languages. In: Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 3–12 (1991)
21. Clinchant, S., Gaussier, E.: Is document frequency important for prf? In: Conference on the Theory of Information Retrieval, pp. 89–100. Springer, Berlin (2011)
22. Clinchant, S., Gaussier, E.: A theoretical analysis of pseudo-relevance feedback models. In: Proceedings of the 2013 Conference on the Theory of Information Retrieval, pp. 6–13 (2013)
23. Cooper, W.S.: Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *Am. Doc.* **19**(1), 30–41 (1968)
24. Croft, W.B., Metzler, D., Strohman, T.: Search Engines: Information Retrieval in Practice, vol. 520. Addison-Wesley Reading (2010)
25. Do Carmo, M.P.: Differential Geometry of Curves and Surfaces: Revised and Updated, 2nd edn. Courier Dover Publications (2016)
26. Fang, H.: An axiomatic approach to information retrieval. Technical report (2007)
27. Fang, H., Tao, T., Zhai, C.: A formal study of information retrieval heuristics. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 49–56 (2004)
28. Fang, H., Tao, T., Zhai, C.: Diagnostic evaluation of information retrieval models. *ACM Trans. Inf. Syst. (TOIS)* **29**(2), 1–42 (2011)
29. Fang, H., Zhai, C.: An exploration of axiomatic approaches to information retrieval. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 480–487 (2005)
30. Fang, H., Zhai, C.: Semantic term matching in axiomatic approaches to information retrieval. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 115–122 (2006)
31. Ferrante, M., Ferro, N., Fuhr, N.: Towards meaningful statements in ir evaluation: Mapping evaluation measures to interval scales. *IEEE Access* **9**, 136,182–136,216 (2021)
32. Ferrante, M., Ferro, N., Fuhr, N.: Response to moffat’s comment on “towards meaningful statements in ir evaluation: Mapping evaluation measures to interval scales” (2022). <https://doi.org/10.48550/ARXIV.2212.11735>
33. Ferrante, M., Ferro, N., Pontarollo, S.: A general theory of ir evaluation measures. *IEEE Trans. Knowl. Data Eng.* **31**(3), 409–422 (2018)
34. Ferro, N., Peters, C.: Information Retrieval Evaluation in a Changing World: Lessons Learned from 20 Years of CLEF, vol. 41. Springer, Berlin (2019)

35. Flach, P.: Performance evaluation in machine learning: the good, the bad, the ugly, and the way forward. In: Proceedings of the AAAI Conference on Artificial Intelligence, 01, pp. 9808–9814 (2019)
36. Fraleigh, J.B.: A First Course in Abstract Algebra. Pearson Education India (2003)
37. Fréchet, M.M.: Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo* (1884–1940) **22**(1), 1–72 (1906)
38. Fuhr, N.: Some common mistakes in ir evaluation, and how they can be avoided. In: ACM SIGIR Forum. 3, pp. 32–41. ACM New York, NY, USA (2018)
39. Gaudette, L., Japkowicz, N.: Evaluation methods for ordinal classification. In: Canadian Conference on Artificial Intelligence, pp. 207–210. Springer, Berlin (2009)
40. Gauss, C.F.: *Disquisitiones Generales Circa Superficies Curvas*, vol. 1. Typis Dieterichianis (1828)
41. Giner, F.: A comment to “a general theory of ir evaluation measures” (2023). [arXiv:2303.16061](https://arxiv.org/abs/2303.16061)
42. Guccione, J.A.: Espacios métricos. Universidad de Buenos Aires, Texto (2018)
43. Han, L., Roitero, K., Maddalena, E., Mizzaro, S., Demartini, G.: On transforming relevance scales. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 39–48 (2019)
44. Hand, D.J.: Statistics and the theory of measurement. *J. R. Stat. Soc. A. Stat. Soc.* **159**(3), 445–473 (1996)
45. Harman, D.: Information retrieval evaluation. *Synth. Lect. Inf. Concepts, Retr., Serv.* **3**(2), 1–119 (2011)
46. Hauff, C., de Jong, F.: Retrieval system evaluation: Automatic evaluation versus incomplete judgments. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 863–864 (2010)
47. Hausdorff, F.: Set Theory, vol. 119. American Mathematical Soc. (2005)
48. Huibers, T.W.C.: An axiomatic theory for information retrieval. Ph.D. thesis (1996)
49. Hull, D.: Using statistical testing in the evaluation of retrieval experiments. In: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 329–338 (1993)
50. Hungerford, T.W.: Algebra, vol. 73. Springer Science & Business Media (2012)
51. Jacobson, N.: Basic Algebra I. Courier Corporation (2012)
52. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst. (TOIS)* **20**(4), 422–446 (2002)
53. Kando, N.: Information retrieval system evaluation using multi-grade relevance judgments-discussion on averageable single-numbered measures. *IPSJ SIG Notes* **63**, 105–112 (2001)
54. Karimzadehgan, M., Zhai, C.: Axiomatic analysis of translation language model for information retrieval. In: European Conference on Information Retrieval, pp. 268–280. Springer, Berlin (2012)
55. Kazai, G.: Report of the inex 2003 metrics working group. In: Initiative for the Evaluation of XML Retrieval (INEX): INEX 2003 Workshop Proceedings, Dagstuhl, Germany (2004)
56. Kazai, G., Lalmas, M.: Inex 2005 evaluation measures. In: Fuhr, N., Lalmas, M., Malik, S., Kazai, G. (eds.) *Advances in XML Information Retrieval and Evaluation*, pp. 16–29. Springer, Berlin (2006)

57. Kekäläinen, J., Järvelin, K.: Using graded relevance assessments in ir evaluation. *J. Am. Soc. Inform. Sci. Technol.* **53**(13), 1120–1129 (2002)
58. Korfhage, R.R.: *Information Storage and Retrieval*. Wiley, USA (1997)
59. Krantz, D., Luce, D., Suppes, P., Tversky, A.: *Foundations of Measurement*, vol. I: Additive and Polynomial Representations (1971)
60. Krantz, D.H.: *Foundations of Measurement*, vol. II. Geometrical, Threshold and Probabilistic Representations (1989)
61. Luce, D., Krantz, D., Suppes, P., Tversky, A.: *Foundations of Measurement*, Vol. III Representation, Axiomatization, and Invariance (1990)
62. Maddalena, E., Mizzaro, S.: Axiometrics: Axioms of information retrieval effectiveness metrics. In: EVIA@ NTCIR (2014)
63. Michell, J.: Measurement scales and statistics: a clash of paradigms. *Psychol. Bull.* **100**(3), 398 (1986)
64. Michell, J.: *An Introduction to the Logic of Psychological Measurement*. Psychology Press (2014)
65. Moffat, A.: Seven numeric properties of effectiveness metrics. In: Asia Information Retrieval Symposium, pp. 1–12. Springer, Berlin (2013)
66. Moffat, A.: Batch evaluation metrics in information retrieval: Measures, scales, and meaning. *IEEE Access* **10**, 105, 564–105,577 (2022)
67. Moffat, A., Bailey, P., Scholer, F., Thomas, P.: Incorporating user expectations and behavior into the measurement of search effectiveness. *ACM Trans. Inf. Syst. (TOIS)* **35**(3), 1–38 (2017)
68. Moffat, A., Zobel, J.: Rank-biased precision for measurement of retrieval effectiveness. *ACM Trans. Inf. Syst. (TOIS)* **27**(1), 1–27 (2008)
69. Montazeralghaem, A., Zamani, H., Shakery, A.: Axiomatic analysis for improving the log-logistic feedback model. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 765–768 (2016)
70. Pollock, S.M.: Measures for the comparison of information retrieval systems. *Am. Doc.* **19**(4), 387–397 (1968)
71. Rahimi, R., Montazeralghaem, A., Shakery, A.: An axiomatic approach to corpus-based cross-language information retrieval. *Inf. Retr. J.* **23**(3), 191–215 (2020)
72. Roberts, F.S.: Measurement theory. *Encycl. Math. Appl.* **7** (1985)
73. Robertson, S.: On gmap: and other transformations. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pp. 78–83 (2006)
74. Robertson, S.: On the history of evaluation in ir. *J. Inf. Sci.* **34**(4), 439–456 (2008)
75. Rocchio, J.: Performance indices for document retrieval systems. In: *Information Storage and Retrieval* p. 83 (1964)
76. Rosset, C., Mitra, B., Xiong, C., Craswell, N., Song, X., Tiwary, S.: An axiomatic approach to regularizing neural ranking models. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 981–984 (2019)
77. Sagara, Y.: Performance measures for ranked output retrieval systems. *J. Jpn. Soc. Inf. Knowl.* **12**(2), 22–36 (2002)
78. Sakai, T.: New performance metrics based on multigrade relevance: their application to question answering. In: NTCIR (2004)
79. Sakai, T.: Metrics, statistics, tests. In: PROMISE Winter School, pp. 116–163. Springer, Berlin (2013)
80. Sakai, T.: Statistical reform in information retrieval? In: ACM SIGIR Forum, vol. 48, pp. 3–12. ACM, New York, NY, USA (2014)

81. Sakai, T.: On fuhr's guideline for ir evaluation. In: ACM SIGIR Forum, vol. 54, pp. 1–8. ACM, New York, NY, USA (2021)
82. Sakai, T., Kando, N.: On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Inf. Retr.* **11**(5), 447–470 (2008)
83. Sakai, T., Oard, D.W., Kando, N.: Evaluating Information Retrieval and Access Tasks: NTCIR's Legacy of Research Impact. Springer Nature (2021)
84. Salton, G.: Automatic Information Organization and Retrieval (1968)
85. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill (1983)
86. Sanderson, M.: Test collection based evaluation of information retrieval systems. *Found. Trends Inf. Retr.* **4**(4), 247–375 (2010)
87. Savoy, J.: Statistical inference in retrieval effectiveness evaluation. *Inf. Process. Manag.* **33**(4), 495–512 (1997)
88. Sebastiani, F.: An axiomatically derived measure for the evaluation of classification algorithms. In: Proceedings of the 2015 International Conference on the Theory of Information Retrieval, pp. 11–20 (2015)
89. Sirotnik, P.: On search engine evaluation metrics (2013). [arXiv:1302.2318](https://arxiv.org/abs/1302.2318)
90. Stevens, S.S.: Mathematics, Measurement, and Psychophysics. Wiley, New York (1951)
91. Stevens, S.S., et al.: On the Theory of Scales of Measurement. Bobbs-Merrill, College Division (1946)
92. Swets, J.A.: Information retrieval systems. *Science* **141**(3577), 245–250 (1963)
93. Urbano, J., Lima, H., Hanjalic, A.: Statistical significance testing in information retrieval: an empirical analysis of type i, type ii and type iii errors. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 505–514 (2019)
94. Van Rijsbergen, C.J.: Foundation of evaluation. *J. Doc.* **30**(4), 365–373 (1974)
95. Vanbelle, S., Albert, A.: A note on the linearly weighted kappa coefficient for ordinal scales. *Stat. Methodol.* **6**(2), 157–163 (2009)
96. Velleman, P.F., Wilkinson, L.: Nominal, ordinal, interval, and ratio typologies are misleading. *Am. Stat.* **47**(1), 65–72 (1993)
97. Voorhees, E.M.: The trec 2005 robust track. In: ACM SIGIR Forum, vol. 40, pp. 41–48. ACM, New York, NY, USA (2006)
98. Voorhees, E.M., Harman, D.K.: TREC: Experiment and Evaluation in Information Retrieval, vol. 63. Citeseer (2005)
99. Voorhees, E.M., et al.: Overview of the trec 2003 robust retrieval track. In: Trec, pp. 69–77 (2003)
100. Wicaksono, A.F., Moffat, A.: Metrics, user models, and satisfaction. In: Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 654–662 (2020)
101. Zhang, F., Liu, Y., Li, X., Zhang, M., Xu, Y., Ma, S.: Evaluating web search with a bejeweled player model. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 425–434 (2017)



Substructure Discovery in Commonsense Relations Using Graph Representation Learning

Ke Shen and Mayank Kejriwal^(✉)

Information Sciences Institute, University of Southern California, Marina del Rey,
California 90292, USA
kejriwal@isi.edu

Abstract. Acquiring commonsense knowledge and reasoning is an important goal in modern natural language processing research. Despite much progress, there is still a lack of understanding (especially at scale) of the nature of commonsense knowledge itself. A potential source of structured commonsense knowledge that could be used to derive insights is ConceptNet. In particular, ConceptNet contains several coarse-grained relations, including ‘HasContext’, ‘FormOf’ and ‘SymbolOf’, which can prove invaluable in understanding broad, but critically important, commonsense notions such as ‘context’. In this article, we present a methodology based on unsupervised knowledge graph representation learning and clustering to reveal and study substructures in three coarse-grained and heavily used relations in ConceptNet. Our results show that, despite having an ‘official’ definition in ConceptNet, many of these coarse-grained relations exhibit considerable *substructure*. In the future, therefore, such relations could be sub-divided into other relations with more refined definitions. We also supplement our core study with visualizations and qualitative analyses.

Keywords: Commonsense · ConceptNet · Context · Knowledge graph · Knowledge graph embedding

1 Introduction

Despite the ubiquity of intelligent agents such as Alexa and Siri in modern life, these agents have yet to capture the human element in natural conversations. Even with advances in Natural Language Processing (NLP), deep learning, and knowledge graphs [14, 37], it is not clear if such agents are fully capable of answering questions (e.g., ‘Should I put my spare change in a piggy bank?’) with incomplete information, or under-specified needs. Such questions tend to require more contextual and implicit knowledge that humans often take for granted when navigating daily situations. Among other things, lack of deep contextual

understanding limits the agents' *commonsense reasoning* abilities. Even the most recent large language models (such as OpenAI's ChatGPT system) have trouble explaining why some of their answers are correct, and which ones require more surrounding context [24, 50, 51].

Commonsense reasoning is the process that involves processing information about a scenario in the world, and making inferences and decisions by using context, implicit and explicit information based on our collective 'commonsense knowledge'. Commonsense knowledge is difficult to define precisely but it is usually assumed to be a broad body of knowledge of how the 'world' works [33]. Generally, such knowledge is essential for navigating social situations and interactions, 'naive' physical understanding (e.g., the simple knowledge that when an object on the table is 'picked up', it is not on the table anymore) and more controversially, knowledge that relies on reasoning about local culture and milieu [28, 42].

Commonsense knowledge and reasoning have both been recognized as essential for building more advanced 'general' AI systems that have human-like capabilities and reasoning ability when facing uncertain, implicit, or even potentially contradictory, information [18]. Recognizing its importance, researchers in several communities have increasingly engaged in improving agent performance on commonsense question answering, abductive reasoning and other tasks pertinent to commonsense reasoning [5, 39, 42, 43].

ConceptNet¹ is a large-scale, freely available knowledge graph (KG) that describes commonsense knowledge as a set of assertions or *triples* [49]. It is designed to represent the common knowledge needed to help machines better understand the meanings of concepts and inter-concept relationships that people rely on in everyday situations. The *graph structure* that represents knowledge in ConceptNet is particularly useful for textual reasoning over natural language documents. An example of how such knowledge is organized in ConceptNet is illustrated in Fig. 1. ConceptNet originated from the Open Mind Common Sense [13] project, itself launched in 1999 at the MIT Media Lab. It was regularly updated to include new knowledge from crowdsourced resources, expert-curated resources, and 'games with a purpose' designed specifically to elicit certain kinds of commonsense annotations from people, such as intuitive word associations.

Our guiding hypothesis in this paper is that, due to its growth and usage over the years, ConceptNet can yield valuable insights about commonsense knowledge itself. These insights can then be used to define more context-friendly systems and benchmarks [19], develop richer theories of commonsense reasoning, and semi-automatically refine ConceptNet.

The intuition behind our approach is relatively straightforward. First, using representation learning, we 'embed' each entity and relation into a dense, continuous real-valued vector space, learned in an unsupervised fashion using a state-of-the-art KG representation learning algorithm. Next, we isolate the *triples* [or labeled edges, such as (people, CapableOf, socialize_at_night) in Fig. 1] of interest from the raw KG. For example, we isolate the triples with relation 'hasContext'

¹ <https://conceptnet.io/>.

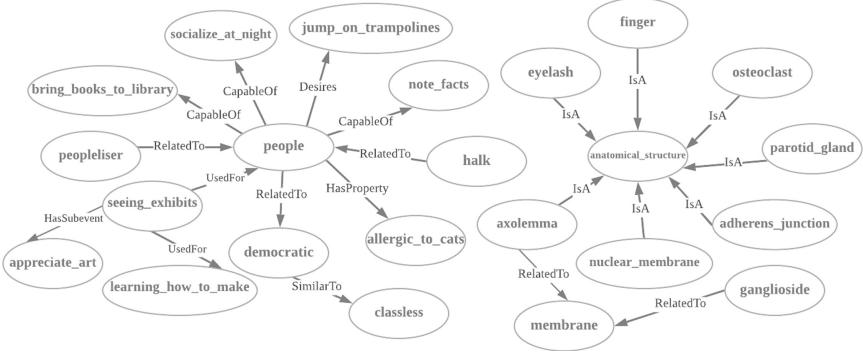


Fig. 1. A fragment of the ConceptNet Knowledge Graph (KG)

if we want to study that relation. Finally, using the learned embeddings from the first step, we derive an embedding for the triple using the notion of *translation* that is more formally described subsequently. Once such embeddings have been obtained for each triple with the relation under study, we cluster them using an established algorithm such as k-Means to detect potential substructures. We also use statistical and quantitative measures to understand the quality and structural properties of these substructures. Finally, visualization and sampling-based qualitative analyses are used to provide more insight into the data. Specific contributions are enumerated below:

- We present a novel methodology for understanding structural aspects of commonsense knowledge by studying three relations ('HasContext', 'SymbolOf', and 'FormOf'). These relations are both *high-volume* and *coarse-grained*, and are known to be significant in commonsense reasoning, explaining their high prevalence in ConceptNet. Our methodology relies on a judicious combination of KG representation learning, clustering and visualization.
- We conduct a detailed set of experiments by applying the proposed methodology on a large sample of triples that contain the three relations mentioned earlier. Using quantitative metrics for measuring cohesion and separation in the embedding space, our results show that, despite having an 'official' definition in ConceptNet, many of these broad relations exhibit considerable *substructure*. In the future, therefore, such relations could be 'divided' into other relations with more refined definitions.
- Using qualitative and sampling-based analyses, we show how a broad notion like 'context' is defined in practice in commonsense KGs such as ConceptNet. These analyses potentially allow us to develop a deeper understanding of the definition and role of context in commonsense knowledge.

While our goal is not to present a complete understanding of commonsense, we study the three relations noted earlier ('HasContext', 'SymbolOf', and 'FormOf') to understand relations that are believed to be critical to commonsense

reasoning. Our methodology and empirical study also relies on established methods from the KG representation learning and clustering literature. An explicit goal behind using established tools, besides their expected reliability, is to enable greater replicability for studying other such relations in large KGs.

2 Related Work

Although both Wikipedia² and the (relatively) recent Defense Advanced Research Projects Agency (DARPA) Machine Common Sense (MCS) program³ define commonsense reasoning as ‘the basic ability to perceive, understand, and judge things that are shared by nearly all people and can be reasonably expected of nearly all people without need for debate’, there is no official, sufficiently broad definition that we can cite outside of the psychology context. Within psychology, [47] has defined commonsense as ‘the system of implications shared by the competent users of a language’. One important commonality that is shared, no matter the definition or field considered, is that commonsense knowledge is (at least to a degree) *implicit*.

There is very little work on the *nature* of commonsense knowledge: one rare example is a recent book [9] that attempts to provide a ‘theory’ of commonsense by breaking down commonsense knowledge into various categories, and present formalisms for those categories. This paper takes an inductive, data-driven approach to understanding commonsense. Our work is potentially complementary to [9], since some of our findings may be used in the future to provide evidence for (or against) some of their purely theoretical claims. Other work in computer science that has taken a similar inductive approach (albeit involving human annotators) includes [40].

Progress has been achieved in specific *kinds* of commonsense reasoning, especially in reasoning about time and internal relations, reasoning about actions and change, and the sign calculus. Semantics have played an important role in some of these successes; in fact, ConceptNet itself has been described as a ‘semantic network’ [12]. We refer the interested reader to works such as [5, 34, 36, 38] for more details. A similar example is WordNet, whose entries are organized in terms of semantic inter-relationships. The easy-to-use network structure lends it to being frequently applied in multiple reasoning systems [3]. Other relevant areas in AI that could potentially benefit from this work include graph-based models and meta-heuristics [31].

Commonsense reasoning systems are typically measured against benchmark tests, such as question answering and abductive reasoning, although many questions have been raised about their efficacy [21, 22, 44, 45]; however, our emphasis in this article is on understanding the nature of commonsense by studying a knowledge base such as ConceptNet, rather than building a reasoning system that performs well on benchmarks, or designing new benchmarks [41].

² https://en.wikipedia.org/wiki/Common_sense.

³ <https://www.darpa.mil/program/machine-common-sense>.

While KGs have been extensively used for representing encyclopedic knowledge and other domains [20], including COVID-19 and even human trafficking [17, 23], ConceptNet and Cyc are the only two examples that we are aware of that serve as commonsense KGs [28, 48]. Owned by Cycorp, Cyc’s knowledge base claims⁴ to be the ‘broadest, deepest, and most complete repository ever developed’, but given its proprietary nature, the claim is difficult to validate. It is not known how similar the commonsense content of Cyc is to that of ConceptNet, but potentially, the same methodology proposed in this paper could be applied to Cyc to replicate, strengthen or refute some of our findings. ConceptNet also expresses knowledge in an easy-to-use format (i.e., as sets of 3-tuple assertions, illustrated as edges in Fig. 1) rather than in higher-order logic notation. Because of these advantages, ConceptNet has rapidly emerged as a practical dataset and resource for various kinds machine learning and NLP applications over the last decade [4, 30, 46].

Compared with WordNet [32], which focuses on preserving lexicographic information and the relationship between words and their senses, ConceptNet maintains a semantic network structure that is designed to capture commonsense assertions. In particular, ConceptNet contains more relations than WordNet. For this reason perhaps, it serves as an important background resource for commonsense reasoning and question answering. However, it has not been studied directly for gaining insights into commonsense knowledge, even though there is precedent. For example, studies on DBpedia and YAGO have been conducted specifically to understand their relational structure and the structural properties of the encyclopedic knowledge that these KGs are known for [7]. We attempt to do the same, but with commonsense knowledge as the focus.

Another line of work highly related to this one is *representation learning*, especially as they pertain to KGs. Such algorithms take as input the KG, including entities and relations, and embed them into continuous vector spaces, as surveyed by [53]. In this article, we use a graph embedding package that builds on the ones above (especially TransE) and is especially designed for graphs with many millions of triples.

Finally, we note recent advances in commonsense reasoning and question answering by using transformer-based language representation learning models such as Bidirectional Encoder Representations from Transformers (BERT) [6], GPT-3 [8], and several others. While these advances have led to state-of-the-art performance across NLP tasks, of which question answering is only one example, they do not help us understand the nature of commonsense reasoning itself. Developing such an understanding is necessary both as a scientific mode of investigation into human commonsense (which is ubiquitous), and to further improve transformers and other neural models to be more *interpretable* when producing answers to questions. By applying a systematic and empirically grounded methodology on a dataset like ConceptNet, this paper aims to provide more insight about commonsense as a fundamental phenomenon.

⁴ <https://www.cyc.com/>.

3 Materials and Methods

Our guiding principle in this paper is that a commonsense KG such as ConceptNet could be used as the basis for understanding some of the structural properties of commonsense knowledge. Conceptually, ConceptNet can be defined as a multi-relational graph $G = (V, R, E)$, where V is the set of entities or *nodes*, R is the set of 49 *relations* and E is the set of triples or *edges* where each triple $e = (h, r, t) \in E \subseteq V \times R \times V$. While G can also be thought of in a non graph-theoretic way as a *set* of triples,⁵ the distinction is not relevant for the purposes of this paper, and we use knowledge bases and graphs interchangeably. However, it is more natural to think about ConceptNet as a graph due to its structural properties. Earlier, Fig. 1 expressed a fragment of ConceptNet as a multi-relational graph with 21 edges, or alternatively, 21 triples.

The symbols, h and t , are respectively referred to as the *source* (or the *head*) and the *destination* (or the *tail*) entity, and r is the relation. For the purposes of maintaining consistent terminology, we use the terms *triple*, *head*, *tail*, and *relation* to refer to the elements e , h , t and r respectively. Where applicable, these head and tail entities are collectively referred to as *entities*.

Entities and relations are projected into a continuous vector space by PyTorch-BigGraph (PBG) for further computation and analysis [29]. Next, we briefly introduce the PBG system. We then describe the raw data and our sampling methodology, followed by the setup and training of the PBG system on our sample.

3.1 PyTorch-BigGraph (PBG) System

PBG is an efficient and recent embedding system for learning real-valued vector representations of nodes and edges (referred to as ‘entities’ and ‘relations’ in the context of KGs like ConceptNet) in large-scale graphs. It was developed and publicly released⁶ by Facebook AI Research. It is able to deal with large-scale graphs because it relies on distributed computing, in addition to other high-scale techniques, such as graph partitioning and batched negative sampling. At present, it also supports GPU training.

PBG operates by first training on an input graph, which can be a knowledge graph with labeled edges or ‘relations’. Similar to other knowledge graph representation learning algorithms like TransE and RESCAL [20], it outputs embeddings by optimizing an objective, whereby unconnected entities are pushed further apart in the vector space, and entities that are ‘adjacent’ (connected via an edge) are pushed closer to each other.

Compared to network embeddings, such as DeepWalk and node2vec [10, 35], PBG supports multi-entity and multi-relation graph embeddings. Its

⁵ This definition is sometimes used in the early natural language community when discussing these datasets as *knowledge bases* rather than (multi-relational) *knowledge graphs*, where triples are usually interpreted as directed edges [16].

⁶ <https://github.com/facebookresearch/PyTorch-BigGraph>.

embedding quality has been found to be comparable with (or even exceed) existing KG embedding systems, evaluated on the Freebase, LiveJournal and YouTube graphs.

3.2 Raw Data, Sampling and Representation Learning

We use ConceptNet 5.7 for the empirical study in this paper.⁷ One important aspect of the dataset is the ratio of the number of unique entities to the total number of triples, which is much higher in ConceptNet (0.833) than in other similarly-sized KGs such as Freebase (0.055) or WordNet. Additionally, while ConceptNet tends to contain more entities than ‘encyclopedic’ KGs such as Freebase or DBpedia, there are fewer unique relations.

In practice, these significant deviations in expected entity/relation ratios can cause problems for KG representation learning packages, even those designed for large-scale KGs such as the PBG system. For example, while embedding the full ConceptNet knowledge base, we found that, during the training process, the PBG algorithm fails with a ‘bus error’ message if the number of input triples exceeds 4 million. This occurs due to the algorithm running out of shared memory (despite the fact that we execute the algorithm on a machine with 60+ GB memory). To address the memory issue and ensure that our results can be extended or replicated in the future using reasonable computation resources, we randomly sampled 4 million triples from ConceptNet for this study.

This set of triples contains more than 3.93 million entities in total, including 2.78 million and 1.38 million head and tail entities respectively. The head entities-set is twice the size of the tail entities-set and the overlap between them is about 235,000 entities. Cursory analysis also showed that the head entity ‘/c/en/person’ and tail entity ‘/c/fr/francais’ were found to have the most triples associated with them. ‘/r/RelatedTo’ was found to be the most frequent relation, occurring in more than 1 million triples. Other relation-specific statistics are tabulated in Table A1 in the online appendix.⁸

We input these 4 million triples into the PBG algorithm for representation learning. We partition the 4 million sampled triples into training, validation and test datasets, containing 75, 12.5, 12.5% of the total triples, respectively. Before doing the sampling, we remove triples with the ‘ExternalURL’ relation. *ExternalURL* is a ‘non-semantic’ relation that is only referring to a URL identifier and cannot be used for structural analysis of the kind proposed in this paper. Finally, we train and validate PBG on a single server in the Amazon cloud with 4 Intel Xeon cores, with one socket and 61 GB of RAM. After training is concluded, the algorithm outputs a single vector for each unique relation and entity in the training dataset. In the next section, we discuss the validation of the quality of these embeddings.

⁷ Downloaded at <https://github.com/commonsense/conceptnet5/wiki/Downloads>.

⁸ Accessed at <https://docs.google.com/document/d/1HCaeOngIQCm3xBVAFFVrgrhQ6SLKuUHvlm3Pk2gK-BA/edit?usp=sharing>.

3.3 Validating Quality of Embeddings

Due to the sampling described earlier, a reasonable question arises as to whether the *quality* of the learned representations or ‘embeddings’ output by PBG can be trusted. We propose a novel quantitative methodology to validate the quality and effectiveness of these embeddings. Specifically, we first compute a *centroid vector* for each relation, as described below. Recall that we denoted the graph using the symbol $G = (V, R, E)$, where E was the set of triples or ‘edges’ in the graph. In a slight abuse of notation, we use the symbol G_E to represent the set E associated with G .

Given a relation $r \in R$, let $G_r \subseteq G_E$ be the subset of triples in G_E with relation r . For each such triple (h, r, t) in G_r , we define the *translation vector* $\mathbf{v} = \mathbf{t} - \mathbf{h}$, where \mathbf{t} and \mathbf{h} are the embeddings output by PBG for entities t and h , respectively. The *centroid vector* \mathbf{r}_c of r is defined simply as the mean of the translation vectors in G_r :

$$\mathbf{r}_c = \frac{1}{|G_r|} \sum_{(h, r, t) \in G_r} (\mathbf{t} - \mathbf{h}) \quad (1)$$

Note that this yields two distinct vectors for r : the vector ‘directly’ output by the graph embedding (denoted as \mathbf{r}) and the centroid vector \mathbf{r}_c . We use the symbol \mathcal{R} to indicate the set of directly output embeddings for all 49 relations and the symbol \mathcal{R}_c to indicate the set of (derived) centroid vectors.

With this technical machinery in place, we validate our 4 million-triples sample as follows. First, we calculate two *similarity lists*, SL_r and SL'_r , per relation, using each of these two notions of embedding a relation. Specifically, SL_r is constructed as a list of the cosine similarities between \mathbf{r} and *each* translation vector⁹ in G_r . The number of entries in SL_r will equal $|G_r|$. Similarly, SL'_r is constructed as a list of cosine similarities between the centroid vector \mathbf{r}_c and each translation vector, and also has size $|G_r|$. Furthermore, if we impose an arbitrary ordering on the triples in G_r the two similarity lists are *aligned* by virtue of the common translation vectors computed over triples in G_r . However, in the general case, the values in SL_r and SL'_r will differ since the former relies on the direct embedding of r in its construction, while the latter relies on the centroid vector \mathbf{r}_c .

Given these two per-relation lists, we establish that the two lists are, in fact, highly correlated. The Spearman’s rank correlation is designed to measure both the strength and direction of association between two ranked variables and ranges from -1 (perfect negative correlation) to 1 (perfect positive correlation). Because of the geometric features of the embedding space, we are interested in *whether* there is correlation (i.e. the strength), rather than the direction of the correlation. For this reason, given the two aligned lists per relation (SL_r and SL'_r), we computed the *absolute value* of the Spearman’s rank correlation for each relation, in Table 1. As expected, some of the correlations are indeed

⁹ Note that, unlike \mathbf{r} , the translation vector (defined earlier as $\mathbf{t} - \mathbf{h}$) clearly depends on the triple.

negative. Specifically, of the 49 relations, 24 relations have a Spearman’s rank correlation greater than 0.6, while 25 other relations have negative correlations (approximately 50%, as would be statistically expected). However, in no case is the absolute value less than 0.4.

Since r_c is a function of the entities in the triples, and never uses the direct embedding r output by PBG, this result serves as an independent check on the quality of the embeddings. The high absolute correlations show that, not only are the embeddings learned on our sample self-consistent, but also that they conform closely to the notion of translation that is an important feature of neural graph embeddings [2]. In contrast, if r had showed little or no correlation (compared to r_c), it would have begged the question about whether the embeddings were learned by PBG in a sufficiently non-random way that, at least approximately, model the translation operation in vector space. Furthermore, to ensure the results are not an artifact of using Spearman’s correlation, we replicated it using an alternate measure (KL-Divergence), with similar conclusions. That is, the distributions of $vecr_c$ and r_c were found to exhibit low KL-Divergence for all relations.¹⁰

3.4 Vectorizing and Clustering Relation-Specific Triples

Certain relations in ConceptNet are deliberately designed to be broad. A good example is the *HasContext* relation, which is defined on the ConceptNet website as: *A HasContext B* is declared in the knowledge base if ‘A is a word used in the context of B, which could be a topic area, technical field, or regional dialect’. In this article, we investigate the hypothesis that, despite being originally defined so broadly, there is considerable *substructure* in such relations. In considering the definition of *HasContext* above, multiple contexts are suggested e.g., *technical field*, *regional dialect*, and presumably, other contexts that may be similar to these explicit cases. Another example is a relation such as *FormOf*, where a triple *A FormOf B* may be declared if ‘A is an inflected form of B; B is the root word of A’. Even the basic official definition *suggests* breadth, since A could either be an ‘inflected’ form of B, or the ‘root word’ of B. Furthermore, there is nothing in the definition that places a strict constraint on such triples, either in theory or in practice.

Since ConceptNet is crowdsourced to a great extent, it is quite likely that many people have interpreted these relations at ‘face value’ i.e., in accordance with what one would understand their ‘everyday’ meaning to be. Therefore, our goal is to measure and describe the *empirical* substructures, if any, in these three specific relations (*HasContext*, *FormOf* and *SymbolOf*¹¹) using a systematic methodology. An important aspect of these three relations is not just that they are defined broadly and are *coarse-grained*, but are also relatively *high-volume*.

¹⁰ As the conclusions are largely identical, we do not reproduce the KL-Divergence table herein.

¹¹ The *SymbolOf* relation is succinctly defined as: the triple *A SymbolOf B* is asserted in the knowledge base if ‘A symbolically represents B’.

Table 1. The Spearman’s Rank Correlation Score between SL_r and SL'_r , for each of the 49 Relations. The Methodology for Constructing these Two (Aligned) Similarity Lists is Described in the Text

Relation	Spearman’s correlation	Relation	Spearman’s correlation
IsA	-0.773	NotDesires	0.954
dbpedia/knownFor	0.795	PartOf	-0.939
HasSubevent	0.882	dbpedia/genus	-0.962
Entails	-0.958	EtymologicallyRelatedTo	-0.385
DerivedFrom	-0.864	HasA	0.891
UsedFor	0.926	Desires	0.946
CapableOf	0.934	dbpedia/leader	0.705
AtLocation	0.600	CreatedBy	0.780
HasContext	-0.516	NotUsedFor	0.639
Antonym	-0.856	DefinedAs	0.812
HasLastSubevent	0.918	SymbolOf	0.861
CausesDesire	-0.946	LocatedNear	-0.951
EtymologicallyDerivedFrom	-0.865	HasPrerequisite	0.797
InstanceOf	-0.947	MadeOf	0.921
dbpedia/influencedBy	-0.475	ReceivesAction	0.979
MannerOf	-0.979	dbpedia/capital	0.946
dbpedia/language	-0.595	Causes	0.987
HasProperty	0.924	NotHasProperty	-0.736
dbpedia/product	-0.880	NotCapableOf	-0.598
HasFirstSubevent	0.818	dbpedia/field	-0.611
dbpedia/genre	-0.983	SimilarTo	-0.918
DistinctFrom	0.756	MotivatedByGoal	0.957
dbpedia/occupation	-0.591	ObstructedBy	0.849
FormOf	-0.708	RelatedTo	-0.937
Synonym	-0.738		

Within our sample of 4 million triples, HasContext, FormOf and SymbolOf are asserted in 133,038, 630,914 and 63,785 triples respectively. This provides an added incentive to study these relations further, since they are clearly central to the knowledge base and its purpose of capturing commonsense knowledge as sets of assertions. While these are not the most voluminous relations,¹² we

¹² For example, *RelatedTo* and *Synonym* have more than a million triples each, the reason for their breadth (and high volume) is more evident than for a relation such as *HasContext* (for example), since context is a much more ambiguous concept in commonsense reasoning. We hypothesize that a relation like *Synonym* will behave

aimed for a set of three relations that are expected to have different practices around them. We leave for future work to replicate our methodology for other such high-volume relations.

An established unsupervised methodology for discovering structure in large collections of data points is *clustering* [15]. The relations in ConceptNet were meant to capture common, informative patterns from various data sources that feed into ConceptNet (along with crowdsourcing). If well-defined clusters exist, there is good evidence to suggest that these coarse-grained relations could be further sub-divided or ontologized (possibly by declaring relation-subtypes). By studying both the consistency of the clusters, as well as the subjective nature of data within them, we can start gaining insight into each relation. These insights allow us to gain an empirical understanding of concepts, such as ‘context’ and ‘form’, that are important in commonsense reasoning and communication, beyond their theoretically broad definitions.

Most established clustering algorithms require the collection and representation of data points to be described in advance. In our case, the goal is to cluster asserted triples of the form (h, r, t) in three independent experiments (with r belonging to one of HasContext, FormOf and SymbolOf in each experiment). However, such a clustering would require us to represent each triple as a vector. If the relation is fixed, as it would be within an experiment, we can represent the triple using translation vector $\mathbf{t} - \mathbf{h}$ that we earlier introduced, and with the entity embeddings \mathbf{t} and \mathbf{h} output by PBG.

For the clustering algorithm itself, we chose to use the classic k-Means algorithm [11]. There were several reasons, including the large numbers of data points (which requires efficient clustering), the lack of a task-specific objective function or training labels, and importantly, the methodological preference for an established and reasonably robust clustering algorithm.

To briefly review k-Means, the algorithm works iteratively to *partition* the dataset into k clusters, each of which is disjoint, owing to the clusters constituting a partition. Let us assume a set $D = \{d_1, \dots, d_n\}$ of n data points, each of which is q -dimensional. We set up the algorithm so that the k means or clusters are randomly initialized, and each of the n data points are assigned to exactly one of the k means, depending on which cluster the point is closest to. Next, the mean for each cluster is re-computed by taking the mean of the vectors assigned to that cluster. The steps above are then repeated: each of the n data points is reassigned to exactly one of the k clusters (ties are broken arbitrarily), depending on which cluster’s mean it is closest to. The means are then re-computed, and so on. We run the algorithm till convergence is achieved, and cluster-assignment of points does not change from one iteration to the next.

Note that k is a hyper-parameter that must be predefined prior to executing the algorithm. There are several ways to obtain the ‘best’ value of k given a collection of points. The underlying commonality between these methods is to compute, for each value of k , an error ‘score’, with lower values implying better

similarly as a relation like *SymbolOf*, although we leave for future work to investigate it.

quality. This score is computed from the clusters obtained after executing k-Means for that k . In practice, k is varied over a predetermined range. By plotting the error score versus k , and looking for sudden shifts in the *second derivative* of the curve,¹³ we can determine a value of k that captures the structure in the data. Intuitively, we are seeking a clear ‘bend’ in the *score vs. k* curve to deduce where the second derivative is being minimized.

A specific method that implements the k -selection principles conceptually described above is the *elbow method* [26], but other methods include the Silhouette Coefficient method, the Davies-Bouldin Index, and the Calinski-Harabasz Index [1, 54]. The Silhouette Coefficient value measures how similar a point is to its own cluster’s centroid (cohesion) compared to other clusters’ centroids (separation). The Index-based measures compute their scores in slightly different ways, but with the same underlying philosophy that clusters should be cohesive and well-separated. Specific details and formulae may be found in the cited works.

In implementing these methods, while both the elbow method and Davies-Bouldin index were found to be inconclusive, the other two methods indicated fairly evident and consistent values for k . In the case of the Silhouette Score, we find that there is an explicit minima at $k = 20$. In general, we found $k = 20$ to be a robust choice for all three relations. These implementations are reproduced in Figures A1, A2 and A3 for all three relations in the online appendix (see footnote 8). The exercise illustrates the methodological utility of using more than one k -selection procedure.

In future work, it may be possible to address the limitation of selecting k heuristically by using hierarchical or agglomerative clustering, and also by using recent clustering algorithms that do not require such hyperparameter selection (e.g., the recent work by [27]). Our main reason for using k-Means (along with hyperparameter selection methods such as elbow and Calinski-Harabasz Index) for this paper is that it is an established and scalable procedure that can be easily replicated without re-implementation or significant hyperparameter tuning. It also provides a reference and benchmark for future research that is looking to explore the use of other, more advanced algorithms for discovering finer-grained sub-structures in the ConceptNet relations (both the ones that are used in this paper, as well as others, such as RelatedTo, that are not considered in this paper).

3.5 Replication and Code Availability

As noted earlier, ConceptNet 5.7 is used as the commonsense knowledge graph for the experiments in this paper and is downloadable from <https://github.com/commonsense/conceptnet5/wiki/>. Identifiers of the 4 million sampled triples are downloadable at the following link.¹⁴ For many of the experiments in the paper,

¹³ Although the curve can be monotonic for some methods, it is not always guaranteed. Hence, it is incorrect to look for a ‘minimum’.

¹⁴ <https://drive.google.com/file/d/1RlHkwvuYmgOMqNf4UT0EDx5MmnbKvX4s/view?usp=sharing>.

we draw on publicly available codebases that have been developed in the larger community including:

1. t-SNE: <https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html>
2. PBG: <https://github.com/facebookresearch/PyTorch-BigGraph>, and
3. Standard clustering and machine learning packages available at <https://scikit-learn.org/stable/>.

4 Results

With the selected value of $k = 20$ in place, we conducted k-Means clustering for each of the three relations, as discussed earlier. Following the clustering, we computed quantitative metrics to measure the *cohesion* and *separation* of the clusters. As its name suggests, *cohesion* measures the extent to which the points in each cluster are tightly grouped together. In other words, a cluster with high dispersion has low cohesion. While this intuitive measure can be quantitatively measured in several ways, we consider a simple, easily interpretable methodology and formulae described below in detail.

First, for each cluster, we compute its centroid and then normalize all points in the clusters, as well as the computed centroid of the cluster.¹⁵ Specifically, let us assume k clusters (with k set to 20, as detailed earlier) $\{C_1, \dots, C_k\}$, which are non-overlapping, non-empty and form a partition over the set D of data points (with each data point $d \in D$ being a vector $\mathbf{d} = [d_1, \dots, d_q]$ with q dimensions, as discussed in Sect. 3.4) being clustered. The centroid \mathbf{c}_m of a cluster C_m is defined using the formula below:

$$\mathbf{c}_m = \frac{\sum_{d_i \in C_m} \mathbf{d}_i}{|C_m|} \quad (2)$$

The centroid always exists, since each cluster is non-empty. The sum in the numerator is element-wise. Following centroid computation for each cluster, we normalize each vector \mathbf{d} in D as well as all k centroids (which are also vectors, with the same dimensionality q as \mathbf{d}), such that the vector now lies on the unit hypersphere, we use the following formula:

$$\mathbf{v}' = \frac{\mathbf{v}}{\sqrt{\sum_{i=1}^{i=q} |v[i]|^2}} \quad (3)$$

Here, \mathbf{v} is any vector from D or a centroid vector, and \mathbf{v}' is the normalized vector. The division is again element-wise, and $v[i]$ is the i^{th} element of the vector \mathbf{v} .

¹⁵ Since the graph embeddings are not normalized to lie on a unit-radius hypersphere, we normalize the embeddings before computing the distance to enable cross-cluster comparisons, as well as comparisons with the (subsequently described) separation measures.

Next, we calculate the average *Euclidean* distance (designated as t_m) between the normalized points in the cluster C_m and the cluster's centroid \mathbf{c}_m (which is also normalized):

$$t_m = \frac{\sum_{\mathbf{d}_i \in C_m} Euc[\mathbf{d}_i, \mathbf{c}_m]}{|C_m|} \quad (4)$$

Here, $Euc[\mathbf{x}, \mathbf{y}]$ between two q -dimensional vectors \mathbf{x} and \mathbf{y} is $Euc[\mathbf{x}, \mathbf{y}] = \sqrt{\sum_{i=1}^{i=q} (x_i - y_i)^2}$, and is a scalar. Since *smaller* distances indicate *greater* cohesion, we subtract the average from 1 to obtain a cohesion $coh_m = 1.0 - t_m$ of the cluster C_m on a scale of 0.0 to 1.0, with 1.0 indicating perfect cohesion i.e., all points inside the cluster coincide after normalization). In Table 2, we report the cohesion for each of the 20 clusters obtained, for each of the three relations. We also report the means and standard deviations, for each of the three relations being studied. Formally, given the cohesions coh_1, \dots, coh_k , the mean cohesion M_{coh} is given by the formula $\frac{\sum_{i=1}^{i=k} coh_i}{k}$, and the standard deviation Std_{coh} is $\frac{\sum_{i=1}^{i=k} (coh_i - M_{coh})^2}{k}$.

Table 2. The *cohesion/separation* of FormOf, HasContext and SymbolOf clusters, along with per-relation mean and standard deviation. Note that cluster ids are assigned arbitrarily and independently across relations, and not ‘aligned’ in any way

Cluster ID	FormOf	HasContext	SymbolOf
0	4.517/2.563	5.953/4.644	5.358/2.421
1	4.576 / 2.911	5.386/3.624	4.815/2.597
2	4.085/2.969	4.661/3.943	5.436/5.611
3	4.254/2.866	4.640/5.811	5.578/2.661
4	4.785/2.681	5.738/4.044	4.703/2.750
5	4.503/2.686	5.555/3.943	4.735/2.607
6	4.660/3.376	3.284/6.016	3.677/4.862
7	3.601/3.105	5.072/3.564	4.854/2.454
8	3.966/2.952	4.051/4.345	4.623/2.245
9	4.547/2.658	3.731/5.738	4.730/2.348
10	4.065/4.092	4.297/4.323	6.319/4.151
11	4.741/2.536	3.371/6.412	4.553/2.585
12	4.214/2.831	4.215/4.620	4.647/2.381
13	3.918/3.235	5.398/3.727	4.031/4.697
14	5.300/4.055	4.331/4.889	4.715/2.269
15	4.451/2.568	3.739/4.935	4.585/2.339
16	4.648/2.719	5.515/3.827	4.443/2.940
17	4.684/3.169	4.987/3.958	5.132/2.613
18	5.151/3.307	3.677/5.083	4.533/2.392
19	4.119/2.422	3.649/4.985	5.149/2.480
Mean	4.439/2.985	4.562/4.622	4.831/2.970
Std. Dev.	3.302/4.014	13.588/13.738	6.084/18.914

Based on the table, we find that the mean cohesion for FormOf, HasContext and SymbolOf clusters is 4.439, 4.562 and 4.831 respectively. While the mean cohesion scores of clusters in these three relations may seem close in value, their standard deviations exhibit significant differences. The standard deviations of HasContext cluster cohesion scores are generally higher than the standard deviations of the other two relations' cohesion scores. This simple result suggests that HasContext may be more diverse (and hence, more *dispersed* in embedding space) than the other two relations. Furthermore, while the deviation is inversely related to the number of triples corresponding to each relation, it is not linear. Finally, it is important to note that the absolute values here are less meaningful than the values relative to one other.

While cohesion is a good measure for characterize clusters, it is not adequate by itself. An ‘optimal’ cohesion can be obtained by assigning each point to its own cluster (in which case, the point becomes the centroid of the cluster). An additional metric, even after controlling for k , is the *separation* of the clusters i.e., how ‘far apart’ the different clusters are in the embedding space. Similar to cohesion, there are multiple mathematical ways to capture this qualitative notion. We employ a simple method that is analogous to the cohesion measure—namely, for a given cluster C_m , we compute its *separation* s_m by computing the average Euclidean distance from its centroid \mathbf{c}_m to each of the *other* $k - 1 = 19$ centroids. For simplicity, let us define the centroid-set $\mathcal{C} = \{\mathbf{c}_1, \dots, \mathbf{c}_k\}$ as the set of (normalized) centroids of all $k = 20$ clusters:

$$s_m = \frac{\sum_{c_i \in \mathcal{C}, c_i \neq c_m} Euc[\mathbf{c}_i, \mathbf{c}_m]}{k - 1} \quad (5)$$

Note that a subtraction from 1.0 is not necessary (as was the case for the cohesion computations), since the higher the average Euclidean distances between the centroids, the higher the separation. Table 2 also reports the separation results for all three relations, along with the mean and standard deviation. The same formulae apply for the mean and standard deviation as noted earlier for cohesion.

We find that, once again, HasContext has highest average separation (4.622). This further suggests that the ‘contexts’ represented by these clusters are well-separated. The FormOf and SymbolOf clusters obtain similar average separations of 2.985 and 2.970, respectively. Unlike cohesion, the standard deviation of cluster separation scores is highest for the SymbolOf relation.

In comparing the cohesion and the separation of clusters for all three relations in Table 2, we find that mean separation of HasContext clusters is close to their mean cohesion. In other words, the mean distance from a cluster centroid to a within-cluster data point is similar to the mean distance from that cluster-centroid to other cluster-centroids. The mean separations of FormOf and SymbolOf clusters are lower than the respective mean cohesions, suggesting that substructures in these two relations may be less independent than those in HasContext.

We can also visualize the clustering results by first performing dimensionality reduction (to two dimensions) using the t-Stochastic Neighbor Embedding

(t-SNE) method, which has emerged as a state-of-the-art neural visualization technique in the machine learning community [52]. Next, we plot these points in 2D space by using a different color to represent each cluster. Results for all three relations are visualized in Fig. 2. For all three relations (and especially, FormOf), there are some homogeneous clusters, where the embeddings are close to each other. However, other clusters can exhibit dispersion. For both SymbolOf and HasContext, some clusters exhibit high dispersion and overlap with other clusters. These dispersed clusters provide an explanation for why HasContext and SymbolOf were found to have much high standard deviations on both the *separation* and *cohesion* measures described earlier (compared with much lower standard deviations for FormOf).

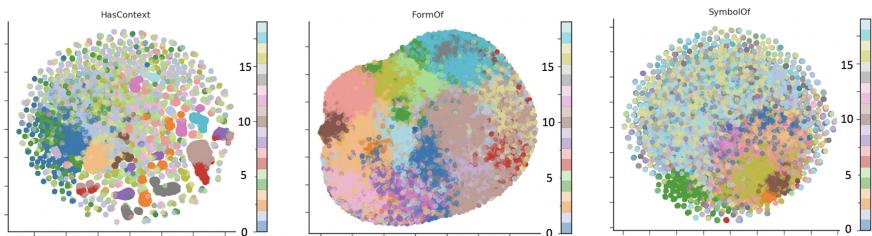


Fig. 2. The visualization of HasContext, FormOf and SymbolOf ‘triples-clusters’ using the t-SNE method. Here, 20 such clusters per relation, discovered using k-means, are represented using different colors. The cluster ids are indicated on the color bar on the right side of each subplot, and are consistent with those used in Table 2, and used subsequently in Table A2 in the online appendix (See Footnote 8). The t-SNE ‘dimensions’ lack intrinsic meaning and are only for visualization purposes

It is also instructive to study the substructures for a given relation by further analyzing triples *qualitatively* within each of the 20 clusters. We conduct such a qualitative study for the HasContext relation. Specifically, we randomly sampled five triples per cluster to determine if we can deduce the ‘theme’ of the cluster from these five triples. These samples are reproduced in Table A2 in the online appendix (see footnote 8). Although we only comment on HasContext below, for the sake of completeness, we also reproduce similarly sampled triples for the other two relations in Tables A3 and A4 in the online appendix.

Even the limited set of sampled triples (per cluster) was found to demonstrate a pattern. For example, some of the clusters embodied scientific ‘domains’ of study such as geography (Cluster 14), chemistry (Cluster 6), medicine (Cluster 17), mathematics (Cluster 18), and physics (Cluster 19). However, there are also ‘mixed’ clusters that seem to be related to more than one theme, at least on the surface. Cluster 3 contains some ‘locality’ triples, even though Cluster 15 is predominantly concerned with localities, and Cluster 3 is mainly concerned with zoology.

We believe that there could be two causes or interpretations for such ‘confusion’. The first is due to the automatic and unsupervised nature of the embedding

and the second is due to ConceptNet itself, both in terms of the noise within ConceptNet and also because some tail entities, such as */c/fr/localites*, may be imbalanced in terms of the head entities associated with them. Some other clusters also include some interesting combinations, such as Cluster 12 which contains triples corresponding to both ‘computing’ and ‘slang’. In the embedding and clustering process, ‘computing’ and ‘slang’-related triples are thought to be close to each other in vector space, although their semantic similarity is not intuitively evident. Yet other clusters, such as Cluster 0, seem to encapsulate the broad notion of *HasContext*, and do not have an evident thematic classification that we can determine.

Additional interpretations of these overlapping clusters are also possible. One interpretation is *historical context*, especially concerning how these fields (‘sub-structures’) have co-evolved over time. For example, fields such as mathematics, physics and even computing have become increasingly entwined over time. The overlap may simply be a consequence of this co-evolution. Another possibility is that the fields share common semantics (including common terms), and this is reflected in overlap as well. Finally, we note that ConceptNet is meant to be a repository of commonsense knowledge, and not necessarily a detailed description of highly specialized domains within science (or other areas). The overlap may be arising not because the fields themselves are highly overlapping but because their *commonsense content* is. We leave for future work to investigate these interpretations more rigorously and quantitatively.

Aside from hypothesis validation of commonsense knowledge and evolution of commonsense semantics in specialized areas, the sub-structures could be used in novel *domain-specific* applications of AI, including domain-specific versions of fuzzy clustering and expert systems [25]. One such application is in knowledge acquisition: our discovered sub-structures could be used for isolating specific portions of, or triples within, ConceptNet that are useful for the application domain being considered. Once isolated, these triples could be used in a domain-specific fuzzy clustering algorithm or expert system to enhance performance. Further research is needed to build and evaluate such applications.

Further investigation of the inter-relationships between these clusters in Fig. 2 yields other insights. For example, the cluster focused on chemistry-related triples overlaps with the cluster containing biology, as well as with astronomy-related, triples, as we would intuitively expect. While some of the overlap in the figure is exaggerated due to dimensionality reduction, it is nonetheless indicative of the low separation between these two clusters in high-dimensional space. It is an indirect acknowledgement of the shared lineage of these scientific disciplines. An interesting avenue for future exploration is to quantify and explain the observed topical overlap between such clusters, by using techniques such as hierarchical clustering.

5 Discussion

In exploring three specific relations (SymbolOf, FormOf, and HasContext), we found and characterized significant ‘substructures’ that are *thematically diverse*

(especially in the case of HasContext), illustrating distinct and complex sub-relations within the overall relation. Sub-structures were also noted in FormOf and SymbolOf, but were less interesting and had clear separations than HasContext.

We have also empirically observed that, while ‘super-class’ semantics tend to be associated with the definition of HasContext,¹⁶ there are significant substructures that can’t be uniformly explained by an ‘umbrella’ term like HasContext. These substructures may help us better understand what the different contexts are in which people interpret pairs of words or entities. Understanding context is critical for building systems that have commonsense, such as chatbots and conversational agents, that need to understand sentences in the specific context in which the sentences are uttered.

6 Conclusion

The release and growth of commonsense knowledge graphs, such as ConceptNet, has provided an opportunity to conduct a computational study of commonsense knowledge using semi-automatic techniques. In this article, we presented and applied a data-driven methodology to understand structure in ConceptNet commonsense assertions through an empirical study of three high-volume, coarse-grained relations, namely, FormOf, SymbolOf and HasContext. All of these relations (and especially, hasContext) are known to be important in everyday commonsense tasks, including communication and conversation. Our analyses suggest that there are at least 20 distinct kinds of context that can be discovered within ConceptNet, some very well-defined (such as a scientific field of study), with others being more diffuse. Similarly, different sub-categories of symbols and forms can be semi-automatically discovered. Discovering even richer sub-categories and organizing them into a hierarchy using semi-automatic methods could be a valuable avenue for future research.

References

1. Bandyopadhyay, S.: *Unsupervised Classification*. Springer Publishing Company, Incorporated (2015)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 26*, pp. 2787–2795. Curran Associates, Inc. (2013)
3. Botschen, T., Sorokin, D., Gurevych, I.: Frame- and entity-based knowledge for common-sense argumentative reasoning. In: *Proceedings of the 5th Workshop on Argument Mining*, pp. 90–96. Association for Computational Linguistics, Brussels, Belgium (2018)

¹⁶ Namely, when head entity h is a word that is used in the context of tail entity t , t tends to be a more general, abstract ‘super-class’ of h , such as a topic area, technical field, or regional dialect, as is also mentioned in the official definition of HasContext.

4. Chen, M., Tian, Y., Chang, K.-W., Skiena, S., Zaniolo, C.: Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment (2018). [arXiv:1806.06478](https://arxiv.org/abs/1806.06478)
5. Davis, E., Marcus, G.: Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM* **58**, 92–103 (2015)
6. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018). [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)
7. Färber, M., Ell, B., Menne, C., Rettinger, A.: A comparative survey of dbpedia, freebase, opencyc, wikidata, and yago. *Semant. Web J.* **1**(1), 1–5 (2015)
8. Floridi, L., Chiriatti, M.: Gpt-3: Its nature, scope, limits, and consequences. *Mind. Mach.* **30**(4), 681–694 (2020)
9. Gordon, A.S., Hobbs, J.R.: *A Formal Theory of Commonsense Psychology: How People Think People Think*. Cambridge University Press (2017)
10. Grover, A., Leskovec, J.: Node2vec: Scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864 (2016)
11. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. *J. R. Stat. Society. Ser. C (Applied Statistics)* **28**(1), 100–108 (1979)
12. Havasi, C., Speer, R., Alonso, J.: Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In: *Recent Advances in Natural Language Processing*, pp. 27–29. Citeseer (2007)
13. Havasi, C., Speer, R., Arnold, K., Lieberman, H., Alonso, J., Moeller, J.: Open mind common sense: Crowd-sourcing for common sense. In: *Proceedings of the 2nd AAAI Conference on Collaboratively-Built Knowledge Sources and Artificial Intelligence*, AAAIWS’10–02, pp. 53. AAAI Press (2010)
14. Hirschberg, J., Manning, C.D.: Advances in natural language processing. *Science* **349**(6245), 261–266 (2015)
15. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. *ACM Comput. Surv.* **31**(3), 264–323 (1999)
16. Kejriwal, M.: Domain-Specific Knowledge Graph Construction. Springer, Berlin (2019)
17. Kejriwal, M.: Knowledge graphs and covid-19: opportunities, challenges, and implementation. *Harv. Data Sci. Rev.* **11**, 300 (2020)
18. Kejriwal, M.: Artificial Intelligence for Industries of the Future: Beyond Facebook, Amazon, Microsoft and Google. Springer Nature (2022)
19. Kejriwal, M.: Knowledge graphs: A practical review of the research landscape. *Information* **13**(4), 161 (2022)
20. Kejriwal, M., Knoblock, C.A., Szekely, P.: *Knowledge Graphs: Fundamentals, Techniques, and Applications*. MIT Press (2021)
21. Kejriwal, M., Santos, H., Mulvehill, A.M., McGuinness, D.L.: Designing a strong test for measuring true common-sense reasoning. *Nat. Mach. Intell.* **4**(4), 318–322 (2022)
22. Kejriwal, M., Shen, K.: Do fine-tuned commonsense language models really generalize? (2020). [arXiv:2011.09159](https://arxiv.org/abs/2011.09159)
23. Kejriwal, M., Szekely, P.: An investigative search engine for the human trafficking domain. In: *The Semantic Web–ISWC 2017: 16th International Semantic Web Conference, Proceedings, Part II 16*, pp. 247–262. Springer, Berlin (2017)
24. Kejriwal, M., Tang, Z.: Evaluating language representation models on approximately rational decision making problems. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 44 (2022)

25. Kidd, A.: Knowledge Acquisition for Expert Systems: A Practical Handbook. Springer Science & Business Media (2012)
26. King, G.J.W., Richards, R.R., Zuckerman, J.D., Blasier, R., Dillman, C., Friedman, R.J., Gartsman, G.M., Iannotti, J.P., Murnahan, J.P., Mow, V.C., et al.: A standardized method for assessment of elbow function. *J. Shoulder Elb. Surg.* **8**(4), 351–354 (1999)
27. Komkhao, M., Kubek, M., Halang, W.A.: Sequentially grouping items into clusters of unspecified number. In: International Conference on Computing and Information Technology, pp. 297–307. Springer, Berlin (2017)
28. Lenat, D.B.: Cyc: A large-scale investment in knowledge infrastructure. *Commun. ACM* **38**(11), 33–38 (1995)
29. Lerer, A., Wu, L., Shen, J., Lacroix, T., Wehrstedt, L., Bose, A., Peysakhovich, A.: PyTorch-BigGraph: a large-scale graph embedding system. In: Proceedings of the 2nd SysML Conference. Palo Alto, CA, USA (2019)
30. Lin, B.Y., Chen, X., Chen, J., Ren, X.: Kagnet: Knowledge-aware graph networks for commonsense reasoning (2019). [arXiv:1909.02151](https://arxiv.org/abs/1909.02151)
31. McDermott, J.: When and why metaheuristics researchers can ignore “no free lunch” theorems. *SN Comput. Sci.* **1**(1), 1–18 (2020)
32. Miller, G.A.: Wordnet: A lexical database for english. *38*(11), 39–41 (1995)
33. Mueller, E.T.: Commonsense Reasoning: An Event Calculus Based Approach, 2nd edn. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2014)
34. Narayanan, S.: Reasoning about actions in narrative understanding. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence (2000)
35. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710 (2014)
36. Pinto, J., Reiter, R.: Reasoning about time in the situation calculus. *Ann. Math. Artif. Intell.* **14**, 251–268 (1995)
37. TianXing, W., GuiLin, Q., Huan, G.: The research advances of knowledge graph. *Technol. Intell. Eng.* **3**, 4–25 (2017)
38. Rajagopal, D., Cambria, E., Olsher, D., Kwok, K.: A graph-based approach to commonsense concept extraction and semantic similarity detection. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 565–570 (2013)
39. Roemmele, M., Bejan, C., Gordon, A.: Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning (2011)
40. Santos, H., Kejriwal, M., Mulvehill, A.M., Forbush, G., McGuinness, D.L., Rivera, A.R.: An experimental study measuring human annotator categorization agreement on commonsense sentences. *Exp. Results* **2**, e19 (2021)
41. Santos, H., Shen, K., Mulvehill, A.M., Razeghi, Y., McGuinness, D.L., Kejriwal, M.: A theoretically grounded benchmark for evaluating machine commonsense (2022). [arXiv:2203.12184](https://arxiv.org/abs/2203.12184)
42. Sap, M., Rashkin, H., Chen, D., Bras, R., Yejin, C.: Social iqa: Commonsense Reasoning About Social Interactions, pp. 4453–4463 (2019)
43. Sap, M., Shwartz, V., Bosselut, A., Choi, Y., Roth, D.: Commonsense reasoning for natural language processing. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, pp. 27–33. Association for Computational Linguistics (2020)
44. Shen, K., Kejriwal, M.; On the generalization abilities of fine-tuned commonsense language representation models. In: Artificial Intelligence XXXVIII: 41st SGAI International Conference on Artificial Intelligence, AI 2021, Proceedings 41, pp. 3–16. Springer, Berlin (2021)

45. Shen, K., Kejriwal, M.: An experimental study measuring the generalization of fine-tuned language representation models across commonsense reasoning benchmarks. *Expert Syst.* e13243 (2023)
46. Shi, B., Weninger, T.: Open-world knowledge graph completion (2017). [arXiv:1711.03438](https://arxiv.org/abs/1711.03438)
47. Smedslund, J.: Common sense as psychosocial reality: A reply to sjöberg. *Scand. J. Psychol.* **23**(1), 79–82 (1982)
48. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: An open multilingual graph of general knowledge. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17, pp. 4444–4451. AAAI Press (2017)
49. Speer, R., Havasi, C.: Conceptnet 5: A large semantic network for relational knowledge. In: The People's Web Meets NLP, Theory and Applications of Natural Language Processing, pp. 161–176 (2013)
50. Tang, Z., Kejriwal, M.: Can language representation models think in bets? (2022). [arXiv:2210.07519](https://arxiv.org/abs/2210.07519)
51. Tang, Z., Kejriwal, M.: A pilot evaluation of chatgpt and dall-e 2 on decision making and spatial reasoning (2023). [arXiv:2302.09068](https://arxiv.org/abs/2302.09068)
52. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. *J. Mach. Learn. Res.* **9**(11) (2008)
53. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **1** (2017)
54. Yuan, C., Yang, H.: Research on k-value selection method of k-means clustering algorithm. *Journal* **2**, 226–235 (2019)



Artificial Intelligence Algorithm for Optimizing PID Parameters to Control Weakly Damped Systems

Roland Büchi^(✉)

School of Engineering, Zurich University of Applied Sciences, Winterthur, Switzerland
bhir@zhaw.ch

Abstract. The step responses of many systems to be controlled show an overshoot behavior. This is the case, for example, with active vibration damping with spring-mass systems. This document provides PID controller tables for overshooting systems. Such systems can be approximated with second-order systems. The parameters were calculated with the simulation and optimized with a search for the best values according to the minimum ITAE criterion. For this purpose, parameter sets were calculated using hill climbing, an approach from artificial intelligence. It minimizes the criterion in transient response over time. The publication provides tables for systems with different system damping. So that the sets can be used in general, a way of easily identifying such systems is also presented. The controller parameters are then verified using the position control of a weakly damped spring-mass system.

Keywords: PID · Artificial intelligence · ITAE · Weakly damped

1 Introduction

Even though many new controller topologies have been developed in recent times, the PID controller is still by far the most frequently used controller in industrial systems. For controlled systems that are inherently stable and do not show any overshoot, several variants of parameter sets have been calculated in the past for controlling them with PI and PID controllers [1, 2]. In recent years, several methods with parameter swarm optimization [3, 4] or others have been used for the calculation of controller parameter sets. These provide a rapid convergence of the solution. From literature, also many further optimization methods for control parameters are known [5–12]. The ‘hill climbing’ method [13] used here is related to particle swarm optimization. And there are also other methods used for the optimization of controllers [14–16].

However, although many systems are stable, they show a system-related overshooting. Dampened spring-mass systems from mechanical engineering can be mentioned among the many examples.

In the following, such a system is identified first. In the second section, the basis for the calculation of the optimal controller parameters according to the ITAE criterion

is dealt with to control such systems. The core of the publication is the table with the controller parameters in the third section. In the fourth section, this table is applied to the practical example with the control of a spring mass system. At the end, the conclusion and further work are discussed.

1.1 Identification of An 2nd Order Overshooting System

Figure 1 shows a step response of an overshooting system. First, based on this figure, simple rules can be defined how to identify the parameters of the transfer function from a given step response of an overshooting system. In general, the transfer function $G(s)$ is according to formula 1.

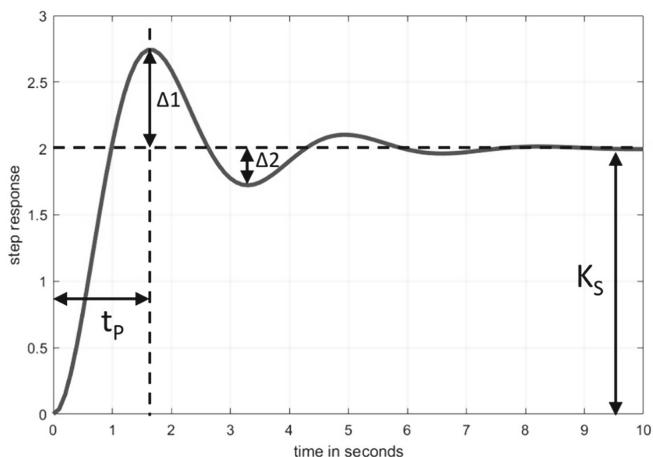


Fig. 1. Step response of a damped 2nd order system.

$$G(s) = \frac{K_s}{T^2 \cdot s^2 + 2 \cdot D \cdot T \cdot s + 1} \quad (1)$$

Not every system is of the 2nd order type, but such systems are very common in practice. In the given system, the first overshoot $\Delta 1$ must be found first. This must be specified in relation to the stationary end value. From the graph in Fig. 1 one reads: $\Delta 1 = 0.37$. The relationship between the first overshoot $\Delta 1$ and the system damping D has already been calculated in many publications.

$$\Delta 1 = \exp\left(-\frac{D \cdot \pi}{\sqrt{1 - D^2}}\right) \quad (2)$$

The same formula solved for the system damping D gives:

$$D = -\frac{\ln(\Delta 1)}{\sqrt{\pi^2 + (\ln(\Delta 1))^2}} \quad (3)$$

This results in a $D = 0.3$ for the given system. In general, it is not mandatory to use formula 3 in practice. The damping can also be read from the graph in Fig. 2. This was calculated from the formula. As a check, you can also identify the second overshoot Δ_2 . This is not necessarily important for the calculation of the damping measure, but it gives an indication of whether the given step response can be identified with a clear conscience as a 2nd order system. The relationship between D , Δ_1 and Δ_2 is calculated for a second order system as

$$D = \frac{\ln\left(\frac{\Delta_1}{\Delta_2}\right)}{\sqrt{\pi^2 + \left(\ln\left(\frac{\Delta_1}{\Delta_2}\right)\right)^2}} \quad (4)$$

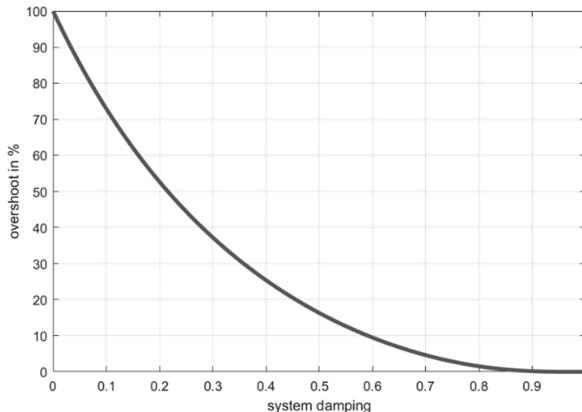


Fig. 2. Step response overshoot as a function of the system damping.

The time constant T in the formula of the general second order transfer function $G(s)$ must also be identified. For this purpose, the time t_p is measured, which elapses between the start of the jump and reaching the maximum overshoot. t_p is the so-called peak time. The time constant T is then directly related to t_p and can be calculated using the formula 5.

For the example of the step response according to Fig. 1, $T = 0.5$ s is calculated. The static gain K_s is also missing in formula 1. K_s can also be read directly from the graphic in Fig. 1. Since the static end value is 2 and the step is a unit step, there is $K_s = 2$ in this example. Overall, the system is identified with the parameters according to Fig. 1, i.e. with $G(s)$ according to formula 6.

$$T = \frac{t_p \cdot \sqrt{1 - D^2}}{\pi} \quad (5)$$

$$G(s) = \frac{K_s}{T^2 \cdot s^2 + 2 \cdot D \cdot T \cdot s + 1} = \frac{2}{0.25 \cdot s^2 + 0.3 \cdot s + 1} \quad (6)$$

In this paper, there are stable second-order systems treated, which have a system damping between 0 and 1. They thus show the overshoot behavior as shown in Fig. 2.

This leads to complex conjugated pole pairs in the classification of the second-order systems. For the sake of completeness, it should be mentioned here that there are also second-order stable systems which do not exhibit any overshoot. These have two negative real pole pairs and a system damping of > 1.0 . Their control is not dealt with in this publication.

2 Controller Parameters Calculated After the Minimized ITAE Criterion

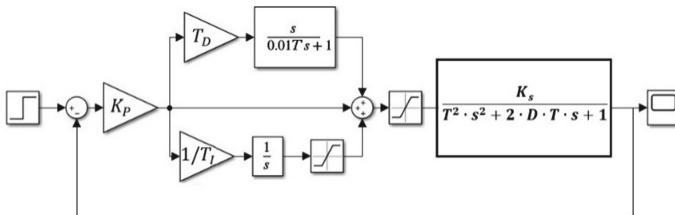


Fig. 3. Closed loop system, as it is used for calculations in this paper.

Figure 3 shows the block diagram of the closed loop of the controlled system. All following considerations and calculations are based on this figure. And they can also be checked based of this. The system described above is shown on the right, which overshoots more or less depending on the damping D. The PID controller is shown on the left. The integrator has an output limitation. When using the optimal parameters of the controller calculated in the following, this is never active, but nevertheless these limitations are always built in for practical systems. This would be relevant, for example, in the case when the control error cannot be eliminated for some reason. Then it must be prevented, that the integrator integrates into infinity. For practical feasibility, a 1st order filter must be inserted after the differential component T_d . It is chosen to be 100 times faster than the time constant T of the second order system to be controlled. The controller output limitation between the controller and the 2nd order system is present in all practical systems. This is taken into account in the parameter sets to be calculated in the following.

$$ITAE : \int_0^\infty |e(t) - e(\infty)| \cdot t \cdot dt \quad (7)$$

Figure 4 and formula 7 show how the ITAE criterion is to be understood. It is the integral of the amount of deviation of a step response of the closed loop system. If the error does not decay, the integral will increase faster over time. It is time weighted because of being multiplied by time.

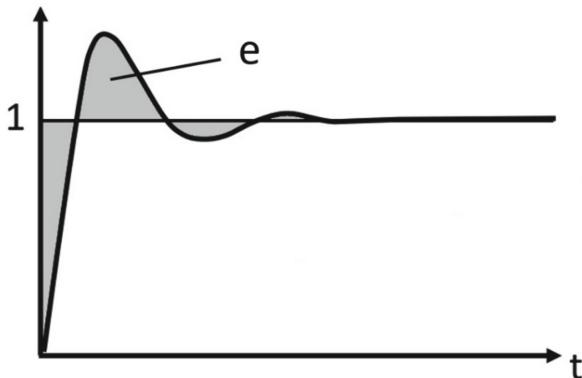


Fig. 4. The ITAE criterion in the time domain.

However, what is described and explained below is an automated calculation process that finds PID parameters and minimizes the ITAE quality criterion. Different controller output limitations are also taken into account.

In theory, the controller parameters according to the minimized ITAE criterion can be easily calculated by simulating all possible parameter combinations of P, I, D (K_p , T_i , T_d) for the 2nd order systems with different system damping D and controller output limitations. In the end, the parameters with the smallest value of the ITAE criterion are the result. The parameters found in this way can then be stored in table form and applied directly to control the practical system after identification.

The problem that arises when calculating the optimal parameters for the PID controller is that the method requires large number of simulations. The three parameters span a three-dimensional space. If they are increased from 0.1 to 10 in steps of 0.1 each, a total of $(10 \cdot 10)^3 = 1'000'000$ simulations are needed. There are two further dimensions for the calculation of the parameter sets with different controller output limitations and system dampings D. Therefore, the parameter search is performed with a method that can find local minima, with ‘hill climbing’.

The parameters of the controller are changed during the calculation according to the following rule: With each new calculation or measurement, the minimum change in the parameters K_p , T_i here 0.1 (respectively 0.05 for T_d), is multiplied by a random value from $(+1, -1, 0)$. And added to the parameters. Then the ITAE criterion is calculated. If the new value of the criterion is smaller, the parameters are fixed. If not, the original parameters are recalculated.

However, since the hill climbing method only finds local minima, many different random tuples of start values of the control parameters were used in practice when searching for parameters. After the calculation has been carried out, so many of the parameter solutions of the converged minimum ITAE criteria are the same ones. Therefore, one can assume with reasonably good certainty that the parameters found are actually the PID parameters K_p , T_i and T_d , which either correspond to the absolute minimum of the quality criteria or at least come very close to them.

3 Optimal ITAE Parameter Table for PID

Table 1 shows the parameter sets calculated in this way for different system damping D and different controller output limitations. The maximum value of the parameters was limited to 10. It is noteworthy that the table is scaled with the static gain K_s and the time constant T of the system. This makes the table usable for all weakly respectively subcritical damped systems. This table is the core of this publication.

Table 1. PID parameters for minimized ITAE criterion.

D	± 2	± 3	± 5	± 10
1.0 0% Overshoot	K _p ·K _s = 10 Ti = 9.6·T Td = 0.3·T	K _p ·K _s = 10 Ti = 7.3·T Td = 0.3·T	K _p ·K _s = 9.6 Ti = 5.4·T Td = 0.3·T	K _p ·K _s = 9.8 Ti = 4.7·T Td = 0.3·T
0.9 0.2% overshoot	K _p ·K _s = 9 Ti = 8.4·T Td = 0.35·T	K _p ·K _s = 9.7 Ti = 7.1·T Td = 0.35·T	K _p ·K _s = 10 Ti = 5.4·T Td = 0.3·T	K _p ·K _s = 9.9 Ti = 4.5·T Td = 0.3·T
0.8 1.5% Overshoot	K _p ·K _s = 9.7 Ti = 8.7·T Td = 0.35·T	K _p ·K _s = 9.9 Ti = 7.0·T Td = 0.35·T	K _p ·K _s = 9.8 Ti = 5.5·T Td = 0.35·T	K _p ·K _s = 9.9 Ti = 4.8·T Td = 0.35·T
0.7 4.6% overshoot	K _p ·K _s = 10 Ti = 8.6·T Td = 0.35·T	K _p ·K _s = 10 Ti = 6.8·T Td = 0.35·T	K _p ·K _s = 10 Ti = 5.4·T Td = 0.35·T	K _p ·K _s = 9.9 Ti = 4.6·T Td = 0.35·T
0.6 9.5% overshoot	K _p ·K _s = 9.8 Ti = 8.3·T Td = 0.4·T	K _p ·K _s = 10 Ti = 6.9·T Td = 0.4·T	K _p ·K _s = 10 Ti = 5.2·T Td = 0.35·T	K _p ·K _s = 9.9 Ti = 4.9·T Td = 0.4·T
0.5 16% Overshoot	K _p ·K _s = 9.9 Ti = 8.1·T Td = 0.4·T	K _p ·K _s = 9.8 Ti = 6.5·T Td = 0.4·T	K _p ·K _s = 9.8 Ti = 5.3·T Td = 0.4·T	K _p ·K _s = 9.9 Ti = 4.7·T Td = 0.4·T
0.4 25% overshoot	K _p ·K _s = 9.7 Ti = 7.6·T Td = 0.4·T	K _p ·K _s = 10 Ti = 6.4·T Td = 0.4·T	K _p ·K _s = 10 Ti = 5.2·T Td = 0.4·T	K _p ·K _s = 9.9 Ti = 4.5·T Td = 0.4·T
0.3 37% overshoot	K _p ·K _s = 9.4 Ti = 7.3·T Td = 0.45·T	K _p ·K _s = 9.7 Ti = 6.3·T Td = 0.45·T	K _p ·K _s = 9.9 Ti = 5.4·T Td = 0.45·T	K _p ·K _s = 9.9 Ti = 4.8·T Td = 0.45·T
0.2 53% Overshoot	K _p ·K _s = 9.7 Ti = 7.3·T Td = 0.45·T	K _p ·K _s = 9.9 Ti = 6.2·T Td = 0.45·T	K _p ·K _s = 9.9 Ti = 5.2·T Td = 0.45·T	K _p ·K _s = 9.9 Ti = 4.6·T Td = 0.45·T
0.1 73% overshoot	K _p ·K _s = 9.9 Ti = 7.5·T Td = 0.5·T	K _p ·K _s = 9.8 Ti = 6.3·T Td = 0.5·T	K _p ·K _s = 10 Ti = 5.5·T Td = 0.5·T	K _p ·K _s = 9.9 Ti = 4.9·T Td = 0.5·T
0.0 100% overshoot	K _p ·K _s = 10 Ti = 7.3·T Td = 0.5·T	K _p ·K _s = 10 Ti = 6.2·T Td = 0.5·T	K _p ·K _s = 10 Ti = 5.3·T Td = 0.5·T	K _p ·K _s = 9.9 Ti = 4.7·T Td = 0.5·T

The limitations ($\pm 2, \pm 3, \pm 5, \pm 10$) mean: (control signal limitation—control signal before the step) divided by (control signal for stationary end value—control signal before step).

The system according to Fig. 1 is being controlled as a first example. The block diagram corresponds to that of Fig. 3 in formula 6, it was found for the system: $D = 0.3$; $T = 0.5s$; $K_s = 2$.

With these values, the parameters K_p , T_i and T_d can be read from the table. One finds, for example, for a controller output limitation of ± 2 :

$$K_p = \frac{9.4}{K_S} = 4.7T_i = 7.3 \cdot T = 3.65T_d = 0.45 \cdot T = 0.23 \quad (8)$$

The transient response of the closed loop system according to the block diagram after Fig. 3 to a unit step at the system input is very nice. They are shown in Fig. 5.

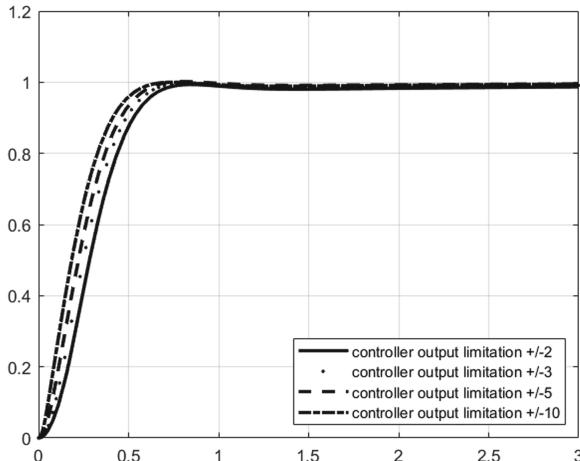


Fig. 5. Closed loop step response for the controlled system of Fig. 1.

4 Position Control of a Weakly Damped Torsion Spring System

The system in Fig. 6 is driven with a DC motor which is controlled with a current. The spring is fixed at the right end. At the output, the angle of rotation is measured with an incremental encoder. Figure 7 shows the response (increments) of the system, measured with an encoder, to a step of the motor current from 0 A to 1.5 A.

The system is identified with the discussion in the introduction. Here, the input is the motor current and the output are the encoder increments. The result is $K_s = 2000/1.5A = 1333 1/A$. The time constant T is identified as 0.188s. The system damping D is close to 0, around 0.02. The transfer function $G(s)$ is thus:

$$G(s) = \frac{K_s}{T^2 \cdot s^2 + 2 \cdot D \cdot T \cdot s + 1} = \frac{1333}{0.035 \cdot s^2 + 0.00752 \cdot s + 1} \quad (9)$$

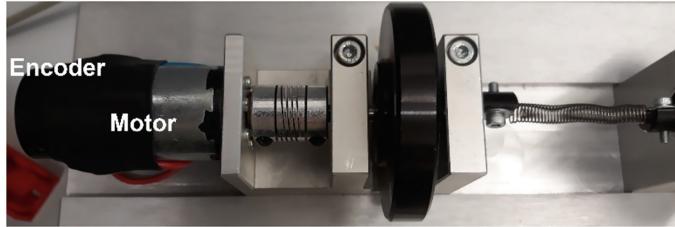


Fig. 6. Mechanical setup of the system.

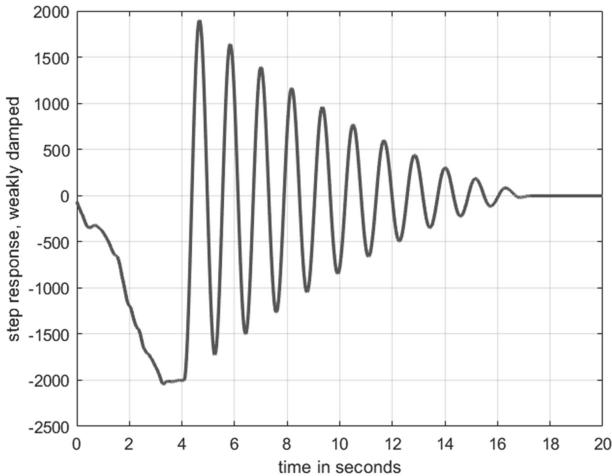


Fig. 7. System response to a current step from 0A to 1.5A.

With this system one would like to realize a position control of the wheel. To reach a position of 2000 increments, the final steady state current would be about 1.5A. The maximum current can reach 3A, resulting in a regulator output limiting factor of $3A/1.5A = 2$.

$$K_P = \frac{10}{K_S} = 0.0075T_i = 7.3 \cdot T = 1, 37T_d = 0.5 \cdot T = 0.094 \quad (10)$$

Thus one finds in the table for $D = 0$ the values of formulas 10. The controller structure is again chosen according to Fig. 3. The resulting closed loop system has a response according to Fig. 8 for a setpoint jump in the increments from 0 to 2000 (respectively, 2000 to 0).

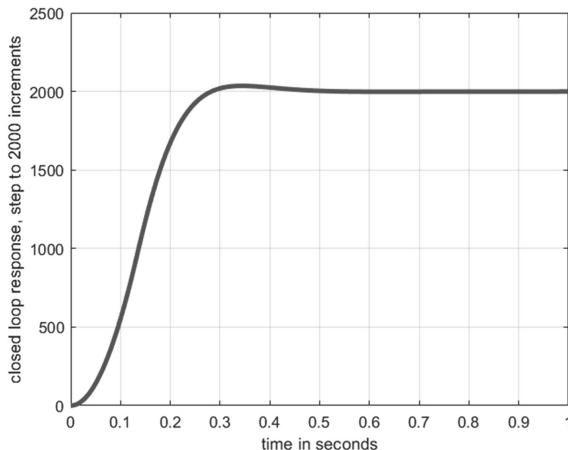


Fig. 8. System response of the closed loop system, a step from 0 to 2000 increments.

5 Conclusion and Further Work

The transient response of the example discussed above is excellent. That is the case, because the parameters of the PID controllers in the table were calculated before and identified as optimum. The discussion of the example shows very well, that PID controllers are not only suitable for controlling systems that do not have any intrinsic overshoot. Adequate parameters table values can also be found for the second-order systems with overshoot discussed here using the ‘hill climbing’ method, as it is presented above. They show a good transient behavior according to the minimized quality criterion ITAE.

In particular, it seems very important to the author to mention that the controller output limitations that always occur in practice were also taken into account in the parameter search. The parameters found are also dependent on the time constant T , the system damping D and the static gain K_s , as it can be read from the Table 1. Due to this fact, the found controller parameters can be applied very generally. They therefore also allow the control of a very large number of overshooting systems that occur in practice.

May this publication contribute to control systems with overshoot behavior in practice based on these calculated table values.

This procedure of the optimal controller parameters pre-calculated with ‘hill climbing’ for the minimized ITAE criterion is used here in a good example of the control of a second-order overshooting system. However, this method can also be applied to higher-order systems in further work. It is to be examined, but also to be expected, that the scalability found here of the table of controller parameters with the parameters time constant T and the system damping D also works with the parameters of any transfer functions. This would lead to pre-computed controller parameter tables which would be applicable at least to SISO systems and general stable transfer functions.

References

1. Ziegler, J.B., Nichols N.B.: Optimum settings for automatic controllers. ASME Trans., **64**, 759–768 (1942)
2. Chien, K.L., Hrones, J.A., Reswick, J.B.: On the automatic control of generalized passive systems. In: Transactions of the American society of mechanical engineers., Bd. 74, Cambridge (Mass.), USA, pp. 175–185 (1952)
3. Qi, Z., Shi, Q., Zhang, H.: Tuning of digital PID controllers using particle swarm optimization algorithm for a CAN-based DC motor subject to stochastic delays. IEEE Trans. Industr. Electron. **67**(7), 5637–5646 (2019)
4. Zamani, M., Sadati, N., Ghartemani, M.K.: Design of an $H\infty$ PID controller using particle swarm optimization. Int. J. Control., Autom. Syst. **7**(2), 273–280 (2009)
5. Liang, H., et al.: High precision temperature control performance of a pid neural network-controlled heater under complex outdoor conditions. Appl. Therm. Eng., 117234 (2021)
6. Büchi, R.: Optimal ITAE criterion PID parameters for PTn plants found with a machine learning approach. In: 2021 9th international conference on control, mechatronics and automation (ICCMA). IEEE, (2021)
7. Joseph, E.A., Olaiya, O.O.: Cohen—Coon PID tuning method, a better option to ziegler Nichols—PID tuning method. Comput. Eng. Intell. Syst., **9**(5), (2018). ISSN 2222-1719
8. Hussain, K.M. et al.: Comparison of PID controller tuning methods with genetic algorithm for FOPTD system. Int. J. Eng. Res. Appl., **4**(2), 308–314 (2014). ISSN: 2248-9622
9. Büchi, R.: State space control, LQR and observer: step by step introduction, with Matlab examples. Books on Demand, (2010)
10. Wahyunggoro, O., Wibawa, H., Cahyadi, A.I.: Speed control simulation of DC servomotor using hybrid PID-fuzzy with ITAE polynomials initialization. In: 2017 International conference on computer, control, informatics and its applications (IC3INA). IEEE, (2017)
11. da Silva, L.R., Flesch, R.C., Normey-Rico, J.E.: Controlling industrial dead-time systems: When to use a PID or an advanced controller. ISA Trans. **1**(99), 339–350 (2020)
12. Zacher, S., Reuter, M.: Regelungstechnik für Ingenieure. 15.Auflage, Springer Vieweg Verlag, (2017)
13. Norvig, P.: Artificial intelligence: A modern approach, 2nd edn. Prentice Hall, Upper Saddle River, New Jersey, pp. 111–114 (2003). ISBN 0–13–790395–2
14. Büchi, R.: Modellierung und Regelung von Impact Drives für Positionierungen im Nanometerbereich (Doctoral Dissertation, ETH Zurich). (1996)
15. Zhang, D., Han, Q.-L., Zhang, X.-M.: Network-based modeling and proportional-integral control for direct-drive-wheel systems in wireless network environments. IEEE Trans. Cybern. **50**, 2462–2474 (2020). <https://doi.org/10.1109/TCYB.2019.2924450>
16. Momani, S., Batiha, I.M.: Tuning of the fractional-order PID controller for some real-life industrial processes using particle swarm optimization. (2020)



Measuring Implicit Bias Using SHAP Feature Importance and Fuzzy Cognitive Maps

Isel Grau^{1,2(✉)}, Gonzalo Nápoles³, Fabian Hoitsma³,
Lisa Koutsoviti Koumeri⁴, and Koen Vanhoof⁴

¹ Information Systems Group, Eindhoven University of Technology, Eindhoven, The Netherlands

² Eindhoven Artificial Intelligence Systems Institute, Eindhoven University of Technology, Eindhoven, The Netherlands
i.d.c.grau.garcia@tue.nl

³ Department of Cognitive Science & Artificial Intelligence, Tilburg University, Tilburg, The Netherlands

⁴ Business Informatics Research Group, Hasselt University, Hasselt, Belgium

Abstract. In this paper, we integrate the concepts of feature importance with implicit bias in the context of pattern classification. This is done by means of a three-step methodology that involves (i) building a classifier and tuning its hyperparameters, (ii) building a Fuzzy Cognitive Map model able to quantify implicit bias, and (iii) using the SHAP feature importance to active the neural concepts when performing simulations. The results using a real case study concerning fairness research support our two-fold hypothesis. On the one hand, it is illustrated the risks of using a feature importance method as an absolute tool to measure implicit bias. On the other hand, it is concluded that the amount of bias towards protected features might differ depending on whether the features are numerically or categorically encoded.

Keywords: Fairness · Implicit Bias · Explainable artificial intelligence · Feature importance · Fuzzy cognitive maps

1 Introduction

Fairness is a requirement that decision-makers are obliged to fulfill in any sector, as the law dictates that it is illegal to discriminate against so-called protected personal traits like gender or ethnicity. Therefore, decision-makers need to be able to ensure that their decision-making process is unbiased. Since Artificial Intelligence-based systems often assist decision-makers, such decision support systems are asked to be interpretable and transparent, which is a challenging task for several reasons. First, fairness has multiple definitions depending on the

Isel Grau and Gonzalo Nápoles: Equal contribution.

case at hand, thus being difficult to quantify even if the related decision-making process is unbiased. Second, discrimination might be implicitly encoded in more than one feature of a dataset in non-linear ways and to different extents. Third, a trade-off between model accuracy and interpretability has been observed, thus making it difficult to understand how the decisions are made. This paper lies in the intersection of these challenges in an attempt to quantify implicit bias in pattern classification contexts.

A distinction is made between explicit and implicit bias. The former occurs when the decision-making outcome is influenced by protected features pre-defined by law [22], whereas the latter implies that the decision-making outcome is influenced by seemingly unbiased unprotected features that nevertheless reflect biased beliefs. An example is the redlining practice, where residents of minority neighborhoods receive less favorable treatment from financial institutions. This means that the place of residence can be used to implicitly discriminate against ethnic minorities [25]. Another distinction is made between individual and group fairness. The first implies that similar individuals should be treated similarly and the second that different groups should be treated equally [15].

The majority of the existing bias measures reported in the literature focus on quantifying explicit bias against protected features. Group-based measures often involve probabilistic approaches that only consider the protected feature and the classification outcome effectively [15], ignoring the rest of the information in the data. Individual-based approaches use distance metrics or regression-based tools that might not be sensitive enough to capture discrimination against a single sensitive feature, as shown in [19]. Therefore, we argue that explicit bias towards a single protected feature/group is a naive way to quantify fairness because discriminatory beliefs can find their way into the data through unprotected features that unexpectedly correlate with protected ones.

Existing measures for implicit bias often rely on statistics using regression or correlation coefficients [26]. These approaches cannot easily capture non-linear and higher-order interactions. Other methods include classification rules [9] and causal models [25] which, despite being relatively interpretable, mainly consider one-way interactions among a selected number of features. Moreover, they offer group-wise approaches [26] and require to assume which group is discriminated against, which might lead to reverse discrimination. In addition, existing approaches make poor use of interpretable machine learning, which is deemed an effective tool to enhance the transparency and fairness of models [8]. Feature importance methods are one way to illustrate how the model arrives at a certain prediction. However, in the context of fairness, these methods are mainly used to look for explicit bias since the features suspected of encoding bias are manually chosen by experts [8]. The works published in [1, 10, 16] rely on Shapley Additive Explanations (SHAP) [14], which is one of the most explored model-agnostic explanation methods, for detecting and from there also mitigating explicit bias. However [5], shows that it is difficult to get real insights on relationships among variables only by examining SHAP values.

In [18], the authors introduced a recurrent neural network-based model able to measure implicit bias with regard to seemingly neutral unprotected features. This model leverages Fuzzy Cognitive Maps (FCMs) [12], a soft computing technique able to model the behavior of complex systems and perform what-if simulations. FCMs are able to capture higher-order associations, feedback loops and dependencies within the data. Therefore, we argue that they are a pertinent tool to capture the way that bias implicitly spreads and diffuses within data. This approach transforms the dataset into a fully connected graph, where each node corresponds to a problem feature. The nodes, or concepts, are connected using weighted edges denoting absolute pairwise correlations between features. Each neural concept is assigned an activation value representing its initial influence within the system. FCMs allow concepts to interact with each other by updating these activation values iteratively using a reasoning function. The final activation values after convergence are used as a proxy for implicit bias since the model considers all possible pathways through which bias can propagate through the system. The theoretical contribution of the work published in [18] is a novel reasoning rule coupled with a normalization-like transfer function. These guarantee convergence to a unique fixed point regardless of the initial conditions or diverse point attractors by adjusting the influence of the reasoning rule's linear component. The main limitation of this model is that it relies on domain knowledge to activate the network.

In this paper, we design an experiment to illustrate that feature importance might hide implicit bias against protected features in decision-making problems. In consequence, we argue that feature importance should not be used as a direct proxy to discard bias in a dataset. This experiment is implemented in a three-step methodology that includes fitting a classifier and optimizing its hyperparameters, using our FCM model to quantify implicit bias, and running what-if simulations using SHAP feature importance to feed the FCM model. This initialization overcomes the limitation of setting the initial vector of the FCM model based on domain knowledge. Consequently, it offers a way to measure how an implicitly biased unprotected feature influences the prediction for an instance. An additional theoretical contribution of this paper is using clustering to automatically discover the groups describing a numeric feature, which allows computing the association between numeric and categorical features using Cramér's V coefficient [6]. This strategy removes the need to define the groups manually while dealing with the limitations of determining the association between numeric and nominal features. Towards the end, we discuss the effect of discretizing features on the implicit bias analysis.

The structure of this paper is as follows. Section 2 introduces the preliminaries concerning fuzzy cognitive mapping and the SHAP method used for computing feature importance. Section 3 elaborates on the proposed methodology and the two-fold research hypothesis. Section 4 presents the simulation results while Sect. 5 presents the concluding remarks.

2 Preliminaries

This section will describe the building blocks of our methodology, namely the FCM model and the SHAP method for feature importance.

2.1 Fuzzy Cognitive Maps

The classic FCM model introduced in [12] consists of a collection of meaningful neural entities called concepts that describe the modeled complex system. The interaction between these neurons is governed by a squared weight matrix such that $w_{ij} \in [-1, 1]$. FCMs are knowledge-based recurrent neural networks, and as such, they perform an iterative reasoning process devoted to updating neurons' activation values given an initial condition.

The traditional FCM model uses monotonically increasing transfer functions (such as the sigmoid and hyperbolic tangent functions) to ensure that neurons' activation values are in the desired bounded interval [21]. Moreover, the recurrent process is closed, meaning that the state of the network in the current iteration solely depends on the previous state. These features cause a wide variety of problems ranging from saturation situations (where the activation values move towards the boundaries of the activation interval) to serious convergence problems. Nápoles et al. [18] expanded the quasi-nonlinear FCM model presented in [20] and introduced a re-scaled transfer function to address these drawbacks. This model will be briefly described next.

Let $\mathbf{A}^{(t)} = (a_1^{(t)}, \dots, a_i^{(t)}, \dots, a_m^{(t)})$ be the activation vector produced by an FCM such that $a_i^{(t)}$ is the activation value of the i -th neuron in the t -th iteration and m is the number of neurons. Moreover, $\bar{\mathbf{A}}^{(t)} = (\bar{a}_1^{(t)}, \dots, \bar{a}_i^{(t)}, \dots, \bar{a}_m^{(t)})$ is the raw activation vector where $\bar{a}_i^{(t)}$ is the raw activation value of the i -th neuron in the current iteration. More explicitly, the vector $\bar{\mathbf{A}}^{(t)}$ is given by $\bar{\mathbf{A}}^{(t)} = \mathbf{A}^{(t)}\mathbf{W}$ where $\mathbf{W}_{m \times m}$ is the weight matrix. Equation (1) shows the quasi-nonlinear reasoning rule using a re-scaled transfer function,

$$\mathbf{A}^{(t+1)} = \phi f(\mathbf{A}^{(t)}\mathbf{W}) + (1 - \phi)\mathbf{A}^{(0)} \quad (1)$$

such that $f(\cdot) : \mathbb{R}^m \rightarrow \mathbb{R}^m$ is defined as follows:

$$f(\mathbf{X}) = \begin{cases} \frac{\mathbf{X}}{\|\mathbf{X}\|_2} & \text{if } X \neq \vec{0} \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

where $\|\cdot\|_2$ represents for the Euclidean norm, and $0 \leq \phi \leq 1$ controls the nonlinearity of the reasoning rule. When $\phi = 1$, the model performs as a closed system where the activation value of a neuron depends on the activation values of connected neurons in the previous iteration. When $0 < \phi < 1$, the model adds a linear component to the reasoning rule concerning the initial activation values of neurons [18]. When $\phi = 0$, the model narrows down to a linear regression where the initial activation values of neurons act as regressors [20].

The recurrent reasoning mechanism stops when either (i) the model converges to a fixed point or (ii) a maximal number of iterations T is reached. In the quasi-nonlinear FCM model, we have the following states:

- **Fixed point** ($\exists t_\alpha \in \{1, \dots, (T-1)\} : \mathbf{A}^{(t+1)} = \mathbf{A}^{(t)}$, $\forall i, \forall t \geq t_\alpha$): the FCM produces the same state vector after t_α , thus $\mathbf{A}^{(t_\alpha)} = \mathbf{A}^{(t_\alpha+1)} = \mathbf{A}^{(t_\alpha+2)} = \dots = \mathbf{A}^{(T)}$. If the fixed point is unique, the FCM model will produce the same state vector regardless of the initial conditions.
- **Limit cycle** ($\exists t_\alpha, p, j \in \{1, \dots, (T-1)\} : \mathbf{A}^{(t+p)} = \mathbf{A}^{(t)}$, $\forall i, \forall t \geq t_\alpha$): the FCM produces the same state vector periodically after the period p , thus $\mathbf{A}^{(t_\alpha)} = a_i^{(t_\alpha+p)} = \mathbf{A}^{(t_\alpha+2p)} = \dots = \mathbf{A}^{(t_\alpha+jp)}$, where $t_\alpha + jp \leq T$.
- **Chaos**: the FCM produces different state vectors.

If $\phi = 1$, the re-scaled FCM model is expected to converge to the unique fixed-point attractor provided that the weight matrix W has an eigenvalue that is strictly greater in magnitude than the other eigenvalues and that the initial activation vector $\mathbf{A}^{(0)}$ has a nonzero component in the direction of an eigenvector associated with the dominant eigenvalue [18]. If these conditions are fulfilled, the network will converge to the dominant eigenvalue.

If $0 \leq \phi < 1$, the fixed point will depend on the initial conditions, thus allowing for what-if simulations. However, limit cycles can appear if $\mathbf{A}^{(t)}\mathbf{W} = \vec{0}$ for some t . Similarly, a cycle will be reached if the function $f(\cdot)$ is evaluated on the discontinuity point $\vec{0}$ during the inference process. Overall, a limit cycle can appear if the following expression is fulfilled:

$$\left(\frac{\mathbf{A}^{(t+p-1)}}{\|\mathbf{A}^{(t+p-1)}\mathbf{W}\|_2} - \frac{\mathbf{A}^{(t-1)}}{\|\mathbf{A}^{(t-1)}\mathbf{W}\|_2} \right) \mathbf{W} = 0.$$

The previous equality holds when the vector resulting from the difference of the fractions is perpendicular to every column of \mathbf{W} or such a difference is equal to the null vector. These situations might happen when

$$\frac{\mathbf{A}^{(t+v-1)}}{\|\mathbf{A}^{(t+v-1)}\mathbf{W}\|_2} = \frac{\mathbf{A}^{(t-1)}}{\|\mathbf{A}^{(t-1)}\mathbf{W}\|_2},$$

and

$$\mathbf{A}^{(t+p-1)} = \frac{\|\mathbf{A}^{(t+p-1)}\mathbf{W}\|_2}{\|\mathbf{A}^{(t-1)}\mathbf{W}\|_2} \mathbf{A}^{(t-1)}.$$

The above expression suggests that $\mathbf{A}^{(t+p-1)}$ is equal to $\mathbf{A}^{(t-1)}$ multiplied by a scalar factor, which is the quotient of the Euclidean norms. This means that if an activation vector is multiple of another activation for the same initial stimulus, then the cyclic behavior appears.

2.2 Feature Importance and SHAP Method

Shapley Additive Explanations (SHAP) [14] is a model-agnostic post-hoc method that computes feature importance as an approximation of Shapley values [24]. Shapley values come from the field of coalitional game theory and represent how much a feature brings in for a prediction, in addition to a given subset of features. Formally, a Shapley value S_c represents the importance of the feature c when included in the model g , that is to say:

$$S_c = \sum_{B \subseteq C \setminus \{c\}} \frac{|B|!(|C| - |B| - 1)!}{|C|!} (g(x_{B \cup \{c\}}) - g(x_B)) \quad (3)$$

where C represents the original feature set, B stand for the possible subsets of $C \setminus \{c\}$, while $g(x_{B \cup \{c\}})$ denotes the prediction for the instance x from the model g when including all features in B plus the feature c , and marginalizing over the rest of the features. Shapley values comply with the efficiency property, i.e., the sum of all feature contributions equals the difference between the prediction for x and the average prediction. In other words, this means that the feature attribution can be aggregated for groups of features.

Shapley values can be calculated for a single instance or globally as an aggregation over all instances. However, this aggregation is computed over all possible combinations of feature subsets (or coalitions), therefore it can be computationally expensive. The SHAP implementation [14] builds upon Shapley values theory and reframes it as an additive feature attribution method, i.e., a linear model. For example, the model-agnostic Kernel-SHAP method uses a local linear regression for estimating the values. Other model-specific implementations of SHAP include Tree-SHAP [13], which is optimized for decision trees, random forests, and gradient-boosted trees, by using the number of training examples traversing the tree to represent the background distributions.

3 Methodology

This section presents our research methodology, which consists of the following steps (i) building a classifier, tuning its hyperparameters, (ii) building an FCM model able to quantify implicit bias, fitting the SHAP explainer, and (iii) running simulations. To build the classifier, we need a training dataset (70%) and a separate validation dataset (20%) to perform hyperparameter tuning. Both pieces of data can be combined to build the FCM model and fit the SHAP explainer once the classifier has been built. Finally, we can select some instances from the test set (10%) for running simulations using the SHAP feature importance scores to activate the neurons in the FCM model. In that way, we can quantify how these feature importance scores translate to implicit bias against protected features. Figure 1 summarizes how the dataset is split toward obtaining the (stratified) training, validation, and test sets mentioned above.

The main hypothesis of our research is two-fold. On the one hand, we state that feature importance is not a suitable measure to determine bias against

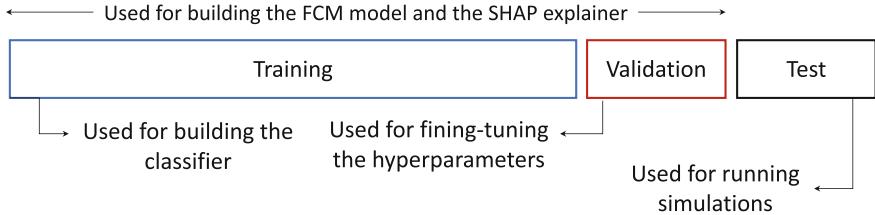


Fig. 1. Blueprint of the data usage in the proposed experiment

protected features. The fact that a protected feature is not regarded as relevant according to the SHAP values does not allow automatically concluding that the decisions are not biased towards that feature. On the other hand, we conjecture that the conclusions about the amount of bias against protected features might differ depending on how the feature is encoded.

3.1 Building the Classification Model

In this paper, we will use a Random Forest (RF) classifier [4] as the core decision model. The aim is to investigate the extent to which protected features are regarded as important (as determined by the SHAP values). In order to ensure the reliability of the analysis, we need to ensure that RF produces the highest prediction rates possible. Therefore, it seems convenient to perform hyperparameter tuning using the training and validation sets.

The hyperparameters to be optimized are the number of estimators, the function used to measure the quality of a split, and the number of features to consider when looking for the best split. In our experiments, we will consider 100, 500, and 1000 estimators, while the maximum number of features can be determined as \sqrt{m} or $\log_2(m)$, with m being the number of features. As for the split quality function, we will consider the Gini impurity and the Shannon information gain. Once the best parameter combination is determined, we will retrain the classifier using all data but the test set.

3.2 Building the FCM-Based Model

The next step in our methodology consists of creating an FCM model to quantify implicit bias based on the model proposed by Nápoles et al. [18].

In the knowledge-based network, each neuron represents a problem feature regardless of whether the feature is nominal or continuous since neurons will be activated using the SHAP values. The weights connecting the neurons denote the degree of association between the features in the dataset. In this step, we need to select the proper method to compute the association between two features based on whether these features are numerical or nominal. Let F_i and F_j denote two problem features denoted by neurons C_i and C_j , respectively. The weight w_{ij} connecting C_i and C_j will be determined as follows:

- **Case 1.** Both F_i and F_j are numeric. In this case, the weight w_{ij} is computed as the absolute Pearson's correlation [23].
- **Case 2.** Both F_i and F_j are nominal. In this case, the weight w_{ij} is computed using the Cramér's V coefficient [6].
- **Case 3.** Either F_i or F_j is numeric. In this case, we first transform the numerical feature into a nominal one by using the fuzzy c -means algorithm [3]. The optimal number of clusters is determined using the fuzzy partition coefficient [2], which measures the amount of overlap among the fuzzy clusters. Once the numeric feature has been discretized, computing the weight w_{ij} narrows down to the second case explained above.

More details on the automatic detection of categories describing a numerical feature are provided next. Firstly, the feature values x_i are represented as symmetric tuples (x_i, x_i) that can be represented in a plane. This suggests that the fuzzy c -means algorithm will discover c fuzzy sets along the identity line, where each fuzzy cluster denotes a category.

The fuzzy component of this algorithm a membership is given by a matrix $\mathbf{U}_{n \times c}$ such that n is the number of data points to be processed (i.e., the number of instances in the dataset). As such, $\mu_{ij} \in \mathbf{U}$ represents the degree to which the i -th data point belongs to the j -th fuzzy cluster. The algorithm returns a matrix of prototypes $\mathbf{Z}_{1 \times c}$ denoting the cluster centers. The fuzziness of fuzzy c -means is controlled by a fuzzification coefficient $\alpha \in [1, \infty]$ where larger values indicate more fuzziness. Equations (4) and (5) display how to compute the membership values and the fuzzy prototypes, respectively:

$$\mu_{ij} = \frac{1}{\sum_{l=1}^c \left(\frac{\|x_i - z_l\|}{\|x_i - z_j\|} \right)^{2/(\alpha-1)}} \quad (4)$$

$$z_j = \frac{\sum_{i=1}^n \mu_{ij}^\alpha \cdot x_i}{\sum_{i=1}^n \mu_{ij}^\alpha}. \quad (5)$$

Since the number of categories needs to be discovered, we need to execute the clustering algorithms several times for different numbers of clusters (normally, from 2 to 10). The setting with the largest fuzzy partition coefficient is adopted to describe the numerical feature. This coefficient measures the amount of overlap between the fuzzy clusters and can be computed as follows:

$$FPC = \frac{1}{n} \sum_{j=1}^c \sum_{i=1}^n (\mu_{ij})^\alpha. \quad (6)$$

It is worth mentioning that the approach to handling numeric-nominal pairs of features is a contribution of this paper. In the method proposed in [18], the authors used the R-squared coefficient of determination [17] to quantify the percentage of variation in the numeric feature that the nominal one explains. While statistically sound, this strategy poses two issues. Firstly, it was assumed that such association is symmetric. Secondly, the results of analyzing the amount

of bias against a feature might change depending on the encoding of that feature. However, automatically discovering groups associated with protected features (as done with the fuzzy c -means algorithm) could remove the subjectivity of defining the protected groups. Ultimately, the fuzzy clustering approach will help study the second hypothesis of our study.

3.3 Initializing the FCM-Based Model

The final step of our methodology for fairness analysis notably differs from the model in [18], where selected neurons were randomly activated to perform the reasoning process depicted in Eq. (1). In our study, the initial activation vector $\mathbf{A}^{(0)} = (a_1^{(0)}, \dots, a_i^{(0)}, \dots, a_m^{(0)})$ used the trigger reasoning will be unitized with the SHAP values for selected instances. Therefore, starting from feature importance, we will quantify how the associations among the variables increase the activation values of protected features. This can be done by exploring the final activation vector $\mathbf{A}^{(T)} = (a_1^{(0)}, \dots, a_i^{(T)}, \dots, a_m^{(T)})$. Such an increase can be understood as implicit bias: the protected feature is not considered relevant by itself when making the decision; however, its patterns are encoded into unprotected features through correlations and associations.

4 Simulations

In the numerical simulations, we use two datasets to illustrate the extent to which protected features are implicitly biasing the final decision of randomly selected individuals. First, we compute the global and local SHAP values after training a classifier. Second, we build an FCM model where weights are a square association matrix characterizing the interaction between the variables. In this model, the initial activation vector uses the local SHAP values of a randomly selected individual. Third, we quantify implicit bias related to specific individuals using w.r.t. interesting protected features.

4.1 German Credit

The first case study in our simulations concerns the German Credit dataset [7]. This binary classification dataset consists of 1000 credit applications, from which 700 are classified as good credit risk, while 300 are labeled as bad credit risk. Applicants are described by 20 qualitative and quantitative features. In this dataset, *age* (F13), *foreign worker* (F20) and *gender* (F9) are considered to be protected and will be the center of our analysis.

Table 1 shows the association values between all features and the protected ones. Notice that some unprotected features have rather strong associations with the protected ones. For example, the unprotected features employment since (F7), residence since (F11) and housing (F15) are strongly associated with *age* (F13) which might be an indication that they implicitly encode *age* bias. Table 1 also displays the global SHAP values using a Random Forest as a classifier. This

measure indicates that checking account (F1) and duration (F2) are key features when making the decision. It is worth mentioning that the weight matrix of our FCM model is symmetric, thus the matrix is diagonalizable and its eigenvalues are real with a dominant eigenvalue.

Table 1. Association values between protected and unprotected features in the German credit dataset. Global SHAP values provide information about feature importance using a random forest as a classifier

ID	Features	Associates with			SHAP
		Gender	Age	Foreign worker	
F1	Checking account	0.03	0.08	0.08	0.093
F2	Duration	0.11	0.04	0.17	0.035
F3	Credit history	0.12	0.13	0.07	0.023
F4	Purpose	0.15	0.14	0.17	0.019
F5	Credit amount	0.08	0.03	0.04	0.026
F6	Savings account	0.07	0.10	0.04	0.026
F7	Employment since	0.22	0.37	0.08	0.012
F8	Installment rate	0.13	0.06	0.13	0.009
F9	Gender	1.00	0.11	0.05	0.015
F10	Other debtors	0.01	0.02	0.12	0.005
F11	Residence since	0.11	0.27	0.08	0.009
F12	Property	0.09	0.17	0.14	0.017
F13	Age	0.11	1.00	0.02	0.013
F14	Other installment	0.05	0.03	0.04	0.018
F15	Housing	0.23	0.21	0.07	0.017
F16	Existing credits	0.10	0.15	0.02	0.004
F17	Job	0.09	0.12	0.11	0.010
F18	People liable	0.20	0.13	0.07	0.003
F19	Telephone	0.07	0.09	0.10	0.010
F20	Foreign worker	0.05	0.02	1.00	0.003

To study our two-fold hypothesis, we will select two randomly selected instances from the test set (one belonging to the good credit risk class and another belonging to the bad credit risk class). After verifying that instances have been correctly classified, we determine their associated SHAP values, which will be used to activate the FCM model. In that way, we can study how the implicit bias behaves given these feature importance scores.

Figure 2 shows the feature importance computed by SHAP for randomly selected positive and negative instances. The model's prediction for the positive instance is 0.98 in terms of the probability of obtaining a good credit risk assessment. The width of the bars corresponds to the magnitude of the feature attribution. For example, the checking account (F1) contributes positively to increasing the probability with a value of 0.11, compared to the average prediction of the dataset for the positive class (0.70). In the same way, the features credit amount (F5) and savings account (F6) contribute 0.04 each, while other features such as other installment (F14), duration (F2), *gender* (F9), and housing (F15) have smaller positive contributions. The value of purpose (F4) reduces the probability of getting a positive outcome for this instance.

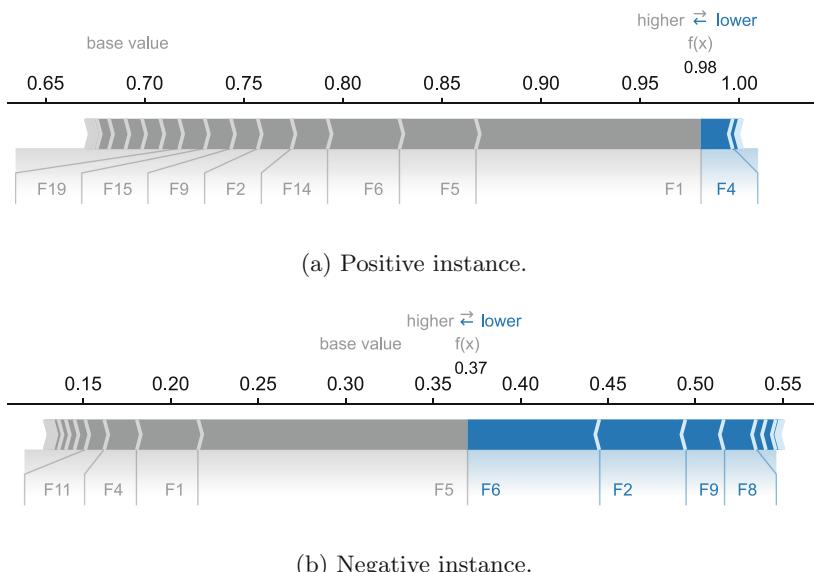


Fig. 2. Force plot depicting feature attribution for randomly selected positive and negative instances of the german credit dataset. The results show that the protected features do not contribute notably to the predictions

In contrast, the model's prediction for the randomly selected negative instance is 0.30, while the prediction for this instance is 0.37. In this case, the difference with the base value is not large, resulting from the combination of positive and negative contributions of several features. For example, credit amount (F5) contributes to increasing the probability, while savings account (F6) decreases the probability by a magnitude of 0.08. Other features contribute negatively and positively to the instance prediction.

Next, we use the SHAP values as an initialization vector in our FCM model for studying the implicit bias starting from the random instances above. For the

first instance, Fig. 3 depicts the activation values of the protected features (*gender*, *age*, and *foreign worker*). Although the initial activation values of neurons denoting these features are rather low (0.01 for *gender* and *age*, and 0 for *foreign worker*), we can see a clear increase in their activation after a few iterations. This pattern is consistent across different values of ϕ , obtaining higher values as ϕ increases. This increment comes from the interaction with other features, suggesting the presence of implicit bias in the dataset.

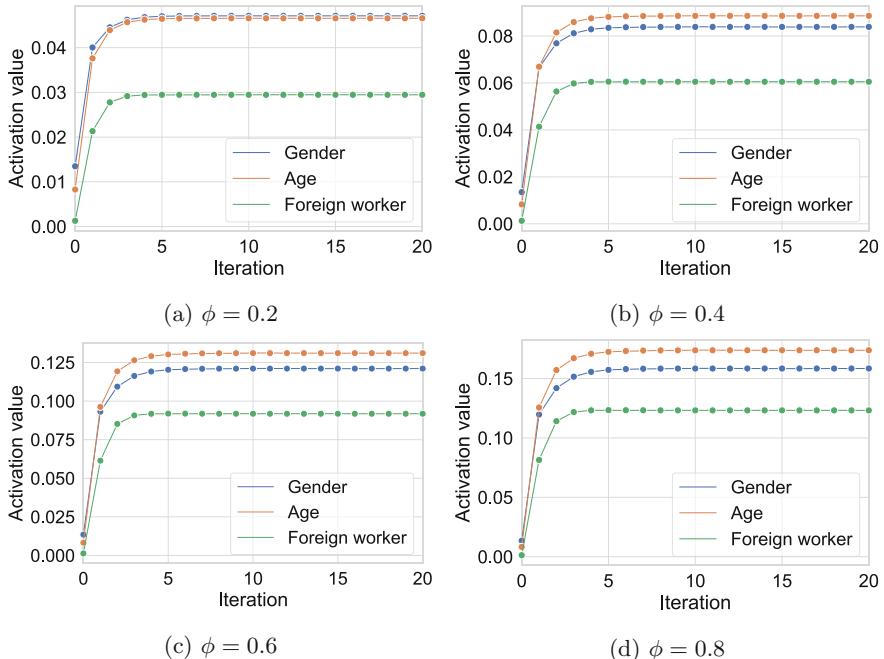


Fig. 3. Activation values of neurons denoting protected features for a positive instance in the german credit dataset for different ϕ Values. Although the neurons associated with the protected features are initialized with very small values, we can observe an increase in their activation values

We repeat the analysis using the SHAP values corresponding to the negative instance as the initialization vector in our FCM model. Figure 4 plots the activation values of the protected features *gender*, *age*, and *foreign worker*. The initial SHAP values for the protected features are small, with *gender* (F9) having an activation value of 0.02 and *age* (F13) and *foreign worker* (F20) having no direct contribution. However, after a few iterations, we can observe an increase in the values of the three protected features, with *gender* (F9) and *age* (F13) as the most excited neurons. This pattern is also consistent across different ϕ values, where the higher the ϕ value, the more pronounced the increment. Recall that,

the smaller the ϕ parameter, the more linear the model, therefore the activation values are more similar to the initial feature importance.

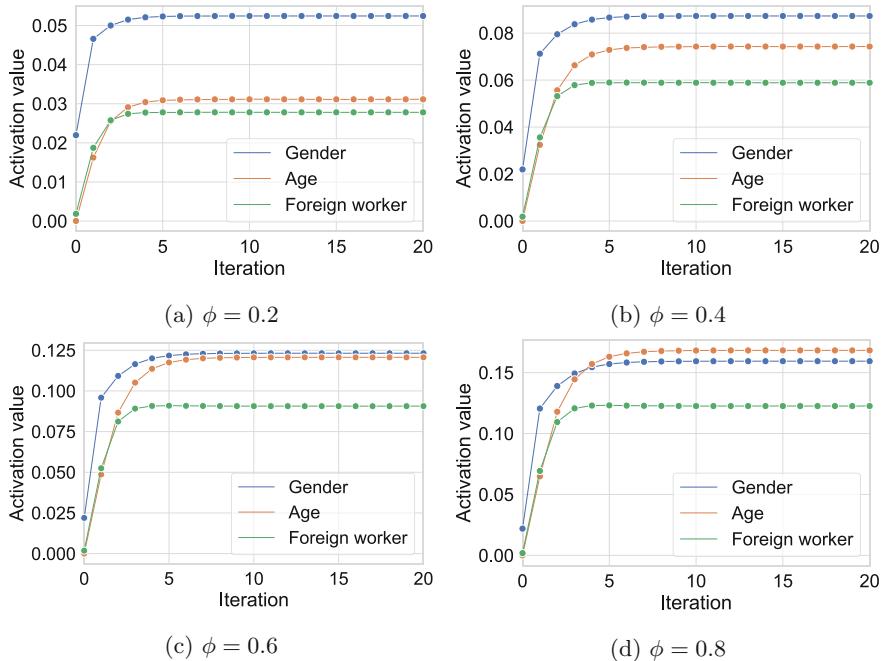


Fig. 4. Activation values of neurons denoting protected features for a negative instance in the german credit dataset for different ϕ values. Although the neurons associated with the protected features are initialized with very small values, we can observe an increase in their activation values

The reader could argue that Figs. 3 and 4 do not allow assessing whether the protected features rank comparably w.r.t. the SHAP values and the outputs of the FCM model. In other words, the activation values of neurons representing protected features might be significantly smaller than those denoting unprotected features. Figure 5 displays the aggregated SHAP values and the normalized activation values produced by the FCM model ($\phi = 0.8$) in the last iteration for the negative instance. This figure shows that *age* (F13) is deemed the least relevant feature when classifying the instance according to the SHAP values. However, the amount of bias captured by the FCM model against *age* is quite significant. At the same time, the protected feature *gender* (F9) was recognized by SHAP and the FCM model as important. Finally, both the SHAP values and the FCM model agree that *foreign worker* (F20) does not seem to play a relevant role when classifying the instance.

The simulations provide evidence of implicit bias even when protected features are not deemed explicitly important according to SHAP values. This hap-

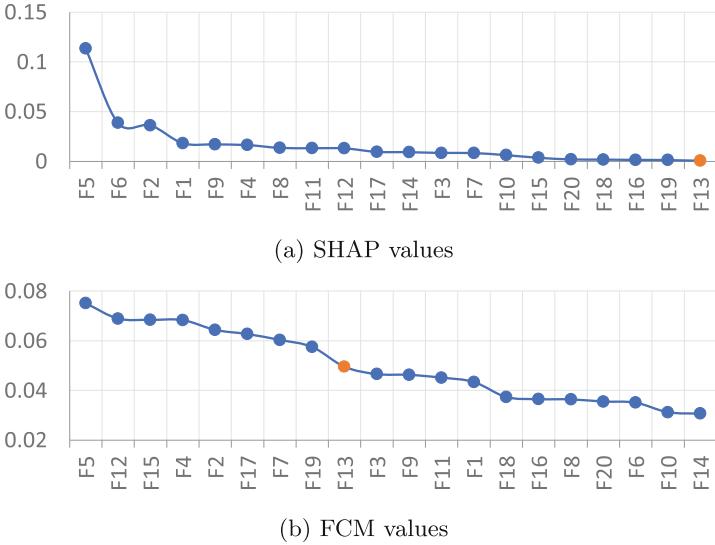


Fig. 5. Absolute SHAP values and bias scores computed by the FCM model for the randomly selected negative instance of the german credit dataset. In this case, the protected feature *age* (F13) is the least relevant according to SHAP, but this feature does involve a moderate amount of bias

pens because features are not independent, and as such, unprotected features can partially encode the information of protected ones.

The final point to be discussed is whether the conclusions concerning implicit bias change if protected numeric features are categorically encoded. In the fairness literature, numeric features are often associated with protected groups such as females or young people. While the results in Figs. 3 and 4 report more bias against *age* than *gender*, the opposite is concluded in [18] and [19] in which protected features are analyzed as a whole. The cause of this difference is that in the approach presented in this paper, the fuzzy c -means algorithm automatically detects such groups. This remark agrees with the results presented in [18], where the authors showed how analyzing bias at a group level leads to different conclusions, even using the same model and metrics.

4.2 Adult Dataset

The second case study concerns the Adult dataset [11]. For this dataset, the pre-processing step only involved the normalization of numeric features. Table 2 shows the association values between all features and protected features *race* (F9) and *sex* (F10) in Adult dataset along with the global SHAP values per feature. We observe that some unprotected features have rather strong associations with protected ones: native-country (F14) is strongly associated with *race* (F9), and the protected feature *sex* (F10) is strongly associated with marital-status (F6),

occupation (F7) and relationship (F8). According to the global SHAP values, marital-status (F6) and relationship (F8) are key features when making the decision, while both protected features are deemed relatively irrelevant.

Table 2. Association values between protected and unprotected features in the adult dataset. Global SHAP values provide information about feature importance using a random forest as a classifier

ID	Features	Associates with		SHAP
		Race	Sex	
F1	Age	0.04	0.07	0.039
F2	Workclass	0.06	0.15	0.011
F3	Fnlwgt	0.11	0.04	0.010
F4	Education	0.07	0.1	0.026
F5	Education-num	0.07	0.09	0.037
F6	Marital-status	0.08	0.46	0.066
F7	Occupation	0.08	0.42	0.040
F8	Relationship	0.1	0.65	0.055
F9	Race	1.0	0.12	0.004
F10	Sex	0.12	1.0	0.014
F11	Capital-gain	0.02	0.03	0.046
F12	Capital-loss	0.03	0.05	0.012
F13	Hours-per-week	0.06	0.26	0.026
F14	Native-country	0.41	0.07	0.003

Figure 6 displays the feature importance computed by SHAP for randomly selected positive and negative instances, which are selected from the test set after verifying they have been correctly classified.

Figure 6 shows that the model’s prediction for the positive class is 0.26, while the prediction for this instance is 0.64. Features occupation (F7), marital-status (F6), and relationship (F8) contribute positively to increasing the probability. In contrast, the *race* (F9) feature reduces the probability of getting a positive outcome for this instance by 0.06.

The overall dataset prediction for the negative class is 0.74, while the prediction for the randomly selected negative instance is 0.82. Feature *race* (F9) has the largest positive contribution (0.09) followed by education-num (F5) and education (F4). Features marital-status (F6), hours-per-week (F13) and relationship (F8) reduce the probability of getting a positive outcome for this instance.

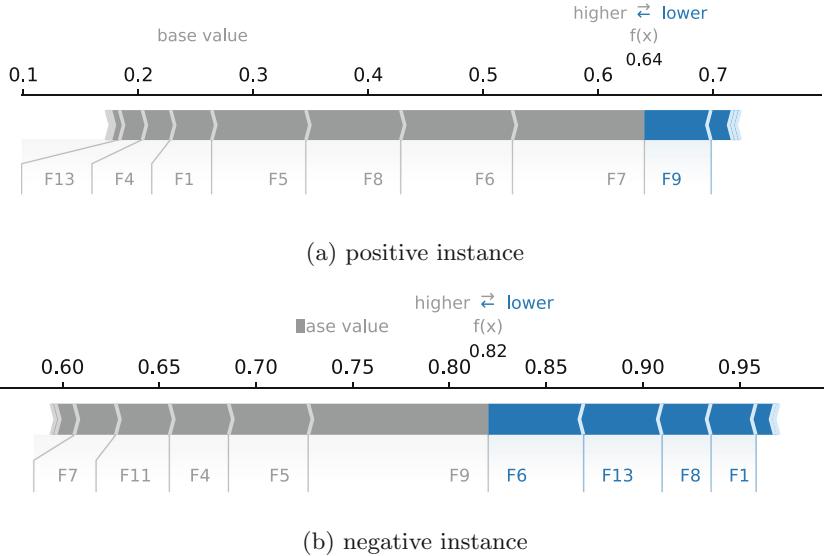


Fig. 6. Force plot depicting feature attribution for randomly selected positive and negative instances of the adult dataset. the results show that the protected feature F9 contributes to the predictions

Next, we use the SHAP values as an initialization vector in our FCM model for studying the implicit bias starting from the two randomly selected instances. Figure 7 depicts the activation values of the protected features (*race* and *sex*). Although the initial activation values of neurons denoting *sex* (F10) start close to zero, we can see a clear increase in their activation values after a few iterations for all levels of ϕ , obtaining higher values as ϕ increases reaching 0.3 at $\phi = 0.8$. Comparatively, *race* (F9) does not deviate much from its initial activation value (starts at 0.05 and reaches 0.14 at $\phi = 0.8$).

We repeat our analysis using the SHAP values corresponding to the negative instance as an initialization vector in our FCM model. Figure 8 plots the activation values of the protected features *race* (F9) and *sex* (F10). We can observe that the initial SHAP value associated with *sex* (F10) is close to zero, while the initial value for *race* (F9) is almost 0.1. Again, after a few iterations, the same pattern as before emerges: *sex* (F10) is twice as important as *race* (F9) thus implicitly biasing the rest of the features in the complex system.

In an effort to assess whether the protected features rank comparably w.r.t. the SHAP values and the outputs of the FCM model, the reader can refer to Fig. 9 which displays the aggregated SHAP values and the normalized activation values produced by the FCM model in the last iteration for the negative instance when $\phi = 0.8$. This figure shows that the protected feature *sex* (F10) is one of the least relevant features according to the SHAP values. However, our FCM ranks *sex* (F10) as the second most relevant feature thus implying that it is

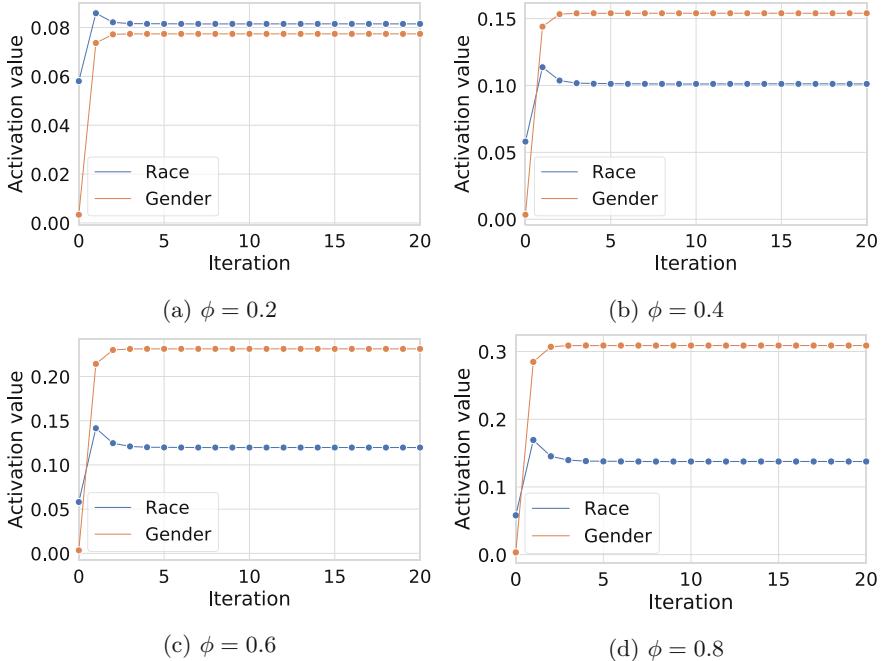


Fig. 7. Activation values of neurons denoting protected features for a positive instance in the adult dataset for different ϕ values

implicitly influencing all other features to a high extent. At the same time, the exact opposite trend is observed regarding the protected feature *race* (F9) thus proving that a linear method, such as SHAP fails to capture the feedback loops and implicit interactions as encoded in the data.

The simulations provide evidence that, if a protected feature seems to be important by the SHAP method, the rest of the unprotected features do not necessarily have to partially encode bias related to that particular protected feature. This is a situation where a protected feature is actually independent relative to this negative instance (namely *gender* here). On the other hand, *race* confirms our main finding: it is a moderately important feature according to SHAP, but its implicit influence in the system is high.

5 Conclusions

This paper developed a methodology to study the relationship between feature importance and implicit bias. Firstly, a classifier was built and optimized in order to predict unseen instances. Secondly, we built a recurrent neural network devoted to quantifying implicit bias from the statistical association patterns between the features. Finally, SHAP feature importance values associated with testing instances were used to trigger the reasoning mechanism.

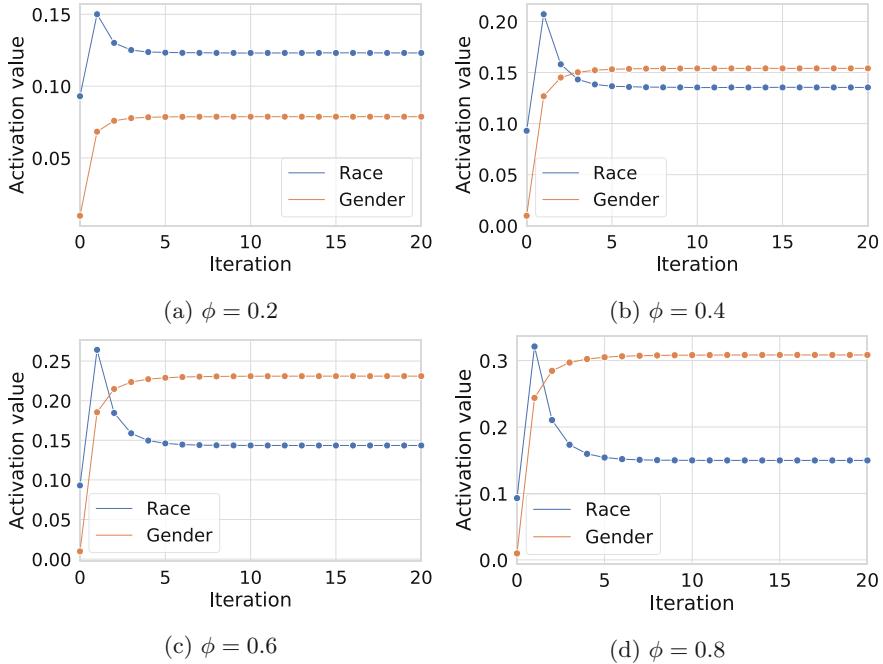


Fig. 8. Activation values of neurons denoting protected features for a negative instance in the adult dataset for different ϕ values

The simulations using the German Credit and the Adult datasets as case studies showed that there might be situations where protected features are not deemed relevant, yet the amount of implicit bias against them was found significant. In this sense, feature importance and the measures for quantifying explicit bias (such as the measure proposed in [19]) fail to capture the extent to which unprotected features encode the implicit bias patterns. The FCM model presented in [18] tackles this limitation by exploiting the statistical associations between variables. However, the strategy to activate the network relied on expert knowledge, which is often difficult to acquire and quantify. In our approach, such knowledge is replaced with the SHAP values, which provide an elegant alternative to quantify the extent to which each variable is active in the model. In this way, we can measure the extent to which an implicitly biased unprotected feature influences the prediction for a single instance.

Another aspect studied in our paper was the extent to which encoding the numeric features (when analyzing the association between categorical and numerical features) would change the bias patterns. With this aim, we contrasted our results with the simulations reported in [18, 19] where more bias against gender than age was found. In our experiments, we observed more discrimination against age than gender after detecting the groups automatically using a clustering algorithm. These differences raise concerns about the consistency of existing

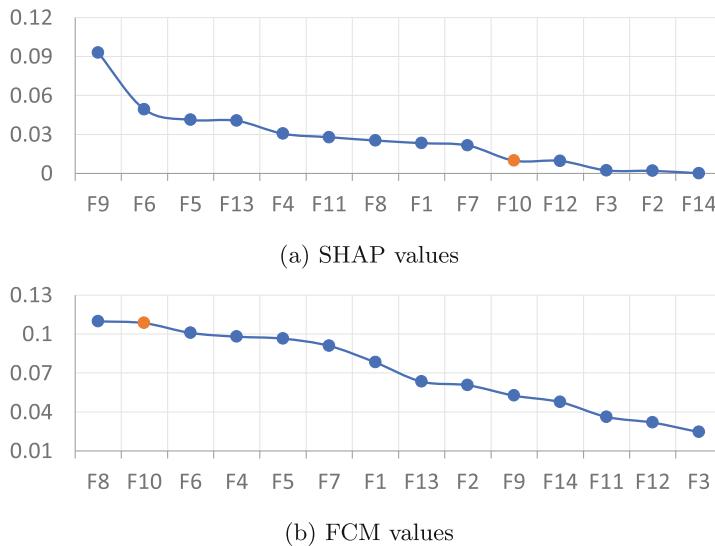


Fig. 9. Absolute SHAP values and bias scores computed by the FCM model for the randomly selected negative instance of the adult dataset. In this case, the protected feature *sex* (F10) is deemed not very relevant according to SHAP, but it does involve a significant amount of bias

approaches for detecting bias since a malicious decision-maker could select one approach over another to justify biased decisions.

References

1. Alves, G., Bhargava, V., Bernier, F., Couceiro, M., Napoli, A.: Fixout: an ensemble approach to fairer models (2020)
2. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Springer, New York, NY (2013)
3. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: the fuzzy c-means clustering algorithm. Comput. Geosci. **10**(2), 191–203 (1984)
4. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
5. Cesaro, J., Gagliardi Cozman, F.: Measuring unfairness through game-theoretic interpretability. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 253–264. Springer (2019)
6. Cramér, H.: Mathematical Methods of Statistics. Princeton University Press (2016)
7. Dua, D., Graff, C.: UCI machine learning repository (2017)
8. Fang, B., Jiang, M., Cheng, P. Y., Shen, J., Fang, Y.: Achieving outcome fairness in machine learning models for social decision problems. In: Bessiere, C., (ed) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, pp. 444–450 (2020)
9. Hajian, S., Domingo-Ferrer, J.: A methodology for direct and indirect discrimination prevention in data mining. IEEE Trans. Knowl. Data Eng. **25**(7), 1445–1459 (2012)

10. Hickey, J.M., Di Stefano, P.G., Vasileiou, V.: Fairness by explicability and adversarial SHAP learning. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, Proceedings, Part III, pp. 174–190. Springer (2020)
11. Kohavi, R.: Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp. 202–207. AAAI Press (1996)
12. Kosko, B.: Fuzzy cognitive maps. *Int. J. Man Mach. Stud.* **24**(1), 65–75 (1986)
13. Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.-I.: From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2**(1), 2522–5839 (2020)
14. Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., (eds) *Advances in Neural Information Processing Systems*, vol. 30, pp. 4765–4774. Curran Associates, Inc., (2017)
15. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. *ACM Comput. Surv. (CSUR)* **54**(6), 1–35 (2021)
16. Meng, C., Trinh, L., Nan, X., Enouen, J., Liu, Y.: Interpretability and fairness evaluation of deep learning models on mimic-iv dataset. *Sci. Rep.* **12**(1), 1–28 (2022)
17. Nagelkerke, N.J.D., et al.: A note on a general definition of the coefficient of determination. *Biometrika* **78**(3), 691–692 (1991)
18. Nápoles, G., Grau, I., Concepción, L., Koumeri, K., Papa, J.P.: Modeling implicit bias with fuzzy cognitive maps. *Neurocomputing* **481**, 33–45 (2022)
19. Nápoles, G., Koumeri, L.K.: A fuzzy-rough uncertainty measure to discover bias encoded explicitly or implicitly in features of structured pattern classification datasets. *Pattern Recogn. Lett.* **154**, 29–36 (2022)
20. Nápoles, G., Salgueiro, Y., Grau, I., Espinosa, M.L.: Recurrence-aware long-term cognitive network for explainable pattern classification. *IEEE Trans. Cybern.* 1–12 (2022)
21. Nápoles, G., Salmeron, J.L., Froelich, W., Falcon, R., Espinosa, M.L., Vanhoenshoven, F., Bello, R., Vanhoof, K.: Fuzzy cognitive modeling: theoretical and practical considerations. In: *Intelligent Decision Technologies 2019*, pp. 77–87. Springer (2020)
22. Ntoutsi, E., Fafalios, P., Gadiraju, U., Iosifidis, V., Nejdl, W., Vidal, M.-E., Ruggieri, S., Turini, F., Papadopoulos, S., Krasanakis, E., et al.: Bias in data-driven artificial intelligence systems—an introductory survey. *Wiley Interdiscip. Rev.: Data Mining Knowl. Discov.* **10**(3), e1356 (2020)
23. Rovine, M.J., Von Eye, A.: A 14th way to look at a correlation coefficient: correlation as the proportion of matches. *Am. Stat.* **51**(1), 42–46 (1997)
24. Shapley, L.S.: A value for n-person games. In: *Contributions to the Theory of Games*, vol. 2, no. 28, pp. 307–317 (1953)
25. Zhang, J., Bareinboim, E.: Fairness in decision-making—the causal explanation formula. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32 (2018)
26. Žliobaitė, I.: Measuring discrimination in algorithmic decision making. *Data Min. Knowl. Disc.* **31**(4), 1060–1089 (2017). <https://doi.org/10.1007/s10618-017-0506-1>



Predicting and Explaining Variations in Software Effort Estimation Using Adaptive Fuzzy-Neural Networks with Clustering

Riyadh A. K. Mehdi^(✉)

Artificial Intelligence Research Centre, College of Engineering & Information Technology,
Ajman, UAE
r.mehdi@ajman.ac.ae

Abstract. Software effort estimation is a significant and critical step in software development; an accurate estimate is vital to a software project's planning, scheduling, budgeting, and successful completion. This paper investigates the effect on prediction accuracy of introducing an initial clustering phase before applying an Adaptive Neuro-Fuzzy Inference System to estimate software efforts. Also, we explore the most significant determinants of variations in software efforts. The China dataset shows that dividing projects into groups using a clustering algorithm has reduced the root mean square error by 34%. Also, neural network sensitivity analysis has revealed that the resources required to complete the project are the most influential factor determining variations in software efforts for small-sized projects, followed by the product delivery rate. However, for medium and large-sized projects, the effect of resources is more significant than the delivery rate. Project duration comes third in importance for medium and large-sized projects; however, the number of function points is more important than project duration for small projects. Other metrics have little influence on software effort variations, with the number of deleted functional requirements having the slightest effect. These findings can significantly affect the accuracy of software effort estimation through proper analysis and computation of the factors that influence software efforts most. Future work will investigate model performance using different attributes, such as project type, for clustering large heterogeneous datasets.

Keywords: Software cost estimation · Fuzzy inference · Clustering · Neural networks · And fuzzy-neural systems

1 Introduction

An accurate estimate of the time required to develop software is one of the most important aspects of managing software development initiatives. It is essential for project managers to forecast software development efforts accurately. Estimating software development efforts remains a difficult problem that attracts many researchers. Numerous models for predicting software effort have been devised [1–4]. Conventional models predict project costs based on cost drivers and input parameters using mathematical formulas. Included

among the predictors are parameters that measure project size based on the number of lines of source code or function points, the number of software developers, and other processes and product-related metrics [4]. The Constructive Cost Model (COCOMO) is the most widely used parametric cost modelling technique for estimating software effort at various phases among software cost estimation techniques [5]. There have also been non-parametric models of software cost estimation based on soft computing techniques such as artificial neural networks (ANN) [3, 6] and fuzzy logic [7–9]. ANNs are adept at modelling complex nonlinear relationships; they function as a massive parallel-distributed processor comprised of basic processing units and neurons that can learn data patterns [6].

An ANN resembles the brain in two ways [6]: (1) it acquires knowledge from its environment through a learning process, and (2) it stores the acquired knowledge as interneuron connection weights. Fuzzy logic is a mathematical paradigm for coping with uncertainty and imprecise information; it maps the input space to the output space using a set of if-then principles developed by a domain expert [7–9].

Kosko [10] demonstrated that an additive fuzzy system could accurately approximate any continuous real-world function. Due to the imprecise and ambiguous nature of software effort cost variables, a software effort estimation model that employs a fuzzy inference approach provides a framework for addressing these properties. Implementing fuzzy systems necessitates that distinct cost factors be represented by fuzzy sets with corresponding membership functions. Below, we describe succinctly the techniques used to estimate software effort.

The adaptive neuro-fuzzy inference system (ANFIS) combines a feedforward neural network and a fuzzy inference system. The ANFIS neural network employs either a gradient-descent learning rule or a backpropagation and least squares learning mechanism [11]. The fuzzy logic component considers the imprecision and uncertainty of the modelled system. In contrast, the neural network component uses its learning algorithm to modify the fuzzy inference system's membership functions [12]. Using this hybrid technique, an initial fuzzy model and its input parameters are derived from rules extracted from the input-output data of the modelled system. The neural network then fine-tunes the principles of the initial fuzzy model to generate the system's final ANFIS model [12]. The ANFIS learning mechanism estimates the parameters in order for the ANFIS architecture to represent both the Sugeno and Tsukamoto fuzzy models [13]. This study attempts to develop a software effort estimation model by first clustering the dataset's projects and then constructing an ANFIS model for each cluster.

2 Literature Review

The calibration of the functional complexity weight (CFCW) algorithm was presented by Hai et al. [14] in order to calibrate the standardised functional complexity weights. Their research aims at (1) estimate a more precise software size that corresponds to particular software applications, (2) reflect software industry trends, and (3) enhance the effort estimation of software projects. Calibration was performed utilizing a standard International Function Point User Group (IFPUG FPA) method based on the Bayesian ridge regressor model. They reported an increase in accuracy of 10.39% in root mean

square error (RMSE) for ungrouped projects and 27.75% in the mean percentage difference between the individual sectors and the dataset for all sectors. As a second step, they upgraded their CFCW algorithm to CFCWO by adding an optimization component based on an ensemble model known as the voting regressor. According to their evaluations, CFCWO outperformed CFCW with a reduction in RMSE of 24.62% for ungrouped initiatives and a mean of 12.28% for individual sectors.

Using generalized fuzzy number software, Azzeh et al. [15] proposed an analogy-based effort estimation based on analogy. To address the inherent uncertainty of attribute measurement, they combined fuzzy set theory and Grey Relational Analysis to create a method for measuring similarity. They reported increased prediction accuracy when utilizing multiple project attributes with weights. In addition, they proposed a method based on Kendall's coefficient of concordance to represent the impact of each attribute on software development efforts. They reported that this procedure had rendered the process of feature selection unnecessary.

Sharma and Vijayvargiya [16] utilized a neuro-fuzzy model to predict the cost of software development efforts. Using the genetic elephant herding optimization (GEHO) algorithm, the training of the model is optimised. They evaluated model performance using five datasets of industrial projects and four error-based performance evaluation metrics: mean relative magnitude error, median relative magnitude error, root mean square error and prediction accuracy. Experiments revealed that the neuro-fuzzy model based on GEHO outperformed other prominent soft computing methods such as linear regression, support vector regression, wavelet ANN, and decision tree-based software effort estimation methods.

Lopez-Martin et al. [17] compared a fuzzy logic model for estimating the effort of small software programs employing triangular, trapezoidal, and Gaussian membership functions to a linear regression model. Using data from 105 small-scale programs, they developed fuzzy logic and linear regression models; they then contrasted the estimates derived by these models using 20 small programs. Their MMRE-based results demonstrate that the fuzzy model is marginally more accurate than the linear regression model.

Wei et al. [18] examined the possibility of integrating a neuro-fuzzy model with the System Evaluation and Estimation of Resource Software Estimation Model (SEER-SEM). They reported that their model is capable of learning, has minimal sensitivity, generalizes effectively, and accepts linguistic input values. According to their model evaluation, combining a neuro-fuzzy model with the SEER-SEM algorithm yields more accurate estimates than using the SEER-SEM algorithm alone. Hodgkinson and Garratt [19] introduced a neuro-fuzzy model as an alternative to algorithmic models for effort estimation. The model predicts project effort based on predictors of project magnitude and duration.

Huang et al. [20, 21] proposed and evaluated a neuro-fuzzy Constructive Cost Model (COCOMO) that incorporates the features of a neuro-fuzzy approach, such as learning capability and inference while retaining the characteristics of the COCOMO model. They reported that the model permits continuous rating and linguistic values for input and effectively handles imprecise and ambiguous data. Their findings, based on data from industry projects, indicate that adding a fuzzy inference with a component of learning

ability can substantially increase estimation accuracy in comparison to the well-known COCOMO model.

Xia et al. [22] developed the Neuro-Fuzzy Function Point model, which calibrates Function Point (FP) models using the neuro-fuzzy technique. This model's objectives were to enhance the FP complexity weight systems using fuzzy logic, modify the weight values of the unadjusted FP using a neural network, and generate a calibrated FP count for more precise measurements. They evaluated the model using the International Software Benchmarking Standards Group (ISBSG) data repository and reported a 22% improvement in software effort estimation accuracy after calibration.

Wong et al. [23] combined neural networks and fuzzy logic to enhance the precision of backfiring size estimations. This study calibrated the conversion ratios using a neuro-fuzzy approach to reduce the error margin. A comparison was made between the calibrated prediction model and the default conversion ratios. As a consequence, the calibrated ratios maintained the inverse curve relationship between the programming language level and the number of function points, and the size estimation accuracy improved marginally. Karunakaran and Sreenath [24] compiled a list of 150 journal and conference papers on software scalability and effort/cost estimation.

3 Research Methodology

This work examines the effect on the accuracy of estimating software efforts by initially clustering the input data, using a k-means algorithm [25], before using ANFIS to predict software efforts. For each cluster, an ANFIS model is built to predict software efforts. The following is a brief overview of k-means clustering and ANFIS. In addition, the ANFIS model is used to execute a sensitivity analysis to determine the most influential factors that explain variations in software effort estimation.

3.1 K-Means Clustering

Clustering typically employs unsupervised techniques to arrange comparable objects [25]. k -means is an analytical technique that, for a given value of k , identifies k clusters of objects based on their proximity to the k n -dimensional centroids for a collection of objects with n measurable attributes. The k -means clustering algorithm consists of the four stages detailed below [26, 27]:

- Select a value for k and the k initial estimates for the k n -dimensional centroids that correspond to it.
- Compute the distance to each centroid for each data point and assign each n -dimensional data point to the closest cluster based on its distance from the cluster's centroid. This relationship determines the initial k clusters.
- Recompute the centroid of each cluster created in the previous phase.
- Repeat stages 2 and 3 until the algorithm achieves convergence. When the computed centroids stabilize or the designated data points oscillate back and forth between iterations, convergence occurs.
- Clustering was implemented using the R programming language's `kmeans()` utility [27].

3.2 ANFIS Architecture

Typical ANFIS architecture is depicted in Fig. 1 [28]. A circle represents a permanent node, while a square represents an adaptive node. A two-rule rule base for a first-order Sugeno fuzzy model is expressed as follows:

$$1. \text{ If } x \text{ is } A_1 \text{ and } y \text{ is } B_1, \text{ then } f_1 = p_1x + q_1y + r_1 \quad (1)$$

$$2. \text{ If } x \text{ is } A_2 \text{ and } y \text{ is } B_2, \text{ then } f_2 = p_2x + q_2y + r_2$$

Assume that the membership functions of fuzzy sets A_i, B_i for $i = 1, 2$ are given as. μ_{A_i}, μ_{B_j} . In this work, the authors used Gaussian membership functions,

$$\mu_{A_i}(x) = \frac{1}{1 + \left(\frac{x - c_i}{a_i}\right)^{2b_i}} \quad (2)$$

A product T-norm (logical and operator) is selected to evaluate the rules' premises parts which results in,

$$w_i = \mu_{A_i}(x)\mu_{B_i}(y), \quad i = 1, 2. \quad (3)$$

Evaluating the implication and the rule consequences results in,

$$f(x, y) = \frac{w_1(x, y)f_1(x, y) + w_2(x, y)f_2(x, y)}{w_1(x, y) + w_2(x, y)} \quad (4)$$

Leaving the arguments out,

$$f = \frac{w_1f_1 + w_2f_2}{w_1 + w_2} \quad (5)$$

The above equation is rewritten as,

$$f = \bar{w}_1f_1 + \bar{w}_2f_2 \quad (6)$$

where,

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}$$

The Sugeno-type neuro-fuzzy system was implemented using MATLAB functions [29, 30].

3.3 Software Effort Estimation Datasets

In this study, the China dataset (19 attributes, 499 projects, effort measured in person-hours) was used as a testing ground [31]. One initiative was terminated as an anomaly. 398 and 100 records were used for training and testing, respectively, using k-fold cross-validation with $k = 5$. Only 12 features were used as input variables, and normalized efforts were used as the output dependent variable. Six attributes were dropped as explained below:

- Project identification number
- Efforts were dropped in favour of normalised efforts
- Development type was dropped because all projects were of the same type
- The four attributes for Product Delivery Rate (PDR) are measured in hours/function points. This work uses a normalised product delivery rate based on adjusted function points NPDR_AFP as the four attributes represent the same quantity in different ways and are consistent with the type of efforts used and the modified function points attribute.

Table 1 describes the attributes used in the study. The root mean square error (RMSE) is the criterion used to determine the accuracy of a prediction:

$$RMSE = \sqrt{\sum_{i=1}^{i=n} (actualEfforts(p_i) - predictedEfforts(p_i))^2 / n} \quad (7)$$

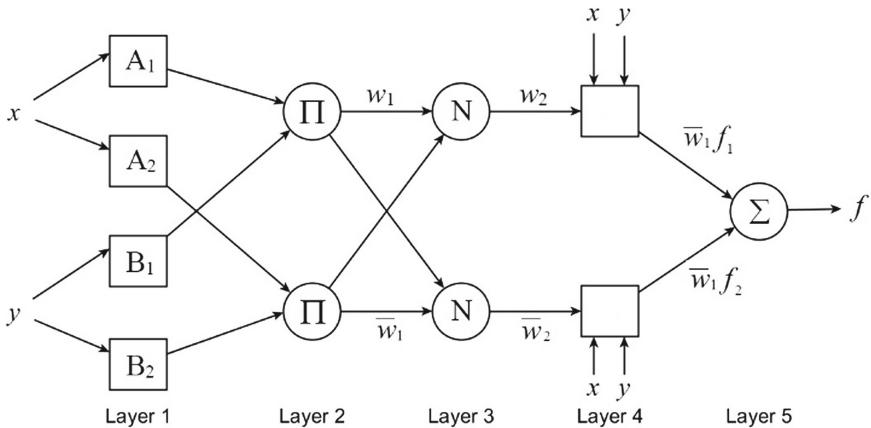


Fig. 1. System architecture for adaptive neuro-fuzzy inference.

4 Experimental Results

4.1 Clustering

Using the Within Sum of Squares (WSS) heuristic, a suitable value for the number of clusters, k , was determined. The WSS is the sum of the distance squares between each data point and its nearest centroid [27]. The clustering algorithm was applied to the dataset with k ranging from 1 to 20. Figure 2 depicts the relationship between k and WSS and indicates that an optimal value of k is five, as there is no significant reduction in WSS for a higher value of k . The number of data points designated to each cluster is provided in Table 2. Due to their small size, clusters 1, 2, and 5 merged with the adjacent clusters, resulting in three overall clusters with 259, 139, and 100 projects, as shown in Table 2. Clusters 1, 2, and 3 henceforth are called small, medium, and large project categories with average adjusted function points of 113, 332, and 1498, respectively.

4.2 Predicting Software Efforts

An ANFIS model was constructed and evaluated using the entire dataset, 498 projects, with 398 projects used to train the model using k-fold cross-validation with k set to 10. The model was evaluated on the remaining 100 projects which belong to the different identified clusters; the RMSE of the predictions was 2444.4. The next step was to build and evaluate an ANFIS model for each cluster. Using the same testing data, Fig. 3 compares actual and predicted software efforts with and without clustering. The RMSE for predictions with clustering is reduced to 1605, resulting in improved prediction accuracy of 34%. Figure 3 depicts the actual and the predicted software efforts with and without clustering; the figure indicates a marked improvement in prediction accuracy, specifically for small and medium-sized projects.

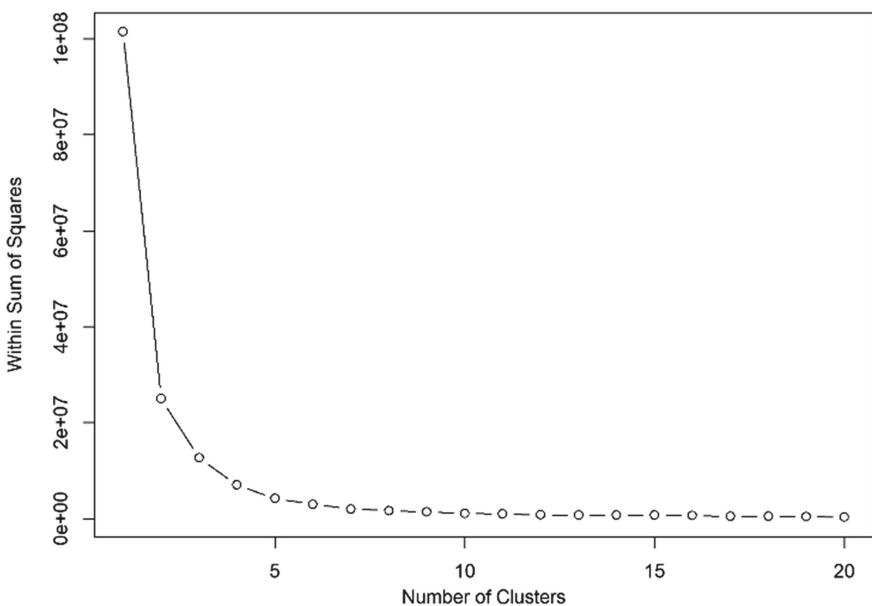


Fig. 2. Within sum of square distances v number of clusters.

4.3 Determining the Factors Affecting Software Efforts Estimation

The significance of each input variable (software metric) was determined through a sensitivity analysis. Numerous methods for neural network-based sensitivity analysis have been proposed [32]. It has been demonstrated that the partial derivative algorithm [33] and the input perturbation algorithm [34] outperform other techniques [35, 36]. Despite this, the partial derivatives method has two significant flaws. First, it cannot implement neural networks with non-differentiable activation functions; second, it is inadequate for calculating the extent of an input variable's effect on output sensitivity assessment [37].

Table 1. Attributes of the china dataset used in the study.

Attributes	Description
AFP	Adjusted function points
Input	The number of input types (I)—A count of the number of different data structure types needed to store the inputs not counting inputs used to control program execution
Output	The number of output types (O)—Outputs that lead to data structure changes (not execution control). Count each output type that has a different format or is treated differently
Enquiry	The number of inquiry types (E)—Input that controls execution flow but does not change the internal data structures such as menu selection and responses to queries
File	The number of logical internal files (L)—Internal data generated, used, and maintained by the system, such as index files
Interface	The number of interfaces (F)—Data input to another application or shared with another application
Added_FR	The number of added functional requirements
Changed_FR	Number of changed functional requirements
Deleted_FR	Number of deleted functional requirements
NPDR_AFP	Normalised product delivery rate (hour/FP)—Adjusted function points
Resource	Indicate the requirement of resources like software, hardware, and machines that complete the project
Duration	Number of days from project commencement to completion

Table 2. Number of projects in each cluster.

Cluster	Before merging					After merging		
	1	2	3	4	5	Small	Medium	Large
Number of projects	42	139	259	29	29	259	139	100
Average project size (function points)						113	332	1498
Average efforts (person-hours)						1761	3772	10994

We chose the perturbation procedure in this study for the aforementioned reasons. This procedure perturbs a specified input variable by adding noise, while leaving the remaining inputs unmodified. The change ratio of the output variable in relation to the perturbation of the input variable is calculated. This procedure is repeated for numerous noise levels.

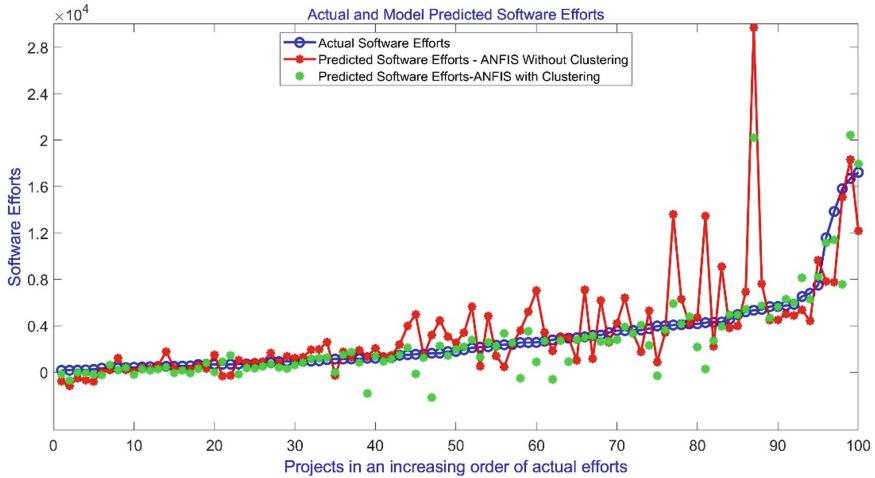


Fig. 3. Actual and predicted efforts using ANFIS without clustering.

The input variable with the greatest change ratio has the greatest explanatory influence on the system's output [38]. However, the critical issues are (i) selecting a suitable index for quantifying the output change and (ii) the range of input perturbation levels.

Runbo et al. [39] investigated several approaches to neural network sensitivity and demonstrated that the formula given by Reddy et al. [39] and described by Eq. (9) accurately measures both the direction and magnitude of a neural network output's sensitivity to a perturbation in a specific input variable value:

$$S_j = \frac{\Delta o}{\Delta u_j} \quad (8)$$

where,

$$\Delta o = \sum_{i=1}^N (\hat{y}_i - y_i) \quad (9)$$

$$\Delta u = \sum_{i=1}^N (\hat{u}_i - u_i) \quad (10)$$

S_j is a sensitivity index of output with respect to input j ,
 N is the number of input training vectors \hat{y} , and y measures the network output with and without perturbation using the training data, and
 \hat{u}_i , and u_i refer to input variable I with and without noise, respectively.

To acquire an objective evaluation of the sensitivity to perturbations in the input variables, the optimal ratio of input perturbation should be determined; if the perturbation is excessively large, the sensitivity spectra may appear truncated.

In general, the greater the deviation of a perturbation from the base case value, the less reliable is the results. Nonetheless, if the perturbation is too small, the sensitivity

spectrum may be devoid of noise and in some cases signal [39]. Runbo [39] discovered that $[-20, 20\%]$ is a reasonable range for the input perturbation range, as there are no substantial disparities in measurement sensitivity within this range. After the neuro-fuzzy model training, the sensitivity spectra values at increasing input perturbation levels ranging from 0 to 20% were calculated in steps of 0.01 according to formula (9) using training data.

4.4 Significant Factors Affecting Software Efforts Estimation

The relative sensitivity index measurements of the model's output for each input variable for the four cases: all projects, small, medium, and large projects are shown in Figs. 4, 5, 6, and 7, respectively. The sensitivity index values computed indicate that the resource metric is the most influential input predictor determining variations in software efforts for projects in general, followed by normalised product delivery rate (hour/FP) using adjusted function points (NPDR_AFP metric). This finding also applies to small-sized projects, as in Fig. 5. However, the importance of these two metrics is reversed for medium and large-sized projects, with NPDR_AFP being the most dominant, as shown by Figs. 6 and 7. Project duration comes third in importance for medium and large-sized projects; however, adjusted function points (AFP software metric) are more important than project duration for small projects. The remaining metrics have little influence on software effort variations, with the number of deleted functional requirements (Deleted_FR) having the slightest effect. A graphical summary of the power of the various software metrics on the software efforts according to project size are presented in Fig. 8.

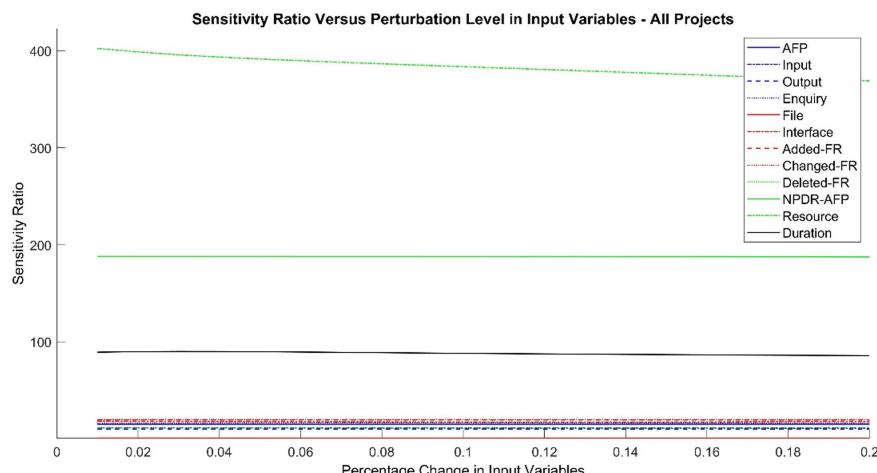


Fig. 4. Sensitivity ratios for software metrics—all projects.

5 Conclusions

In this paper, we have shown that clustering software projects into groups based on their size can significantly improve the prediction accuracy of software effort estimation models. We have used the k-means algorithm to cluster software projects in the China dataset into groups based on their size in terms of the number of function points. Building and using a Segeno adaptive neuro-fuzzy model for each cluster, we have found that the overall RMSE of software efforts estimation has improved by 34% compared to using a single estimation model for the entire set of projects.

We have also investigated the software metrics that have the most significant effect on software effort variations. The sensitivity index values computed indicate that the resource metric is the most influential in determining variations in software efforts for projects, irrespective of size; normalised product delivery rate (hour/FP) using adjusted function points (NPDR_AFP metric) comes second. These findings also apply to small-sized projects, as in Fig. 5. However, the effect is reversed for medium and large-sized projects, with NPDR_AFP being the most dominant, as shown by Figs. 6 and 7. Project duration comes third in importance for medium and large-sized project; however, adjusted function points (AFP software metric) is more important than project duration for small projects. The remaining metrics have little influence on software effort variations, with the number of deleted functional requirements (Deleted_FR) having the slightest effect. Through appropriate analysis and computation of the most influential factors on software effort estimation, these findings have the potential to substantially impact the precision of software effort estimation. Figure 8 represents a graphical summary of the influence of the various software metrics on the software efforts according to project size. These are tentative results, and further investigation is required using other larger software effort datasets of heterogenous projects in size. Future work will investigate the effects on the model's accuracy of using different attributes, such as project type or domain, for clustering a sizeable heterogeneous dataset. Another objective is to examine the generalizability of the approach to other larger software effort datasets.

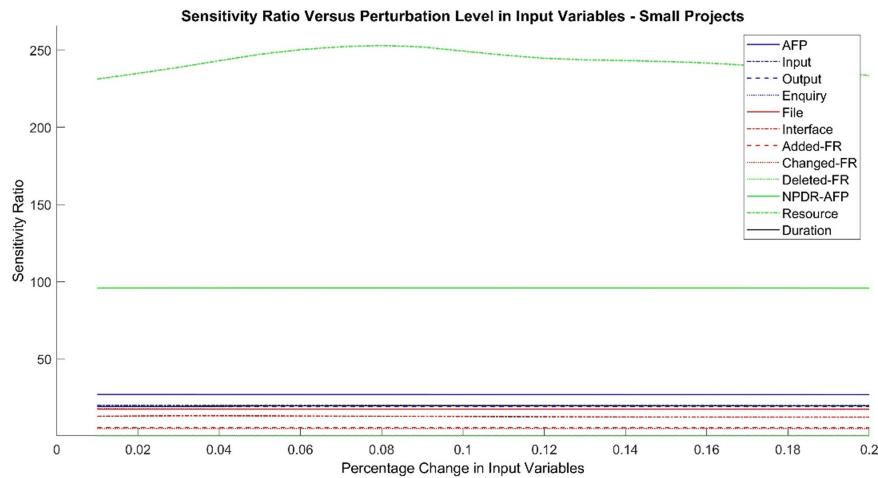


Fig. 5. Sensitivity ratios for software metrics—small projects.

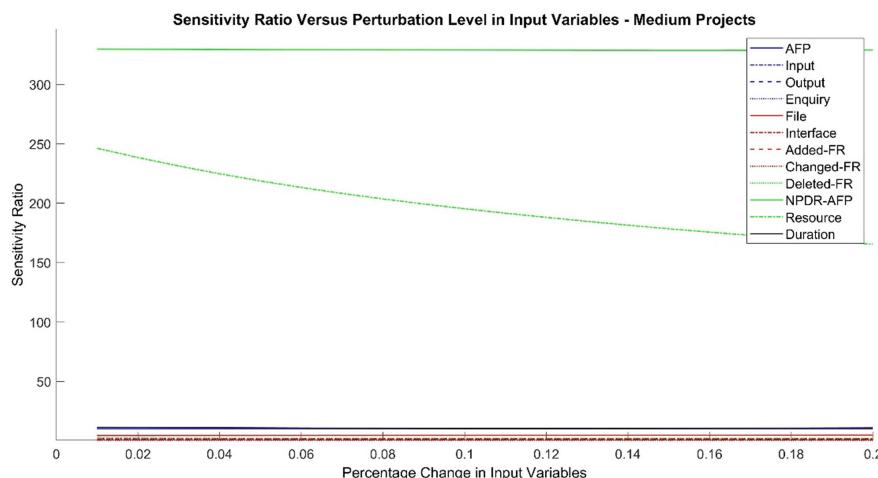


Fig. 6. Sensitivity ratios for software metrics—medium projects.

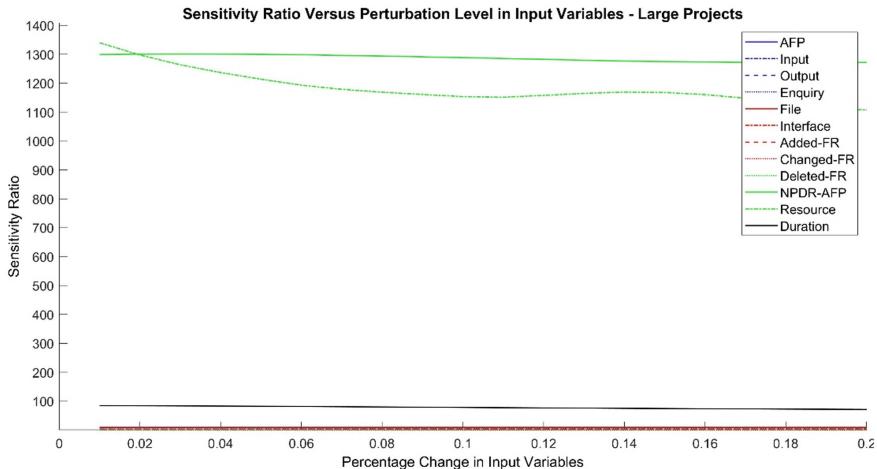


Fig. 7. Sensitivity ratios for software metrics—large projects.

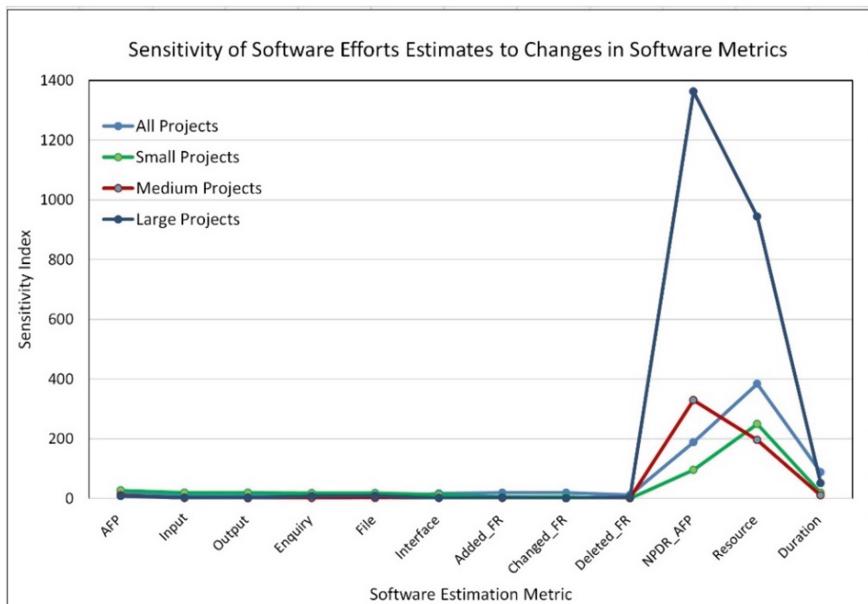


Fig. 8. Sensitivity ratios at 10% perturbation for different project sizes.

References

1. Abbas, S.A., Liao, A.R., Azam, A.: Cost estimation: a survey of well-known historic cost estimation techniques. *J. Emerg. Trends Comput. Inf. Sci.* **4**(1), 612–636 (2012)
2. Malhotra, R., Jain, A.: Software effort prediction using statistical and machine learning methods. *Int. J. Adv. Comput. Sci. Appl.* **2**(1) (2011)

3. Heat, A.: Comparison of artificial neural network and regression models for estimating software development effort. *Inf. Softw. Technol.* **44**, 911–922 (2002)
4. Sharma, R.: Survey: non-algorithmic models for estimating effort. *Eur. Int. J. Sci. Technol.* **2**(3) (2013)
5. Kaushik, A., Chauhan, A., Mittal, D., Gupta, S.: COCOMO estimates using neural networks. *Int. J. Intell. Syst. Appl.* **9**, 22–28 (2012)
6. Reddy, C.S., Raju, K.: An optimal neural network model for software effort estimation. *Int. J. Softw. Eng.* **12**(1), 66–78
7. Zadeh, L.A.: From Computing with numbers to computing with words-from, from manipulation of measurements to manipulation of perceptions. *Int. J. Appl. Math. Comput. Sci.* **12**(3), 307–324 (2002)
8. MacDonell, S.G., Gray, A.R., Calvert, J.M.: Fuzzy logic for software metric practitioners and researchers. In: The Proceedings of the 6th International Conference on Neural Information Processing ICONIP, Perth, pp. 308–313 (1999)
9. Ryder, J.: Fuzzy modeling of software effort prediction. In: Proceeding of IEEE Information Technology Conference, pp. 53–56. Syracuse, New York (1998)
10. Kosko, B.: Fuzzy systems as universal approximators. *IEEE Trans. Comput.* **43**(11), 1329–1333 (1994). <https://doi.org/10.1109/12.324566>
11. Jang, J.S.: Anfis: adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man Cybern.* **23**, 665–685 (1993); [2] Wolberg, W.H., Mangasarian, O.L.: Multisurface method of pattern separation for medical diagnosis applied to breast cytology. In: Proceedings of the National Academy of Sciences, vol. 87, pp. 9193–9196, USA (1990)
12. Buragohain, M.: Adaptive network-based fuzzy inference system (ANFIS) as a tool for system identification with special emphasis on training data minimization. Ph.D. thesis, Indian Institute of Technology Guwahati, India (2008)
13. Tsukamoto, Y., Gupta, M.M., Ragade, R.K., Yager, R.R.: An approach to fuzzy reasoning method. In: Gupta, M.M., Ragade, R.K., Yager, R.R. (eds.) *Advances in Fuzzy Set Theory and Application*, pp. 137–149. North-Holland, Amsterdam (1979)
14. Hai, V.V., Nhung, H.L.T.K., Prokopova, Z., Silhavy, R., Silhavy, P.: A new approach to calibrating functional complexity weight in software development effort estimation. *Computers* **11**(2), 15 (2022)
15. Azzeh, M., Neagu, D., Cowling, P.I.: Analogy-based software effort estimation using fuzzy numbers. *J. Syst. Softw.* **84**(2), 270–284 (2011)
16. Sharma, S., Vijayvargiya, S.: An optimised neuro-fuzzy network for software project effort estimation. *IETE J. Res.* 1–12 (2022)
17. Lopez-Martin, C., Yanez-Marquez, C., Gutierrez-Tornes, A.: Predictive accuracy comparison of fuzzy models for software development effort of small programs. *J. Syst. Softw.* **81**(6), 949–960 (2008)
18. Wei Lin, D., Ho, D., Capretz, L.F.: Improving software effort estimation using neuro-fuzzy model with SEER-SEM. *Glob. J. Comp. Sci. Technol.* **10**(12), 51–63 (2010)
19. Hodgkinson, A.C., Garratt, P.W.: A neuro-fuzzy cost estimator. In: Proceedings of the Third Conference on Software Engineering and Applications, pp. 401–406 (1999)
20. Huang, X., Ho, D., Ren, J., Capretz, L.F.: A soft computing framework for software effort estimation. *Soft Comput.* 170–177 (2005)
21. Huang, X., Ho, D., Ren, J., Capretz, L.F.: Improving the COCOMO model using a neuro-fuzzy approach. *Appl. Soft Comput.* 29–40 (2006)
22. Xia, W., Capretz, L.F., Ho, D., Ahmed, F.: A new calibration for function point complexity weights. *Int. Softw. Technol.* **50**(7–8), 670–683 (2008)
23. Wong, J., Ho, D., Capretz, L.F.: Calibrating functional point backfiring conversion ratios using neuro-fuzzy technique. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **16**(6), 847–862 (2008)

24. Karunakaran, Sreenath.: Survey on software effort estimation technique. *A Rev. Int. J. Sci. Eng. Res.* **6**(12) (2015)
25. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14, pp. 281–297 (1967)
26. Kaufman, L., Rousseeuw, P.J.: *Finding Groups in Data: an Introduction to Cluster Analysis*, vol. 344. Wiley (2009)
27. EMC Education Services.: *Data Science & Big Data Analytics: discovering, Analyzing, Visualizing, and Presenting Data*, p. 120. Wiley, Indianapolis, USA (2015)
28. Negnevitsky, M.: *Artificial Intelligence: a Guide to Intelligent Systems*. Addison Wesley, Harlow, UK (2017)
29. MathWorks.: Fuzzy Inference Systems Modeling (2021). <https://ch.mathworks.com/help/fuzzy/genfis.html>
30. Shirabad, S.J., Menzies, T.J.: The PROMISE repository of software engineering databases. In: *School of Information Technology and Engineering*. University of Ottawa, Canada (2005). <http://promise.site.uottawa.ca/SERepository>
31. Cao, M., Alkayem, N.F., Pan, L., Novák, D.: Advanced methods in neural networks-based sensitivity analysis with their applications in civil engineering. In: *Artificial neural networks—Models and applications*, pp. 335–353. IntechOpen (2016). <https://doi.org/10.5772/64026>
32. Dimopoulos, Y., Bourret, P., Lek, S.: Use of some sensitivity criteria for choosing networks with good generalisation ability. *Neural Process. Lett.* **2**(6), 1–4 (1995). <https://doi.org/10.1007/BF02309007>
33. Zeng, X., Yeung, D.S.: A quantified sensitivity measure for multilayer perceptron to input perturbation. *Neural Comput.* **15**(1), 183–212 (2003)
34. Gedeon, T.D.: Data mining of inputs: analysing magnitude and functional measures. *Int. J. Neural Syst.* **8**(2), 209–218 (1997). <https://doi.org/10.1142/S012906579700227>
35. Wang, W., Jones, P., Partridge, D.: Assessing the impact of input features in a feedforward neural network. *Neural Comput. Appl.* **9**(2), 101–112 (2000). 10.1007/PL00009895
36. Cheng, A.Y., Yeung, D.S.: Sensitivity analysis of neocognitron. *IEEE Trans. Syst. Man Cybern.-Part C: Appl. Rev.* **29**(2), 238–249 (1999)
37. Lamy, D.: Modelling and sensitivity analysis of neural network. *Math. Comput. Simul.* **40**, 535–548 (1996)
38. Runbo, B., Zhang, D., Jia, H.: Factor sensitivity analysis with neural network simulation based on perturbation system. *J. Comput.* **6**(7) (2011)
39. Reddy, N.S., Lee, C.S., Kim, J.H., Semiatin, S.L.: Determination of the beta-approach curve and beta-transus temperature for titanium alloys using sensitivity analysis of a trained neural network. *Mater. Sci. Eng. A* **434**(1–2), 218–226 (2016)



Automated Spatiotemporal Modeling for Real-Time Data-Driven Actionable Insights

Hugo Latapie¹(✉), Mina Gabriel¹, Sidarth Srinivasan¹, Ramana Kompella¹,
Kristinn R. Thórisson^{2,3}, and Pei Wang⁴

¹ Cisco Research, Cisco Systems, San Jose, CA, USA
hlatapie@cisco.com

² Center for Analysis & Design of Intelligent Agents, Department of Computer
Science, Reykjavik University, Reykjavik, Iceland

³ Icelandic Institute for Intelligent Machines, 102 Reykjavik, Iceland

⁴ Department of Computer and Information Sciences, Temple University,
Philadelphia, PA 19122, USA

Abstract. Significant increases in industry requirements for network bandwidth are seen year after year. The exponential growth in streaming data is matched by an increase in the use of machine learning and deep learning to glean actionable insights from these data—ideally in real-time. Demand for artificial intelligence (AI) solutions to a variety of computational needs are likely to increase significantly over the coming years and decades. Meanwhile, the capacity of AI and data scientists to meet current requirements with contemporary approaches, which require continual updating and retraining, is falling short of industry demands for automation on dimensions of critical importance, including training speed, accuracy, trustworthiness, and explainability. In this paper we introduce a hybrid AI approach to computational intelligence which features new self-supervised learning mechanisms, a knowledge model engineered to include support for machine generated ontologies, as well as traditional human-generated ontologies, and interfaces to AGI systems such as OpenNARS, AERA, ONA, and OpenCog. Our hybrid AI system is capable of self-supervised learning of machine-generated ontologies from millions of time series, to provide real-time data-driven insights for large-scale deployments including data centers, enterprise networks, and video analytics. Preliminary results across all the use cases we have attempted to date are promising, but more work is needed to fully map out both the approach’s benefits and limitations. This Hybrid AI project, and associated data, are expected to be available as open source in April 2023.

Keywords: Artificial intelligence · AI · Hybrid AI · Video analytics

1 Introduction

Hybrid AI refers to the integration of multiple artificial intelligence (AI) technologies and approaches, such as machine learning, natural language processing, computer vision, and symbolic reasoning systems, to create a more sophisticated and comprehensive AI system.

In a hybrid AI system, different AI technologies are combined in a way that allows them to complement each other's strengths and compensate for each other's weaknesses. For example, machine learning algorithms can be used to analyze large amounts of data and identify patterns, while symbolic reasoning systems can provide human-like reasoning and decision-making capabilities.

The goal of hybrid AI is to create AI systems that are more capable, flexible, and adaptable to a wide range of tasks and situations, and to overcome the limitations of individual AI technologies.

High-resolution cameras and other high bandwidth sensors are proliferating, as are other types of real-time data such as network telemetry and business related time series. Gartner estimates the value of providing real-time data-driven insights at over trillion dollars in the coming years.¹ However, approaches based on artificial neural networks (ANNs) may be insufficient in terms of functionality, flexibility, accuracy, explainability, and robustness.² This is driving research and development of artificial intelligence (AI) solutions that combines methods and techniques from various sub-fields of artificial intelligence such as machine learning, deep learning, reasoning, knowledge graphs, and various ideas from research on general machine intelligence—referred here to as ‘hybrid AI.’ Driven by real-world needs, an industry-academia collaboration is making numerous pragmatic advances in hybrid AI solutions. As a general category of systems, a contemporary hybrid AI system may include several different technologies such as reasoners, matrix profile time series analysis, traditional machine learning (ML), and deep learning (DL), and more. It may also include any number of more experimental technologies from research on general machine intelligence, such as cumulative learning, causal modeling, and empirical reasoning. In our own approach to hybrid systems design, we emphasize the systems’ ability to learn from experience and reason over what has been learned, leveraging the benefits of symbolic systems to address the various limitations of deep learning and statistical approaches in general, such as lack of autonomous learning and explainability.

Falling into the admittedly broad category of hybrid AI systems, this paper describes the specific elements of our recent research on hybrid AI systems development. We also discuss preliminary results for smart city, retail, and networking use cases. It should be noted that hybrid AI systems are in the early development phase and remain complex to develop and test. Because the objective here is to solve problems requiring capabilities such as cumulative learning and

¹ [Gartner Top Trends in Data and Analytics](#)—accessed Nov. 18th, 2022.

² [Even after \\$100 Billion Self Driving Cars Are Going Nowhere](#)—accessed Nov. 18th, 2022.

machine generated ontologies, which remain generally unsolved, there are few if any suitable datasets and generally accepted leader boards that can be used for comparison. Our plan is to contribute more reproducible research, open source, open datasets, open problems and challenges to help progress the field. For these reasons, those seeking incrementally better solutions are advised to not pursue hybrid AI solutions. However, we believe hybrid AI is an important field that can help address many currently unsolved real-world use cases. The use cases discussed and referenced in this paper are not an exhaustive list of the possibilities.

2 Related Works

In this section, we review the literature on hybrid AI systems that combine machine learning (ML), deep learning (DL), reasoning, and symbolic knowledge representation. Our aim is to provide an overview of the most relevant works that have contributed to the development of the novel hybrid AI system we propose in this paper. We organize the related works into four main categories: (1) early hybrid AI systems, (2) neural-symbolic integration, (3) graph neural networks, and (4) transfer learning and few-shot learning.

2.1 Early Hybrid AI Systems

Early efforts to create hybrid AI systems aimed at combining symbolic reasoning with machine learning techniques. SHRDLU, developed by Terry Winograd in the 1970s, was one of the first AI systems to integrate natural language processing with a knowledge representation system and a reasoning mechanism [29]. Similarly, CYC, developed by Douglas Lenat and his team at MCC, aimed to create a comprehensive ontology and knowledge base that could support common-sense reasoning [14]. These early systems demonstrated the potential of hybrid approaches to AI but were limited by their handcrafted knowledge bases and rule-based systems.

2.2 Neural-Symbolic Integration

Neural-symbolic integration focuses on combining the strengths of both neural networks and symbolic reasoning in a unified framework. Key works in this area include the Knowledge-Based Artificial Neural Networks (KBANN) proposed by Towell and Shavlik [22], which demonstrated how existing symbolic knowledge could be used to initialize and constrain the weights of a neural network. Later, Garcez et al. [4] introduced the Connectionist-Symbolic Integration (CSI) approach, which combined propositional logic with recurrent neural networks, allowing the system to perform reasoning and learning tasks simultaneously.

Another significant contribution to neural-symbolic integration is the Differentiable Inductive Logic Programming (DILP) framework proposed by Evans and Grefenstette [2]. DILP uses a differentiable first-order logic layer to bridge the gap between neural networks and symbolic representations, enabling the system to learn and reason over complex relational structures.

2.3 Graph Neural Networks

Graph Neural Networks (GNNs) have recently emerged as a powerful framework to model structured data, offering a natural way to represent knowledge and reason over it. GNNs have been successfully applied to various domains, including knowledge graph completion [17], molecular property prediction [5], and social network analysis [10]. The success of GNNs in these applications has shown their potential for combining learning and reasoning in a single framework.

In particular, the Relational inductive biases, deep learning, and graph networks paper by Battaglia et al. [1] provides a comprehensive overview of the graph network framework and its potential for integrating learning, reasoning, and symbolic representation.

2.4 Transfer Learning and Few-Shot Learning

Transfer learning and few-shot learning techniques have gained significant attention in recent years for their ability to adapt knowledge learned from one task or domain to new tasks or domains with limited training data. Key works in this area include the Model-Agnostic Meta-Learning (MAML) algorithm proposed by Finn et al. [3], which learns a model initialization that can be fine-tuned quickly on new tasks with limited data. Another notable approach is the Prototypical Networks by Snell et al. [18], which learns a metric space for classification tasks in which new classes can be incorporated with few examples.

These techniques have shown promise for integrating learning and reasoning in AI systems, as they enable the efficient adaptation of knowledge across tasks and domains

3 Methodology

Our hybrid AI system’s technical architecture was designed and assessed using in-house test environments. This architecture was developed through a collaborative effort between Cisco, Temple University, Reykjavik University, and the OpenCog teams. In this section, we present the various components of the hybrid AI system.

While the evaluation strategies for each development can be found in the papers discussing the specific use cases, it is essential to note that the system’s objectives included:

- automatic learning of spatiotemporal semantics in smart cities
- reasoning-based detection and alerting of potential safety-related events
- automatic learning of retail inventory semantics
- automatic learning of large-scale (≥ 300 K time series) multivariate time series for descriptive and predictive event analysis
- integration of neural and symbolic approaches.

During the system’s development, there were no widely accepted ground truth datasets and leaderboards for these objectives. However, the team is currently working on creating suitable open datasets and fostering reproducible research through the Cisco Deep Vision open-source project.

3.1 Self-supervised Learning from Spatiotemporal Information

As the term implies, self-supervised learning is a type of learning in which the learning system acquires knowledge autonomously; as the data that animals learn from is neither pre-selected nor controlled by a teacher, input data must be chosen by the system dynamically and the learning progress controlled continually by the learning system itself. Contemporary machine learning (ML) has significantly less autonomy in this respect: A model is trained using pre-selected unlabeled input data and a set of pre-defined rules or objectives. One way in which such self-supervised learning can progress is through the use of spatiotemporal information, e.g. video and sensor data. In this case, the training objective is to predict some properties of the data based on its spatial and temporal relationships.

A specific application of self-supervised learning using spatiotemporal information is semantic segmentation of time series data. In such work, the goal is to partition the data into semantically meaningful segments, such as the different objects or actions in a video sequence. This can be done by training a model to predict the boundaries of these segments, given the surrounding data as input.

Our approach for processing raw input spatiotemporal data often termed ‘sub-symbolic,’ is based on matrix profile time series semantic segmentation [11]. This approach has been found to be efficient in processing large amounts of time series data (e.g. millions of time series in real-time). The inductive bias for ML/DL can typically be described as pattern recognition, and we build on this inductive bias in several ways. At the core of our approach is the hypothesis that any large scale system being observed consists of events that may be interrelated in complex ways, producing a time series with all manner of observed and unobserved correlational relations, resulting from unknown (and unobserved) causal relations. Therefore, when a number of time series’ from multiple different sources all experience a simultaneous shift from one structural mode to another, known as a regime change, it may mean there is a significant event of interest occurring. The system continues to observe the nature of this correlation over time to quantify the degree of correlation for these hypothesized events of interest. In our experiments, we have found that accurate descriptive and predictive models for networking use cases can be learned in this manner.

Once the data has been segmented, the next step is to align the semantic segments across different time steps. This can be done by training a model to predict the transformation that aligns each segment with its corresponding segment in the next time step. This alignment step can help the model learn to track and reason about the movements of objects and actions over time.

In addition to the semantic segmentation and alignment of time series data, self-supervised learning can also be used to learn the rate of change of these semantic segments. This can be done by training a model to predict the speed and direction of motion of the segments, given their previous positions and movements as input. This can help the model understand the dynamics of the scene, allowing it to make more accurate predictions and decisions.

In the case where there are small numbers of ephemeral time series, such as in the case of privacy preserving behavioral video analytics for moving objects, we found that the addition of another inductive bias based on the rate of change of the regime changes is a useful metric for events of interest. This inductive bias for animate and inanimate objects is based on the hypothesis that the dynamics of these systems result in structural cohesion in the temporal domain which is normally broken as the system moves from one state to another. This mode changes from, for example, walking to sitting to standing doesn't normally occur at a high rate. Our initial experiments for human behavior indicate this inductive bias is a good way to find behaviors of interest such as aggression, medical incidents, and so forth, in a self-supervised manner.

One important aspect of the rate of change of semantic segments is its ability to identify events of interest. For example, human behavior typically has a normal maximum rate of change from standing to sitting to walking to running. When the rate of change of semantic segments increases dramatically over baseline levels, this can be a reliable indicator of interesting events across a wide range of time series data, including sensor data, network telemetry, and video analytics. By using self-supervised learning to learn the rate of change of semantic segments, a model can automatically identify and respond to these events in real time.

Overall, self-supervised learning from spatiotemporal information is a powerful and versatile approach for training machine learning models. By leveraging the inherent structure and relationships within the data, it can enable models to learn useful features and patterns without the need for explicit labels or supervision. This can lead to more efficient, scalable, and robust machine learning systems.

3.2 Machine-Generated Ontologies

The use of machine-generated ontologies is a key enabler of lifelong learning in artificial intelligence. By providing a structured representation of knowledge that can be easily integrated into machine learning models, ontologies can help to overcome the catastrophic forgetting problem and enable AI systems to adapt and learn over time.

The self-supervised learning mechanism above allows us to categorize and classify input data in terms of spatiotemporal semantic segments. By grouping all the observed semantic segments across time series on the basis of similarity metrics, the system is able to categorize and rank the input data into the most and least commonly seen semantic segments. These categories are organized into hierarchical structures by identifying representative actual or synthetic samples. These machine generated ontologies may also be associated with natural language labels via active learning or other more automated techniques. These machine generated ontologies based on input data processed by the system form the foundations of our hybrid AI approach to symbol grounding. In essence, this type of hybrid AI literally “makes sense” of symbols by building

them on representations obtained from prior sensory information. A key challenge here, and generally when an AI system builds knowledge from data, is that the knowledge will not be perfect right away. Inconsistencies can happen, and some task-relevant knowledge may need to be kept up to date or needs yet to be acquired. Hereby, also representation shifts can occur when the methods utilized for learning and detecting relevant features are updated with new data from observation. This demands a stance on reasoning which is unlike historical approaches built on First Order Predicate Logic variants which have turned out to be too brittle. If the benefits of logical reasoning are to be exploited in realistic contexts with clear dependence on ML and DL methods for processing of high-dimensional input data, experience-grounded truth and meaning become a necessity [25]. This way representation learning and updating, building and maintaining hypotheses, and working with them in a logical sense can all happen simultaneously. This is different from the simplistic and less practical solution of just wiring together a DL model with a deductive reasoning system, as it demands this special kind of reasoning to be able to cope with the learning requirements of the overall system. Our solutions based on combining Non-Axiomatic Reasoning System (NARS) [26] with ML/DL techniques have turned out to be particularly successful in this direction, and exploiting these new technological capabilities in various applications forms a key part of our efforts. The following section is dedicated to capturing the learning part from the reasoner's side which includes relevant details of Non-Axiomatic Reasoning.

Before we go there, we also address the problem of taking human background knowledge into account via some kind of seed ontology. Here, there are essentially two major approaches:

- to treat seed ontology as axiomatic in the sense that knowledge items cannot be revised, yet the knowledge base can be extended through learning
- to treat the seed ontology as initial knowledge, which is open for revision as any other acquired knowledge.

NARS allows a good tradeoff between the two, by allowing elements in the seed ontology to have different initial confidence values. Hereby, low confidence items will essentially be quickly overridden by learned evidence, while high confidence items are more stable when facing new evidence. Furthermore, representation-wise, the Narsese formal language the system uses for I/O purposes, makes encoding of human-provided domain knowledge easy. This language is also used as the internal language of the system, which makes learned representation interpretable due to the usage of logical connectors and copulas. The formation of more complex representations (compound term formation, especially in perception) can make interpretation also more difficult. However, due to the logical structure of the compounds, these do not need the kind of post hoc analysis which is required to make black box ANN models interpretable.

Overall, the use of machine-generated ontologies is a powerful and versatile approach for enabling lifelong learning in AI systems. By using techniques such as clustering and self-supervised learning, AI systems can learn to extract useful features and patterns from data and encode this knowledge in an ontology,

allowing them to adapt and continue learning over time. This can lead to more efficient, scalable, and robust AI systems.

3.3 Integrated Learning and Reasoning

While learning and reasoning have been widely recognized as major functionalities of intelligence and cognition, historically they have been studied separately, without much effort in combining them [28]. In our Hybrid AI approach, learning and reasoning are integrated into several ways that are complementary to one another.

NARS is a general-purpose model of intelligence that is designed to work with insufficient knowledge and resources and to adapt to its environment [26, 27]. From this perspective, the conceptions of learning and reasoning are fundamentally different from the conventional ones. Though NARS is designed in the framework of a reasoning system with formal grammar and inference rules, “reasoning” is not taken to mean theorem proving. Instead, NARS is non-axiomatic, meaning that the knowledge or beliefs of the system are not representations of facts or state of affairs, but summaries of the system’s own experience, that is, records of the system’s interaction with the environment.

Concretely speaking, in NARS each *concept* is identified by a *term* that can be as simple as a string, while the content of a concept corresponds to a segment of the system’s experience that has been abstracted to a certain level, and the experience can be sensorimotor, linguistic, or their combinations. As the meaning, or content, of a concept is grounded in the system’s experience, it is not a “symbol” whose meaning depends on an interpretation that maps it to an external object or event [25]. Also based on this experience-grounded semantics, each belief of the system typically represents a relation between two concepts, especially the degree to their corresponding experience segments are substitutable by each other, or how much one can be *seeing as* the other [9]. Since each concept, as an abstraction, usually corresponds to different (concrete) experience segments, its substitutability by another concept is usually a matter of degree, as it has both positive and negative evidence. Consequently, the truth-value of a belief is not simply true or false, but indicated by a pair of numbers, indicating the evident support the belief gets from experience [25].

Since truth-value of a belief indicates its relation with available evidence, various types of inference (deduction, induction, abduction, revision, etc.) are justified uniformly as different ways for new conceptual substitutability to be derived from the available ones. This is fundamentally different from traditional logic systems, where the truth-value of a proposition indicates whether it corresponds to a fact under a given interpretation. In that situation, non-deductive inference, such as induction and abduction, cannot be justified, as they may derive false conclusions from true premises. This is not the case in NARS, as the truth-value of a belief merely indicates its degree of agreement with past experience, but that with future experience, so inductive and abductive conclusions are still justifiable, and if they turn out to be “wrong” according to new observations, their truth-values are adjusted by taking the new evidence

into consideration. Since induction and abduction typically carry out the cognitive function of *generalization* and *explanation*, respectively [16], these types of *reasoning* can also be considered as *learning*.

The above unification of *reasoning* and *learning* also appears in other forms, such as compound-term compositions that lead to the creation of new concepts. Overall, in NARS reasoning and learning are two aspects of the same underlying process. When the process is studied as reasoning, the focus is on the logical relation between the premises and the conclusions of every single step; When the process is studied as learning, the focus is on the accumulated effects of the process on the system's beliefs and other forms of knowledge, such as skills and desires.

Though NARS has provided general-purpose reasoning and learning functions, it can still benefit from the performance and efficiency provided by the special-purpose techniques, especially on sensorimotor mechanisms for which NARS depends on plug-in devices. This is based on the belief that intelligence should be able to handle different types of sensors and actuators by learning the sensorimotor contingency associated with the operations realized by the sensor/actuator. To realize this design, NARS has an interface that can connect various types of devices with their commands registered in NARS as operations, which are statements under a procedural interpretation [27].

Previously, we have built such an application in the SmartCity domain [7]. In that project, an open-source implementation of NARS, *OpenNARS*, is connected to machine learning tools with object recognition and tracking functions. In that project, we demonstrate how even crude interfacing of an ML/DL-based object tracking system, with OpenNARS was able to significantly improve the generalizability of video analytics systems. By starting with a small amount of a priori seed knowledge which includes some basic spatiotemporal assumptions about the world and what mattered amounting to around 1 page of rules, the system is able to “understand” a new camera view of an environment within five minutes, assuming an average amount of activity and begin reporting events of interest such as potential accidents between any two moving objects, actual accidents, jaywalking, and more.

The same system was given a slightly different seed knowledge focused on retail inventory use cases and was able to automatically learn shelf locations, product candidate placements, etc. [20].

3.4 Ontology-Based Problem Decomposition with Attention

A combinatorial explosion occurs when the complexity of a problem grows exponentially, this growth rate is a well-known problem in symbolic space, it occurs when the number of rules and facts in a system increases, leading to an exponential growth in the amount of computation required to reason about the knowledge. This can greatly limit the practical application of reasoners to real-world problems since most real-world problems involve thousands or even millions of rules and facts.

One potential solution to the combinatorial explosion problem is the use of hierarchical ontologies. In a hierarchical ontology, the knowledge is organized into a tree-like structure, with high-level concepts at the top and more specific concepts at the lower levels. This allows the knowledge to be decomposed into smaller, more manageable pieces, which can be reasoned about separately.

By using hierarchical ontologies, the combinatorial explosion problem can be avoided or greatly reduced. This is because the hierarchical structure of the ontology allows the reasoning process to be divided into smaller, independent sub-problems, each of which can be solved separately. This can significantly reduce the amount of computation required, keeping the growth rate of resources in the linear region.

In addition to solving the combinatorial explosion problem, the use of hierarchical ontologies has several other advantages. For example, it can improve the efficiency and scalability of reasoning systems, since the knowledge can be organized and accessed in a more structured and efficient manner. It can also improve the accuracy and reliability of the reasoning process since the hierarchical structure of the ontology can help to prevent inconsistencies and errors.

Overall, the use of hierarchical ontologies is a key solution to the combinatorial explosion problem in the symbolic space. By decomposing the knowledge into smaller, more manageable pieces, hierarchical ontologies can enable reasoners to handle larger and more complex problems, leading to more powerful and capable AI systems.

Retail use case used this to avoid combinatorial explosion issues in the reasoning space and to improve static confidence metrics. We used a spatial semantics model focused on containment and relative positioning of shelves and objects [13].

3.5 Machine Ontology to Human Ontology Interoperation

The current focus of our research is on interoperability between machine and human generated ontologies. While manual methods, such as active learning, are significantly more efficient than supervised learning-requiring only one active learning-acquired label to replace potentially thousands or tens of thousands of traditional ground truth labels-our goal is to leverage large language models and other autonomous approaches to minimize the need for active learning time from human domain experts.

When using machine-generated ontologies, one challenge is to align and match these ontologies with traditional, human-generated ontologies. This is important because human-generated ontologies are often the basis for many applications, such as information retrieval and natural language processing. By aligning and matching machine-generated ontologies with human-generated ontologies, it is possible to leverage the strengths of both to improve the performance and reliability of AI systems.

There are several mechanisms that can be used to align and match machine-generated ontologies with human-generated ontologies. One approach is to use generative models, which can learn to generate ontologies that are similar to

human-generated ontologies. For example, a generative model might be trained on a large corpus of human-generated ontologies, learning to predict the structure and content of these ontologies. This can help the model generate ontologies that are similar to human-generated ontologies, allowing them to be easily aligned and matched.

Another approach is to use similarity metrics, such as Jaccard similarity or cosine similarity, to measure the similarity between machine-generated and human-generated ontologies. These metrics can be used to compare the structure and content of the ontologies, allowing the most similar ontologies to be identified and matched. This can help to ensure that the aligned ontologies are consistent and compatible, improving the performance and reliability of AI systems.

Overall, aligning and matching machine-generated ontologies with human-generated ontologies is an important challenge in the field of artificial intelligence. By using mechanisms such as generative models and similarity metrics, it is possible to leverage the strengths of both types of ontologies, leading to more powerful and capable AI systems.

4 Results

This work focuses on the incremental development of real world, customer-facing robust proof of concepts built on industry standard open source building blocks: docker containers, Redis, Redis time-series, GStreamer, OpenCV, Grafana dashboards, DL frameworks and libraries, matrix profile time series analysis, Neo4J graph database, and Kubernetes.

4.1 Smart City

The results for smart city analytics of complex road intersections that include bus lanes, commuter train tracks, and crosswalks show that with a small amount of seed knowledge, the system can, in constructivist AI fashion, rapidly adapt and learn to provide a wide range of safety related analytics [6, 19]. In addition to generic Q&A related functionality, this approach can detect various traffic anomalies which can appear at runtime via reasoning and does not need to be trained with question-answer pairs or prior training with anomaly cases. This is especially useful when the solution is deployed on a wider scale, whereby each place where it is utilized has special characteristics, not all of which could have been captured by offline training before deployment. As a concrete example, let's consider the case of Jaywalking. Using semantic segmentation it is quite reliable nowadays to train a system to distinguish an area dedicated to pedestrians from an area of street utilized by vehicles. However, in real life, construction work, etc. can lead to certain areas being temporally re-purposed, e.g. a segment of streets not being accessible to cars anymore, or a pedestrian area temporally being used for vehicle traffic. The real-time learning approach as in [6] allows us to notice the change in the use of an area and is hence able to make more useful reports of jaywalking incidents. This however is not limited to this situation: successes of

offline-trained solutions in real-world applications are generally limited compared to systems that can adapt at runtime. Making the latter more reliable opens up the possibility to use AI in real-world environments in which unexpected changes are to be expected.

Runtime adaptation in [6] was especially achieved via Inductive Reasoning as described in [26, 27], which is in line with the considerations regarding Machine-generated ontology. To also take human expertise into account in an effective way, a seed ontology was utilized which was adjusted and extended by the AI system. This seed ontology described anomaly cases of interest to the human operator, while the grounding of these cases was achieved through real-time learning. For instance, for a jaywalking scenario to be detected, locations which are identified as streets which are not commonly utilized by pedestrians need to be considered. However, whether a certain location qualifies as such, cannot be utterly pre-defined by a designer but needs to be revisable on a case-by-case basis at runtime. While it is possible to take the output from offline-trained models (such as a semantic segmentation model) into account, the output information cannot be taken as the only information source, as usage of the location (which can be specific to a specific road segment and can depend on time), etc. need to be considered in addition. By obtaining evidence about the nature of specific locations of interest (such as a specific pedestrian location) in a cumulative way, more reliable conclusions relevant to the detection of related anomalies are increasingly reached by the system, while the initial performance is limited to the performance of the offline-trained models and the reliability of human-provided background knowledge.

4.2 Retail

Using a different spatial semantics seed of knowledge, our hybrid AI system for retail analytics was able to configure itself to all manner of retail scenes and provide inventory analytics [13].

4.3 Networking

In the networking domain, our hybrid AI system was able to process over 300 K time series in real time on a single 8-core CPU and produce descriptive and predictive models in a self-supervised manner. These models were tested with 30 previously unseen failure modalities and exhibited state-of-the-art accuracy [12].

4.4 Privacy Preserving Behavioral Analytics

Finally, our results in computer vision based human behavior analytics demonstrate our self-supervised learning is highly sensitive to subtle changes in individual human behavior indicative of potential interest. We demonstrate the ability to detect flash mob formation in its early stages while being able to detect all manner of incidents by the subtle changes in the behavior of the people around the incident, even if the incident itself is completely off camera.

5 Capturing User Intent

The Hybrid AI system we are presenting here focuses on goal-oriented AI, in which understanding the user’s intent is key. To this end, we have explored the use of large language models, specialized in capturing intent related to desired insights from video data. OpenNARS and AERA, two elements used by the Hybrid AI system, have independent implications for unconstrained, open-ended AI applications, but require some technical expertise to use due to their highly expressive and precise domain-specific languages.

Recent advances in neural networks (NNs) have enabled automatic querying of large volumes of video data with high accuracy. While these deep NNs can produce accurate annotations of an object’s position and type in video, they are computationally expensive and require complex, imperative deployment code to answer queries. With no-code approaches becoming the preferred norm among both technical and non-technical users, we propose a model that leverages Natural Language to perform querying on video analytics. In short, we propose a pipeline that does the following; (a) Generate an image/video output based on the user’s description of the video analytics they are interested in natural language (b) Generate a SQL query of the user’s description of the video analytics querying they are interested in natural language (c) Identify behaviors in the SSL model that closely matches the output from (a) and (b) and output those behaviors.

5.1 Natural Language based Image/Video Generation

- The system takes the user’s description of the video analytics they are interested in, using natural language.
- Using the user’s input prompt, text to image based DDPM models generate images/videos (see Fig. 1).
- The user can then refine the generated output using a combination of natural language and advanced image/video editing and mask features.

Features

- Users can input their description through Natural Language and provide the system with a reference image to complement the text prompt.
- During the feedback process, the user can create masks on specific portions of the generated images that need to be reworked on. They can provide instructions via text that gets applied on the masked region alone.
- Supports Inpainting/Out-painting and weightage that the system assigns to the text prompt, more weightages would result in images strictly adhering to the text prompt.



Fig. 1. User personas for stable diffusion based image generation.

5.2 Natural Language Based Query Generation

- The user inputs their query of the analytics they are interested in natural language.
- The system then generates equivalent SQL code based on the input prompt.

Features

- Users can input their predefined template about the table such as their fields and provide a text prompt to generate a SQL query based on the input table templates. If no such templates are given, the model assumes fields based on the input prompt and generates queries accordingly.

Use cases

The following examples were generated using the OpenAI Code Completion feature.³

³ [OpenAI Codex](#).

```
SELECT * FROM vehicles
WHERE speed > 60 AND registration_status = 'expired';
```

Listing 1. The previous SQL statement finds all cars with expired registrations and speeds over 60 mph

```
#Table cars, columns = [LicenseNo, Carspeed,
#ownerfirstName, OwnerLastName, CarMake, CarColor, RegCode]
#Create a MySQL query to display the LicenseNo and RegCode
#for all cars going over 80 mph

query = ``SELECT LicenseNo, RegCode FROM cars WHERE Carspeed > 80''

#Table cars, columns = [LicenseNo, Carspeed, OwnerFirstName,
#OwnerLastName, CarMake, CarColor, RegCode]
#Create a MySQL query to display the LicenseNo
#and RegCode for all cars going over 80 mph and are red

query = ``SELECT LicenseNo, RegCode
FROM cars
WHERE Carspeed >80 AND CarColor ='Red' '',
```

Listing 2. The previous SQL statements find car information for all cars going over 80 mph using *where* clause

```
#Count the number of people in a given image

def create_query(frame_id):
    query = ``SELECT COUNT(*) FROM people WHERE frame_id = ''
+ str(frame_id)
    return query
```

Listing 3. Using SQL aggregation functions to count the number of people in a given frame

5.3 Link to Self-supervised Models

From video frames of a given environment, let's consider a public place, for instance, we would have different classes of behaviors such as people walking, running, skateboarding, and falling. With the environment being diverse in terms of its behaviors (different classes of action), a supervised learning approach would be cumbersome due to the lack of labeled data. Instead, we rely on a self-supervised learning approach where-in the model would identify the set of behaviors in each video frame but lacks the natural language label to denote the behavior.

Our model functions as an AI tool for data scientists and analysts by capturing their intent in natural language and generating images/videos along with the SQL queries of the analytics they are interested in. We then apply the generated image/video to our SSL model and identify if the generated behavior fits any of the existing behaviors captured by the model.

5.4 Generating Ground Truth

Often while training supervised ML models, we encounter the problem of limited ground truth labels and even techniques such as active learning do not improve the model's performance substantially. We believe our Natural Language based Image/Video generation module can be leveraged to generate ground truths for niche use-cases. For instance, Cisco was involved in a CSR project in building video analytics systems for anti-poaching in African wildlife. Lack of ground truth labels can be a potential roadblock for niche applications like these and hence generating ground truth samples, in this case, generating image/video samples of miscreants poaching can be used for training the supervised models.

In general, the users would be able to generate samples that could be used to further train an existing model to handle edge cases or train a model from scratch based on the synthetic data. This functionality opens whole new possibilities.

5.5 Unifying the Intent between the User and Data Scientists

As discussed above, our model functions as an AI tool for data scientists by capturing their intent in natural language of the analytics they are interested in. It also acts as an interface between the end user and the analysts/scientists thereby unifying their intent. Often during ML model development, it is difficult for the end user to communicate their requirements to the data scientists and vice versa. Since our model interfaces with natural language, it would assist both end users and data scientists to generate content using natural language and iterate until both parties agree on.

6 Discussion

The Smart City use case showcased some of the advantages of employing a hybrid AI system, such as:

- Self-supervised learning of spatio-temporal scene semantics
- Generation of pertinent events through high-level abstractions.

However, to ensure real-time operation, the system needed to limit the total number of tracked objects to less than 100 at any given moment. This constraint became crucial when the same system was applied to retail inventory use cases, which could involve thousands of objects in a scene. To overcome this issue, we implemented a hierarchical knowledge representation based on spatial semantic abstraction, enabling a straightforward decomposition of the problem space and addressing combinatorial explosion concerns.

In our retail inventory project, we observed some promising capabilities, such as:

- Self-supervised “point and click” operation for retail scenarios involving products on shelves
- Precise identification of shelves, products on shelves, and comparable situations.

Nevertheless, the symbolic space processing for spatial semantics had limited applicability beyond the specific retail inventory use case. Although the subsymbolic potential product detector was generic, the bidirectional integration of the subsymbolic and symbolic spaces was far from being universally applicable.

Regarding our multivariate time series experiments, which demonstrated the ability to generate accurate descriptive and predictive models for key events in a self-supervised manner, this system has not yet been integrated with general reasoning and prior human domain expert knowledge.

While we have investigated various components of hybrid AI and achieved some promising initial results, much work remains to combine all these elements cohesively. Cisco’s open-source Deep Vision framework aims to facilitate further research and development in the field of hybrid AI.

7 Conclusion

Our hybrid AI system, developed in collaboration with Pei Wang, Patrick Hammer, Kristinn Thorisson, and other academics and AGI researchers, as well as with the support of key Cisco partners and customers, provides preliminary empirical evidence of the potential capabilities of hybrid AI systems. These capabilities include cumulative learning, generational learning, self-supervised learning, neurosymbolic integration, reasoning, and goal-oriented constructivist AI. Hybrid AI systems may offer viable solutions for a wide range of use cases and can be tailored to address complex and ever-changing business requirements. Cisco is making this work available as open-source through the Cisco Deep Vision project to ensure reproducible results and encourage further theoretical and empirical advancements in hybrid AI systems.

There remains a significant amount of work to be done in developing the underlying theory and mathematical models for hybrid AI systems, which would encompass new models of abstraction, neurosymbolic integration, cognitive synergy, meta-learning, and more. Additionally, there is a considerable need for further empirical validation using various existing datasets. Future work will concentrate on industry-standard datasets for time series and object detection [8, 15, 21, 23, 24].

References

1. Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al.: Relational inductive biases, deep learning, and graph networks. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 1–11 (2018)
2. Evans, R., Grefenstette, E.: Learning explanatory rules from noisy data. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, pp. 1–10 (2018)
3. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1126–1135 (2017)
4. d’Avila Garcez, A.S., Gabbay, D.M., Lamb, L.C.: Connectionist-symbolic integration: From unified to hybrid approaches. *Synthese* **170**(1), 143–166 (2009)
5. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1263–1272 (2017)
6. Hammer, P., Lofthouse, T., Fenoglio, E., Latapie, H.: A reasoning based model for anomaly detection in the smart city domain. In: NARS Workshop in AGI-19, Shenzhen, China, pp. 1–10 (2019)
7. Hammer, P., Lofthouse, T., Fenoglio, E., Latapie, H., Wang, P.: A reasoning based model for anomaly detection in the smart city domain. In: Arai, K., Kapoor, S., Bhatia, R., (eds) Intelligent Systems and Applications, pp. 144–159. Springer International Publishing, Cham (2021)
8. Hart, D., Goertzel, B.: Opencog: a software framework for integrative artificial general intelligence. In: Wang, P., Goertzel, B., St Franklin (eds) Proceedings of AGI2008, Memphis, Tennessee, pp. 468–472, USA (2008)
9. Hofstadter, D.R.: On seeing A’s and seeing As. *Stanf. Humanit. Rev.* **4**, 109–121 (1995)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: Proceedings of the International Conference on Learning Representations (2017)
11. Latapie, H., Kilic, O., Thórisson, K.R., Wang, P., Hammer, P.: Neurosymbolic systems of perception and cognition: the role of attention. *Front. Psychol.* **2105**(3), 2105 (2022)
12. Latapie, H., Kilicand, O., Liu, G., Kompella, R., Lawrence, A., Sun, Y., Srinivasa, J., Yan, Y., Wang, P., Thórisson, K.R.: A metamodel and framework for artificial general intelligence from theory to practice. *J. Artif. Intell. Conscious.* **8**(2), 205–227 (2021)
13. Latapie, H., Kilic, O.: A metamodel and framework for AGI. *2008(12879)* (2020)
14. Lenat, D.B.: Cyc: a large-scale investment in knowledge infrastructure. *Commun. ACM* **38**(11), 33–38 (1995)
15. Nivel, E., Thórisson, K.R.: Replicode: a constructivist programming paradigm and language. Technical RUTR-SCS13001. Reykjavik University School of Computer Science (2013)
16. Peirce, C.S.: Collected Papers of Charles Sanders Peirce, vol. 2. Harvard University Press, c-m (1931)
17. Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Extended Semantic Web Conference, vol. 10843, pp. 593–607 (2018)

18. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems, pp. 4077–4087 (2017)
19. Thórisson, K.R.: From constructionist to constructivist A.I. In: Tech Report FS-09-01. AAAI Fall Symposium Series: biologically Inspired Cognitive Architectures, pp. 175–183 (2009)
20. Thórisson, K.R.: Seed-programmed autonomous general learning. In: Proceedings of Machine Learning Research, pp. 32–70, Cambridge, MA, USA (2020)
21. Thórisson, K.R.: A new constructivist AI: from manual construction to self-constructive systems. In: Wang, P., Goertzel, B., (eds) Theoretical Foundations of Artificial General Intelligence, pp. 145–171. Springer, NY (2012)
22. Towell, G.G., Shavlik, J.W.: Knowledge-based artificial neural networks. *Artif. Intell.* **70**(1–2), 119–165 (1994)
23. Wang, P.: Rigid Flexibility: the Logic of Intelligence. Springer, Dordrecht (2006)
24. Wang, P.: Non-axiomatic Logic: a Model of Intelligent Reasoning. World Scientific, Singapore (2013)
25. Wang, P.: Experience-grounded semantics: a theory for intelligent systems. *Cogn. Syst. Res.* **6**(4), 282–302 (2005)
26. Wang, P.: Rigid Flexibility: the Logic of Intelligence. Springer, Dordrecht (2006)
27. Wang, P.: Non-axiomatic Logic: a Model of Intelligent Reasoning. World Scientific, Singapore (2013)
28. Wang, P.: A unified model of reasoning and learning. In: Thórisson, K.R., Robertson, P., (eds) Proceedings of the Second International Workshop on Self-Supervised Learning, Proceedings of Machine Learning Research, vol. 159, pp. 28–48. PMLR (2022)
29. Winograd, T.: Understanding natural language. *Cogn. Psychol.* **3**(1), 1–191 (1972)



Autoadaptive Networks of Coherent Domains for “Intelligent” Quantum Computation and Quantum Information

Luigi Maximilian Caligiuri^(✉)

Foundation of Physics Research Center (FoPRC), 87100 Cosenza, IT, Italy
luigimaxmilian.caligiuri@foprc.org, max.caligiuri@gmail.com

Abstract. It has been shown that liquid water can be considered as two-phases system composed of a coherent phase, in which the molecules oscillate in tune with an electromagnetic field within macroscopic regions called “coherent domains” and non-coherent vapor-like fraction, including uncorrelated molecules. Furthermore, such domains, due to the coherent dynamics, are the pools of quasi-free electrons forming cold coherent vortices upon an energy intake from the outside. Such electrons, in turn, form a plasma that, under suitable conditions, can itself become coherent and oscillate in tune with electromagnetic fields trapped inside it. We will show in this paper that if two or more plasma contained in water coherent domains interact each other, they can adaptively self-adjust their own oscillations in order to synchronize so creating a network of coherent domains oscillating in tune. Depending on the system’s parameters, different clusters of coherent domains, at different space-time scales, can appear. We finally show how this feature could be exploited and used to realize novel advanced quantum information and computational systems and quantum neural networks.

Keywords: QED coherence in matter · Water · Quantum computation · Synchronization · Quantum information

1 Introduction

The QED coherent behavior of liquid water has been the object of several investigations so far. In particular, basing on the work of Preparata [1], Del Giudice and co-workers [2–6], a first formulation of the structure of liquid water within the framework of QED has been achieved. According to this model, a ‘two-phases’ structure of liquid water emerges composed by a vapor-like component in which water molecules perform uncorrelated motion and a coherent liquid component in which the matter field associated to it oscillates in tune with a auto-generated electromagnetic field. More specifically, when suitable conditions about temperature and density of water are fulfilled, the system undergoes a quantum phase

<http://www.foprc.org>.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
K. Arai (Ed.): IntelliSys 2023, LNNS 822, pp. 799–819, 2024.
https://doi.org/10.1007/978-3-031-47721-8_53

transition towards a more stable state, characterized by a lower energy compared to the non-coherent state (quantified by a energy gap per water molecule) in which the electron clouds perform oscillations between their molecular ground state and a excited state (selected by the dynamics itself) about 0.4 eV below the ionization threshold. This coherent oscillation mode causes the formation of defined macroscopic spatial domains, named “coherent domains” (CDs), in which the coherent oscillation between matter and a self-trapped electromagnetic field unable to be irradiated outside takes place, whose size is determined by the wavelength of such common oscillations (in the case of liquid water about $0.1\text{ }\mu$ in size).

The coherent state of matter and e.m. field inside water coherent domains is a quantum superposition of the electron ground state and the excited one with a relative weight of about 0.1. For this reason, every water CD contains a lot of quasi-free electrons (one electron for water molecule on the average) that are easily excitable upon energy intake from the outside in the form of cold vortices [4,7] whose magnetic dipoles align along with the external magnetic fields (typically the Earth’s one). The coherent vortices of quasi-free electrons, forming the excited spectrum of the water coherent domains, have no internal friction, behaving like Cooper’s pairs in a superconductor, and can accumulate the energy supplied from the environment (provided it is lower than the energy gap/molecule associated to the coherent state), summing up to form a single vortex whose frequency is the sum of the frequencies of the single component vortices. In this way the excited coherent domain is a “device” able to convert low-frequency and high entropy energy of the environment to high-frequency and low entropy energy inside the coherent domains. At a given absolute temperature $T \neq 0$, the energy supplied to coherent state by thermal collisions could be able to reduce the coherent fraction of water molecules $F_{coh}(T)$ (defined as the ratio of the molecules in the coherent state to the total number) by pushing out some of them from the coherent state. However, it has been show that, for interfacial water (namely the water very close to a hydrophilic surface) the $F_{coh}(T)$ can be stabilized at a value close to unity even at room temperature so ensuring the coherence of the state [12,13]. Furthermore, we have already shown [7–11] that, when coherent domains are sufficiently close to each other, they can interact through the exchange of coherent virtual photons belonging to them, resulting in the modification of the excited levels of the coherent domains. Such feature allows water coherent domains to be used to realize logical quantum gates for quantum computation [10,11]. Furthermore, the reduction of entropy and the long-range order associated to the formation of coherent domains means they can be considered as very powerful storage devices of quantum information especially in the case of water [14,15]. In the aqueous systems further modes of QED coherent oscillations are allowed. In particular we’ll consider in the following the oscillations of the plasma of the negative electric charges (quasi-free electrons resulting by the coherent oscillations of electron clouds above considered) with respect to the positive electric charges (quasi free protons: the widows of quasi-free electrons). As shown in [1], like electron clouds, when the right conditions

are fulfilled, also the plasma of charged particles can access to a coherent state, with lower energy, in which charged particles perform oscillation (around their equilibrium positions) in tune with an electromagnetic fields confined inside the plasma. The parameters characterizing such coherent oscillations, namely their amplitude and frequency, depend, as we'll show in the following, on the excited state of the involved coherent domains so that when two CDs interact, modifying their energy levels, they also influence the coherent oscillations of the plasma they belong to so that they can then synchronize. This allows the arising of a self-adapting synchronizing mechanism able to generate different time-scale patterns of synchronized coherent domains according to the values of system parameters. The emerging patterns of coherent domains can be then used as a dynamical memory schemes as well as a learning system very similar to self-adapting neural networks.

We'll show how all these very special features of coherent water coherent domains could be exploited to realize advanced novel self-adapting macroscopic quantum systems with natural and revolutionary applications in the fields of quantum computation, information, artificial intelligence and data cryptography. Finally, the model here discussed also give a well-grounded explanation of the phenomenon of QED supercoherence, namely the coherence between coherent domains of liquid water.

In Sect. 2 we'll briefly review the key concepts of QED coherence in water with special regard to excited spectrum of coherent domains and their interaction. In Sect. 3 the coherent oscillations in plasma of quasi-free electrons of water has been considered. The transition towards synchronization of an extended networks of water coherent domains will the object of Sect. 4, then, in Sect. 5 we'll suggest some possible applications of such dynamics to "intelligent" quantum computation and storage of quantum information. Finally, in Sect. 6, we sketch out a summary and some final remarks.

2 QED Coherence in Water: A Brief Overview

From a quantum field theory (QFT) viewpoint, condensed matter can be described in terms of the interaction between quantum matter field and gauge fields as, in particular, the electromagnetic field. Within this framework, Preparata [1] showed that if the density of matter system overcomes a threshold value and temperature is below a critical value the system undergoes a spontaneous quantum phase transition towards the so-called coherent ground state (CGS) in which the oscillation of matter field is phase-locked with that of electromagnetic field inside well-defined spatial regions for this reason named "coherent domains" (CDs). The phase-locked regime appears as a consequence of the synchronization of the oscillations of all the elementary matter components between two energy levels belonging to their spectrum (typically the atomic/molecular ground state E_{gs} and an excited state E_{exc} selected by coherent dynamics itself) that also determines the "size" of the coherent domain given by (from this point

on we'll adopt the system of natural units $\hbar = c = 1$)

$$d \simeq \frac{2\pi}{\Delta E} \quad (1)$$

where $\Delta E = |E_{exc} - E_{gs}|$. A very remarkable consequence of the coherent behaviour is the renormalization of the common oscillation frequency to the value

$$\omega_r = \left| 1 - \dot{\phi} \right| \omega_0 \quad (2)$$

where $\omega_0 = \Delta E$ and ϕ is the phase of the electromagnetic field in the coherent state that is phase locked with that of the matter fields describing the atomic/molecular ground state $|gs\rangle$, given by θ_0 , and that of the excited one $|exc\rangle$ given by θ_1 , so that the following constraint holds

$$\dot{\phi} = \dot{\theta}_0 - \dot{\theta}_1 \quad (3)$$

The transition from the non-coherent state to CGS is ruled by the value assumed by a specific parameter g , that is a function of the system features, compared with its “critical value” g_c , namely

$$g \geq g_c \quad (4)$$

where

$$g = \frac{\omega_P}{\omega_0} \sqrt{2\pi f_{01}} \quad (5)$$

ω_P being a plasma frequency given by $\omega_P = e\sqrt{N/Vm_e}$ (N is the number of atoms/ molecules in the coherent state, V is the volume covered by the coherent oscillation and m_e the electron mass), f_{01} the oscillator strength for the transition $|gs\rangle \iff |exc\rangle$ and

$$g_c = \frac{8}{27} + \frac{2}{3}\mu + \left(\frac{4}{9} + \frac{2}{3}\mu \right)^{3/2} \quad (6)$$

in which μ is a constant that depends on the spectrum of the excited levels of the atom/molecule involved in the coherent oscillation. The coherent state is the true ground state of the system since it is more stable compared with the non coherent state being characterized by a negative energy gap per atom/molecule $\Delta E/N$ due to the release of energy during the “condensation” process accompanying the quantum phase transition towards the CGS. The latter is also a macroscopic quantum object described by a single macroscopic wavefunction with a well defined phase, namely

$$\Psi = \psi e^{i\Theta} \quad (7)$$

whose squared amplitude $|\psi|^2$ gives the local density of elementary quantum oscillators in the coherent state and the phase Θ is the eigenvalue of a suitable quantum phase operator [16], that characterize the coherent quantum behaviour of the system arising form the phase-locking condition (3). The physical meaning

of (7) is in all similar to that of the wavefunctions used to describe the macroscopic quantum state of superfluids and superconductors [17]. In the very special case of water the coherent state determined by the oscillations of electron cloud in water molecule is the superposition

$$|coh\rangle = \cos\alpha |gs\rangle + \sin\alpha |exc\rangle \quad (8)$$

with $\cos^2\alpha = 0.873$ and $\sin^2\alpha = 0.127$, so that the excited state has an associated probability of more than 10% to occur. In the case of water this state is just about 0.4 eV lower than the ionization threshold, meaning that, on the average, about one electron per water molecule inside a CD can be considered as quasi-free. This feature makes the coherent state of water very special since it allows the CD to have a spectrum of excited energy levels. The latter occur in the form of vortices of quasi-free electrons whose magnetic dipoles align to the external magnetic fields in the environment as the Earth's one. They can be considered as a charged fluid performing "ridig" rotations characterized by a quantized angular momentum given by

$$\mathbf{L} = I\boldsymbol{\omega} \quad (9)$$

where I is the momentum of inertia of the ensemble of quasi free electrons at a given configuration and $\boldsymbol{\omega}$ their angular velocity. The corresponding energy of a given vortex can be obtained by the expression

$$E = \frac{L^2}{2I} - \gamma \mathbf{L} \cdot \mathbf{B} \quad (10)$$

where γ is the gyromagnetic ratio and \mathbf{B} the external static magnetic field. When the water CD is supplied by certain amounts of energy E_i from the environment, provided that for each energy contribution $E_i < |\Delta E/N|$, such energy can be accumulated by the CD in the form of a unique coherent vortex whose energy $E = \sum E_i$ is given by Eq. (10). The coherent vortices cannot lose energy thermally since they have no internal friction and the resulting metastable states, in absence of interaction with the outside environment, have very long lifetime, their vorticity being included in the macroscopic quantum phase of Eq. (7). As already show by this author [8–11, 17], the existence of excited coherent states of water allows the CDs to interact each other by exchanging virtual photons belonging to the evanescent tails of their respective coherent electromagnetic fields "trapped" inside them. This especially occurs when two or more coherent domains are sufficiently close each other. Through this process a coherent domain can then absorb energy from the environment to excite its energy level and then "release" it to a neighbor one by exciting it. As a result the values of macroscopic quantum phase associated to the interacting CDs is able to change. In this case every couple of coherent domains can be considered as a unique two-levels oscillating system able to exchange energy with the environment. As shown in Refs. [8, 10, 11], the interaction between two coherent domains can be parameterized through the interaction strength Γ ruling the Hamiltonian of the system. In particular, due to such interaction, the transition probability of a

given coherent domain from an initial ($t = 0$) state $|1\rangle$ with energy E_1 to a state $|2\rangle$ with energy E_2 (at $t > 0$) is given by

$$P_{12} == \frac{\Gamma^2}{\Gamma^2 + (E_1 - E_2)^2} \sin^2 \left(\sqrt{\frac{(E_1 - E_2)^2 + 4\Gamma^2}{4}} t \right) \quad (11)$$

that is equivalent to a “Rabi oscillation” in which a photon wave packet of given wave number is coupled back and forth between two coherent domains. In the following section we’ll show how such interaction, in the presence of a further coherent mode of oscillation allowed inside water coherent domains (coherent plasma oscillations), can eventually synchronize the oscillations of two or more of them, giving rise to a “supercoherence”, namely a coherence between a network of coherent domains able to produce very surprising and interesting effects with possible applications in the fields of quantum computation, quantum information and artificial intelligence.

3 QED Coherence in Plasma of Water Quasi-Free Electrons

The ensemble of quasi-free electrons produced inside water coherent domains can be considered as a plasma of negative electric charges able to perform oscillations around their equilibrium position with a pulsation ω_R . In general, the coherent oscillations of the electron clouds, giving rise the coherent domains, make the electrons to oscillate between the molecular ground state $|gs\rangle$ and the excited state $|exc\rangle$, whose amplitude and frequency are functions of the difference $|E_{exc} - E_{gs}|$. In the case of the coherent oscillation mode above described (namely those involving the electron clouds only), the oscillation have the pulsation ω_{coh} and an amplitude that can be obtained by conceptually picturing the oscillating electron like a quantum harmonic oscillator whose “position” x satisfies the following equations

$$\langle x \rangle = 0 \quad (12)$$

$$\langle x^2 \rangle = \frac{E_n}{m\omega^2} \quad (13)$$

where E_n is the energy of the considered state and ω the pulsation. On the other hand, by applying the correspondence principle with a classical harmonic oscillator, we can also write

$$\langle x^2 \rangle = \frac{E_n}{m\omega^2} = \frac{A^2}{2} \quad (14)$$

where A has the meaning of the classical amplitude of the oscillation that, in the quantum context, gives an estimation of $\langle x_{\max}^2 \rangle$, namely

$$\langle x_{\max}^2 \rangle = \frac{2E_n}{m\omega^2} \quad (15)$$

In the case of a water coherent domain, the energy E_n also depends on the given excited state in which it finds itself, namely on the energy of the corresponding vortex of quasi-free electrons given by Eq. (10). The energy available to each quasi-free electron to perform plasma oscillations can be then obtained as

$$E_{plasma} = (E_{exc} + E_{vort}) - E_{gs} \quad (16)$$

where E_{vort} is the energy of the vortex containing the quasi-free electron.¹ By specifying Eq. (10) for a single quasi free electron we have

$$E_n = mr \left(\frac{1}{2} r \omega_n^2 - \gamma \boldsymbol{\omega}_n \cdot \mathbf{B} \right) \quad (17)$$

where r is the distance of the given electron from the rotation axis. By inserting the latter in Eq. (16) we have, for the plasma oscillations of quasi-free electrons inside water coherent domains

$$E_{osc} = mr \left(\frac{1}{2} r \omega_n^2 - g \boldsymbol{\omega}_n \cdot \mathbf{B} \right) + \Delta E \quad (18)$$

with $\Delta E \equiv E_{exc} - E_{gs}$. Equations (18) and (15) together then give

$$\langle \mathbf{x}_{max}^2 \rangle = \frac{2}{m \omega_R^2} \left[mr \left(\frac{1}{2} r \omega_n^2 - g \boldsymbol{\omega}_n \cdot \mathbf{B} \right) + \Delta E \right] \quad (19)$$

Two key points must be now underlined:

1. the oscillation amplitude and frequency of the quasi-free electrons plasma depend on the parameters of the excited state of the coherent domain;
2. the maximum value of oscillation amplitude is bounded from the above by the energy of the excited state.

As shown in Ref. [1], if the suitable conditions occur, this plasma of quasi-free electrons can give rise to a further coherent oscillation mode in which all the electric charges in the plasma perform correlated oscillations through the action of an electromagnetic field. More specifically, by starting from small amplitude uncorrelated plasma oscillations at frequency ω_R , the system undergoes a “run-away” towards a coherent state where all the charges oscillate in phase at a common renormalized frequency given by

$$\omega_{coh} = \omega_R \left(1 - \dot{\phi} \right) \quad (20)$$

where, as above, ϕ indicates the phase of the coherent e.m. field keeping tuned the plasma oscillations. As for the amplitude of the related oscillation amplitude, it is ruled by the following equation

$$A^2 \left(1 - \dot{\phi} \right) + \alpha^2 = 0 \quad (21)$$

¹ In order to speak of plasma of quasi-free electrons the energy transferred by the vortex to the electron should be less than the ionization threshold.

whose solution is given by [1] ($x = \alpha/A$):

$$\dot{\phi} = 1 + \sqrt{1 - 2gx} \quad (22)$$

$$x^2 = \sqrt{1 - 2gx} \quad (23)$$

$$A = \frac{\alpha_{\max}}{x} \quad (24)$$

where α_{\max} is the maximum value of the plasma charges displacement and

$$g^2 = \frac{2\pi}{3} \left(\frac{\omega_P}{\omega_R} \right)^2 \quad (25)$$

Liquid water can then host simultaneously several coherent oscillation modes like, in the case here considered, the one related to the electron clouds oscillations and the other corresponding to the plasma oscillations. Both these modes generate corresponding energy gaps compared with their respective non-coherent modes. In particular, according to Ref. [1], the plasma coherent state can be achieved as a stable physical state only if the displacement amplitude of charges oscillations is bounded from above as occurs in the present case due to the Eq. (19), so we can write

$$\alpha_{\max} = \sqrt{\langle \mathbf{x}_{\max}^2 \rangle} \quad (26)$$

ensuring that, provided that the runaway condition $g \geq g_c$ ($g_c = \sqrt{16/27}$) is satisfied, the system reaches the coherent state. It is very important to remark the amplitude and pulsation of the coherent mode given by Eqs. (21)–(24) depend, through Eq. (19), on the energy of the excited state in which the coherent domain is found by the plasma oscillations. This means that the coherent mode could be able to self-adjust its own oscillation parameters by exchanging energy with the environment, according to the mechanism we have suggested above, namely by its interaction with other coherent domains.

4 The Transition to Synchronized State and “Supercoherence”

When two or more coherent domains are sufficiently close each other they can interact by exchanging energy among them and with the environment. As regards the coherent plasma, described above, the energy absorption and release are able to modify the amplitude and frequency of oscillation compared to an isolated coherent domain. We now make the assumption that, at macroscopic scale, an individual coherent domain can be conceptually pictured as a “Stuart-Landau oscillator” whose runaway towards the CGS corresponds to the arise of a stable limit cycle due to the passage of the system through a supercritical Andronov-Hopf bifurcation. Such class of oscillators can be described by the equation [18, 19]

$$\dot{Q} = (i\omega + \alpha) Q - \beta |Q|^2 Q \quad (27)$$

where Q is a complex oscillation amplitude, ω is the natural frequency of the oscillator, α and $\beta > 0$ are real parameters. Without entering into a detailed description of Eq. (27) that can be found in literature, the point to emphasize now is the existence of a stable limit cycle, corresponding, within our framework, to the coherent oscillation of the isolated domain, for $\alpha > 0$ such as

$$Q(t) = \sqrt{\frac{\alpha}{\beta}} e^{i\omega t} \quad (28)$$

that in all resembles the macroscopic wavefunction (9) describing, in the QFT approach, the macroscopic coherent state of water, with $\omega = \omega_{coh}$. For an extended ensemble of N interacting oscillators (water coherent domains), grounding on our previous results [8–11, 17], we modify Eq. (27) by introducing the interaction between coherent domains so that, for the k -th domain, we can write

$$\dot{Q}_k = (i\omega_k + \alpha) Q_k - \beta |Q_k|^2 Q_k + \frac{\Gamma}{N} \sum_{j=1}^N Q_j \quad (29)$$

where Γ is the coupling strength parameter (supposed, in first approximation, to be the same for all the coherent domains) and ω_k is pulsation of the k -th coherent domain that is, as above discusses, a function of the energy of the excited state in which it is found. We note that, in writing Eq. (29), we have considered the interaction between CDs to be mediated by a “mean-field” in which all the oscillators, as far as the coupling, are considered as equivalent. In our preliminary dynamical model of CDs interaction, the latter (namely the connection strength or the coupling between them) depends, in first approximation, both on the (intercentrums) distance between coherent domains and their structural features as well as on the (difference between) excited energy levels involved in the oscillations of the interacting coherent domains, in turn, controlling the amplitude and frequency of plasma oscillations ruling the synchronization process. Consequently, we can assume the strength of the local mean field generated by all the other interacting oscillators to be constant (the extension of the model to time-space varying connection strength will be discussed in forthcoming publications) so that we can assume as reasonable the assumption leading to Eq. (29). We then search for a solution of Eq. (29) in the form

$$Q_k = q_k e^{i\phi_k} \quad (30)$$

by substituting Eq. (30) in Eq. (29) we obtain, after some easy manipulations

$$\dot{q}_k = (\alpha - \beta q_k^2) + \frac{\Gamma}{N} \sum_{j=1}^N q_j \cos(\phi_j - \phi_k) \quad (31)$$

$$\dot{\phi}_k = \omega_k + \frac{\Gamma}{N} \sum_{j=1}^N \frac{q_j}{q_k} \sin(\phi_j - \phi_k) \quad (32)$$

An examination of Eq. (28), compared with our model in which the squared amplitude of wavefunction (7) quantifies the local density of oscillating quasi-particles, gives the correspondence

$$\frac{\alpha}{\beta} \rightarrow |\Psi|^2 = n = \frac{N_p}{V} \quad (33)$$

where N_p is the total number of oscillating elementary components of the system and V its volume. In the QFT approach to condensed matter physics that leads to the coherent dynamics above outlined this, requires that $N_p \rightarrow \infty$ keeping the ratio N_p/V finite. This also agrees with the meaning of such ratio as local density of particles that must remain finite. By taking the above limit, Eq.(31) becomes

$$\dot{q}_k = \beta \left(\frac{\alpha}{\beta} - q_k^2 \right) q_k \quad (34)$$

so that, for every k (namely for every domain), q_k doesn't depend on the system and it reaches a stable fixed point given by

$$\bar{q}_k = \sqrt{\frac{\alpha}{\beta}} \quad (35)$$

namely the amplitude of every domain is the same and depends only on its own structural features in agreement with our assumptions. By inserting Eq. (34) in (32) we finally have for the phase equation

$$\dot{\phi}_k = \omega_k + \frac{\Gamma}{N} \sum_{j=1}^N \sin(\phi_j - \phi_k) \quad (36)$$

that is formally identical to the Kuramoto model equation [18–20]. Equation (36) has N degrees of freedom (the values of ϕ_k for each coherent domain) and its phase space is a N -dimensional torus $T^N = [0, 2\pi]^N$ on which we can define the phase sum

$$\phi_S = \sum_{k=1}^N \phi_k \quad (37)$$

so that we have

$$\dot{\phi}_S = \sum_{j=1}^N \omega_j = N\bar{\omega} \quad (38)$$

where $\bar{\omega}$ is the average frequency of the ensemble of N oscillators. It is worth noticing that the above model is also invariant under a phase and a frequency shift. We are interested in the case of a very large number of oscillators (namely an extended network of interacting water coherent domains) to study its behavior as regards their synchronization and coherent dynamics. Being the ensemble of coherent domains is a huge collection of macroscopic quantum objects (the same

situation that conceptually occurs in Quantum Field Theory as well), we can characterize their collective dynamics by means of an order parameter given by

$$\varphi_0(\mathbf{x}, \gamma; t) = \langle \Omega | \hat{\Psi}(\mathbf{x}, \alpha; t) | \Omega \rangle \quad (39)$$

where $|\Omega\rangle$ is the state of the quantum field and $\hat{\Psi}(\mathbf{x}, \alpha; t)$ the quantum field operator acting on it. The order parameter given by Eq. (39) is the quantum wavefunction associated to the field Ψ that, in our case, has just the form given by Eq. (30). We then introduce the order parameters $r(t)$ and $\Theta(t)$ such as

$$r(t) e^{i\Theta(t)} = \frac{1}{N} \sum_{j=1}^N e^{i\phi_j(t)} \quad (40)$$

where $r(t) \in [0, 1]$. The aim of Eq. (40) is to measure the degree of synchronization or the coherence of the whole ensemble of oscillators. In fact, in the case of complete synchronization, that is $\phi_1 = \phi_2 = \dots = \phi_N$, we obtain $r(t) = 1$ and $\Theta(t) = k\pi$ ($k = 0, \pm 1, \pm 2, \dots$) in agreement with the fact the system has a well-defined amplitude and phase. In the general case of a partial synchronized system, the phase Θ represents an “average phase” and the value of r gives the “degree” of synchronization of the system. Equation (31), when written as a function of the order parameter, becomes

$$\dot{\phi}_k = \omega_k + \Gamma r \sin(\Theta - \phi_k) \quad (41)$$

that, coupled with Eq. (40), gives a system of self-contained equations depending on Γ , N and the frequencies ω_k . This equation describes the forced dynamics of any single oscillator k as subjected to an external force due to the mean field generated by any other. More specifically, when the frequencies ω_k are uniformly and randomly distributed then, at a given time, also the phases ϕ_k are the same in the interval $[0, 2\pi)$ and the average contribution to a mean field is zero. On the other hand, when some oscillators are synchronized on the same frequency, their contributions to the mean field sum coherently giving a non-zero field that, in turn, affects other oscillators locking them with the previous ones so further increasing the main field and so on. If we suppose the arising mean field to be periodic, Eq. (41) is equivalent to the phase equation of a periodically driven oscillator and we can search for solutions of the order parameter in the form

$$r(t) = \hat{r} \quad (42)$$

$$\Theta(t) = \Omega t \quad (43)$$

This choice perfectly fits our model, according to which the whole ensemble of elementary macroscopic oscillators (each represented by a water coherent domain) would be described, in turn, by a unique quantum macroscopic wavefunction, namely it shows a super-coherent behavior in which all the coherent domains oscillate in tune with the same pulsation Ω , a constant phase shift (as

compared to Θ) and amplitude \hat{r} . The idea underlying the Ansatz given by Eqs. (42) and (43) is to obtain the mean field parameters in analogy to the mean field theory of second-order phase transition. The solution of Eq. (41) for the average frequency $\bar{\omega}$ can be generally very complex but if we suppose, as a reasonable assumption, that the distribution of natural frequencies of oscillators $g(\omega)$ is symmetric around its average value $\bar{\omega}$ then we can assume $\Omega = \bar{\omega}$. By introducing the phase difference

$$\psi_k = \phi_k - \Theta = \phi_k - \bar{\omega}t \quad (44)$$

so that $\dot{\psi}_k = \dot{\phi}_k - \bar{\omega}$, Eq. (41) can be written as

$$\dot{\psi}_k = \omega_k - \bar{\omega} + \Gamma\hat{r}\sin(\Theta - \phi_k) \quad (45)$$

For an oscillator synchronized with the mean field, we must have $\dot{\psi}_k = 0$ and then, from Eq. (45) we obtain

$$\psi_k = \arcsin\left(\frac{\omega_k - \bar{\omega}}{\Gamma\hat{r}}\right) \quad (46)$$

Equation (46) states that, for every oscillator entrained by the mean field (oscillating with the frequency $\bar{\omega}$), the phase difference ψ_k is constant and depends, in particular, on its natural frequency ω_k so that the phase ϕ_k is given by²

$$\phi_k(t) = \arcsin\left(\frac{\omega_k - \bar{\omega}}{\Gamma\hat{r}}\right) + \bar{\omega}t \quad (47)$$

Equations (46) and (47) hold for a synchronized oscillator whose natural frequency ω_k must be not too different from $\bar{\omega}$ (so that the mean field could be able to entrain it for a given value of coupling constant Γ), namely if

$$|\omega_k - \bar{\omega}| \leq \Gamma\hat{r} \quad (48)$$

If the condition (48) is not satisfied, the oscillator is not synchronized with the mean field and the phase difference ψ_k rotates non-uniformly according to Eq. (45). It is clear from (48) that the higher the coupling constant Γ , the higher is the width of the frequency interval of oscillators able to be synchronized by the mean field. The values of $\bar{\omega}$ and \hat{r} characterizing the mean field can be obtained in a self-consistent way by determining the contributions to the mean field respectively due to the synchronized and not-synchronized sub populations of oscillators in the system. To this aim we determine the distributions of oscillators n_s and n_{ns} respectively belonging to such two sub-populations in the limit $N \rightarrow \infty$ ³ [18]. Equation (46) tells that, for the synchronized oscillators, the phase ψ_k is time-independent and is a function of ω_k so that $n_s(\psi)$ can be calculated from $g(\omega)$

² We assume $\Theta(0) \equiv \phi^0 = 0$ for simplicity.

³ In this case we can assume ψ_k and ϕ_k to be continuous variables and omit the subscript k .

by means of the equation

$$n_s(\psi) = g(\omega) \left| \frac{d\omega}{d\psi} \right| \quad \psi \in \left[-\frac{\pi}{2}; \frac{\pi}{2} \right] \quad (49)$$

based on the assumption that, for a synchronized oscillator, the phase ψ is a function of ω according to Eq. (45) with $\dot{\psi}_k = 0$, then we can write

$$n_s(\psi) = \Gamma \hat{r} g(\omega + \Gamma \hat{r} \sin \psi) \cos \psi \quad (50)$$

As regards as the distribution $n_{ns}(\psi)$, using the symmetry of g , namely $g(\omega) = g(-\omega)$, we obtain [18]

$$n_{ns}(\psi) = \int_{-\Gamma \hat{r}}^{\infty} \frac{g(\bar{\omega} + x)}{\pi} \frac{x \sqrt{x^2 - \Gamma^2 \hat{r}^2}}{(x^2 - \Gamma^2 \hat{r}^2 \sin^2 \psi)} dx \quad (51)$$

where $x = \omega - \bar{\omega}$. In the continuum limit, Eq. (40) takes then the form

$$\Gamma e^{i\bar{\omega}t} = \int_{-\pi}^{\pi} e^{i(\psi + \bar{\omega}t)} [n_s(\psi) + n_{ns}(\psi)] d\psi \quad (52)$$

so, by inserting Eqs. (50) and (51) in Eq. (52) and taking the real and imaginary parts we achieve, after few manipulations, the two self-consistent equations for $\bar{\omega}$ and \hat{r}

$$\hat{r} = \hat{r} \Gamma \int_{-\pi/2}^{\pi/2} \cos^2 \psi g(\omega + \Gamma \hat{r} \sin \psi) d\psi \quad (53)$$

$$0 = \hat{r} \Gamma \int_{-\pi/2}^{\pi/2} \sin 2\psi g(\omega + \Gamma \hat{r} \sin \psi) d\psi \quad (54)$$

Equation (54) allows for the determination of Ω and, according to our assumption about the symmetry of $g(\omega)$, it is verified for $\Omega = \bar{\omega}$ as predicted. On the other hand, Eq. (53) gives the amplitude of the mean field \hat{r} that, in general, depends on the specific form of the distribution function g . However it has been shown [18, 19] the integral in (53) can be calculated exactly for some special forms of distribution such, for example, the uniform as well as unimodal distribution. In the latter case a non-trivial mean field exists if the coupling Γ exceeds a critical value Γ_c whose value depends on the frequency distribution $g(\omega)$ and the expression of the mean field amplitude is of the type

$$\hat{r} \sim (\Gamma - \Gamma_c)^{\frac{1}{2}} \quad (55)$$

For not too high values of Γ an explicit expression of Γ_c can be calculated since, in this case, only the frequencies around $\bar{\omega}$ are synchronized and we can expand the function $g(\omega)$ in Taylor series around $\omega = \bar{\omega}$, namely

$$g(\omega) \simeq g(\bar{\omega}) + \frac{1}{2} g''(\bar{\omega}) \omega^2 = g(\bar{\omega}) + \frac{g''(\bar{\omega})}{2} \hat{r}^2 \Gamma^2 \sin^2 \psi \quad (56)$$

that, inserted in Eq. (53), gives

$$\hat{r} \simeq \left[\frac{8g(\bar{\omega})}{|g''(\bar{\omega})|\Gamma^3} (\Gamma - \Gamma_c) \right]^{\frac{1}{2}} \quad (57)$$

with

$$\Gamma_c = \frac{2}{\pi g(\bar{\omega})} \quad (58)$$

The synchronization process in all resembles a second-order phase transition and the supercoherent state is the result of such transition from a network of uncorrelated oscillating water coherent domains towards a macroscopic state characterized by a cluster of oscillators that are phase-locked with the mean field generated by their interaction. This process is just the “Kuramoto transition” for an ensemble of generic self-sustained interacting oscillators.

4.1 Phenomenology of the Transition Towards the Water Supercoherence State and its Main Features

As well as a single water coherent domain is the result of a quantum phase transition from a state of uncorrelated oscillations of its elementary quantum matter oscillators (atoms and/or molecules) to a state in which matter and e.m. field perform tuned oscillations described by a macroscopic quantum wavefunction of well-defined amplitude and phase, in an all-similar fashion the supercoherent state, composed by a very high number of interacting coherent domains oscillating in phase, is described by a macroscopic order parameter (having the meaning of a wavefunction in the coherent quantum state) quantified by the complex number as in Eq. (40). This similarity between the two processes is also emphasized by the existence, for both of them, of a critical value of a typical parameter (the critical coupling strength Γ_c in one case and the critical value g_c in the other) that rules the phase transition of the system. Even more interesting is the fact that the higher the value of such parameter, the higher is the degree of coherence of the asymptotic state of the system. As regards the latter, the phenomenology implied by Eq. (36) generally confirms such behavior [18–22]. Leaving to forthcoming publications a more detailed analysis, we can stress some key points as follows: for $\Gamma < \Gamma_c$ the mean field is too low to be able to synchronize the oscillators and the ensemble remains in a non-coherent state in which all the coherent domains perform uncorrelated oscillations with their “natural” frequencies. If Γ then remains lower than a given value, corresponding to the solution $\hat{r} = 1$ in the Eq. (57), the system will not synchronize at all, even for $t \rightarrow \infty$. For $\Gamma \geq \Gamma_c$ the mean field described by the order parameter starts to grow becoming able to entrain more and more oscillators, through a sort of “nucleating” process (that resembles the “condensation” of quasiparticles that give rise to the formation of a single coherent domains), so creating a cluster of synchronized oscillators that, for a given value of Γ , increases with time until it reaches its asymptotic state (signalled by the value of \hat{r}). The cluster of synchronized oscillators begins to form around the central frequency $\bar{\omega}$ of the distribution $g(\omega)$ whose phases

vary, as a function of ω , according to Eq. (46). The amplitude of the parameter \hat{r} defines the degree of synchronization of the asymptotic state and its value increases with the coupling strength Γ . For low values of Γ , the phases ϕ_k are then almost uniformly distributed in the interval $[0, 2\pi]$ while, by increasing Γ , most phases group around a single value with a decreasing spread around this. We have then outlined a feasible physical mechanism able to explain the arising of the supercoherent state of a network of water coherent domains based on the model of interaction between coherent domains already proposed by this author [7–10] and the synchronization process described by the Kuramoto model. The cluster of synchronized oscillators, in which all the water coherent domains, in turn, oscillate coherently with each other, is a “supercoherent” cluster namely a system in which the coherence between coherent systems takes place at different space-time scales. In other words, the supercoherent state reproduces, at greater scales, the coherent oscillations occurring inside a single coherent domain. It is important to note the occurrence of synchronization phase transition depends on the chance that the single coherent domains could self-adjust their oscillation frequencies as the result of their mutual interaction that is just what allowed by the coherent plasma oscillations of quasi-free electrons inside water coherent domains as described above. It is then very important, at this stage, to further emphasize such synchronization should be considered as a mode of self-organization of a complex system in our case grounded on the capability of the interacting coherent domains of water to self-adjust their oscillation frequencies. In particular, as we have seen, the coherent domains, already synchronized, tend to entrain more water domains compared with those entrained by the uncorrelated ones according to a mechanism just resembling the same at work in the neural networks of both natural and artificial type. On the other hand, neural networks can also be described as networks of coupled phase oscillators in which the role of relative spike timing is played by phases of the involved individual oscillators [21] according to a complimentary approach in which neuronal activity is modeled by replacing neurons by periodic oscillators. In this connection a more specific framework in which the interaction strength between water coherent domains is a space/time function is already in progress and will be the subject of future investigations.

4.2 Memory Storable in a Self-adjusting Network of Supercoherent Water Domains

It has been shown [14,15] a QED coherent system, like a single water coherent domain or a network of interacting water coherent domains, is able to store an amount of information due to its energy gap, also corresponding to a reduction of the entropy of the coherent state, compared to the non-coherent one, due to the arising of a long-range order. On the other hand, the increase of interaction strength Γ generally induces a wider energy gap associated to the system compared with an ensemble of non-interacting coherent domains as calculated in Ref. [14]. Such result also agrees with the model of supercoherence as oscillators

synchronization since the greater the value of Γ , the greater the number of oscillators belonging to the synchronized cluster. In Refs. [14, 15] the total amount of information storables in a network of interacting water coherent domains in their ground states has been calculated as

$$I_{tot} = -\frac{32}{9\pi^2} \left(\frac{\Delta E_1}{N} \right)_{r=0} \frac{F_{coh}(T) N}{(ln2) k_B T} \frac{n_{CD} (n_{CD} - 1)}{2} \quad (59)$$

where $(\Delta E_1/N)_{r=0}$ is the energy gap per atom/molecule of a single water coherent domain (supposed spherically symmetric) at its center, N the number of elementary components in the coherent domain, T the absolute temperature, $F_{coh}(T)$ the coherent fraction of water at temperature T , k_B the Boltzmann constant and n_{CD} the number of interacting water coherent domains in the ensemble.

The overall amount of information storables in the supercoherent state then depends on the fraction of synchronized oscillators of the system at a given time t . By using the above results, we can estimate such a number $n_{CD} = N_{t,s}$ through the equation

$$N_{t,s} = N - N_{t,ns} \quad (60)$$

where $N_{t,ns}$ is the number of oscillators of the system that, at time t , are not synchronized. By inserting Eq. (49) in (59) we have

$$N_{t,s} = N - \int_{-\infty}^{\infty} \int_0^{2\pi} g(\omega) \frac{\sqrt{(\omega - \bar{\omega})^2 - \Gamma^2 \bar{r}^2}}{2\pi |\omega - \bar{\omega} - \Gamma \hat{r} \sin(\theta - \Theta)|} d\theta d\omega \quad (61)$$

that is a function of Γ in a complex manner. Equations (61) and (46) show that both I_{tot} and ψ_k depend, in particular, on the distribution function of natural frequencies $g(\omega)$ that, in this way, determines the total information stored in the supercoherent state as well as the distribution of the phase difference ψ_k in this state. This means that, by suitably choosing the function $g(\omega)$, it could be possible in principle to control the quantity I_{tot} and, much more interestingly, the phase ϕ_k of the oscillators in the synchronized state so allowing to store a quantity of information that will be codified in the phase structure of the synchronized cluster. The physical model here proposed then states the correspondence

$$I \rightarrow g(\omega) \leftrightarrow \{\phi_k\} \quad (62)$$

so that an information scheme could be then selectively codified in the initial (not synchronized) state of the water domains ensemble through the action of an external oscillating driving force of a given frequency able to entrain selected sub-populations of water coherent domains by constructing a precise distribution of initial phase differences φ_k . The set-up of such phase distribution could be achieved by one of the methods already proposed by this author, namely by exploiting the electric version of the Aharonov-Bohm effect [23] or the electromagnetic memory effect. Both the procedures, in fact, allow the fast and precise

setting of the phase of water coherent domains so that their natural frequencies can be adjusted according to a specific experimental configuration. The subsequent self-adjusting evolution of the system towards the synchronization (supercoherent) state changes the initial memory configuration into a stationary phase pattern, based on the latter, whose structure depends on Γ and the degree of coherence of the network (measured by \hat{r}). It should be clear now that the synchronized state composed by the cluster of supercoherent water domains could also represent the result of a quantum computation performed by the interacting coherent domains of water. Such calculation outcome, stored in the final configuration of the synchronized cluster, could be either the result of a well-defined set-up strategy or generated by the self-organizing evolution of the system, even in response to its interaction with the surrounding environment. Furthermore, due to the peculiar dynamics involved in the synchronization process of water coherent domains, that is able to “naturally” codify the final state of the system, this can be also considered, as we’ll suggest in a forthcoming study, to realize a quite novel method to generate, communicate and store quantum information. Indeed, the final configuration of the supercoherent state, apart from the frequency distribution $g(\omega)$, also depends on the form of the mutual interaction between the oscillator supposed at work in the Eq. (36). In fact, if the sinusoidal function is obviously the simplest form of periodic function, any other 2π -periodic function could be in principle considered according to which different final configurations are possible. For example, in [22], the following interaction function has been considered

$$q(\phi) = -\frac{1}{\delta} \tan^{-1} \left[\frac{\delta \sin \phi}{1 - \delta \cos \phi} \right] \quad (63)$$

with $\delta = -1$ and $\Gamma = 1$. In this case the final state is characterized by a three-clusters structure and a multistable behavior showing different pattern of synchronization composed by three sub-populations of oscillators each characterized by a definite value of phase ϕ_k . The consideration of other interaction functions, different than the sinusoidal one, within our model of interacting water coherent domains will be discussed in a forthcoming paper and it will open even more interesting perspectives of research and applications. As regards as the stability of the supercoherent state, that is revealed crucial in order to exploit it as a computational and storing “device”, we must remind that a water coherent domain is as itself more stable than the corresponding non-coherent state since it cannot lose energy by radiating it outward. On the other hand, its stability can be compromised by the thermal noise. Nevertheless, the effect of thermal fluctuations of the environment, if not too strong, is counteracted by the increase of the overall stability of the supercoherent system (also corresponding to a increased overall energy gap compared to the not-synchronized state) due to the synchronization process itself (already numerically proven for an ensemble of generic interacting oscillators [20, 21]), as well as by the feasible introduction of suitable meta-materials to enclose the water coherent domains as shown in [7].

4.3 A Possible Architecture of Autoadaptive Network of Coherent Domains for Quantum Computation

The theoretical model previously shown could be used to realize a physical network able to perform quantum computations by using interacting wave domains in an autoadaptive way.

A possible design of such a network would consist in an ensemble of coherent domains arranged according to a geometrical configuration able to maximize their mutual interaction. If we assume, for simplicity, the coherent domains to have spherical symmetry, this requires to achieve the minimum value of their mutual intercenter distance, namely the close-packing of N spheres of radius $r = R_{coh}$. This can be realized according to either the so-called “face-centered cubic” (FCC) or “hexagonal close-packed” (HCP), consisting of sheets of spheres arranged at the vertices of a triangular tiling differing in the way the sheets are stacked upon one another. In both the cases every sphere is surrounded by 12 other spheres at distance (between centers) of $d = 2R_{coh}$ with an average density of spheres given by

$$\frac{\pi}{3\sqrt{2}} \simeq 0.74048 \quad (64)$$

so that we can estimate the average amount of information in a given volume of the network through Eq. (59) with

$$n_{CD} = \frac{\pi}{3\sqrt{2}} V \quad (65)$$

where V is the volume of the network. A given node of such network will be then constituted by one or more coherent domains characterized by a given macroscopic wavefunction. As we have seen, the coherent domains (and, consequently, the network nodes) can be put in oscillation by supplying energy from the outside and let them interact with each other. The initial state of each node can be defined by precisely setting the values of the macroscopic quantum phases belonging to the coherent domains including it, by using the method suggested, for example, in [23], and the correspondence given by Eq. (62). The final configuration of the network, reached after the quantum dynamical evolution has taken place and the synchronized state of coherent domains has been established, gives us the output of the computation and its measure gives the quantum information stored in it. In this connection, it can be noted also the network topology could represent, within a more sophisticated model, a key feature to consider since it could generally affect the final configuration of the network that represents the calculation result itself due to the network space-time evolution. One of the most interesting properties of such a network is its ability to merge classical features, like those typical of classical neurocomputation (implemented in artificial neural networks), with quantum features like quantum computation (or, better, quantum hypercomputation) and quantum information capabilities, being composed by coherent domains that are macroscopic objects (like classical systems) but described by quantum wavefunction. For this reason, the suggested network

composed by quantum coherent domains could be able to realize advanced functions as, for example, pattern recognition, quantum associative memory, etc., making it suitable for the realization of powerful novel quantum neural networks (QNN) with important and revolutionary applications even in the field of artificial intelligence as will be further discussed in forthcoming publications.

5 Outlook and Conclusion

In this paper we have analyzed the collective dynamics of an ensemble of macroscopic quantum objects constituted by the so-called “coherent domains” of water, namely spatial domains, predicted by the theory of QED coherence in matter, in which water molecules oscillate in tune with a self-trapped electromagnetic field. Every coherent domain is described by a macroscopic wavefunction characterized by well-defined amplitude and phase. Furthermore, in the case of water, such domains admit excited energy levels in the form of coherent vortices of an ensemble of quasi-free electrons that can be also considered as a plasma, able to perform coherent oscillations as well. A single coherent domain can be then considered as a self-sustained oscillator whose oscillation frequency depends on the coherent modes of oscillations it allows for. When two or more coherent domains interact to each other they can modify their own oscillation frequencies as occurs in the case of ordinary self-sustained oscillators.

In particular, the coherent plasmas belonging to the interacting coherent domains can adjust their oscillations in order to become synchronized. In this paper we have then shown that, if we consider a large ensemble of interacting coherent domains, provided that the interaction strength is greater than a critical value, they undergo a second-order phase transition toward a state, named “supercoherent”, characterized by the formation of a cluster of synchronized coherent domains entrained by a “mean” field of interaction due to all the synchronized oscillators. Water coherent domains belonging to the cluster are phase locked with the mean field and their phase structure depends on the frequency distribution of the oscillators in the uncorrelated initial state as well as on the form of interaction and initial values of the phases.

The number of oscillators entrained by the mean field and belonging to the synchronized cluster depends on the coherent domains interaction strength and defines the information that can be stored in the cluster itself. This information is quantum in principle since it is codified in the quantum phases associated to each water coherent domain. In the supercoherent state the information is then stored in the phase distribution of the locked oscillators and the evolution of the system towards the synchronized state is driven by the self-organization dynamics of the interacting coherent domains according to the interaction features as well as the initial conditions (about uncorrelated oscillations frequencies and phases). Then, by suitably adjusting the related parameters, we can let the system to perform an “intelligent” computation (also quantum in nature), through its self-organizing dynamics (the more synchronized are the water domains, the more they are able to entrain other not yet synchronized oscillators and vice

versa), and store its result in the phase structure of the supercoherent state. A network of interacting water coherent domain can be then considered as an auto-adaptive macroscopic quantum network similar to a neural network able to perform quantum computation and store quantum information.

We should emphasize that the proposed model is still quite generic at this stage since requires a closer analysis of several theoretical and operational properties such as, for example, the consideration of a space-time varying interaction strength between coherent domains, different interaction functions other than sinusoidal ones, multistability and multi-cluster configuration of the supercoherent state, stability of the synchronized cluster and so on.

Despite the proposed model should be considered to be in a preliminary stage, nevertheless it shows how, at least in principle, an ensemble of interacting water coherent domains predicted by the theory of QED coherence in matter could be used to simulate a quantum self-adapting network able to perform “intelligent” quantum computation and the storage of quantum information. Our proposal could represent a first step towards the realization of quantum neural network, based on the use of water molecules, as well as a model able to give very interesting and novel insights in the understanding of biological neural networks, in which water, as known, represents a key element.

References

1. Preparata, G.: *QED Coherence in Matter*. World Scientific, Singapore (1995)
2. Del Giudice, E., Preparata, G.: A new QED picture of water: understanding a few fascinating phenomenon. In: Sassaroli, E., Strivastava, Y., et al. (eds.) *Macroscopic Quantum Coherence*, pp. 108–129. World Scientific, Singapore (1998)
3. Arani, R., Bono, I., Del Giudice, E., Preparata, G.: QED coherence and the thermodynamics of the water. *Int. J. Modern Phys. B* **9**, 1813–1841 (1995)
4. Del Giudice, E., Tedeschi, A.: Water and autocatalysis in living matter. *Electromagn. Biol. Med.* **28**, 46–52 (2009)
5. Del Giudice, E., Spinetti, P.R., Tedeschi, A.: Water dynamics at the root of metamorphosis in living organisms. *Water* **2**, 566–586 (2010)
6. Bono, I., Del Giudice, E., Gamberale, L., Henry, M.: Emergence of the coherent structure of liquid water. *Water* **4**, 510–532 (2012)
7. Caligiuri, L.M.: Quantum (Hyper)Computation by means of water coherent domains part I: the physical level. In: Caligiuri, L.M. (ed.) *Frontiers in Quantum Computing*, pp. 1–37. NOVA Science Publisher, New York (2020)
8. Caligiuri, L.M.: QED coherence and super-coherence of water in brain microtubules and quantum hypercomputation. In: Bandyopadhyay, A., Ray, K. (eds.) *Rhythmic Advantages in Big Data and Machine Learning. Studies in Rhythm Engineering*, pp. 225–262, Springer, Singapore (2022)
9. Teo, T.W., Choy, B.H.: in. In: Tan, O.S., Low, E.L., Tay, E.G., Yan, Y.K. (eds.) *Singapore Math and Science Education Innovation. ETLPSSIP*, vol. 1, pp. 43–59. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-1357-9_3
10. Caligiuri, L.M.: Quantum (hyper)computation by means of water coherent domains part II: the computational level. In: Caligiuri, L.M. (ed.) *Frontiers in Quantum Computing*, pp. 57–102. NOVA Science Publisher, New York (2020)

11. Caligiuri L.M.: Quantum (hyper)computation through universal quantum gates in water coherent domains. *J. Phys.: Conf. Ser.* **2162**, 012003 (2022)
12. Pollack, G.H., Clegg, J.: Unexpected linkage between unstirred layers, exclusion zones and water. In: Pollack, G.H., Chin, W.C. (eds.) *Phase, Transitions in Cell Biology*, pp. 143–152. Springer, Berlin (2008)
13. Buzzacchi, M., Del Giudice, E., Preparata, G.: Coherence of the glassy state. *Int. J. Mod. Phys. B* **16**(25), 3771–3786 (2001)
14. Caligiuri L.M.: QED coherence in matter, syntropy and the coherent domains as storing devices. *J. Phys.: Conf. Ser.* **2197**, 012004 (2022)
15. Caligiuri, L.M.: Coherence domains in condensed matter as storage “Devices” of quantum information. In: Arai, K., (ed.) *Proceedings of the Future Technologies Conference (FTC) 2022*, vol. 3. *FTC 2022. Lecture Notes in Networks and Systems*, vol. 561. Springer, Cham (2023)
16. Caligiuri, L.M.: The quantum phase operator and its role in quantum computing. In: Caligiuri, L.M. (ed.) *Frontiers in Quantum Computing*, pp. 39–56. NOVA Science Publisher, New York (2020)
17. Caligiuri, L.M.: Quantum computation by means of josephson junctions made of coherent domains of liquid water. In: Arai, K. (ed.) *Intelligent Computing. SAI 2022. Lecture Notes in Networks and Systems*, vol. 506. Springer, Cham (2022)
18. Pikovsky, A., Rosenblum, M., Kurths, J.: *Synchronization*. Cambridge University Press, Cambridge (2010)
19. Boccaletti, S., Pisarchik, A.N., Del Genio, C.I., Amann, A.: *Synchronization*. Cambridge University Press, Cambridge (2018)
20. Rodrigues, F.A., Peron, T.K.D., Ji, P., Kurths, J.: The Kuramoto model in complex networks. *Phys. Rep.* **610**, 1–98 (2016)
21. Seliger, P., Young, S.C., Tsmiring, L.S.: Plasticity and learning in a network of coupled phase oscillators. *Phys. Rev. E* **65**, 041906 (2002)
22. Okuda, K.: Variety and generality of clustering in globally coupled oscillators. *Phys. D* **63**(3–4), 424–436 (1993)
23. Caligiuri L.M.: Fast and accurate control of gates for quantum hypercomputation in coherent domains of water. *J. Phys.: Conf. Ser.* **2162**, 012025 (2022)



HyMO-RF: Automatic Hyperparameter Tuning for Energy Theft Detection Based on Random Forest Classification

Francisco J. S. Coelho^{1(✉)}, André L. M. Alcântara², Allan R. S. Feitosa¹, Jessica T. Takeuchi³, Ronaldo F. Lima³, and Abel G. Silva-Filho¹

¹ Federal University of Pernambuco - Informatics Center, Recife, PE 50740-560, Brazil

fjsc@cin.ufpe.br

² Eldorado Research Institute, Campinas, SP 13083-898, Brazil

³ CPFL ENERGIA, Campinas, SP 13.087-397, Brazil

Abstract. To curb energy theft and identify fraudsters and other non-technical losses, power distribution companies have used Machine Learning algorithms and a large amount of data with high granularity from electricity consumption units. Those data are collected from consumers through Advanced Metering Infrastructure (AMI) like Smart Meters (SM) being remotely collected in real time and several samples per day. In emerging countries like Brazil, most energy meters are technologically limited or electromechanical. Those devices can measure only aggregated values of energy consumption in a monthly basis. This work proposes HyMO-RF, one strategy of using a multi-objective search algorithm to improve the performance of a machine-learning model in the detection energy theft in a scenario of limited resources (AMI and SM). The proposed approach in this paper (HyMO-RF) is based on hyperparameter tuning and the main contribution is to associate multi-objective algorithms to determine an optimized combination of hyperparameters in order to maximize the classification model's performance. A real life corporate dataset was provided by a Brazilian power distribution company CPFL ENERGIA™. The data was properly anonymized and securely stored. We used NSGAII multi-objective algorithm for hyperparameters tuning, improving the Random Forest (RF) classifier's performance. Results achieved in terms of Precision and F1-score metrics were 0.83 and 0.73 respectively. An additional field study showed that the solution proposed already impacted the operation of the fraud detection team in a positive way by having an accuracy of 74% of the suspect consumer units inspected.

Keywords: Energy theft · Non-technical losses · Hyperparameter tuning · Machine learning · Multi-objective algorithm

1 Introduction

Electrical system losses are divided into two types: technical losses and non-technical losses. Technical losses involve physical effects such as the joule effect and dielectric losses, among others. Non-technical losses are caused by faulty energy meters, measurement errors, or energy theft [1].

According to [2], energy theft costs the energy industry as much as \$96 billion yearly. Brazil, for example, has more than 80 million consumer units, and in 2021 had an yearly cost of approximately BRL 6.5 billion due to energy theft [3]. In addition to economic losses, energy theft represents a risk to public safety, as it alters the characteristics of the electrical network and is one of the biggest causes of fires and deaths from electrical discharges. According to [4], almost 40% of fires and 28% of electric shock victims in China are linked to electricity theft.

Modern techniques for detecting suspicious consumption patterns require high granularity data collected weekly, daily, hourly, or even more frequent sampling rates [4]. This data is organized into consumption profiles and later classified by Machine Learning (ML) models. However, only Advanced Metering Infrastructure (AMI) can provide consumption data with high sampling rates. The AMI enables two-way communication between utility companies and consumers, allowing a wide application of Smart Meters (SM) and frequent measurement of energy usage [5].

The SM can measure active and reactive power, voltage levels, current, power factor, and consumption interruption, among other important variables for assemble of consumption patterns. However, this type of device is only a reality in some parts of the world. In emerging countries like Brazil, most electricity meters are technologically limited or electromechanical types. In other words, they only measure aggregated consumption in kilowatt-hours (kWh). The consumption reading process is done manually by an employee of power distribution company every 30 days.

In this scenario, the primary technique to identify fraudulent customers is on-site inspection of energy meters. However, this approach is very costly from a financial and human point of view, given the need to travel with trained professionals for in site assessment. Furthermore, it is a technique with low assertiveness. Generally, inspections based on denunciation have better assertiveness. However, the interval between new denunciations may be substantial compared to the number of inspections that could be carried out in the same period.

Another well-known strategy is the identification of a consumption drop, characterized by the partial or total reduction of the consumption of a consumer unit. The main advantage of this strategy is that the identification of the abnormal pattern of the consumer unit does not require, *a priori*, field inspection teams.

Methods like those mentioned above have fallen into disuse. On the other hand, approaches based on machine learning (ML) have gained ground given that they only require scanning databases with consumption history to infer profiles of fraudulent consumer unit. This strategy has shown good assertiveness and lower cost when compared to the exhaustive search strategy or incursion

motivated by suspicions and gut feeling. On the other hand, for these algorithms to achieve good performance, data with enough quantity and quality with different consumption profiles is required.

Most of the works found in literature (see Related works section below) propose such theft detection in robust data collected with advanced infrastructure. There is a gap of approaches proposed by simple metering infrastructure as the ones present in not developed countries.

State-of-the-art shows that the training process of these classifiers is usually based on public datasets from AMI or SM. An observed constraint of public datasets is that fraudsters' consumption patterns are rare and often unlabeled. In this sense, the authors started from the following research question: Is it possible to also detect energy theft based on data collected without advanced metering infrastructure by leveraging machine learning?

To answer this question and find a solution that allows achieving relevant results for the power distribution company, the authors based themselves on the following hypothesis: the hyperparameter optimization, in order to reach the right combination of hyperparameters will maximize the classification model performance.

The aim of this paper is to investigate whether hyperparameter tuning using a multi-objective algorithms approach improves the identification of energy theft for a power distribution company. In a nutshell, the main contributions of this work are:

- Our method focuses on a scenario without AMI and SM, where the consumption patterns are composed of the monthly accumulated in kWh;
- Different from related works, we used a multiobjective optimization approach in hyperparameter tuning of classification algorithms;
- Our approach allows the choice of different non-Pareto-dominated solutions to adjustment of model behavior concerning the performance metrics and constraints of the addressed problem.

The data used in this work were provided by Paulista Power and Light Company (CPFL ENERGIA), a power distribution company which operates in 687 Brazilian cities and has around 9.6 million customers.

This work is organized as follows: Sect. 2 presents a literature review with related works. Section 3 describes our methodology and details experiments. The results obtained are discussed in Sect. 4. After that, we present the conclusions of our work in Sect. 5.

2 Related Works

Smart Meter (SM) is only a reality in some parts of the world. In emerging countries like Brazil most electricity meters are technologically limited. For these cases, the meters are electromechanical, and the accumulated energy consumption is measured only once by a company employee every 30 days. Currently,

costs related to energy theft (non-technical losses) represents 15% in Brazil, and still remain a major villain in energy distribution by companies.

Even facing this reality, recent works implement machine learning models to detect energy theft using datasets with high sampling rates from smart meters. However, these datasets generally do not represent the reality of the consumer unit where the model will be applied.

In [6], the authors used the Irish Smart Energy Trial (ISET) [7]. This dataset was created from consumption samples taken from over 5000 Irish consumers. They proposed an approach based on an adaptive neuro fuzzy inference system classifier to detect energy theft. However, the used dataset provides consumption data with a sampling rate of 30 minutes and authors generate synthetic data to simulate unusual electricity consumption patterns.

The authors of [5] also used the ISET. In this work, a deep-learning-based model was proposed (MFEFD) aiming detecting electricity theft considering an advanced metering infrastructure. Despite the results showing good assertiveness, the performance of the models decreases with data whose sampling is greater than 2 h.

In [8], the authors used a dataset provided by the Spanish electric utilities Endesa, with consumption data from smart meters installed in industries and commercial customers. The consumption samples have sampling rate of 15 min. However, the authors used only five samples per day due to privacy reasons that can arise with such large granularity. The authors focused on more simplistic algorithms, such as K-Nearest Neighbors (KNN), Logistic Regression (LR), Support Vector Machines (SVM), and Extreme Gradient Boosted Trees (XGBoost). This approach was implemented in field using the XGBoost classifier for training and it currently reaches a precision of 21% in terms of new on-field inspections approximately.

The authors of [4] applied TextCNN to the problem addressed. This classifier is widely used in Natural Language Processing (NLP), text classification, and emotion analysis. Based on Convolutional Neural Network (CNN), TextCNN has good sensitivity and a simple structure. In training the model, two datasets from a Chinese province were used, with labeled data collected by SM and the ISET. The last was used to create synthetic data.

Data from Spanish Endesa customers was used by [9] to improve the detection of non-technical losses linked to consumer unit with consumption close to zero. The authors assembled a dataset using contractual information from customers, such as type of tariff, economic activity, billing period, postal code, contracted power, and comments made in inspections. The data were extracted from 101,215 consumers connected to smart meter, with at least 24 consumption samples. The algorithm applied to the problem combined NLP, Classification and Regression Tree (C&R), and a Self-Organizing Map (SOM). This approach raised the success rate of Endesa's inspections from 5 to 14.75%.

The consumption history is interpreted as a time series in work developed in [10]. The authors created an algorithm that combines a Convolutional Neural Network (CNN) and a Long short-term memory (LSTM). A dataset from the

State Grid Corporation of China was used to train the model, composed of consumption samples collected via SM. The missing dataset values were inputted based on historical consumption averages. To balance the dataset, the authors generated synthetic data. The results show that the CNN-LSTM architecture may have reduced accuracy when using synthetic data in model training. Results were compared to SVM and logistic regression and an overall 89% classification accuracy was achieved.

In [11] used consumption data from Brazilians consumer unit from the metropolitan region of Minas Gerais. This work considered contractual information, location, voltage, inspection notes, and other data, totaling 14 features. The sample consumption data of the unit consumer analyzed was limited to a column with an average of the last 12 months. In addition, the authors included information about the location and debts due to non-payment of electricity bills. However, this data type has a potential to harmfully bias the classifier since neighborhoods with lower-income people, such as slums and peripheries, tend to have higher occurrence of fraud and cuts in supply due to non-payment. The authors trained an Artificial Neural Network Multilayer perceptron (ANN-MLP). The classifier achieved an accuracy of 65.03%.

In [12], a scheme called LFPR-DNN is proposed. The model is organized into two parts. In the first stage, a one-dimensional Convolutional Neural Network, 1DCNN extracts feature from the consumption samples. In the second stage, a neural network fully connected was used as a binary classifier. To improve the model's performance, the authors used a mono-objective optimization method to adjust the classifier architecture.

The approach of [13] uses the ISET to train shallow models. The metrics used to evaluate the models were Precision, False Positive Rate, and AUC. The authors generate synthetic data to simulate six consumption patterns representing non-technical losses.

The authors of [14] integrate the benefits of AlexNet and AdaBoost. The AlexNet network extracts features from the consumption curves while dealing with the dimensionality curse problem, and AdaBoost is used in the classification step. To balance the dataset, a subsampling method was applied. One approach To improve model performance optimization, the authors applied a mono-objective optimization method to hyper-tuning in AdaBoost.

Some of the related works use public datasets and models based on deep learning. However, these datasets may not have labeled samples, in addition to a reduced number of records, as is the case of the Irish Smart Energy Trial, used in [4,6,13]. Regarding the algorithm, Deep Learning models can demand a large amount of data and a high computational cost in the training of models. Mono-objective optimization algorithms like those used in [12,14] can lead to misinterpretation of metrics such F1-Score, because it can camouflage low values of metrics used in his equation, like Precision or Recall.

Compared to previous approaches, our work focuses on training models with a dataset with accumulated monthly consumption samples of electromechanical meters. Synthetic samples will not be used in the training of classifiers. Instead,

optimization algorithms will be applied to hyperparameter tuning of the algorithm to improve the accuracy of the classifier. For bias mitigation, attributes that refer to the geographic location of consumer unit or debt history will not be considered.

3 Methodology

In this section, we introduce how the data provided by CPFL ENERGIATM was preprocessed, how the proposed technique was implemented and compared with a baseline version, and how optimization algorithms were used for hyperparameter tuning.

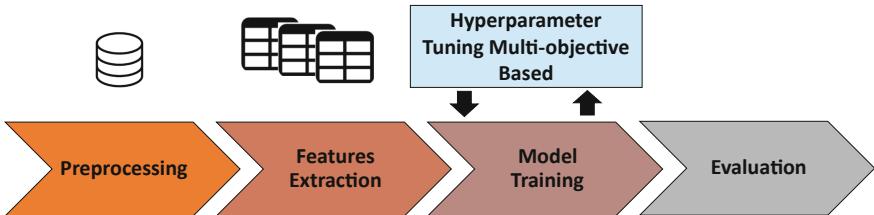


Fig. 1. Proposed methodology.

The following sections describe the steps of Fig. 1.

3.1 Preprocessing

Dataset preprocessing consists of translating raw data into usable information. Generally, this method includes steps like data cleaning, identifying and correcting inaccuracies, transformation, normalization, and converting data into a format suitable to machine learning algorithms. The preprocessing steps applied in the data are detailed below:

Dataset: The dataset was provided by CPFL ENERGIATM from their historic database of electricity usage from more than 1.2 million of consumer units containing more than 15.2 million samples of monthly consumption measurements in kilowatt hour (KWh). Those data were provided together with more than 358 thousand of fraud inspection reports from those consumer units. The original dataset contains information about contracts, consumption history and inspections carried out in those consumer units together with the information whether a fraud was detected or not.

Missing Data: The concept of incomplete data used was records that did not have data from one of three big blocks: location, consumption or inspection. Given that location and consumption were used to generate the features and

inspection was used to generate the supervised labels. Therefore, after removing incomplete records there were 309,780 viable monthly samples left in the database.

Labeling Process: The labeling process was performed by taking the inspection reports from the month of the sample and attributing positive (fraud) in case of reported detection during the inspection. As there is no guarantee that all the units were inspected in all the months, there was a special focus on the precision of the model, to make sure that the positive values were indeed true positives. Correct identification of positives generated a business gain already (by decreasing unnecessary inspections). At the end of labeling process, from the total of 309,780 samples, 12,2% were labeled as 1 positive, what can be taken as an imbalanced dataset relatively to the target variable.

The inspections detect several different situations that are not necessarily frauds. Therefore, only the ones that are from the fraud group were used to label samples. Those type of inspections revealed: clandestine connection to the grid, clandestine wiring to by-pass the consumption recorder device (parallel connection), wires broken after the consumption recorder device, jumper from the output of the recorder to its own input, intentionally damaged recording device, device with coil stopped, device with disk stopped, device with internal mechanisms changed, device with changes in internal circuit, and more.

Features Extraction: Feature extraction transforms raw data into specific formats and adequated to be used as inputs to an inference model. The features selected from the database are shown in Table 1.

Table 1. Features selected from the database.

Variable	Description
Voltage	Voltage range category
Billing type	Customer's billing category
Reading motif	If and why needed an especial reading
Calculation class	Type of calculation for billing
Sector	When commercial what is the sector
Type of installation	Number of phases

The numerical values containing information about the consumption history and categorical values regarding local information of the consumer units like type of connection and others.

Besides to capture the patterns of consumption from the ECUs along time, a strategy was used to generate features from the consumption curves based on univariate statistical indicators from the whole consumption time series of the client.

As consumption measurement is done in a monthly basis, there is the need to aggregate this data to be used as model's input. In [6] mean, median, entropy, skewness, standard deviation, kurtosis, variance, energy, and load factor were extracted from a consumer unit's consumption data. Inspired on this approach, we applied univariate descriptive statistics metrics to extract features from the consumption curves of each consumer unit in the dataset. Table 2 shows them.

Table 2. Univariate metrics extracted from the consumption historic time series.

Meta-variables
Median
Variance
Deviation
Coefficient
Min
Max
Kurtosis
Skewness
MAD
Mode
Count

Table 2 shows the metrics extracted from both registered and charged consumption time series to represent the usage behavior of the user along the months. The objective of it is to generate features that represent the shape of the consumption curves.

3.2 Model Training

More recent methods for detecting energy theft are based on robust Machine Learning techniques such as Deep Learning (DL). Such training often requires larger amounts of data and computational resources to achieve good performance levels. Therefore, due to the fact that large amount of data for detection of energy theft can be hard to find [4] we decided to implement one classical machine learning approach.

Our classification strategy is based on Random Forest (RF), a classical machine learning approach. RF is based on bagging strategy and decision trees and has relatively lower computational costs when compared to deep learning methods. Other advantages of RF are: no sensibility to imbalanced datasets and presents good assertiveness, as demonstrated by [6]. Despite finding non-linear patterns in data, the classifier trained by the RF can be transparent regarding the strategy used in classifying the instances, this is due to the decision trees's decision process based on rules.

To quantify the performance and validate our proposal, Random Forest training was performed in two stages. In the first one, Canonical Random Forest [16] was tested to predict the fraudulent consumer units, and in the second, we added a hyperparameter tuning step using multi-objective optimization.

Canonical Random Forest (RF) Random forest is a machine learning algorithm used for classification and regression tasks. It operates by constructing multiple decision trees, as shown in Fig. 2, each tree is grown from a bootstrapped sample of the training data. Finally, the prediction of the trees is combined through majority voting or averaging.

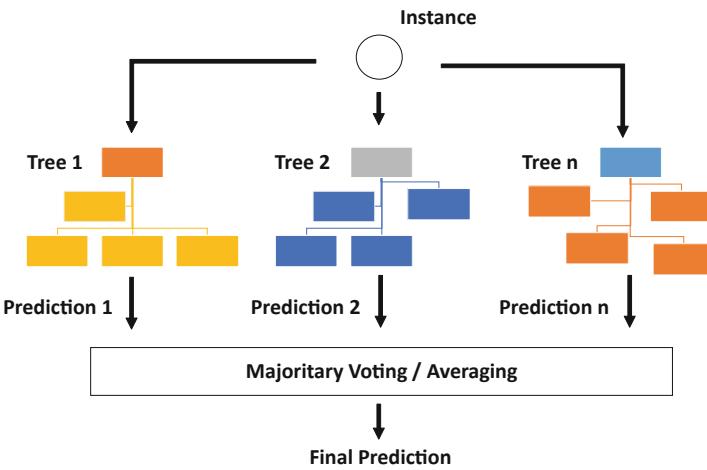


Fig. 2. Representation of a random forest classifier.

This strategy of combining multiple estimators (decision trees in case of RF) is called Ensemble. This method leverages multiple estimators' strengths while mitigating their weaknesses, resulting in improved predictive performance.

Hyper Multi-objective Random Forest Classifier (HyMO-RF): The proposed approach focuses on improving model performance by using search and optimization in the hyperparameter tuning process. This method aims to adapt the behavior of the classifier to a specific set of data. Hyperparameters are values that cannot be directly estimated from data learning and commonly set before training. Grid search and random search are popular algorithms to do such search. However, they require higher computational and time cost, due to the fact that they exhaustively go through an arbitrary set of possibilities in the hyperparameters space.

Evolutionary multi-objective optimization algorithms (MOO) can be an alternative method to do a more efficient search in the hyperparameters space.

These algorithms are inspired by Charles Darwin's theory of natural evolution and can provide near-optimal solutions. One way to evaluate their solutions is using the Pareto dominance concept and the Pareto front generated at the end of the iterations. The solutions from the resulting Pareto front represent balancing between the concurrent objective functions that MOO must maximize or minimize. These algorithms can even handle constrained relationships among these objectives.

Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [15] is a MOO strategy with recognized effectiveness in optimizing problems in engineering, finance, and others. They work by combining Genetic Algorithms and non-dominated sorting. The former is an evolutionary single objective algorithm, and the later is a technique for ranking solutions based on their level of Pareto domination by other solutions. The algorithm implements operators inspired by what happen with chromosomes in genetic, such as mutation, crossover, and selection. Those operations along the iterations reveal the best solutions that maximize or minimize one specific objective function.

Given the need to search for hyperparameter in more efficient ways, we propose the combination of NSGA-II and RF algorithms for automatic hyperparameters search. NSGA-II is responsible for finding combinations of hyperparameters that result in better Random Forest classification performance. The strategy of using NSGA II to tune Random Forest's hyperparameters is named Hyper Multi-objective Random Forest Classifier (HyMO-RF). Figure 3 shows the diagram of HyMO-RF.

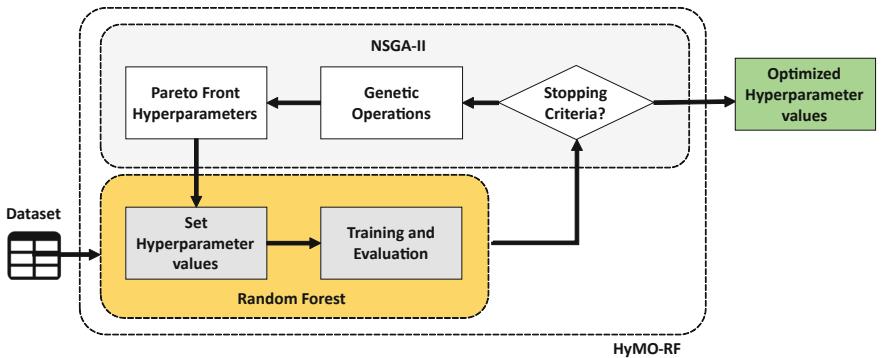


Fig. 3. Diagram of HyMO-RF.

The composition of a solution candidate is a set of Random Forest hyperparameters. Table 3 shows those parameters. The objective functions to calculate the fitness of the solution candidates are F1 score and Precision of RF using the hyperparameters from the solution candidate described in Eqs. 2 and 3. Therefore, both are to be maximized. More details about the choice for those metrics can be found in the next section.

The input parameters of HyMO-RF are the dataset and the list of HPs. The algorithm was implemented to accept integer, categorical, or floating point HPs. In addition to these parameters, the user can inform the size of the population and the maximum iterations. However, these configurations are defined by default. Based on the input data, the algorithm generates its initial population, composed of HPs, trains and evaluates an instance of Random Forest for each individual in the population. The performance evaluation of each model is based on F1-Score and Precision metrics. These values are used to define the fitness of HPs (individuals in the population) and for the sort of individuals.

The HyMO-RF then checks if the stop criteria has been reached. If not, it makes the current population of individuals go through the process of mutation, crossover, and all the logic of evolution and selection of the best individuals inherent to the NSGA-II mechanisms. In this way, the best individuals are preserved, and new ones are added to the population of candidate solutions. The Pareto Front with a new population is then decoded as different sets of hyperparameters, and the process is repeated until the stop criteria is reached.

Table 3. Set of hyperparameters used in tuning.

Hyperparameters	Type
n_estimators	Integer
Criterion	Categorical
max_depth	Integer
max_features	Categorical
oob_score	Binary

3.3 Evaluation

The evaluation metrics used as both objective functions of NSGA II and performance comparison of the classifiers were: Precision and F1-Score. F1-Score evaluates the overall performance of a model. It is the harmonic mean between Recall and Precision [10]. Recall can be defined as the proportion of classified positive results in the actual positive samples [4]. From the point of view of an electric utility, the Recall score represents the share of lost revenue due to illegal actions that are recoverable by on-site inspections [17]. According [10], F1-Score is beneficial in cases where the dataset is unbalanced. Recall, Precision and F1-Score metrics are defined as:

$$Recall = \frac{TruePositive(TP)}{TruePositive(TP) + TrueNegative(TN)} \quad (1)$$

$$Precision = \frac{TruePositive(TP)}{TruePositive(TP) + FalsePositive(FP)} \quad (2)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (3)$$

Precision metric gauges the classification accuracy in the classified positive samples [4]. Prioritizing this metric is essential when a false positive can generate high costs, such as sending inspection teams to consumer unit that are not defrauding. Maximizing the Precision scores translates to increased revenue recovery with minimized inspection-related costs [17].

Receiver operating characteristic (ROC) curve was used to evaluate the performance of both canonical Random forest and HyMO-RF in terms of sensitivity and specificity. Area under curve was also calculated for both approaches.

4 Results and Discussion

The experiments were carried out in virtual machines from AzureTM cloud environment and implemented using Python programming language, with the libraries Scikit-learn [11], Pandas and Matplotlib.

4.1 Preliminary Experiments

Preliminary experiments were carried out to determine the optimal population size NSGA-II. The evaluation was done by F1 score and precision of the attached Random forest under five different population sizes (10, 20, 30, 40 and 50). For those experiments, a stopping criteria was defined as ≥ 50 generations.

Figure 4 shows the evolution of Random Forest F1 Score along NSGA II generations with different population sizes. Given those results it was possible to define 30 as the population size and 30 generations as the number of iterations. This is the minimal population size that reaches the best F1 score at the same time as the others.

The convergence of the optimization model tends to increase the quality of the classifier along the generations. This is the evidence that the optimization model can maximize the function by finding better combinations of hyperparameters along its execution.

A Pareto front resulting from NSGA II is shown in Fig. 5. The solution marked with a red cross would be the one that maximizes precision, to enforce the quality of the positives and avoid sending teams for inspections that are more likely to be false positives, and keeps an F1 Score level around 73%.

The choice for the target metrics and for one of the solutions from the Pareto front may vary according to the specific application of the algorithm and what are acceptable levels of False positive or False Negative tolerance. The Pareto front solutions resulting after 30 generations allow the user to chose among different solutions with different values of precision and recall.

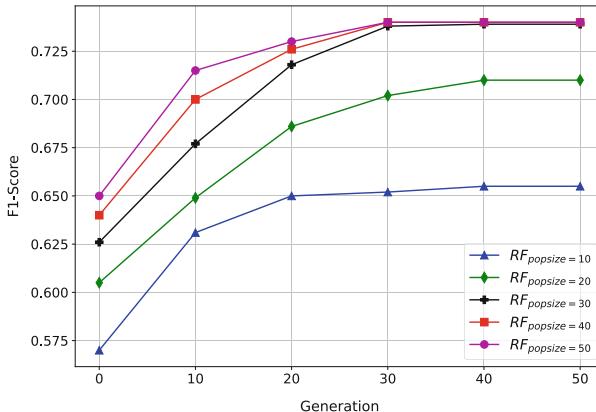


Fig. 4. Evolution in terms of metric F1-score as a function of the numbers of generations for different population sizes to RF algorithm.

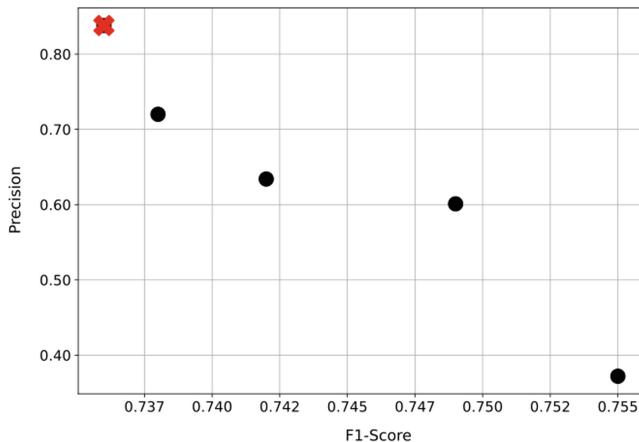


Fig. 5. Pareto front of F1-score versus precision metrics, considering 30 generations and a population size equal to 30.

4.2 Hyperparameters Tuning (HyMO-RF)

The scikit-learn library was developed specifically for the practical application of machine learning and brought a series of algorithms along with their respective hyperparameter configuration possibilities. In this library, Random forest currently has 18 adjustable hyperparameters. In this first version of our proposal, we focus on the most influencing 5 hyperparameters listed in Table 3.

Table 4 shows the performance comparison between the canonical Random Forest approach and HyMO-RF. There is a difference of approximately 10% of F1 score. This difference shows the importance of investigating and optimizing the hyperparameters used when training a Random Forest algorithm.

Table 4. Performance comparison between canonical and HyMO-RF.

Model	F1 score (%)	Precision (%)	Recall (%)	AUC
Canonical random forest	63,1	71,9	56,2	0.70
HyMO-RF	73,6	83,8	65,6	0.75

Table 5 show the optimized set of hyperparameters discovered by HyMO-RF for the task of fraud detection in consumer units.

Table 5. Optimized set of hyperparameters found by HyMO-RF.

Hyperparameters	Value
Estimators	528
Criterion	Entropy
Max depth	1195
Max features	Sqrt
OOB score	False

Figure 6 shows the generated ROC curve from both approaches. It is possible to notice that the superiority of HyMO-RF goes along the biggest part of the score variation, suggesting that the proposed model has superior sensitivity and specificity in different cutoff points.

Given the results shown in Fig. 6 it is possible to notice that using the optimization model to tune the hyperparameter can surface better results than using the library's default parameters. Manual adjust of the parameters was not evaluated because it can be done in different ways based on the experience of the person who is training the algorithm.

4.3 Field Validation

To validate our approach, the CPFL ENERGIA™ used the final fraud detection model to point consumer units suspected of committing fraud in two randomly chosen cities. The cutoff point for selecting those units was 95%. In other words only consumer units with a score bigger than 95% were selected for inspection. From the list of selected consumer units, 594 were randomly chosen. Three inspection teams were assigned to carry out field investigations. From this amount, only 311 inspections were realized, and from those realized, 231 consumer were caught committing energy theft. That is, our approach achieved an accuracy of 74% in the field test.

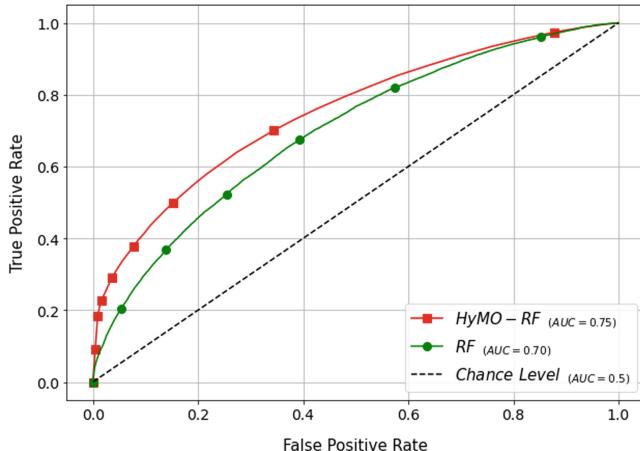


Fig. 6. Receiver operating characteristic curve of both canonical and HyMO-RF.

5 Conclusion

This work investigated the application of a multi-objective approach to improve the performance of a machine-learning model in the task of detecting energy theft. Our approach was based on a scenario without Advanced Metering Infrastructure, where the consumption data are composed of monthly samples from energy meters technologically limited. Based on the scenario mentioned above, a dataset was created with billing data from CPFL ENERGIA™ consumers. The algorithm NSGA-II was used to provide an optimal combination of hyperparameters to algorithm RF, leading the classifier generated by this combination to achieve 0.73 and 0.83 of F1-Score and Precision metrics, respectively. The alternatives to hyperparameters adjustment are using the default parameters from the libraries and manual hyperparameter tuning. The proposed approach has potential to be an automated of hyperparameter tuning. Our approach's main limitations are: that we focus on a specific scenario where meters are technologically limited, our pre-processing data methodology does not include techniques that allow recovering missing data, and multi-objective optimization methods are computationally more expensive than single-objective approaches.

A field study showed that the solution proposed already impacted the operation of the fraud detection team in a positive way by having an accuracy of 74% of the suspect consumer units inspected.

The study was able to show that using multiobjective optimization on top of classifiers can improve the quality of energy theft detection in the context of Brazilian consumers. In addition this can be performed for simple metering data, as in the dataset used.

Our approach's main limitations are: that our pre-processing data methodology does not include techniques that allow recovering missing data, and multi-objective optimization methods may be computationally more expensive than single-objective approaches.

Acknowledgment. The authors would like to thank CPFL Energia for technical and financial support, by means of the Research and Development project PD-00063-3080/2021, with resources from ANEEL's R&D program and cooperation with Eldorado Research Institute. They would also like to thank the CNPQ and FADE-UFPE.

References

1. Carr, D., Thomson, M.: Non-technical electricity losses. *Energies*. **15**, 2218 (2022)
2. Khan, I., Javaid, N., Taylor, C., Ma, X.: Robust data driven analysis for electricity theft attack-resilient power grid. *IEEE Trans. Power Syst.* **38**, 537–548 (2023)
3. Campos, A.: Agência Brasil Losses from fraud in Brazil added up to BRL 336 bi in 2021. <https://agenciabrasil.ebc.com.br/en/geral/noticia/2022-08/losses-fraud-brazil-added-brl-336-bi-2021>. Accessed 05 Dec. 2023
4. Feng, X., Hui, H., Liang, Z., Guo, W., Que, H., Feng, H., Yao, Y., Ye, C., Ding, Y.: A novel electricity theft detection scheme based on text convolutional neural networks. *Energies* **13**, 5758 (2020)
5. Hu, T., Guo, Q., Shen., Sun, H., Wu, R., Xi, H.: Utilizing unlabeled data to detect electricity fraud in AMI: a semisupervised deep learning approach. *IEEE Trans. Neural Netw. Learn. Syst.* (2019)
6. Blazakis, V., Kapetanakis, N., Stavrakakis, S.: Effective electricity theft detection in power distribution grids using an adaptive neuro fuzzy inference system. In: *Inference System. Energies*, MDPI (2020)
7. ISSDA.: Irish social science data archive. <https://www.ucd.ie/issda/data/> commisionforenergyregulationcer. Accessed 01 May 2023
8. Buzau, M., Tejedor-Aguilera, J., Cruz-Romero, P., Gomez-Exposito, A.: Detection of non-technical losses using smart meter data and supervised learning. *IEEE Trans. Smart Grid* **10**(3), 2661–2670 (New York) (2018)
9. Guerrero, I., Monedero, I., Biscarri, F., Biscarri, J., Millan, R., Leon C.: Non-technical losses reduction by improving the inspections accuracy. In a power utility. *IEEE Trans. Power Syst.* **33**, 1209–1218 (2018)
10. Hasan, M., Toma, N., Nahid, A.-A., Isla, M., Kim, J-M.: Electricity theft detection in smart grid systems: a CNN-LSTM based approach. *Energies* (2019)
11. Costa, C., Alberto, L., Portela, A., Madruo, W.: Fraud detection in electric power distribution networks using an ann-based knowledge-discovery process. *Int. J. Artif. Intell. Appl.* **4**, 17–23 (2013)
12. Gu, D., Gao, Y., Chen, K., Shi, J., Li, Y., Cao, Y.: Electricity theft detection in AMI with low false positive rate based on deep learning and evolutionary algorithm. *IEEE Trans. Power Syst.* **37**, 4568–4578 (2022)
13. Yan, Z., Wen, H.: Electricity theft detection base on extreme gradient boosting in AMI. *IEEE Trans. Instrum. Meas.* **70**, 1–9 (2021)
14. Ullah, A., Javaid, N., Asif, M., Javed, M., Yahaya, A.: AlexNet, AdaBoost and artificial bee colony based hybrid model for electricity theft detection in smart grids. *IEEE Access* **10**, 18681–18694 (2022)

15. Deb, K., Prat, A., Agarwal, S., Meyarivan, T.:A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolut. Comput.* **6**(2), 182–197 (Institute of Electrical and Electronics Engineers (IEEE)) (2002)
16. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B.: Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
17. Bhat, R. R., Trevizan, R. D., Sengupta, R., Li, X., Bretas, A.: Identifying nontechnical power loss via spatial and temporal deep learning. In: 15th IEEE International Conference (2016)



E-ELPV: Extended ELPV Dataset for Accurate Solar Cells Defect Classification

Marco Grisanti^(✉), Maria Ausilia Napoli Spatafora, Alessandro Ortis,
and Sebastiano Battiato

Department of Mathematics and Computer Science, University of Catania, Catania,
Italy

macro.Grisanti@phd.unict.it

Abstract. Solar modules are subject to a range of atmospheric events such as rain, wind, and snow and for this reason, they are usually built with protection frames. Nevertheless, these measures are insufficient to prevent damages, especially from the mechanical ones (e.g., the fall of tree branches) decreasing the power efficiency of solar modules. Then it is necessary to monitor their healthy conditions and replace or repair defective units. Electroluminescence, a useful inspection modality of solar modules, makes it possible to detect even the finest defects on the surface of solar modules. However, the analysis of these images is usually carried out by human operators, making this inspection practice expensive, time-consuming and it requires very specific knowledge. In state-of-the-art there are several works that distinguish between a healthy cell and defective cell, but a public dataset of possible defects in solar cells has never been published. For this reason, we propose a new dataset and a preliminary benchmark to make an automatic and accurate classification of defects in solar cells. The dataset includes five classes of defects and the pre-trained ResNext50 network reaches 0.07 Hamming Distance.

Keywords: Solar energy · Solar modules · Electroluminescence imaging · Convolutional neural networks · Deep learning · Visual inspection · Defect detection · Defect classification

1 Introduction

Luminescence is a physical phenomenon that consists in the emission of photons of light by materials excited by causes other than the increase in temperature [7]. It arises from the property of some materials to absorb certain quantities of energy, which are subsequently returned in the form of light photons. Electroluminescence is a particular type of luminescence that characterizes some materials capable of emitting light under the action of an electric field, or rather when crossed by an electric current [8].

The electroluminescence (EL) test applied to photovoltaic panels is based on the reverse process of photovoltaics: a voltage is applied to the modules to check

the current flows, while a camera with special sensors makes the infrared light emitted by the cells visible to the naked eye. Working cells will appear bright, while damaged ones will look dark [2]. The test is to be considered positive only for modules that demonstrate uniform current distribution. It is possible to run electroluminescence tests both in the laboratory and in the field. In the latter case, since the infrared radiation emitted by the sun is much higher than that of a solar cell, this type of test can only be performed at night. Generally, solar cell defects can be divided into two broad defect categories: intrinsic and extrinsic defects. Figure 1 shows an example of a cell extracted from an EL image of a photovoltaic module.

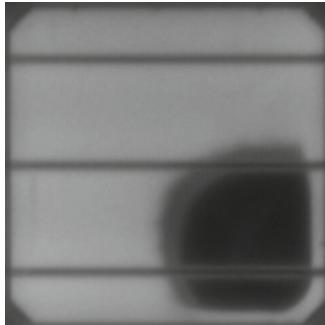


Fig. 1. The electroluminescence test applied to a photovoltaic panel cell. Note as the cell presents a dark area in the bottom-right part

There is an increasing interest towards the deep detection of defects in several industrial products (e.g. Sarpietro et al. [10] developed a deep pipeline for classification of defect patterns applied in Silicon technology). This interest motivated us to propose a new dataset and its benchmark for the classification of defects in solar cells. The rest of the paper is structured as follows: the Sect. 3 describes the proposed dataset and method; in Sect. 4 our results are reported. Finally, there are conclusions and future works in Sect. 5.

2 Related Works

Starting from 44 EL images of photovoltaic (PV) modules, which consisted in 18 monocrystalline modules and 26 polycrystalline modules, the work in [3] proposed a segmentation strategy in order to extract the various cells from the modules. By this process, the authors were able to extract 2624 cells. Subsequently, the authors of [4] dealt with the automatic classification of the various cells by a classifier reaching an average accuracy of 88.42% by considering a CNN based solution. This classification task aimed at indicating how likely a defect is present without specifying any details. The dataset (ELPV Dataset)

used for the classification of the cells with the associated labeling has been publicly released. Using the same dataset, but with a little different labeling, the work in [1] implemented an isolated CNN, that is not pre-trained, for the classification of the cells, which achieves an average accuracy of 93.02%. The authors of [5] with a Deep Feature-Based Support Vector Machine (DFB-SVM) technique outperformed the works in [1, 4] obtaining an average accuracy of 89.63% and 94.52%, respectively. The authors of [13] classified with an average accuracy of 83% two kind of defects: micro-cracks and finger-interruption. However, it used both the public ELPV Dataset and its own dataset, which makes the experiments not replicable.

The aim of this research is to develop a cell classifier able to classify the specific defect detected on the input image, if any, on the basis of a set of predetermined defects. Although previous research works in this field shown high quantitative performances, they only focused on the task of defect detection. This motivated us to extend the ELPV Dataset with a new labeling, concerning five specific defect classes and healthy classes (for the samples without defects). The labeling of an already existing large-scale dataset will be useful for the community, as ELPV represents a standard in the field. In this sense, the main contribution of the paper is the release of the new and extended labeling of the ELPV dataset, named E-ELPV (i.e., Extended-ELPV), and a benchmark evaluation for the task of defect classification. To our knowledge, no prior studies addressed the task of defects classification at this level of detail, nor similar public datasets have been released.

3 Methods

3.1 Dataset

The ELPV Dataset¹ [4] consists of 2624 EL cells of monocrystalline and polycrystalline photovoltaic modules labeled by an expert, to whom for each cell, in addition to answering the question “Is the cell defective?”, also took care of answering the question “Are you sure?”. By doing so, if the evaluator indicated with certainty that a cell was defective, a probability of defectiveness equal to 100% was assigned; if the evaluator indicated without certainty that a cell was defective, a probability of defectiveness equal to 67% was assigned; if the evaluator indicated with certainty that a cell was healthy, a probability of defectiveness equal to 0% was assigned; if the evaluator indicated without certainty that a cell was healthy, a probability of defectiveness equal to 33% was assigned. Table 1 resumes the original labeling strategy of the ELPV Dataset with the cardinality of each class. Figure 2 shows a monocrystalline cell and a polycrystalline cell, whereas Fig. 3 shows a comparison between a cell with a probability of defectiveness equal to 0% and a cell with a probability of defectiveness equal to 100%.

¹ <https://github.com/zae-bayern/elpv-dataset>.

Table 1. The original labeling strategy of the elpv dataset with the cardinality of each class

Condition	Confident?	Probability of defectiveness	Cardinality
Healthy	Yes	0	1508
Healthy	No	0.33	295
Defective	No	0.67	106
Defective	Yes	1	715
		Total	2624

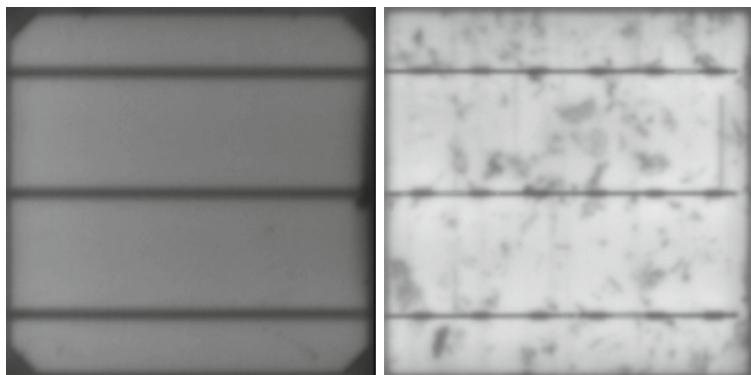


Fig. 2. On the left is shown a monocrystalline cell. On the right is shown polycrystalline cell. Both images have a probability of defectiveness of 0%. The monocrystalline cell appears cleaner than the polycrystalline cell, which appears to be dirty

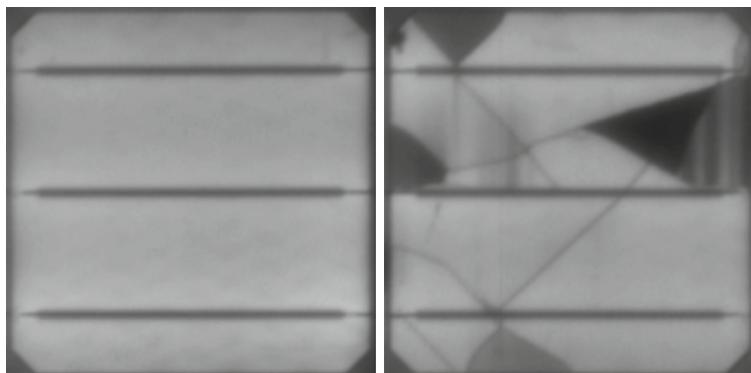


Fig. 3. On the left is shown a cell with a probability of defect equal to 0. On the right is shown a cell with a probability of defect equal to 1

Cells that have been assigned a defect probability of 0% are marked as *Healthy* on our labeling. The remaining cells have been labeled as follows: *Crack* if the cell presents one or more cracks (not too much otherwise the cell is considered as totally broken); *CellBreakage* if the image presents cracks covering more than 50% of the cell's surface; *DarkArea* if the cell has one dark area; *HotSpot* if the cell has one or more hot spots; *OtherDefect* if the cell has a defect different from the previous ones. Then, the associations in Table 2 were obtained. The number of associations is greater than the number of images because to each defective cell could contain one or more defects. In particular, 2551 images are associated to one label and 73 images are associated to two labels. Figure 4 shows the four types of defects studied in this work.

Table 2. Our custom labelling of the ELPV dataset

Label	Associations
Healthy	1508
Crack	430
CellBreakage	133
DarkArea	62
HotSpot	171
OtherDefect	393
Total	2697

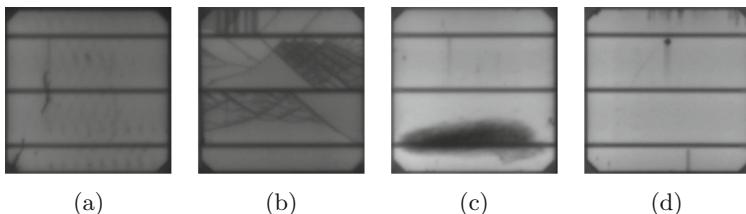


Fig. 4. **a** Cell with crack defect; **b** cell with a cellbreakage defect; **c** cell with DarkArea defect; **d** cell with a HotSpot defect

3.2 Proposed Method

Our methodology consists in the use of two classifiers: the first deals with classifying the health of a cell, indicating whether it is healthy or defective. In the latter case, the cell is given to the second classifier who is responsible for establishing which defect is present. Furthermore, we have developed an additional classifier

that treats the healthy images together with the defective images, the latter labeled with the respective defects. Our purpose is to answer to the following questions: (1) Is training a Healthy VS Not-Healthy binary classifier better than training a multiclass classifier with the same architecture? (2) Does adding the Healthy class to the other classes in the same training improve the classification rate of the other classes?

All developed classifiers use the same image processing and image augmentation. They are developed with PyTorch library, using Python as Programming Language. Furthermore, they are tested and compared with the following networks: ResNext50 [14], VGG-11 [11], Inception-V3 [12] and DenseNet-121 [6]. These neural networks have different characteristics and depth.

Images are normalized using the following formula:

$$\overline{\text{image}} = \frac{\text{image} - \text{mean}}{\text{std}} \quad (1)$$

where $\overline{\text{image}}$ is the normalized image; image is the original image; mean and std are respectively the mean and the standard deviation of ImageNet [9] challenge database. This is due to the fine-tuning of a pretrained model on this dataset. Data augmentation is used. Offline image augmentation simply consists of rotating each image 90° , 180° and 270° . Online image augmentation consists of the following transformations: Gaussian Blur (Kernel $5 * 5$); Color Jitter; Random Horizontal Flip ($p = 0.5$); Random Vertical Flip ($p = 0.5$); Random Rotation ($-3, +3$); Random Translation ($0.02, 0.02$).

3.3 First Classifier (Healthy Classifier)

Given an image of a cell, the first classifier has the purpose of indicating whether the cell is healthy or has defects, without indicating in the latter case which defects are present. The metric used to measure the performance was Overall Accuracy. This classifier was trained with the following hyperparameters: learning rate is 0.0001 and batch size is 32.

3.4 Second Classifier (Defects Classifier)

This task deals with a multilabeling problem. It means that given an image of a cell marked by the first classifier as unhealthy, the second classifier is intended to indicate which defects are present between Crack, CellBreakage, DarkArea, HotSpot and OtherDefect. For this reason we have chosen a proper metric for the multilabeling problem such as the Hamming Distance calculated as follows for each *batch*:

$$HD = 1 - \frac{|\text{predictions} \cap \text{groundtruths}|}{|\text{predictions}|} \quad (2)$$

The loss function used for this classifier is *BCELoss* for ResNext50 and *BCEWithLogitsLoss* for the other networks. For this classifier the training setting was learning rate 0.0001 and batch size 32.

3.5 Third Classifier

To assess the need of two different classifiers, we also trained a third classifier that performs the classification including all the classes. Given an image of a cell, the third classifier is intended to indicate if the cell is Healthy or if there are defects between Crack, CellBreakage, DarkArea, HotSpot and OtherDefect. The metric and the loss function are the same of the second classifier. The setting employed for the third classifier was learning rate 0.001 and batch size 32.

4 Results

We have chosen the mentioned neural networks (ResNext50, VGG-11, Inception-V3 and Densenet-121) to benchmark our dataset. First we computed a baseline for each neural network and then we fine-tuned them. We have chosen this strategy to measure the advantage of training the network instead of using its pre-trained weights on ImageNet. The results of the first classifier, the second classifier and the third classifier are shown in Tables 3, 4 and 5, respectively. The dataset described in Sect. 3.1 is clearly unbalanced but we did not balance it because balancing strategies cause overfitting. We can note that with the ResNext50 network, it is possible to pass from a Hamming Distance of 0.41 (Baseline) to 0.08 in only two epochs for the second classifier; moreover, the Baselines of the VGG11, InceptionV3 and DenseNet121 networks have quite low Hamming Distances of 0.08, 0.10 and 0.18 respectively, making the training of the various networks unable to improve performances. Instead, the third classifier reaches convincing Hamming Distances after a significant number of epochs. The best result is obtained with the ResNext50 network passing from a Hamming Distance of 0.50 (Baseline) to 0.07 only after 100 epochs. It is important to underline that the other networks also seem to work well, reaching a Hamming Distance of 0.08. For each class and for each classifier Table 6 shows information about True Positives, True Negatives, False Positives and False Negatives. Instead, Table 7, 8 and 9 illustrate example of matching and mismatching. Additional materials reporting the classification results for each involved class, the overall dataset (and related labeling) is available at the companion website.² A machine with an Nvidia Quadro RTX 6000 was used to carry out the various experiments.

5 Conclusions

Until now, the scientific community has concentrated its energies on finding techniques that will allow to automatically classify the health or the defect of a photovoltaic cell, without specifying the type of defect or, if it had been searched to classify the types of defects it would have been necessary to resort to private datasets, making the work not very transparent and replicable to researchers

² <https://iplab.dmi.unict.it/EELPV/>.

Table 3. First classifier results. The best and the running up results are highlighted in bold and underline, respectively

Network	Best epoch	Overall accuracy
ResNext50 baseline	/	0.41
ResNext50	86	0.79
VGG11 baseline	/	0.55
VGG11	98	0.77
InceptionV3 baseline	/	0.47
<u>InceptionV3</u>	<u>88</u>	<u>0.78</u>
DenseNet121 baseline	/	0.38
DenseNet121	91	0.77

Table 4. Second classifier results. The best and the running up results are highlighted in bold and underline, respectively

Network	Best epoch	Hamming distance
ResNext50 baseline	/	0.41
ResNext50	2	0.08
VGG11 baseline	/	0.08
VGG11	97	0.08
InceptionV3 baseline	/	0.10
InceptionV3	16	0.08
DenseNet121 baseline	/	0.18
DenseNet121	54	0.08

Table 5. Third results are highlighted in bold and underline, respectively

Network	Best epoch	Hamming distance
ResNext50 baseline	/	0.50
ResNext50	100	0.07
VGG11 baseline	/	0.17
<u>VGG11</u>	<u>92</u>	<u>0.08</u>
InceptionV3 baseline	/	0.17
<u>InceptionV3</u>	<u>99</u>	<u>0.08</u>
DenseNet121 baseline	/	0.22
<u>DenseNet121</u>	<u>99</u>	<u>0.08</u>

Table 6. Information about true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) for each classifier and for each class

Classifier	Class	TP	TN	FP	FN	Total
First classifier	Healthy	258	156	45	65	524
Second classifier	Crack	13	88	40	60	201
	Cell breakage	2	122	58	19	201
	Dark area	2	164	26	9	201
	Hot spot	0	167	3	31	201
	Other defect	1	110	13	77	201
Third classifier	Healthy	297	165	36	26	524
	Crack	52	443	8	21	524
	Cell breakage	21	499	4	0	524
	Dark area	5	506	7	6	524
	Hot spot	15	481	12	16	524
	Other defect	55	389	57	23	524

Table 7. Some examples about true positives, true negatives, false positives and false negatives for each class of the first classifier

Class	True Positive	True Negative	False Positive	False Negative
Healthy				

interested in this field of study. In fact the ELPV Dataset, that is the most famous public dataset of photovoltaic panel cells on which the most well-known works in the state-of-the-art are based, does not have specific labels regarding defects, indicating only a probability of defect. In our work we have extended this dataset by adding for each cell at least one type of defect among four common and well-known types, giving the scientific community the possibility of dedicating future energy to classify the various types of defects efficiently. Just to give a starting point, we performed benchmarks using pre-trained neural networks. Initially, we tried to use two classifiers, the first to distinguish healthy cells from defective cells and the second to identify various defects in the cells marked by the first classifier as defective. Subsequently, with the aim of making the discussion more complete and with more food for thought, we decided to implement another classifier that would automatically classify healthy cells and defective cells, indicating for the latter also the various defects present. By doing so, perhaps for the greater quantity of images taken together under examination, we have achieved concrete results, reaching a Hamming Distance of 0.07

Table 8. Some examples about true positives, true negatives, false positives and false negatives for each class of the second classifier

Class	True Positive	True Negative	False Positive	False Negative
Crack				
CellBreakage				
DarkArea				
HotSpot				
OtherDefect				

with the RexNext50 network. Therefore, we can affirm: (1) Training a Healthy VS Not-Healthy binary classifier is better than training a multiclass classifier with the same architecture. (2) Adding the Healthy class to the other classes in the same training setting further improves the classification rate of the other classes. Due to the lack of a common dataset and benchmark in the field of the classification of defects in solar cell, we could not compare our results with other works. Although this is a limitation for us, we have made a contribution to the scientific community with a new dataset and a benchmark on it that is a good starting point for future comparisons. As future works, we planned to apply on our task methods for defects detection in other industrial sectors such as Silicon technology.

Table 9. Some examples about true positives, true negatives, false positives and false negatives for each class of the third classifier

Class	True Positive	True Negative	False Positive	False Negative
Healthy				
Crack				
CellBreakage				None
DarkArea				
HotSpot				
OtherDefect				

References

1. Akram, M.W., Li, G., Jin, Y., Chen, X., Zhu, C., Zhao, X., Khaliq, A., Faheem, M., Ahmad, A.: CNN based automatic detection of photovoltaic cell defects in electroluminescence images. *Energy* **189**, 116319 (2019)
2. Boulhidja, S., Mellit, A., Voswinckel, S., Lughji, V., Ciocia, A., Spertino, F., Massi Pavan, A.: Experimental evidence of PID effect on CIGS photovoltaic modules. *Energies* **13**, 537 (2020)
3. Deitsch, S., Buerhop-Lutz, C., Sovetkin, E., Steland, A., Maier, A., Gallwitz, F., Riess, C.: Segmentation of photovoltaic module cells in uncalibrated electroluminescence images. *Mach. Vis. Appl.* **32**, 1–23 (2021)
4. Deitsch, S., Christlein, V., Berger, S., Buerhop-Lutz, C., Maier, A., Gallwitz, F., Riess, C.: Automatic classification of defective photovoltaic module cells in electroluminescence images. *Sol. Energy* **185**, 455–468 (2019)
5. Demirci, M.Y., Beşli, N., Gümüşçü, A.: Efficient deep feature extraction and classification for identifying defective photovoltaic module cells in electroluminescence images. *Expert Syst. Appl.* **175**, 114810 (2021)
6. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
7. Obodovskiy, I.: Radiation—Fundamentals. Risks, and Safety, chapter Luminescence. Elsevier, Applications (2019)
8. Peng, H., Sun, X., Weng, W., Fang, X.: Polymer Materials for Energy and Electronic Applications. Academic Press (2016)
9. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**(3), 211–252 (2015). <https://doi.org/10.1007/s11263-015-0816-y>
10. Sarpietro, R.E., Pino, C., Coffa, S., Messina, A., Palazzo, S., Battiatto, S., Spampinato, C., Rundo, F.: Explainable deep learning system for advanced silicon and silicon carbide electrical wafer defect map assessment. *IEEE Access* **10**, 99102–99128 (2022)
11. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
12. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
13. Tang, W., Yang, Q., Xionga, K., Yana, W.: Deep learning based automatic defect identification of photovoltaic module using electroluminescence images. *Sol. Energy* **201**, 453–460 (2020)
14. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1492–1500 (2017)



Application of Deep Q Learning with Simulation Results for Elevator Optimization

Zheng Cao^(✉), Raymond Guo, Caesar M. Tuguinay, Mark Pock, Jiayi Gao,
and Ziyu Wang

University of Washington, Seattle, WA 98195, USA
{zc68,rpg360,ctuguina,markpock,jerrygao,ziyuw5}@uw.edu

Abstract. This paper presents a methodology for combining programming and mathematics to optimize elevator wait times. Based on simulated user data generated according to the canonical three-peak model of elevator traffic, we first develop a naïve model from an intuitive understanding of the logic behind elevators. We take into consideration a comprehensive list of features including capacity, acceleration, and maximum wait time thresholds to adequately model realistic circumstances. Using the same evaluation framework, we proceed to develop a Deep Q Learning model in an attempt to match a naïve algorithmic approach for elevator control. Throughout the majority of the paper, we work under a Markov Decision Process (MDP) schema, but later explore how the assumption fails to characterize the highly stochastic overall Elevator Group Control System (EGCS).

Keywords: Deep Q learning · Optimization · Simulation · Markov decision process · Temporal difference · Elevator group control systems

1 Introduction

Elevators figure strongly into the daily life of the ordinary urbanite. The role of elevator wait times may seem negligible, but the current state of elevator algorithms comes as a result of decades of optimizations and improvements in EGCS facilitated by a wide array of fields. The minimization of elevator wait times becomes especially crucial during the down-peak and up-peak when many elevators are crowded. Poor algorithmic design results in frustrated and tired workers cramming around elevator doors, wasting valuable time.

This research group has approached elevator optimization via two angles, addressed by two separate teams—explicit mathematical modeling and machine learning for approximate optimization. This paper focuses on the latter approach. Our team’s source code is contained in the “Elevator Project” GitHub [4]. We first generate data according to the canonical three-peak model, and use it to build a base-case model to analyze the performance of traditional elevator design. We subsequently turn towards Deep Q Learning to attempt improvements over the naïve base-case.

1.1 Framing and Literature

We frame our discussion of elevator optimization through the hierarchical paradigm of Elevator Group Control Systems (EGCS), the central mechanisms in multi-elevator buildings which control and monitor elevator motion. Where elevators stay by default, which elevators will be dispatched to various hall calls, etc. are managed by EGCS. Its importance to internal transportation has led to an array of research where innovations from across engineering disciplines have been combined and synthesized to produce the modern elevator. Throughout the paper, we discuss an abstract implementation of an EGCS, recording an overall state for elevators, creating algorithms that respond to that state, and state transition functions in a particular building.

Several authors before us have attempted to apply machine learning to EGCS—particle swarm optimizations [1], Convolutional Neural Networks (CNNs), and neuro-fuzzy systems [5], amongst other approaches. Combining machine learning approaches with rigorous mathematics has been a particularly fruitful approach for other problems—for example, Zheng Cao’s previous paper “Application of Convolutional Neural Networks with Quasi-Reversibility Method Results for Option Forecasting” [3].

1.2 Strategy and Motivations

Our approach to optimization is straightforward—an application of Deep Q Learning to what we characterize as a classification problem taking in a simplified version of a building’s current state and outputting commands to the elevator(s) therein.

A Deep Q Network (DQN) model was chosen to be the machine learning model to optimize the environment because there exist numerous examples of successful DQN decision-making models trained and tested in similar environments to our EGCS. The original DQN Atari Paper [4] has shown that DQNs can be trained in environments with large continuous State Spaces and discrete Action Spaces. The EGCS State Space is mildly large and continuous, and the Action Space is discrete. We will continue the DQN discussion in Sect. 5.

2 Theoretical Background

This section introduces a brief theoretical background of Reinforcement Learning, Markov Decision Processes, and Q-Learning.

2.1 Reinforcement Learning, Markov Decision Processes

The generic Reinforcement Learning (RL) problem is framed as an interaction between an agent and an environment. At each time step, the agent selects an action out of a set of possibilities. The environment responds by shifting probabilistically to a different state and presenting that state to the agent alongside a

scalar reward. This back-and-forth interaction continues until the environment reaches a terminal state (there are also RL problems involving environments without a terminal state, but the problem discussed here is not one of them). A complete sequence of actions from the agent and responses from the environment, from start to terminal state, is known as an “episode”. We denote the n th state by S_n , the n th action by A_n , and the n th reward by R_n (where S_0 is the initial state, A_0 is the first action, and R_0 is the reward given in response to that action).

In this sense, RL problems can be interpreted as a series of classification problems where the agent, at time step n , is tasked with choosing the action that maximizes some function of the rewards following the n th action. This function is usually (and is here),

$$\sum_{i=n}^t \lambda^{i-n} S_i$$

known as the discounted return, where t is the time step after which the terminal state is reached. $0 \leq \lambda < 1$ is the “discount factor” and chosen as a hyperparameter in training, where lower values will make the model prioritize increasing immediate rewards, and higher values will make the model have a more “long term” value.

A finite Markov Decision Process (MDP) is a special case of an RL problem where the number of states and possible rewards is finite, the number of actions that can be chosen in response to each state is finite, and most importantly, the probability of any state-reward pair given in response to any action and previous state is dependent only on that action and previous state (and not any of the actions and states that preceded them). This is known as the “Markov Property” and can be expressed symbolically by

$$\Pr\{R_{t+1} = r, S_{t+1} = s | S_0, A_0, R_1, \dots, S_{t-1}, A_{t-1}, R_t, S_t, A_t\} = \Pr\{R_{t+1} = r, S_{t+1} = s | S_t, A_t\}$$

for any r, s that lie in the set of possible rewards and states respectively. A more formal definition is given in the appendix. MDPs are important because most proofs that provide convergence guarantees for RL algorithms only function in the MDP case, although empirically great success has been achieved in applying these methods to non-MDP RL problems. Its importance will be expounded upon in a later section [8].

2.2 Q-Learning

Finite MDPs are solved by finding a good policy $\pi(a|s)$, which is a probabilistic function that defines the actions of the agent. In particular, $\pi(a|s)$ denotes the probability of the agent choosing action a next if the last state were to be s . With respect to such a policy π , we can define a value function $v_\pi(s)$, which defines the expected return of an agent starting at state s and following policy π . Similarly,

we can define an action-value function $q_\pi(s, a)$ which defines the expected return of an agent starting at state s , taking action a , and then following policy π thereafter. For a finite MDP, there is at least one optimal policy [8] π_* , whose value function v_* has the property that for any other policy π , $v_*(s) \geq v_\pi(s)$ for all states s . Its action-value function is denoted q_* . The optimal action-value function must satisfy the Bellman equations [8], shown below:

$$\forall \text{ states } s, q_*(s, a) \mathbb{E} \left[R_0 + \gamma \max_{a'} q_*(S_1, a') | S_0 = s, A_0 = a \right]$$

In fact, the optimal action-value function is the only action-value function capable of satisfying this [8].

Many RL methods try to approximate solutions to this equation through alternating phases of policy evaluation (an approximate computation of the action-value function of the current policy) and policy improvement (changing the policy to improve it).

Early versions of Monte-Carlo methods performed policy evaluation by computing a collection of episodes and computationally approximating the action-value function at state s and action a by the average of the return from all episodes passing through s using a . They then performed policy improvement by setting $\pi(s)$ to be the argmax of $q(s, a)$ over all possible actions a , where q is the approximate action-value function found in policy evaluation.

These methods have the problem of only updating predictions after (possibly extremely long) episodes are complete, so they are improved upon by Q-Learning [8], which instead, in policy evaluation, uses the update rule

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha [R_{t+1} + \lambda \max_a Q(S_t, a) - Q(S_t, A_t)]$$

where R_{t+1} is the actual reward given by the environment, Q is our current estimate for the action-value function, and α is a hyperparameter that decides how heavily new data should be weighted. It has been proven that Q-learning on an MDP converges almost surely to the optimal value function [7]. This method is still impractical because it requires storing estimates for $Q(s, a)$ for every state-action pair s, a , or at least for the most frequently traversed ones. In our case, and many others, there are simply far too many state-action pairs for this to be done.

As a result, we instead rely on a method first introduced in the Atari paper [6] which utilizes a Neural Network to evaluate Q . The state, actions, reward, and next state from each time step are stored in a queue, and every time a finite number of time steps have occurred, the Neural Network is trained on data in the queue with the loss function

$$Q(S_n, A_n) - (r_n + \max_{a'} Q(S_{n+1}, a'))$$

where Q is the current action-value function estimate, S_n is the current state, A_n is the current action, r_n is the reward given for that action, and S_{n+1} is the next state. We note that $(r_n + \max_{a'} Q(S_{n+1}, a'))$ is the update value from

Q-learning with α set to 1. The policy we follow is dictated by a second neural network, and policy improvement is performed by updating the second neural network's weights with a weighted average of weights from both networks, essentially "nudging" the policy in the right direction. More details are shown below.

3 Simulation

Before tackling EGCS, we seek to develop a system for the generation of user traffic data to feed into the system. We modularize this generation by generating a collection of individual people who select times to make hall calls from a truncated Poisson distribution.

We chose to simulate data rather than record real-world observations in order to have the flexibility to observe the three-peak model with an arbitrary number of users. Moreover, we are able to adjust the characteristics of the building with which we are working rather than being bound to observed characteristics.

3.1 Individual Person Values

To have the widest application of our model, we generated raw data based on a standard 9–5 work schedule with a 30-min lunch break in between. We assumed that all the workers would arrive and leave the building with a mean time of 32400 s (9 AM) and 61200 s (5 PM) with a standard deviation of 1800 s (30 min). To simulate real life, we also randomly generated each person's weight based on a normal distribution based on an individual's sex.

3.2 Office Building Values

The parameters of our simulated building are as follows: There are 8 floors in the building. There are approximately 200 workers per elevator in each building. It takes approximately 15 s for an elevator to open, load people, and close. It takes approximately 5 s for an elevator to ascend or descend a single floor.

3.3 Simulated Table

Table 1 presents some sample simulated user data for the modeling process.

4 Modeling

This section illustrates the modeling process of building up the elevator system and the environment, introduces the model's interactions with the environment, and proposes the Naïve model.

Table 1. Sample simulated user data

Time (s)	Start floor	Destination floor	Weight
...
27000	1	2	77.9...
...
27042.9...	1	7	78.3...
...
60754.1...	8	1	101.6...
...

4.1 Elevator

We formulate a basic representation of a single-elevator control system which will serve as the later basis for our expansions to more complex EGCS.

The single-elevator case is both the easiest to mathematically model and the easiest to optimize. Therefore, we decided to concentrate on this case.

When there are more elevators, our results can be naively extended by imagining that the ECGS arbitrarily assigns each hall call to an individual elevator, and each individual elevator operates independently in response to its assigned hall calls using our proposed single-elevator algorithm.

4.2 Environment

The environment consists of an object called a Time List containing all the events, and an object called a State which describes certain aspects of the simulation.

In particular, the State consists of the upward and downward calls of passengers looking to enter the elevator and go a specific direction, the time of the simulation, the elevator's speed, the wait time that it takes to open and close the door to let passengers in, and the location of the elevator.

The environment details will be expounded upon in the following naïve model.

4.3 Model Interactions

Attached to the environment is a model which decides chooses an action based on the current State. Each action will update the State and Time List accordingly. The State may also be updated prior to each action based on events within the Time List.

At the start of the simulation, we initialize the state so that the elevator starts on the first floor and the state's time starts at 27000 s (7:30 AM). During the simulation, we iterate through the Time List until there are no more events within the Time List and when there are no more passengers within the State

that wish to be moved. When this occurs, the simulation ends and we calculate the total time all it took for all the passengers to reach their destinations.

The model takes actions based on only the following three inputs:

1. The state of the up/down button of each floor—Whether it is pressed or not.
2. The state of the buttons in the elevator—whether they are pressed or not.
3. The current floor the elevator resides on.

4.4 Design

Our system centers around the management of a global state variable by a main event loop which processes events in a Time List, an ordered procedure for all the relevant events in the period over which the simulation occurs—for us a day—generated beforehand.

More specifically, we determine before the start of a simulation when people will enter and exit the building and which floors they will want to go to using the classical three-peak model and truncated Poisson distributions around each peak. The various hall calls we generate as a result are fed into our Time List to be processed by the event loop as the simulation progresses.

Treating each Time List event as a stimulus, we manage the current State and request an action from the model, passing in our State as a parameter, to precipitate the succeeding state. Any model we use for this single-elevator system has a simple choice between three commands—Idle, Move, and Open Doors. This classification problem forms the premise of the optimization we attempt to achieve. Our naïve model deals with State in a hard-coded and ‘common sense’ way—one which is of course deeply inapplicable to more complex EGCS,’ but allows us to verify the functionality of the system.

See Naïve Model with Environment 1.0.

4.5 Naïve Approach to Elevator Control

We created a basic algorithm for elevator control that approximates how many elevators practically function. The algorithm alternates between two phases. In phase 1, there are no passengers in the elevator, and the elevator travels to the closest floor with a hall call (if there are no hall calls at all, the elevator will instead idle). Once the elevator travels to a floor with a hall call, it will randomly choose one of the (at most two) directions in which hall calls have been made on that floor, open its doors, and signal its intent to travel in that direction.

This begins phase 2, where the elevator will continuously travel in the direction it signaled, letting people off at their intended destination and picking people up when it reaches a floor with hall calls in the signaled direction. This phase continues until there are no remaining passengers on the elevator, in which case phase 1 will start again. We tested the results of this model on both the 1.0 (see image Naïve Model with Environment 1.0) and 2.0 (see image Naïve Model with Environment 2.0) versions of the naïve model.

5 Modeling Via Deep Q Network

To further optimize total run time, we set a Deep Q Network (DQN) to be the EGCS's inner decision-making model.

5.1 EGCS Data Encoding and Decoding

The following describes how data flows in (Fig. 1) and out (Fig. 2) of the inner decision-making DQN model within the EGCS.

Input Encoding

```

Get capacity, current weight, current position, up buttons, down buttons, and buttons
pressed from State.
tensor = [cacacity], float(current weight), float(current position)]
for entry in buttons pressed (8 entries) do
    append float(entry) to tensor
end for
for entry in up buttons (8 entries) do
    append float(entry) to tensor
end for
for entry in down buttons (8 entries) do
    append float(entry) to tensor
end for
(Note: float(boolean) == 0 for false and float(boolean) == 1 for true)

```

Fig. 1. Input encoding

Output Decoding

```

Get Action Value: Natural Number between 0 and 4.
if Action Value == 0 then
    Action = Idle
else if Action Value == 1 then
    Action = Open Close Doors for Up Calls
else if Action Value == 2 then
    Action = Open Close Doors for Down Calls
else if Action Value == 3 then
    Action = Move Up
else if Action Value == 4 then
    Action = Move Down
end if

```

Fig. 2. Output decoding

5.2 DQN Action Model

Figure 3 describes how the EGCS DQN Action Model is trained.

Figure 4 describes how the EGCS DQN Action Model is used for inference.

Training the EGCS Action Taking DQN

Set ϵ and λ values: Number between 0 and 1.
 Initialize Sample Hall Calls (SHC).
 Initialize Updating Q^* Steps as C .
 Initialize Replay Memory as D.
 Initialize Action Value Model Q with weights W .
 Initialize Target Action Value Model Q^* with weights $W^* = W$.
 Set epoch range as RANGE.
for Epoch in RANGE **do**
 Initialize Time List and Initialize State S_t with SHC.
 Total Time = 0, Number Steps = 0.
while no more events in Time List **do**
 if action can be taken **then**
 if $\epsilon >$ random number between 0 and 1 **then**
 A_t = Choose Random Action.
 else
 $A_t = \text{ArgMax}_A Q(S_t, A, W)$.
 end if
 Take action A_t and observe $Q_{\text{Val}} = Q(S_t, A, W)$, and S_{t+1} .
end if
 Update Total Time, Number Steps, and S_t .
 R_t based off of environment: $R_t = -1 * \text{number of people waiting} * \text{sum added time}$.
 Store transition (S_t, A_t, R_t, S_{t+1}) in D.
 Sample Random Minibatch of Transitions (S_j, A_j, R_j, S_{j+1}) from D as Mini.
 Minibatch Count = 0.
for (S_j, A_j, R_j, S_{j+1}) in Mini **do**
 $M_j = \lambda * \max_A Q^*(S_{j+1}, A, W^*)$ or 0 if S_{j+1} is terminal. Then set $Y_j = R_j - M_j$.
 Loss = $(Y_j - Q(S_j, A_j, W))^2$.
 Perform a Gradient Descent Step with Loss Value, with respect to $Q(W)$.
 Minibatch Count += 1.
 if Minibatch Count % $C == 0$ **then**
 $Q^*(W^*) = Q(W)$.
end if
end for
end while
end for

Fig. 3. EGCS DQN action model training

Inference with the EGCS Action Taking DQN

Initialize Sample Hall Calls (SHC).
 Initialize Action Value Model Q^* with trained weights W^* .
 Initialize Time List and Initialize State S_t with SHC.
 Total Time = 0, Number Steps = 0.
while no more events in Time List **do**
 if action can be taken **then**
 $A_t = \text{ArgMax}_A Q(S_t, A, W)$.
 Take action A_t and observe $Q_{\text{Val}} = Q(S_t, A, W)$, and S_{t+1} .
end if
 Update Total Time, Number Steps, and S_t .
end while

Fig. 4. Inference with the EGCS action-taking DQN

6 Results and Evaluation

This section presents some modeling results of Naïve Model with Environment 1.0 (Fig. 5) and 2.0 (Fig. 6), as introduced in the previous section.

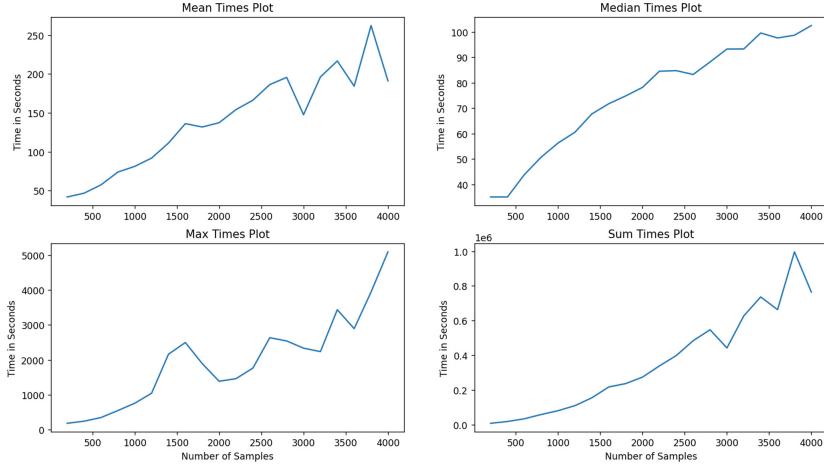


Fig. 5. Naïve model with environment 1.0

6.1 Model Results

Lastly, Fig. 7 summarizes the output logs displaying the training of the DQN Model.

6.2 Model Evaluation

As seen by the above output logs, there is no improvement in the Number of Events, Number of People Moved, Mean Total Time, Median Total Time, Max Total Time, Sum Total Time, when examining the model over ten epochs/full simulation cycles.

We predict that the DQN failed because the EGCS we are working with is not an MDP. Observe the following: Take any state s such that the immediate state after, s' , contains a Hall Call not previously introduced into the EGCS. The Hall Call is placed within the EGCS and the state s is updated to s' at random with no effect from action a . Then, there does not exist a function P_a for s, s', a as described by Definition 3. Therefore, the EGCS is not an MDP.

We know that such states s, s' exist by examining the simulation at the beginning of time: Time 27000 s (7:30 AM). Progress through the simulation by taking a set of sequential actions $A = \{a_1, \dots, a_n\}$ with each action being taken

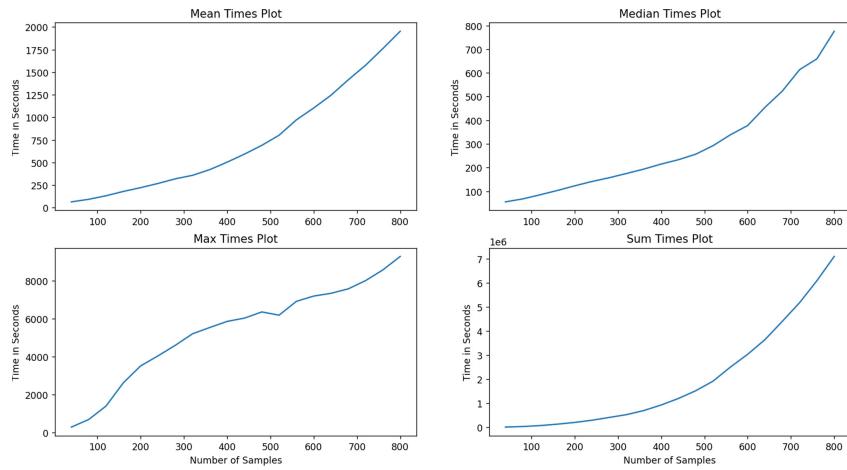


Fig. 6. Naïve model with environment 2.0

<p>Epoch: 1 Number of Events: 1421 Number of People Moved: 13 Mean Total Time: 8672.116597370954 Median Total Time: 6188.91577775008 Max Total Time: 17717.78789404274 Sum Total Time: 112737.5157658224</p> <p>Epoch: 2 Number of Events: 1441 Number of People Moved: 31 Mean Total Time: 5176.042491227574 Median Total Time: 1370.7218384404623 Max Total Time: 17032.94629413835 Sum Total Time: 160457.3172280548</p> <p>Epoch: 3 Number of Events: 1435 Number of People Moved: 12 Mean Total Time: 10404.46914897443 Median Total Time: 11418.724968017284 Max Total Time: 25499.116156212527 Sum Total Time: 124853.62978769315</p> <p>Epoch: 4 Number of Events: 1502 Number of People Moved: 9 Mean Total Time: 3829.7981426569863 Median Total Time: 3370.4148562258997 Max Total Time: 9698.927909479957 Sum Total Time: 34468.18328391288</p> <p>Epoch: 5 Number of Events: 1400 Number of People Moved: 5 Mean Total Time: 1557.9121716438124 Median Total Time: 1371.2478649237346 Max Total Time: 2599.8629089451897 Sum Total Time: 7789.560858219062</p>	<p>Epoch: 6 Number of Events: 1453 Number of People Moved: 7 Mean Total Time: 9148.33341554914 Median Total Time: 2376.5181699321 Max Total Time: 20295.279809866006 Sum Total Time: 64038.333908843975</p> <p>Epoch: 7 Number of Events: 1402 Number of People Moved: 10 Mean Total Time: 4165.139390428786 Median Total Time: 4404.5944891503805 Max Total Time: 9510.815931217185 Sum Total Time: 41651.39390428786</p> <p>Epoch: 8 Number of Events: 1381 Number of People Moved: 2 Mean Total Time: 4989.917602234955 Median Total Time: 4989.917602234955 Max Total Time: 5525.029682037108 Sum Total Time: 9979.83520446991</p> <p>Epoch: 9 Number of Events: 1417 Number of People Moved: 27 Mean Total Time: 10868.958153285035 Median Total Time: 12403.827923114157 Max Total Time: 20542.751971216043 Sum Total Time: 293461.87013869593</p> <p>Epoch: 10 Number of Events: 903 Model Has Moved No People.</p>
---	--

Fig. 7. Output logs displaying the training of the DQN model

on a set of sequential states $S = \{s_1, \dots, s_n\}$ until we reach a single Hall Call h and no $s \in S$ contains h . The Hall Call will then be placed in the next state: s' .

An MDP is necessary for any analytic assurance of convergence within Q-learning [7]. Our DQN is an implementation of Q-learning where we approximate the reward function with a neural network. Extrapolating from [7], there is no assurance for convergence within the DQN model proposed in this paper [7]. Additionally, the authors of the original DQN Atari Paper [6] similarly write that their method lacks sufficient convergence guarantees.

6.3 Possible Solutions

To fix the problem of having no analytic assurance of convergence, a modified EGCS where the inner decision-making model has knowledge of all the future Hall Calls can be created.

The complication with this is that the EGCS no longer models real-life scenarios: Elevators should not know when and how many people will make Hall Calls. Therefore, this model will not be useful outside of simulations.

A possible fix to this modified EGCS inner decision-making model previously described is to create a model which will predict future Hall Calls. This will attach to the inner decision-making model and will feed it to predicted future Hall Calls. It could back-propagate and update its weights based on the Hall Calls the EGCS actually receives during each run.

7 Conclusion

In this project, we created a simulated environment to test the effectiveness of a given elevator control algorithm and demonstrated its correctness with a naïve control algorithm that achieved the expected results. The environment is strongly modularized and provides clear and detailed outputs for the purposes of training Reinforcement Learning models. To demonstrate its efficacy, we attempted to employ a Deep Q-Learning approach to create an ML model that improves upon the naïve algorithm discussed above. This methodology did not yield results surpassing those of the naïve algorithm due to a failure to converge, and we strongly suspect this is caused by the application in question not being a sufficiently strong approximation of a MDP.

In the future, we plan to use this environment to test various other Reinforcement Learning approaches to the elevator optimization problem. We similarly hope that other groups will utilize our environment as groundwork for Reinforcement Learning-based attempts to solve the same problem.

Both projects produce independent results, and potential combinations of both approaches may help achieve better optimization in future research. Nevertheless, we acknowledge the help the main authors of the additional paper have serviced in creating this reinforcement learning approach: Wanchaloem Wunkaew, Xiyah Chang, and Benjamin Davis.

Acknowledgments. The original inspiration for elevator optimization came from Zheng Cao and his experiences with the antiquated elevators on the University of Washington campus. From this catalyst, we have attempted to approach elevator optimization using reinforcement learning and explicit mathematical modeling. Our second approach, facilitated by our math research team, we are currently outlaying in a separate paper, “Application of Spatial Process and Gibbs Random Field Approaches for Dumbwaiter Modeling for Elevator Optimization,” which applies methods from pure math such as Spatial Process and Gibbs Random Field approaches to optimize elevator wait times [2].

Appendix: Markov Decision Processes

A Markov Decision Process (MDP) is defined as a four-tuple (S, A, P_a, R_a) with the following properties:

Definition 1. S is a set of states, which is called a State Space.

Definition 2. A is a set of actions, called the Action Space. The Action Space is (usually) derived from S .

Definition 3. $\forall a \in A$ and $\forall s, s' \in S$, $P_a(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$; the probability of taking action a at state s and transitioning to state s' .

Definition 4. $\forall a \in A$ and $\forall s, s' \in S$, $R_a(s, s', a)$ is the immediate reward received after taking action a and transitioning from state s to state s' .

References

1. Bolat, B., Altun, O., Cort’s, P.: A particle swarm optimization algorithm for optimal car-call allocation in elevator group control systems. *Appl. Soft Comput.* (2013)
2. Cao, Z., Chang, X., Davis, B., Wunkaew, W.: Elevator optimization: application of spatial process and gibbs random field approaches for dumbwaiter modeling and multi-dumbwaiter systems. arxiv.org/abs/2209.12401
3. Z Cao, W Du, K V. Golubnichiy, Application of convolutional neural networks with quasi-reversibility method results for option forecasting. [arxiv.org/pdf/2208.14385](https://arxiv.org/pdf/2208.14385.pdf)
4. C M. Tugunay, R Guo, M Pock, J Gao, Z Wang, Z Cao, Elevator project (2022). <http://github.com/ctugunay/ElevatorProject>
5. Y L. Dong, X M. Wang, The intelligent scheduling method of elevator group control based on fuzzy neural network. *Appl. Mech. Mater.* (2012)
6. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arxiv.org/pdf/1312.5602.pdf
7. Regehr, M.T., Ayoub, A.: An elementary proof that Q-learning converges almost surely. arxiv.org/pdf/2108.02827.pdf
8. Sutton, R., Barto, A.: Reinforcement Learning: an Introduction. <http://inst.eecs.berkeley.edu/cs188/sp20/>



Pix2Pix Hyperparameter Optimisation Towards Ideal Universal Image Quality Index Score

Dirk Hölscher^{1,2(✉)}, Christoph Reich^{1,2}, Martin Knahl^{1,2}, Frank Gut^{1,2},
and Nathan Clarke^{1,2}

¹ Institute for Data Science, Cloud Computing and IT Security,
Furtwangen University, 78120 Furtwangen, Germany

{dirk.hoelscher, christoph.reich, martin.knahl, frank.gut}@hs-furtwangen.de

² Centre for Cyber Security, Communications and Network Research,
Plymouth University, Plymouth, UK
n.clarke@plymouth.ac.uk

Abstract. Generative models and their possible applications are almost limitless. But there are still problems that such models have. On one hand, the models are difficult to train. Stability in training, mode collapse or non convergence, together with the huge parameter space make it extremely costly and difficult to train and optimize generative models. The following paper proposes an optimization method limited to a few hyperparameters with grid-search and early stopping which selects the best hyperparameter combination based on the results obtained with the Universal Image Quality Index (UIQ) by creating a copy of the source image and comparing it with the generated target. The proposed method allows to directly measure the impact of hyperparameter tuning by comparing the achieved UIQ score against a baseline.

Keywords: Hyperparameter tuning · Generative models · Pix2Pix · Universal image quality index · Quality assessment

1 Introduction

The sheer potential of Generative Adversarial Networks (GANs) [11] for augmentation, text-to-image [21], image-to-image translation [16, 32] or upscaling GANs [19] will help to further advance research in various fields. With the emergence of GANs, generating images of high quality has achieved impressive results. Besides subjective measurements from human observers, objective measurement techniques are hard to find, which assesses a generated image whether it represent the real world or not. In this paper we define images representing the real world as **realistic images**.

The challenge GANs are facing, is the missing ground truth image (reference image), making it challenging to directly measure the reality of a single image. A step towards image assessment is measured by metrics, like Inception Score

(IS) [23], Frechet Inception Distance (FID) [14] or classification performance, where the generated images are used in a classification task and the performance is evaluated. Measuring the distribution of generated images Inception Score (IS) where the generated images are run through a pretrained Inceptionv3 model to calculate the score, or as an addition the distribution of generated samples compared against the distribution of real images used for training (FID). Additional research from 2020 with Microsoft proposing a new research field of generated image quality assessment (GIAQ) [12], highlighting the importance of as well as the need for quality assessment. Image quality and parameters are closely correlated to each other. Hyperparameters have a direct impact on stability and quality. Some parameters when chosen to high or low will make it harder for the GAN to generate meaningful result and will lead to mode collapse or vanishing gradients [26]. The overall quality of images will also suffer when using the wrong parameter configuration. In addition, GANs have a huge and complex parameter space which is impossible to explore in a meaningful timeframe.

The goal is to determine the optimal way to train the GAN by finding the best possible hyperparameters for a specific subject area, like manufacturing. For subjective measurements conducted by humans, the image might fulfill a humans quality requirements but used as input for another network for a detection or segmentation task the result can be very different. One problem is the missing reference images (ground truth images), as the tasks of GANs is to generate new unseen image samples. Current metrics are dependent on reference images to measure the similarity between a pair (reference and target image), based on the deviation between pixels or in more sophisticated metrics the deviation of contrast, saturation, luminescence and other key image characteristics.

As shown in our previous work [15], we showed the versatility of the conditional GAN Pix2pix [16] as a powerful tool for augmentation in the metal working industry. Generating new samples by confusing the generator with inputs that can translate into various targets forcing the generator to start interpreting what to do while keeping the overall structure consistent with the input.

The following work utilizes parameter tuning for Pix2Pix to find the optimal hyperparameter set within a specific domain, optimized to achieve the best possible score, maximizing the Universal Image Quality Index (UIQ) [26]. UIQ is as shown in Sect. 4 a complex and strict metric including distortion as well as illumination. As Pix2Pix learns a transformation from one domain to another domain, utilizing this behaviour to generate copies of the original image enables to directly measure the generated image using UIQ. In addition, it makes it possible to measure the impact of hyperparameters when training. Given Pix2Pix's standard parameter and measuring the performance with UIQ, a baseline quality is obtained which can be compared to the output of generated images when using new parameters. Parameters are adjustable values in the architecture and impact training performance of GANs. For example, the number of layers, hidden neurons, activation function, optimizer and their learning rate, filters or weights are changeable parameters in Pix2Pix, impacting training and the quality of the

generated images. The approach utilizes a grid search with pre-selected values for the learning rate, the learning rate's beta value as well as the leaky alpha for the encoder and discriminator. Furthermore, an early stop was implemented to speed up the hyperparameter finding process. The scope of the presented work is to find a suitable parameter combination within a short period of time (depending on the application domain) with the best possible UIQ score and to show that good early results can be transferred to longer training cycles while maintaining the quality.

The paper is organized as follows: Sect. 2 presents related work about hyperparameter tuning methods and optimisation for GANs. Section 3 introduces Pix2Pix and describes how the GAN works. Afterwards, Sect. 4 presents and describes the chosen metric for evaluation UIQ in more detail. The next Sect. 5 explains and illustrates the conducted experiments along with the achieved results. Thereafter, Sect. 6 describes the proposed algorithm in more detail. The last Sect. 7 concludes this paper and presents a brief outlook into future work.

2 Related Work

The following section will give a brief overview of research in hyperparameter tuning and optimisation in GANs.

Liashchynskyi and Liashchynskyi [27] compare the three most common search strategies to find the best hyperparameter set namely grid search, random search and genetic algorithms. Grid search completely searches a space for a subset of hyperparameters for the given network. A boundary can be set to limit the number of searches as grid search suffers from high dimensional spaces leading to long search times when the process is not parallelized. Instead of completely searching the whole space, random search randomly selects a number of values within the hyperparameter space. Random search works especially well if there is only a small number of hyperparameters. Genetic algorithm is an evolutionary search strategy, where the start is a randomly selected population (randomly selected hyperparameter). Afterwards, their fitness using cross validation is determined. The worst performing combinations are replaced with new ones generated through recombination and mutations. This process is repeated until a stopping criterion is reached, or the performance is not increasing any further. Using the CIFAR-10 dataset they established that grid search took nearly twice as long as random search to achieve nearly the same accuracy of 86%. The genetic algorithm took nearly as long as grid search and achieved the same result as random search.

As shown, random search can outperform grid search and genetic algorithms. In our approach the search space was established beforehand through thorough testing, limiting the search space and enabling the usage of grid search. In addition, this was further done, to determine a threshold after as few training cycles as possible to establish an abort termination criterion for future searches.

Xiao and Yan et al. [28] present a genetic algorithm with variable length for deep learning network architecture. With varying number of convolutional

layers in convolutional neural networks the number of parameters also increases. Deeper models require a variable length of chromosomes for the genetic algorithm. A chromosome contains hyperparameters defining a set of parameters for the model. As with any genetic algorithm the first step is to generate the initial population with randomized parameters divided into population \mathbf{p} and generation \mathbf{g} , where each population contains individuals with parameters. Afterwards, the fitness (performance accuracy) of each individual of the current generation is determined. Based on the accuracy each individual is sorted and the individuals with the highest score are taken into the next generation while also keeping a few of the less fit ones. Some of the fittest individuals in the new generation are randomly selected to mutate changing one its hyperparameter values. Afterwards, new individuals are produced from them two are selected as parents and produce children containing randomly selected hyperparameters from the parents. The process is repeated until the population is back to \mathbf{p} (population). This is repeated until the number of generations has reached \mathbf{g} (generation). Then the best individual is chosen and a population with longer chromosomes is produced. Each of these new individual's chromosomes have the parameters from the best individual chosen beforehand and the rest is filled randomly. This is repeated until convergence and then the best overall individual is selected and trained on more epochs. The algorithm was tested with the CIFAR-10 dataset with a stopping condition set to stop if the validation does not increase with additional layers or if the newly achieved fitness score is lower as the previous one. 20 individuals evolved over 5 generations while trained for 5 epochs. Evolution stops at phase 14 with the stopping condition kicking in. Each phase, the number of layers was increased. The results, with a GPU limited to 30 h runtime, achieved with variable length was 88.92%. Large scale evolution performs the worst on time constraints starting with small models and only mutations on chromosomes are available. Without time constraints large scale evolution would yield really good results. Genetic algorithms with no variable length found a high accuracy due to the small search space and a fixed number of convolutional layers of three. When the ideal number is unknown the proposed process of chromosomes with variable length is better especially if resources for computation are limited. Generic algorithms and GANs are an interesting field for hyperparameter optimisation as shown above and additional research done in [6–8, 13, 17].

As Pix2Pix has a variable length when training the generator due to the skip connections, each training cycle is using a different amount of neurons leading to the variable length. The above described extension of generic algorithm can be interesting for the future to test how well a genetic approach can utilized with Pix2Pix and if the performance and overall results are comparable to the presented approach in this paper.

In [5] the authors present an architectural modification called self-modulation for GANs. Self-modulation improves the performance of GANs for different data sets, architectures, losses, regularizers and hyperparameters. The generator of the GAN is conditioned on the generator's own input and only requires a simple change that can be applied to all common generator architectures used in GANs.

A self-modulating layer has hidden activations modulated as the latent space of z (input noise). Modulation allows the model to re-weight the features map when applied feature-wise as a function. Batch normalisation is an integral part of generative models and can be used for self-modulation. The process involves transforming the activations of a layer in batch normalisation, in two ways. First, it does a transformation of the activation values of a perceptron without side information, where two parameters are changed to input dependent parameters parametrized, using a neural network with a multi layer perceptron modulation function applied to the generator's input. The other way involves accessible side information such as labels. With present labels, these conditions can be integrated by connecting the label with the input using a learnable function. For demonstration, they use a bi-linear interaction of the input and two trainable embedding functions of the label. The experimental setting includes two loss functions: (i) non-saturating loss and (ii) hinge loss. The Lipschitz constant is controlled by gradient penalty and spectral normalisation. As architecture, the authors chose ResNet and Spectral Normalisation Deep Convolutional GAN (SNDCGAN). Hyperparameter optimisation is done using three different sets of parameters with the adam optimizer and 100k generator steps. Where the beta values for the optimizer are the most popular ones $(0, 0.9, 1)$, $(0, 0.9, 2)$, $(0.5, 0.999, 1)$, with the last value setting discriminator steps. Datasets include CIFAR10, CELEBA-HQ, LSUN-BEDROOM, and IMAGENET. In the unpaired setting, the proposed approach outperforms the baseline in 30 of 32 cases. With a decrease of 4.3 and 33% outperforms the baseline everytime, when using ResNet and improves in 87.5% for all cases when using SNDCGAN. For the paired setting, the authors evaluated how effective the method performs when applied to an existing model using the same hyperparameter. The result is 36 settings for each dataset and 144 comparisons outperforming the baseline in 124/144 settings.

In [18] the authors present a paper about the current landcspae of GANs while discussing the current existing problems when training GANs. The problems among others are hyperparameter tuning, architecture engineering, loss functions, regularisation and normalisation. They present common loss functions used in GANs such as minimax, non-saturating, Wasserstein and least-squares loss. The minimax loss minimizes the log-likelihood for binary classification for the discriminator, whereas non-saturating the probability maximizes that a generated sample is real. Wasserstein GANs loss is an improvement of minimax loss correlating with G's convergence. Least-square loss is the Pearson divergence between the true distribution of images and the model's distribution. When optimising, normalisation of the discriminator can help to get a more stable gradient flow and overall a stable optimisation. For optimisation two techniques are regualrly used to optimise GAN performance batch-normalisation and layer normalisation. In addition nowadays there are several different architecture patterns for the generator and discriminator. One of those classes are Deep Convolutional GANs and the previous mentioned SNDCGAN and the other class are ResNet. The metrics evaluated are Inception Score, Frechet Inception Dis-

tance and Multi-scale Structural Similarity for Image Quality (MS-SSIM), for one class setups to see if mode collapse and mode dropping happened, the problem and thus why its a metric best suited for one class problems is the unknown class corresponding to the generated image. Search space for GANs are huge and expensive and taking all hyperparameters into account is not feasible at all. Therefore, in the study they perform they validate different steps, changing only the architecture, only the loss and so on, to determine the impact of each of those changes. Hyperparameter are as shown by Miyato et al. [20]. The impact of the loss function on CELEBA-HQ-128 and LSUN-BEDROOM datasets showed that the non-saturating loss is stable for both, psepectral normalisation improves the quality of both sets, with high computational costs gradient penalty can help to improve quality but finding it is complicated and costly. For regularisation and normalisation the loss is fixed to non-saturation. With the various procedures they show that batch-normalisation hurts the discriminator, gradient penalty is better but does not stabilise training. Spectral normalization helps to improve quality and computational time. Architectures are validate with non-saturating loss, gradient penalty and spectral normalization. Both benefit from regularisation and normalisation, and is even better for deeper models. Spectral normalization is better in both cases outperforming the baseline. Hyperparameter optimisation for deep models is an important part for good training results as shown in further research studies [10, 20, 25, 29].

3 Pix2Pix: Conditional Generative Adversarial Network (cGAN)

Pix2Pix [16] is a paired image-to-image translation conditional Generative Adversarial Network (cGAN), with the ability to learn the features of an input image and translate them into a different output without changing the composition and shapes of the original image (ground truth). The condition with pix2pix is the input image itself.

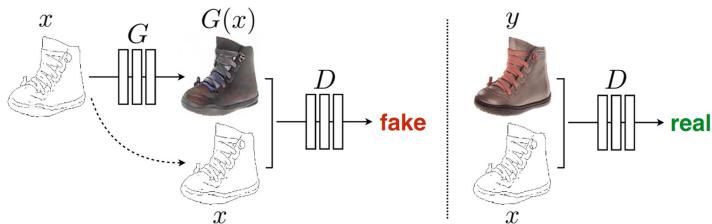


Fig. 1. cGAN architecture in Pix2Pix [16]

Figure 1 shows a simplified Pix2Pix architecture with a generator G and a discriminator D . Generator and discriminator are trained in turns. The input

image used as a condition is x and the generated image created by the generator $G(x)$. The label or target to which the network should translate x to is y (the target domain). The discriminator is trained on the real pairs of x and y to classify them as real and the generated image $G(x)$ as fake. The generator generates a fake image $G(x)$ with the goal of getting classified as real by the discriminator. Therefore, the generator is build as a U-Net [22] an encoder-decoder network with integrated skip connections between layers of the encoder and decoder. Using the U-Net the generator preserves the shapes within the image and transfers them from the input to the output with the desired changes between x and y . One problem with U-Net is the input's missing stochasticity, meaning the output would always be the same as the input, with the generator as well as the discriminator relying on the training data to learn the distribution of data points. Adding random noise does not work, since the generator starts ignoring it at a certain point.

Therefore, the generator is fitted with multiple dropout points throughout it's layers. This leads to a kind of ensemble learner, instead of using the whole network, the dropouts always skip a random number of nodes in the network, and thus always trains on different nodes, this is done to prevent over-fitting. As Pix2Pix is trained to translate an image from the input domain to the target domain binary cross-entropy as the sole loss function is not feasible. This would lead the generator to generate similar images but ones non-compliant with the target as the generated shapes and features differ from the condition. In addition, L1 distance is utilized to calculate the loss. The L1 distance calculates the mean absolute pixel value differences of y and $G(x)$. The discriminator is a PatchGAN a convolutional neural network tasked to binary classify the output of the generator into real and fake. With the L1 loss it is guaranteed that the overall structure (shapes of objects) of the image is correct, meaning the discriminator can focus on classifying each patch (patch size is convolutional kernel size) as real or fake and average the results to determine if the discriminator's input is real or fake.

The paired approach of Pix2Pix2 meaning that each input image, has an corresponding target image, with the same shapes and features but different Style, is a promising start. This can be used as a first step towards image quality assessment for GANs with the possibility of creating copies and evaluating the performance using various metrics such as Mean Squared Error (MSE) [9], Structural Similarity Index (SSIM) [31], Universal Image Quality Index (UIQ) [26], Feature Similarity Index (FSIM) [30] or Mean Absolute Error (MAE) [24].

4 Image Assessment with Universal Image Quality Index (UIQ)

Similar to the L1 loss, the Pix2Pix is trained with, other measurements can be used to determine the quality of the output. In this case exact copies are created to determine how well the model performs. The problem is of missing references. The distribution of images in the input will have similar images as the ones generated, but with different characteristics. Similarity measurements

rely on a reference and target image to measure how similar both images are. Different colouring, changed shapes or completely new interpretations makes it difficult to measure how similar two images are. One benefit of Pix2Pix is the translation keeping the essence of images such as shapes and objects while changing other characteristics. Therefore, images generated with Pix2Pix can to a certain degree be assessed with similarity measurements. To go one step further is to understand how hyperparameter influence the training process and how the generator works, what is easy for the generator and what can be a problem. To better understand the translation process is used to generate exact copies of the input domain and then validate what and how it changed.

Various tests with MSE, SSIM, UIQ, FSIM and MAE showed that the Universal Image Quality Index (UIQ) [26] by Wang and Bovik turned out to be the most strict and precise metric. UIQ scores lower results for non-realistic images compared to other metrics, while having the best realistic images, if the metric scored high. Instead of summing up errors, UIQ is evaluating image distortion using loss of correlation as well as luminance and contrast distortion. With this more detailed look we can receive a better understanding how the generated image changed. The metric is defined as follows [26]: The original image's signals \mathbf{x} are:

$$\mathbf{x} = \{x_i \mid i = 1, 2, \dots, N\} \quad (1)$$

And the test image's signals \mathbf{y} are:

$$\mathbf{y} = \{y_i \mid i = 1, 2, \dots, N\} \quad (2)$$

The quality index is then defined as follows:

$$Q = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2 + \sigma_y^2)[(\bar{x})^2 + (\bar{y})^2]} \quad (3)$$

Where \bar{x} is the sum of $\frac{1}{N}$ times x_i calculated for all signals of the original image. The same calculation is done for the target image summarized in \bar{y} .

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i, \quad \bar{y} = \frac{1}{N} \sum_{i=1}^N y_i \quad (4)$$

The next part of the equation is calculating σ_x^2 and σ_y^2 which is the sum of $\frac{1}{N-1}$ times $(x_i - \bar{x})^2$ and vice versa for the target image.

$$\sigma_x^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2, \quad \sigma_y^2 = \frac{1}{N-1} \sum_{i=1}^N (y_i - \bar{y})^2 \quad (5)$$

The last part of the equation is σ_{xy} the sum of $\frac{1}{N-1}$ times $(x_i - \bar{x})(y_i - \bar{y})$

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y}) \quad (6)$$

The range of Q can be $[-1, 1]$ with 1 being the highest possible score achieved if $y_i = x_i$ for all signals $i = 1, 2, \dots, N$ and -1 is achieved if $y_i = 2\bar{x} - x_i$ for all i . With the aforementioned three different factors of loss of correlation and luminance and contrast distortion the equation can be rewritten as the product of the factors:

$$Q = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \cdot \frac{2\bar{x}\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \cdot \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \quad (7)$$

The first part is the correlation coefficient for x and y measuring the linear correlation resulting in a result between $[-1, 1]$, with 1 being the highest achievable result if $y_i = ax_i + b$ for all signals i with $a > 0$ and b as constants. The second part of the equation measures $[0, 1]$ the luminance between x and y and is 1 if \bar{x} and \bar{y} are equal. The last part calculates contrast distortion and has the same range as the luminance part with 1 as the best result achieved if σ_x and σ_y are equal. Applied to images the metric starts in the top left and has defined sliding window with a size of $Bx B$ moving vertically and horizontally to the bottom right pixel by pixel. Calculating the quality for M steps as follows:

$$Q = \frac{1}{M} \sum_{j=1}^M Q_j \quad (8)$$

5 Hyperparameter Optimization for Pix2Pix

The approach of this paper is to introduce and demonstrate a prelimited grid-search based hyperparameter optimisation process with early stopping. Through several trial runs the best and most impactful parameters were determined, adam learning rate, adam beta and leaky relu for the discriminator and encoder. The next step was to determine when a good stopping point would be. Therefore, the results of the first epochs were evaluated to see if there would be a good stopping point. After 5 epochs there is an intersection point where good pairs achieved similar results. This initial training and performance evaluation was limited to 5 epochs with a reduced training time, a training loop of 5 epochs only takes 5 min with our hardware with a training set containing 373 samples.

The timeframe of 5 min includes the whole process (loading dataset, training, loading model, generating evaluation pair and evaluation) on a Titan X and 64 GB of RAM. Written in python for the combinations iter-tools was used, numpy to prepare, save and load dataset as well as evaluation set stored using the npz format. Keras was used for Pix2Pix and image_similarity_measures.quality_metrics was used for the UIQ implementation to evaluate the results. With the above shown values for the parameters there is a total number of 5600 possible parameter combinations. The number of parameters were limited to four as those showed the highest impact on performance. A total of 2000 pairs were tested. After each parameter set the performance was evaluated against the baseline score achieved with standard parameters and a

high threshold was chosen to get the most promising parameter sets. The threshold was set to a 15% increase from the baseline score of UIQ. If this threshold was reached the network could continue with additional epochs.

The achieved results after 5 epochs were sufficient enough and achieved in a justifiable timespan of 5 min. As shown in Fig. 2 the lowermost line is the trained baseline with standard parameters whereas two of the three others are well performing parameter sets highlighted in Fig. 2.

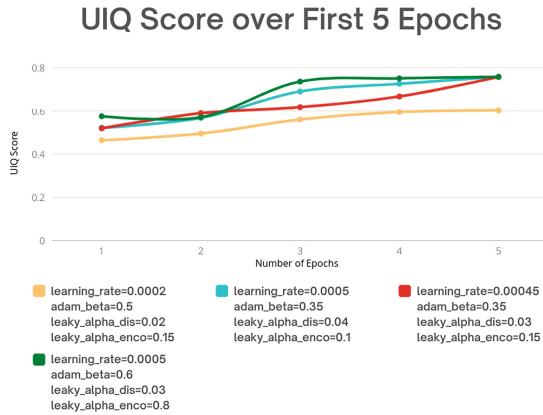


Fig. 2. UIQ for the first 5 epochs

The algorithm works as follows and in addition is shown as a flow chart in Fig. 3:

1. A hyperparameter set is randomly selected from the 5600 samples.
2. The set is checked if it was already trained.
3. If the set was already trained, training is aborted if not training can start and a folder for the model is created.
4. Pix2Pix is trained for 5 epochs with the current parameter settings.
5. After finishing the 5th epoch the performance of the model is evaluated by:
 - (a) Loading the evaluation image.
 - (b) Loading the model trained on 5 epochs.
 - (c) Generating an image based on the evaluation image.
 - (d) Calculating the UIQ score for the reference and generated image.
6. If the UIQ increased by more than 15% to the baseline the training continues for 45 more epochs (can be specified beforehand). If not training is stopped.
7. Good sets are stored in separated text file to be further archived.
8. Repeat steps 1–7 until all combinations were trained.

With the chosen dataset the first part is how to determine the best way to optimize the results to get the best possible UIQ score. As shown UIQ is a fairly complex metric calculating the three factors of image distortion using loss of

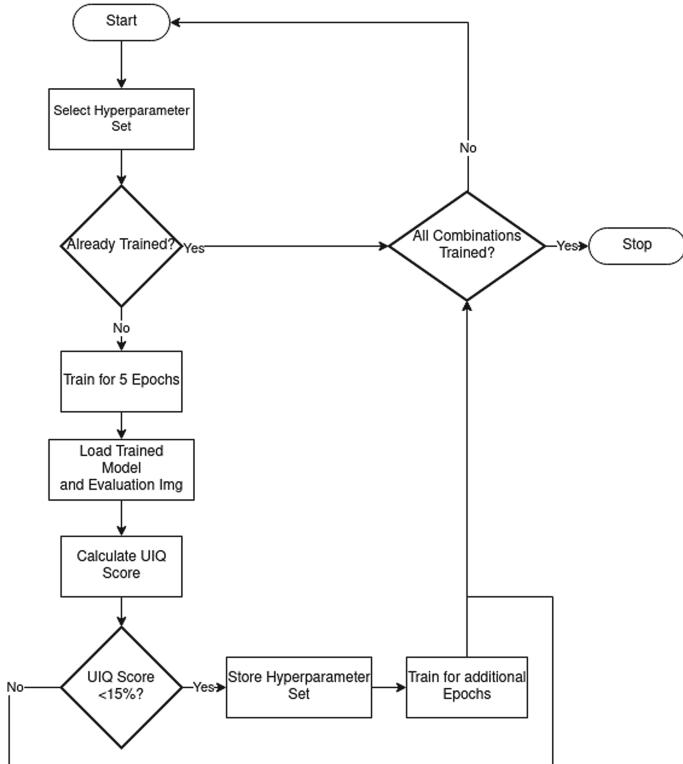


Fig. 3. Flow chart of the hyperparameter optimisation process

correlation, luminance and contrast distortion to evaluate an image other than MSE or SSIM (based on UIQ) and MAE but requires more time to calculate. Pix2Pix's loss is calculated using binary crossentropy and MAE. When considering time requirements, with the higher computation time training Pix2Pix with UIQ instead of MAE for error calculation and optimisation is not feasible. For the presented test cases, testing the time requirement for the above mentioned metrics between MAE, MSE, SSIM and UIQ, showed that the first three metrics are calculated almost instantly, with FSIM taking a mean calculation time of 2 s per image pair. In contrast UIQ took a rounded mean time of 31 s per image pair. For training, this would mean an increased training time by a factor of 30 to train other than with MSE or MAE. Instead of integrating UIQ into the training loop, the output result should yield the best possible result when evaluated with UIQ. This is an indirect form of optimising the training for the best possible UIQ score without sacrificing performance.

6 Experiment: Hyperparameter Optimization

Finding the best parameters for training is performed as a grid search. Various pre-runs determine the most influential parameters as well as value ranges for each parameter.

6.1 Setup of the Experiment

As shown in Table 1 the chosen parameters were adam learning rate, adam beta and the leaky alpha value for the discriminator of Pix2Pix as well as the encoder, with increasing and decreasing values using bigger steps of 0.0005, 0.05 and 0.001. For these experiments, kernel dimensions, stride size, layers and padding stayed untouched.

Table 1. Value pool for parameter optimisation

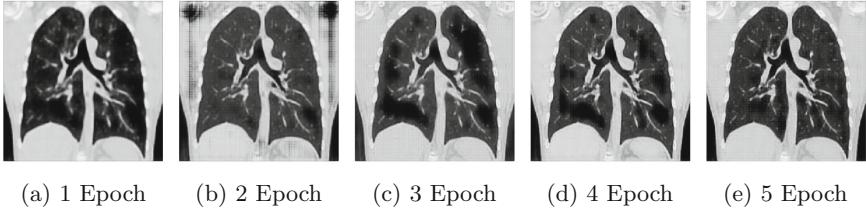
Parameter	Values
adam learning_rate	0.0001, 0.00015, 0.0002, 0.00025, 0.0003, 0.00035, 0.0004, 0.00045, 0.0005, 0.0005
adam_beta	0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7
leaky_alpha discriminator	0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07
leaky_alpha encoder	0.1, 0.15, 0.20, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.80

As mentioned by Isola [16] the encoder has a bottleneck layer with a lower neuron count enabling the network to compress feature representation. When using batch normalization, the activation's would be zeroed for the bottleneck, leading to skipping of the innermost layer. Therefore, batch-norm is replaced with leakyRelu.

6.2 Experiment: Stop Condition

Searching for the best parameter pair should be fast and transferable to a longer training period. Therefore, the first conducted experiment was to look for an early indication if a combination of parameters is good or not and if there is a breaking point.

Figure 4 shows the results of the standard parameter pair of Pix2Pix2 to highlight the training process over the first 5 epochs. With the trials run we determined a good starting point with measurable good and stable results: 5 epochs are good to stop to evaluate model performance till this point. The stopping condition after 5 epochs is the achieved UIQ score when compared with the baseline and if the score is lower 15%, the next parameter set is tested.

**Fig. 4.** Training progress over the first 5 epochs

6.3 Experiment: Overall Result

The next experiment was to evaluate how well and if the results achieved by training the GAN for 5 epochs can be transferred to a higher number of epochs at all, without losing quality and preferable an additional increase in quality. Of those 2000 sets, only 17 pairs achieved an UIQ score with an increase higher 15% than the baseline evaluation. Those sets were chosen and retrained with 50 epochs to see if a good result in the beginning can be transferred to good results in later training stages. The number of epochs was chosen by conducting several tests with different datasets as shown in Fig. 5 and Table 2.

The test dataset was selected from a range of medical datasets including brain MRI [2], brain tumors [1], chest X-Rays [4] and chest CT [3]. The tests were run with set sizes between 350 and 400 greyscale images and were once done for 5 epochs and 50 epochs to find the worst performing dataset for the planned optimisation task. Table 2 shows the results of the four tested datasets and their UIQ performance when using standard parameters for 5 and 50 epochs.

Table 2. UIQ performance with standard parameters

Dataset	UIQ score (5 epochs)	UIQ Score (50 epochs)
Brain tumor	0.7856828755485061	0.8843351633411528
Brain MRI	0.5732014988277478	0.710674772335905
Chest X-ray	0.8703172538516432	0.9434373595897657
Chest CT	0.60254851638818	0.616699817254575

As Table 2 shows the baseline evaluation for two (Brain tumor and chest X-Rays) of the four datasets is already high when trained on 5 epochs while brain MRI increases steadily over the next epochs, the chest CT dataset only shows a small gain when increasing training epochs showing promising potential for parameter tuning.

Figure 5 shows the evaluation images of each dataset used to calculate the UIQ score and to illustrate the structure of images of each dataset used for training.

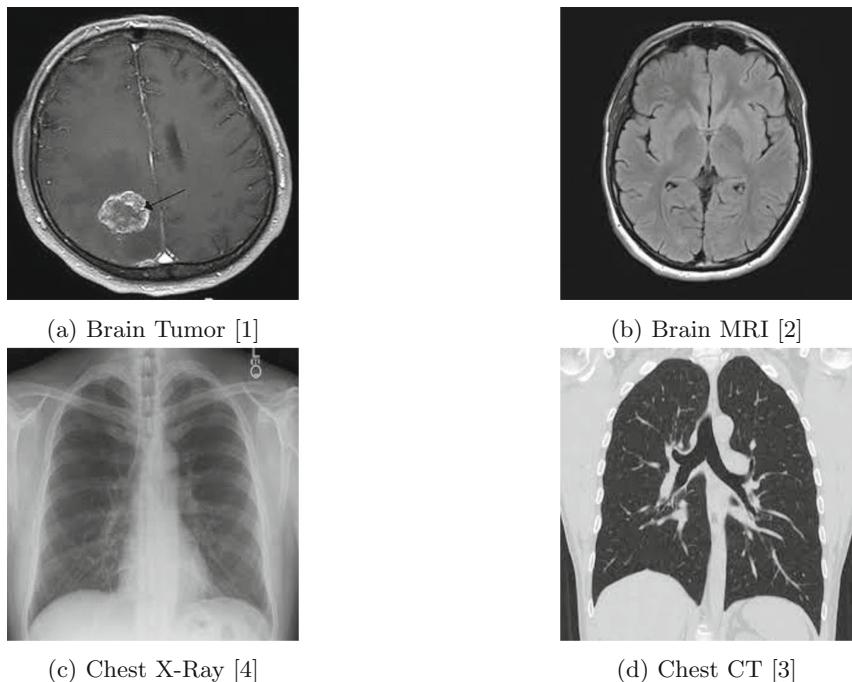


Fig. 5. Evaluation images of the four tested datasets

The first test is always done with standard parameters to get a baseline performance of Pix2Pix for the domain and to evaluate if it is even necessary to look at additional parameter settings. The main question is how well the obtained results when training for only 5 epochs can be transferred to training 50 or more epochs? Table 3 gives a clear indication by showing the baseline UIQ score with the standard parameters and then a selection of results of trained models were the UIQ increased by at least 15% compared to the baseline score.

The table shows that the gained increase in quality can also be transferred to 50 epochs in most but not all cases. In some cases the overall quality gain dropped below 15% when compared to with the 50 epoch baseline. In some cases the final results resulted in a loss of quality when compared to the baseline performance. The question lingers how does the quality between 5 and 50 epochs compare. The following Fig. 6 shows the difference between 5 and 50 epochs of training for the parameter set of learning_rate:0.005, adam_beta:0.45, leaky_alpha Discriminator:0.5 and leaky_alpha Encoder:0.8, with an increased UIQ score of 15.1505183310155 for 5 epochs and 17.6645757511829 for 50 epochs.

Figure 5d shows the reference image used to calculate the UIQ score. The generated copies are used as the target for UIQ. At first glance they seem rather similar, but as Fig. 6c, d show there are several different errors in both copies. The copy created after 5 epochs still has big black patches where the lung's structure

Table 3. UIQ increase from baseline evaluation

Parameter				UIQ (5 epochs)	UIQ (50 epochs)
learning_rate	adam_beta	Leaky Alpha		Difference	Difference
		Discriminator	Encoder		
0.0002	0.5	0.02	0.15	0.60254851638818	0.616699817254575
0.0005	0.35	0.4	0.1	15.4529109163067	-9.1031538828073
0.00045	0.35	0.3	0.15	15.7166571977217	11.2022105085012
0.0005	0.7	0.3	0.75	15.1998759363002	16.2304351892729
0.00045	0.45	0.02	0.8	15.2437085631522	-7.14231863588131
0.0005	0.6	0.3	0.8	15.4735930896487	16.7343419246012
0.0005	0.35	0.06	0.8	15.1308739679999	14.6494554398256
0.0005	0.4	0.06	0.1	15.2558091121152	-17.4374515197859
0.0005	0.45	0.05	0.8	15.1505183310155	17.6645757511829
0.00035	0.6	0.6	0.8	15.228319501323838	12.2102827107861
0.00045	0.5	0.07	0.8	15.9194014775902	17.4478573255486
0.0004	0.35	0.05	0.75	15.192080669392393	16.2733437992696

was wrongfully modeled. When compared to the generated image after training Pix2Pix for 50 epochs the lung's structure is more clearly shown and the black patches are smaller but the overall error count is higher. One typical error which occurs quite often when training for 50 epochs is a black patch in the top right corner. Depending on the use case such errors might be negligible, for tumor detection in the shown use case this can be a problem as the black patches can be identified as an unnatural structure within the lung. For outlying errors as seen in the top right corner this error can be neglected as the important part of the lung is not affected.

Both cases are sub-optimal when used for other tasks as the quality within the important structures of the image are subpar. The resulting follow-up question is how the quality evolves over the span of 50 epochs. In Table 3 the filled row shows an initially really good result with an increase of around 15% but after training with the same parameters for 50 epochs the result dropped and performed worse than the baseline for 50 epochs, loosing 17% to the baseline score. Several training cycles with these parameters showed varying results and dropping performance when training for 50 epochs. The initial idea was to monitor the UIQ score each 5 epochs and drop the training when the score drops significantly under the initial score or under the till then highest achieved UIQ score during the current training cycle.

This approach has not proved feasible as shown in Fig. 7. With the above mentioned parameters three additional training cycles were conducted to see how big of a variance there is when training and to validate if this particular parameter pair always performs worse when trained with more epochs and summarized as a chart in Fig. 7. As shown, the quality of 5 epochs is within a difference of 5% (68% lowest and 73% highest). The 10th epoch shows a slight increase in accuracy, before it drops again. This up and down trend continues through-

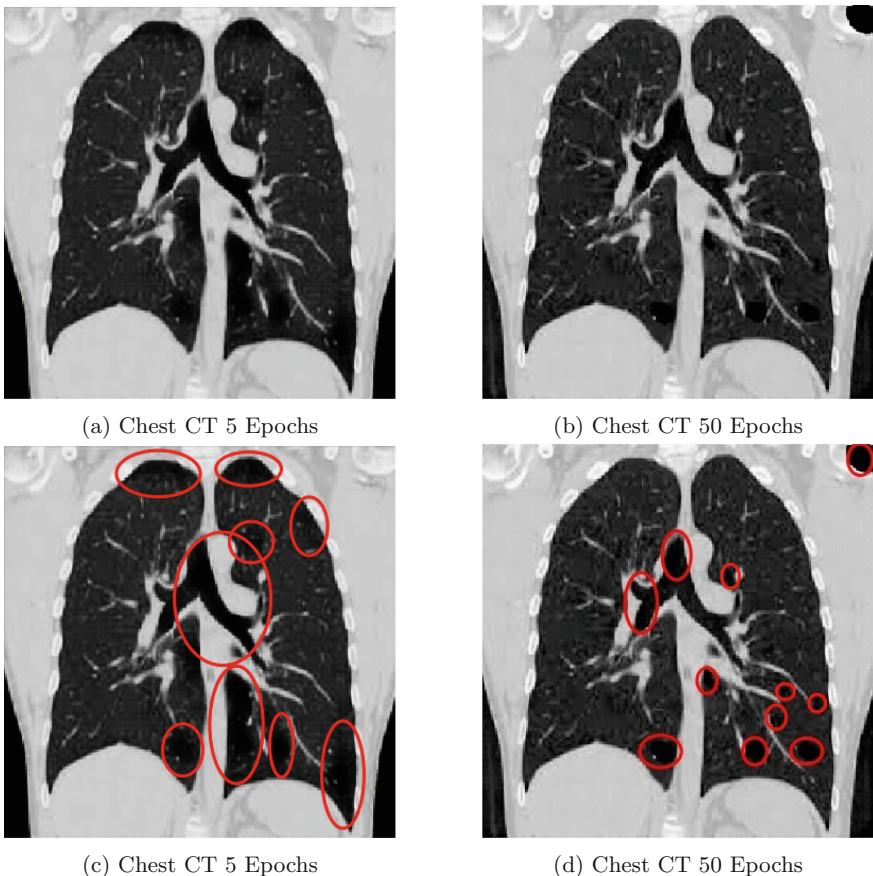


Fig. 6. Highlighted errors occurring after 5 and 50 epochs

out the training cycle with reaching the peak at 40 epochs. Two of the three training cycles achieved their best results with a score of 0.898626244257946 and 0.911978064824998. Therefore, it is beneficial to validate the results regularly and not presume that 50 epochs will automatically yield the best results every time and training is highly fluctuating where the same parameters can yield really good as well as really bad results. A lot of cases were the quality was above 15% in the beginning showed good results when trained for 50 epochs. The chosen threshold for UIQ performance above 15% were the highest recorded increases observed for the 2000 combinations trained. Contrary to other Generative Adversarial Networks, Pix2Pix does not use a noise vector as input. Instead it uses the image from domain A as an input and condition for the generator and is thus required to keep details from the images throughout training. Even without noise present for training, learning the distribution between input and

Table 4. Evaluation of parameter sets under the threshold

Parameter				UIQ (5 epochs)	UIQ (50 epochs)
Learning rate	Adam beta	Leaky alpha		Difference	Difference
		Discriminator	Encoder		
0.00035	0.45	0.06	0.2	-6.340807455137309	19.310198198
0.00015	0.7	0.06	0.2	-6.646971615100073	3.906560308
0.0002	0.45	0.02	0.25	2.1152737341727845	15.847274037
0.00025	0.65	0.02	0.2	0.5157123225243865	7.535355747
0.00035	0.7	0.03	0.2	-0.5772053284294976	-0.818741142
0.0001	0.35	0.06	0.75	-8.215397996743267	15.250770349
0.00015	0.7	0.07	0.15	-8.844959517903185	10.048540373
0.00025	0.45	0.06	0.15	7.112968267892084	0.061771457
0.00035	0.45	0.06	0.2	7.548824779843688	30.810785408
0.0001	0.35	0.03	0.45	-10.225888341511336	0,236863618
0.00045	0.4	0.05	0.8	13.36063958589131	-8.927112764
0.0001	0.4	0.04	0.15	-13.254805877990211	23.130491157
0.00015	0.6	0.03	0.45	3.4023786025367264	-3,682748416
0.00015	0.6	0.05	0.35	2.2737109428935587	11,012554483
0.0002	0.7	0.01	0.2	-4.287236086333667	-8,69692404

expected output is highly fluctuating as learning is different each time, leading to variance in the overall quality.

With fluctuating but overall good results above the 15% threshold, other scores under the current threshold should be evaluated as well to determine if the possibility of growth is higher when the score is lower in the beginning. Therefore, 10 additional parameter sets were selected at random and further extended with sets scoring similar to the randomly selected but overall scored near or way lower as the current threshold. Parameter pairs with a similar UIQ score were paired in the table for a better comparison. As shown in Table 4 the outcome when trained for 50 epochs is highly fluctuating with no clear indication if the training will be good or not.

As shown there are sometimes even better when compared with sets inside the defined threshold (1st row achieving 19%, 9th row 30% and 11th row 23%) scoring the highest scores seen in the complete evaluation. Unfortunately, there is not the same correlation as with the high scoring pairs, as seen in the presented table some of the presented combinations work extremely well while others with a similar UIQ score in the beginning do not perform nearly as well. Due to the high and unpredictable nature of these parameters sets, the chosen sets above the threshold even when not guaranteed to achieve a good result the odds are much higher although as seen it can mean to miss the best possible set of parameters. As previously mentioned parameter tuning for GANs is costly and time consuming. With previous testing limiting the search space saves time and resources. Another important factor is early stopping. Early stopping is important to further limit search time and allows the tradeoff between finding good results within a meaningful timeframe. But this also means that in most cases

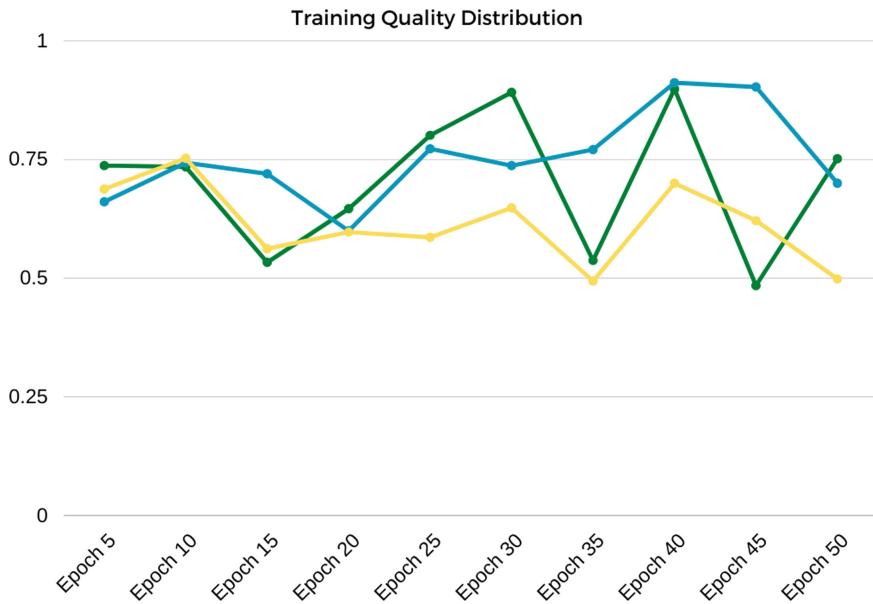


Fig. 7. Training quality distribution

the best parameter combination will not be found. In the above described use case one parameter combination took around 5 min with a rather small sample size. Bigger datasets will take increasingly longer to train and to determine the UIQ score. For the 5600 combination searching the given parameter space would take 20 days with our setup and with the fluctuating results that occur there is no given guarantee that the optimal solution is found and if the time needed is exceedingly higher when compared to the results and time requirements using early stopping. The achieved results justify early stopping to find the best possible result within a meaningful timeframe. In the above case it took around 12 h to find a combination with a 15% increase.

7 Conclusion

This paper proposes a time saving grid based search strategy with early stopping for Pix2Pix, evaluating the initial performance after 5 epochs with UIQ and determine a threshold with the highest chance of good training results after resuming training. After an initial training cycle to determine the performance with standard parameters used to calculate the baseline performance of Pix2Pix within the domain, each trained parameter set is evaluated against the baseline score and if the score is higher than 15% these parameters are used for further training with more epochs. In addition, the paper summarizes the performance achieved by parameter sets above and under the threshold. Although due to the randomness of Pix2Pix's generator as shown, the results can vary by a big

margin, but the overall results show a higher chance to preserve the obtained quality when the threshold was met after 5 epochs. To this effect, although the results are steady there is the possibility of missing the best parameter set. However the required effort to find and identify set is again time consuming and not guaranteed at all.

In our future work, we will explore additional tuning methods and add an approximation method for our currently used values, to explore the immediate neighbourhood and evaluate even better parameter possibilities, as well as include kernel size and stride into the parameter tuning process. Furthermore, the current method will be applied to several other domains with initially low scoring results, as well applied to use case specific transformations within the domain.

Acknowledgments. The authors would like to acknowledge the financial support from the German Federal Ministry of Research and Education (Bundesministerium für Bildung und Forschung) under grant CoHMed/PersonaMed A for this research.

References

1. Br 35H : Brain Tumor Detection 2020—Kaggle
2. Brain MRI Images for Brain Tumor Detection—Kaggle
3. Chest CT-Scan images Dataset—Kaggle
4. Chest X-Ray Images (Pneumonia)—Kaggle
5. Chen, T., Lucic, M., Houlsby, N., Gelly, S.: On self modulation for generative adversarial networks (2018). [arxiv:1810.01365](https://arxiv.org/abs/1810.01365)
6. Cho, H.-Y., Kim, Y.-H.: Stabilized training of generative adversarial networks by a genetic algorithm. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '19, pp. 51–52. Association for Computing Machinery, New York, NY, USA (2019)
7. Cho, H.-Y., Kim, Y.-H.: A genetic algorithm to optimize smote and gan ratios in class imbalanced datasets. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, GECCO '20, pp. 33–34. Association for Computing Machinery, New York, NY, USA (2020)
8. Costa, V., Lourenço, N., Correia, J., Machado, P.: Improved evolution of generative adversarial networks. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '21, pp. 145–146. Association for Computing Machinery, New York, NY, USA (2021)
9. Distante, A., Distante, C.: Handbook of Image Processing and Computer Vision: volume 1: from Energy to Image (2020)
10. Dumont, V., Ju, X., Mueller, J.: Hyperparameter optimization of generative adversarial network models for high-energy physics simulations (2022)
11. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., (eds.) Advances in Neural Information Processing Systems, vol 27, pp. 2672–2680. Curran Associates Inc., (2014)
12. Shuyang, G., Bao, J., Chen, D., Wen, F.: Generated image quality assessment, Giqa (2020)

13. He, B., Kita, E.: Ga-based optimization of generative adversarial networks on stock price prediction. In: 2021 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 199–202 (2021)
14. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local Nash equilibrium. Technical report (2017)
15. Hölscher, D., Reich, C., Knahl, M., Gut, F., Clarke, N.: Surface quality augmentation for metalworking industry with pix2pix. In: Procedia Computer Science, Knowledge-Based and Intelligent Information & Engineering Systems: proceedings of the 26th International Conference KES2022, vol. 207, pp. 897–906 (2022)
16. Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5967–5976
17. Korde, C.G., Reddy, K.M., Vasantha, M.H., Nithin Kumar, Y.B.: Training of generative adversarial networks with hybrid evolutionary optimization technique. In: 2019 IEEE 16th India Council International Conference (INDICON), pp. 1–4 (2019)
18. Kurach, K., Lucic, M., Zhai, X., Michalski, M., Gelly, S.: Losses, architectures, regularization, and normalization, The GAN landscape (2019)
19. Ledig, C., Theis, L., Huszar, F., Caballero, J., Aitken, A.P., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network (2016). [arxiv:1609.04802](https://arxiv.org/abs/1609.04802)
20. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: International Conference on Learning Representations (2018)
21. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: Meila, M., Zhang, T., (eds.) Proceedings of the 38th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 139, pp. 8821–8831. PMLR (2021)
22. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation (2015). [arxiv:1505.04597](https://arxiv.org/abs/1505.04597)
23. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. In: Advances in Neural Information Processing Systems, pp. 2234–2242 (2016)
24. Sammut, C., Webb, G.I., (eds.): Mean Absolute Error, pp. 652–652. Springer US, Boston, MA (2010)
25. Wan, Z., He, H., Tang, B.: A generative model for sparse hyperparameter determination. IEEE Trans. Big Data **4**(1), 2–10 (2018)
26. Wang, Z., Bovik, A.C.: A universal image quality index. IEEE Signal Process. Lett. **9**(3), 81–84 (2002)
27. Xiao, X., Yan, M., Basodi, S., Ji, C., Pan, Y.: Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm (2020). [arxiv:2006.12703](https://arxiv.org/abs/2006.12703)
28. Xiao, X., Yan, M., Basodi, S., Ji, C., Pan, Y.: Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm (2020). [arxiv:2006.12703](https://arxiv.org/abs/2006.12703)
29. Yao, C., Cai, D., Jiajun, B., Chen, G.: Pre-training the deep generative models with adaptive hyperparameter optimization. Neurocomputing **247**, 144–155 (2017)
30. Zhang, L., Zhang, L., Mou, X., Zhang, D.: FSIM: a feature similarity index for image quality assessment. IEEE Trans. Image Process. **20**(8), 2378–2386 (2011)

31. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
32. Zhu, J.-Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks (2017). [arxiv:1703.10593](https://arxiv.org/abs/1703.10593)

Author Index

A

- Abdelhak, Benhamada 360
Abdullah, Afsah 597
Agostinho, Carlos 201
Ahiagble, Agbodze Pascal 75
Ahmed, Ishrat 542
Alcântara, André L. M. 820
Alférez, Germán H. 617
Ali, Zafar 477
Alimardani, Maryam 186
Aliyev, Murad 59
Alizada, Parvin 102
Alkandari, Dhari 597
Alsaber, Ahmad 597
Antal, Péter 386
Askounis, Dimitris 201
Atat, Rachad 234
Avikhana, Maya 174

B

- Bahamonde, Diego 496
Balaneji, Farshid 577
Barbato, Jessica Amianto 84
Battiatto, Sebastiano 837
Bense, Hermann 326
Blom, Fenna 186
Botero-Valencia, Juan S. 265
Büchi, Roland 735

C

- Caligiuri, Luigi Maximilian 799
Cao, Zheng 849
Chandra Jetty, Rajiv 642
Cheddad, Abbas 1
Cheddad, Zohra Adila 1
Chung, Chong Chee 129
Clarke, John-Paul 413
Clarke, Nathan 862

- Coelho, Francisco J. S. 820
Cremaschi, Marco 84

D

- Da Col, Giacomo 508
Damadi, Saeed 223
Davis, Katherine R. 234
Deck, Andy 400
Doush, Iyad Abu 597
Duin, Peter 145

E

- Esteves, Antonio 36

F

- Fabri, Lina 174
Faulds, Anthony 674
Feitosa, Allan R. S. 820
Figueiras, Paulo 201

G

- Gabriel, Mina 780
Gao, Jiayi 849
Gdanitz, Natalie 75
Giner, Fernando 692
Grau, Isel 745
Grisanti, Marco 837
Guo, Raymond 849
Gut, Frank 862

H

- Habibbayli, Tunjay 59
Haq, Kalim Ul 477
Hara, Shoichiro 301
Heiden, Bernhard 448
Hernández-García, Ruber 265
Hidalgo, Rafael 642
Hoitsma, Fabian 745

Hölscher, Dirk 862

Hosier, Jordan 542

Hýbl, Ac 617

I

Iqbal, Akif 284

Isik, Murat 212

Ismail, Muhammad 234

J

Jagadeesh, George Rosario 129

Janzen, Sabine 75

Jetty, Anupama 642

K

Kadoic, Nikola 371

Kefalas, Pavlos 477

Keita, Khadidiatou Wane 316

Kejriwal, Mayank 714

Kendall, Andrew 413

Khaliq, Lotfy H. Abdel 75

Khayut, Ben 174

Khodabakhshian, Ania 521

Kim, Yong Seog 17

Knahl, Martin 862

Kompella, Ramana 780

Kontzinos, Christos 201

Kornienko, Jurijs 345

Koumeri, Lisa Koutsoviti 745

Kunal, Thakkar 129

Mouzakitis, Spiros 201

Murphy, Emma 607

N

Nápoles, Gonzalo 745

Nikiforova, Oksana 345

O

O'Sullivan, Dympna 607

Ogunsakin, Rotimi 433

Oldland, Matthew 212

Oreski, Dijana 371

Ortis, Alessandro 837

P

P, Vishal 284

Petrosian, Ovanes 659

Pilosta, Bruno 371

Pock, Mark 849

Q

Qi, Dongfang 659

R

Re Cecconi, Fulvio 521

Reich, Christoph 862

Rönnback, Ronja 186

Rula, Anisa 84

Rzayev, Ramin 59, 102

S

Salah, Nesreen 642

Sándor, Dániel 386

Sarafanov, Egor 245

Sarasvathi, V. 284

Sarkar, Anupam 477

Serpedin, Erchin 234

Sharma, Nikhita 542

Shen, Jinglai 223

Shen, Ke 714

Sheridan, Helen 607

Silva-Filho, Abel G. 820

Souare, Ibrahima 316

Spasić, Irena 577

Spatafora, Maria Ausilia Napoli 837

Srinivasan, Sidarth 780

Sultan, Khalid 597

Sun, Qiushi 659

M

Ma, Jing Liu Ruimin 659

Maass, Wolfgang 75

Maringer, Dietmar 577

Marios, Christonasios Antonios 145

Maurino, Andrea 84

Mehdi, Riyadh A. K. 765

Mehdiyev, Tahir 102

Mejia-Herrera, Mateo 265

Mohammed, Kherarba 360

Morillo, Paulina 496

T

- Takeuchi, Jessica T. 820
Takiddin, Abdulrahman 234
Tapia, Wilian 496
Teng, Teck-Hou 129
Teppan, Erich 508
Thierry, Nimbeshaho 477
Thórisson, Kristinn R. 780
Tonino-Heiden, Bianca 448
Tuguinay, Caesar M. 849

U

- Ullah, Irfan 477
Uppara, Veeresh 284

V

- Valente, Nuno 36
Valilai, Omid Fatahi 245
van Eijndhoven, Stef 145
Vancoof, Koen 745
Varde, Aparna S. 642
Vinay, M. V. 284

W

- Wang, Pei 780
Wang, Ziyu 849
Watanabe, Yuji 301
Wicaksono, Hendro 245
Wiese, Lena 156

X

- Xu, Mohan 156
Xue, Huiwen 561

Y

- Yuan, Zhengqing 561

Z

- Zabiniako, Vitaly 345
Zhang, Chao 561
Zhang, Yuyi 659
Zheng, Jiageng 552
Zhou, Lifeng 212
Zhou, Yu 542
Zhu, Qinan 433