

Rozaida Ghazali · Nazri Mohd Nawi ·
Mustafa Mat Deris · Jemal H. Abawajy ·
Nureize Arbaiy *Editors*

Recent Advances on Soft Computing and Data Mining

Proceedings of the Sixth International
Conference on Soft Computing and Data
Mining (SCDM 2024), August 21–22,
2024

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, EI Compendex, 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).

Rozaida Ghazali · Nazri Mohd Nawi ·
Mustafa Mat Deris · Jemal H. Abawajy ·
Nureize Arbaiy
Editors

Recent Advances on Soft Computing and Data Mining

Proceedings of the Sixth International Conference on Soft Computing and Data Mining (SCDM 2024), August 21–22, 2024

SCDM2024



Springer

Editors

Rozaida Ghazali

Faculty of Computer Science and Information
Technology

Universiti Tun Hussein Onn Malaysia

Parit Raja, Malaysia

Nazri Mohd Nawi

Faculty of Computer Science and Information
Technology

Universiti Tun Hussein Onn Malaysia

Parit Raja, Malaysia

Mustafa Mat Deris

Faculty of Business and Information
Technology

Universiti Muhammadiyah Malaysia

Padang Besar, Perlis, Malaysia

Jemal H. Abawajy

School of Information Technology
Deakin University

Wandana Heights, VIC, Australia

Nureize Arbaiy

Faculty of Computer Science and Information
Technology

Universiti Tun Hussein Onn Malaysia

Parit Raja, Johor, Malaysia

ISSN 2367-3370

ISSN 2367-3389 (electronic)

Lecture Notes in Networks and Systems

ISBN 978-3-031-66964-4

ISBN 978-3-031-66965-1 (eBook)

<https://doi.org/10.1007/978-3-031-66965-1>

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

If disposing of this product, please recycle the paper.

Preface

Advancements in data storage and accessibility have fueled the growth of data science, which employs various methods to extract insights and patterns from data. Despite the demand for skilled data scientists, extracting actionable insights remains challenging, particularly with complex data systems. To address this, data mining has emerged as a crucial approach, offering potential for discovering patterns across diverse data types. By leveraging data and soft computing techniques, researchers can explore extensive databases to uncover hidden patterns. Ongoing research focuses on developing advanced statistical interpretations and innovative technologies. Soft computing techniques address imprecision and uncertainty, enhancing tractability and robustness. These techniques, whether used individually or combined, are emerging as robust options for various tasks in fields such as business and marketing, healthcare, finance, e-commerce, manufacturing, telecommunications, transportation, environmental science, agriculture, education, and more. They aim to transform data into innovative solutions that offer new value propositions for customers.

Following the successful organization of five previous SCDM conferences from 2014 to 2022, we are pleased to continue this journey of achievements with our sixth international conference, SCDM 2024. This year's conference, held in a virtual format on August 21–22, 2024, facilitated global participation, providing live interactive networking opportunities and content access to attendees worldwide. We received 75 paper submissions from 15 countries, each of which underwent rigorous screening and peer-review processes. Ultimately, 42 papers of the highest quality and merit were selected for oral presentation and publication in this volume proceeding, representing an acceptance rate of 56%.

We would like to express our sincere appreciation to the conference organizer, Faculty of Computer Science & Information Technology, UTHM, and the Soft Computing & Data Mining research group, as well as to the Steering Committee, Conference Chair, Program Committee Chair, Organizing Chairs, and all Program and Reviewer Committee members of SCDM 2024. Their invaluable contributions to the review process have ensured the highest quality of selected papers for the conference.

We also extend our appreciation to our esteemed keynote speakers, Associate Professor Dr. Harish Garg from the Thapar Institute of Engineering & Technology, Punjab, India, and Mr. Azhar Kassim Mustapha from Nervesis, Malaysia. Special thanks are due to Dr. Thomas Ditzinger for facilitating the publication of the proceeding in Lecture Notes in Networks and Systems, Springer. We acknowledge the Organizing Committee members for their significant contributions, particularly those in pivotal roles.

Finally, we want to extend our heartfelt appreciation to all authors for their valuable contributions and to all participants for their enthusiastic engagement. We are truly

grateful for your dedication and commitment, which have greatly contributed to the success of this conference.

Rozaida Ghazali
Nazri Mohd Nawi
Mustafa Mat Deris
Jemal H. Abawajy
Nureize Arbaiy

Conference Organization

Patron

Ruzairi bin Abdul Rahim

Vice Chancellor, Universiti Tun Hussein Onn
Malaysia, Malaysia

Advisory Committee

Ajith Abraham

Machine Intelligence Research Labs, USA

Hamido Fujita

Iwate Prefectural University, Japan

Junzo Watada

Waseda University, Japan

Nikola Kasabov

KEDRI, Auckland University of Technology,
New Zealand

Rajkumar Buyya

University of Melbourne, Australia

Witold Pedrycz

University of Alberta, Canada

Steering Committee

Mustafa Mat Deris

Universiti Tun Hussein Onn Malaysia

Jemal H Abawajy

Deakin University, Australia

Nazri Mohd Nawi

Universiti Tun Hussein Onn Malaysia

Rozaida Ghazali

Universiti Tun Hussein Onn Malaysia

Hairulnizam Mahdin

Universiti Tun Hussein Onn Malaysia

Conference Chair

Nazri Mohd Nawi

Universiti Tun Hussein Onn Malaysia

Conference Co-chair

Norhalina Senan

Universiti Tun Hussein Onn Malaysia

Proceeding Chair

Rozaida Ghazali

Universiti Tun Hussein Onn Malaysia

Program Committee Chair

Nureize Arbaiy

Universiti Tun Hussein Onn Malaysia

Website, Promotion, and Publicity Chair

Mohd Norasri Ismail

Universiti Tun Hussein Onn Malaysia

Organizing Committee

Nurezayana Zainal

Universiti Tun Hussein Onn Malaysia

Zuraida Bosri

Universiti Tun Hussein Onn Malaysia

Suziyanti Marjudi

Universiti Tun Hussein Onn Malaysia

Rabatul Aduni Sulaiman

Universiti Tun Hussein Onn Malaysia

Zam Zarina Zainal Abidin

Universiti Tun Hussein Onn Malaysia

Mohd Zanes Sahid

Universiti Tun Hussein Onn Malaysia

Shahreen Kasim

Universiti Tun Hussein Onn Malaysia

Norashid Hassan

Universiti Tun Hussein Onn Malaysia

Mohd Helmy Abd Wahab

Universiti Tun Hussein Onn Malaysia

Program Committee

Adila Firdaus Arbain

Universiti Teknologi Malaysia

Adnan Abid

University of Management and Technology,
Pakistan

Afnizanfaizal Abdullah

Universiti Teknologi Malaysia

Agouti Tarik

Cadi Ayyad University, Morocco

Ahmed A. Elngar

Beni-Suef University, Egypt

Aida Mustapha

Universiti Tun Hussein Onn Malaysia

Alessandro D'Amelio

University of Milan, Italy

Alessandro Giuliani

University of Cagliari, Italy

Ali Ahmadian

Universiti Putra Malaysia

Ali Mohammadi

Isfahan University of Technology, Iran

Amelia Zafra Gomez

University of Cordoba, Spain

Ammar Awad Mutlag	Universiti Teknikal Malaysia Melaka
Ayodele Lasisi	Augustine University, Nigeria
Bazeer Ahamed Bagrudeen	University of Technology and Applied Sciences Al Musannah, Oman
Carlos Pereora	ISEC, Portugal
Chuah Chai Wen	Universiti Tun Hussein Onn Malaysia
Cik Feresa Mohd Foozy	Universiti Tun Hussein Onn Malaysia
Daoudi Najima	ESI, Morocco
El Habib Benlahmar	University of Hassan II, Casablanca, Morocco
Elena Benderskaya	St.Petersburg State Polytechnic University, Russia
Ender Özcan	University of Nottingham, UK
Ezak Ahmad	Universiti Tun Hussein Onn Malaysia
Fairouz Zendaoui	Institut National de la Poste et des TIC, Eucalyptus, Alger, Algérie
Fatima Zahra Fagroud	Hassan II University, Casablanca, Morocco
Hanaa Hachimi	Ibn Tofail University, Morocco
Hardeo Kumar Thakur	Manav Rachna University, Faridabad, India
Hazlina Hamdan	Universiti Putra Malaysia
Isredza Rahmi Ab Hamid	Universiti Tun Hussein Onn Malaysia
Jawad Ali	COMSATS University ISB Lahore Campus, Pakistan
José Ramón Villar	University of Oviedo, Spain
Jyotir Moy Chatterjee	Lord Buddha Education Foundation, Kathmandu, Nepal
Kashif Hussain	University of Electronic Science and Technology of China
Katsuhiro Honda	Osaka Prefecture University, Japan
Khalil Ghathwan	University of Technology, Iraq
Mario José Diván	National University of La Pampa (UNLPam), Argentina
Maslina Zolkepli	Universiti Putra Malaysia
Md. Raihan Uddin	Daffodil International University, Bangladesh
Mohamad Aizi Salamat	Universiti Tun Hussein Onn Malaysia
Mohammad Zubair Rehman	Agriculture University Peshawar, Pakistan
Mohd Amin Mohd Yunus	Universiti Tun Hussein Onn Malaysia
Mohd. Farhan Md. Fudzee	Universiti Tun Hussein Onn Malaysia
Mohd. Najib Mohd. Salleh	Universiti Tun Hussein Onn Malaysia
Mohit Jain	NSIT (University of Delhi), India
Mouad Banane	University Hassan 2, Morocco
Muhammad Adnan Khan	Riphah International University, Pakistan
Muhammad Faheem Mushtaq	The Islamia University of Bahawalpur, Pakistan
Nadjet Kamel	University Ferhat Abbas Setif 1, Algeria

Noor Azah Samsudin	Universiti Tun Hussein Onn Malaysia
Noor Zuraidin Mohd Safar	Universiti Tun Hussein Onn Malaysia
Nor Azura Husin	Universiti Putra Malaysia
Nordiana Rahim	Universiti Tun Hussein Onn Malaysia
Norfaradilla Wahid	Universiti Tun Hussien Onn Malaysia
Noryusliza Abdullah	Universiti Tun Hussein Onn Malaysia
Nureize Arbaiy	Universiti Tun Hussein Onn Malaysia
Nurezayana Zainal	Universiti Tun Hussein Onn Malaysia
Okan Duru	Nanyang Technological University, Singapore
Oumaima Hourrane	University of Hassan II, Casablanca, Morocco
Palaniappan Shamala	Universiti Teknologi MARA
Pei-Chun Lin	Feng Chia University, Taiwan
Pramit Brata Chanda	Kalyani Government Engineering College, Kalyani
Rabei Al-Jawary	American University of Ras Al Khaimah, UAE
Rachid Saadane	LETI, EHTP, Morocco
Radiah Mohamad	Universiti Tun Hussein Onn Malaysia
Radzi Ambar	Universiti Tun Hussein Onn Malaysia
Rahayu Hamid	Universiti Tun Hussein Onn Malaysia
Rahmat Hidayat	Politeknik Negeri Padang, Indonesia
Rajdeep Chowdhury	West Bengal, India
Riyaz Ahamed	International University of Malaya-Wales
Sadiq Abdelalim	Ibn Tofail University, Morocco
Salama A. Mostafa	Universiti Tun Hussein Onn Malaysia
Sanaa El Filali	Hassan II University, Casablanca, Morocco
Saoud Sahar	National Business School, Ibn Zohr University, Agadir, Morocco
Sasalak Tongkaw	Songkhla Rajabhat University, Thailand
Sathya Bursic	University of Milan, Italy
Shahreen Kasim	Universiti Tun Hussein Onn Malaysia
Shamsollah Ghanbari	Islamic Azad University, Ashtian Branch, Iran
Sofia Najwa Ramli	Universiti Tun Hussein Onn Malaysia
Suhaimi Abd Ishak	Universiti Tun Hussein Onn Malaysia
Szymon Lukasik	Cracow University of Technology, Poland
Tadashi Nomoto	National Institute of Japanese Literature, Japan
Uma N. Dulhare	Muffakham Jah College of Engineering & Technology (MJCET), Hyderabad, India
Venkatesh Gauri Shankar	Manipal University Jaipur, India
Vitalii Nitsenko	Odessa I.I.Mechnikov National University, Odessa, Ukraine
Vittorio Cuculo	University of Milan, Italy
Waddah Waheed	University of Agder, Norway

Wamiq Raza	University of Trento, Italy
Waseem Mohssen Alhasan	Al-Sham Private University, Syria
Yana Mazwin Mohmad Hassim	Universiti Tun Hussein Onn Malaysia
Youness Tabii	Labo ADMIR–ENSIAS UM5 Rabat, Morocco
Zubaile Abdullah	Universiti Tun Hussein Onn Malaysia

Organizer

Faculty of Computer Science & Information Technology, Universiti Tun Hussein Onn Malaysia



Contents

Prediction of OPEC Carbon Dioxide Emissions Using <i>K</i> -Means Clustering and Ensemble Algorithm	1
<i>Ayodele Lasisi, Nur Ariffin Mohd Zin, Rozaida Ghazali, and Modupe Agagu</i>	
Detection of Phishing Websites from URLs Using Hybrid Ensemble-Based Machine Learning Technique	11
<i>Modupe Agagu, Ibrahim Abayomi Ogunbiyi, Ayodele Lasisi, and Osaremwinda Omorogiuwa</i>	
Minimal Data for Maximum Impact: An Indonesian Part-of-Speech Tagging Case Study	23
<i>Chi Log Chua, Tong Ming Lim, and Kwee Teck See</i>	
Alleviating Sparsity to Enhance Group Recommendation with Cross-Linked Domain Model	33
<i>Yui Chee Xuan, Rosmamalni Mat Nawi, Nurul Aida Osman, and Nur Ziadah Harun</i>	
Evaluating Deep Transfer Learning Models for Detecting Various Face Mask Wearings	43
<i>Pei-Jin Goh, Meei-Hao Hoo, and Kok-Chin Khor</i>	
Classification of Stunting Events: Case Study in West Java, Indonesia	53
<i>Ummi Azizah Rachmawati, Puspa Setia Pratiwi, Yusnita, K. Rama Abirami, and Farrel Yuda Praditya</i>	
The Effects of Data Reduction Using Rough Set Theory on Logistic Regression Model	64
<i>Izzati Rahmi, Riswan Efendi, Nor Azah Samat, Hazmira Yozza, and Muhammad Wahyudi</i>	
Robust Heart Disease Prognosis: Integrating Extended Isolation Forest Outlier Detection with Advanced Prediction Models	74
<i>Irfan Javid, Norlida Hassan, Rozaida Ghazali, Yana Mazwin Mohmad Hassim, Tuba Batool, Noor Aida Husaini, and Syed Irteza Hussain Jafri</i>	

Overlapping Granular Clustering: Application in Fuzzy Rule-Based Classification	84
<i>Muhammad Zaiyad Muda and George Panoutsos</i>	
Improved Rough-Multiple Regression for Unemployment Rate Model in Indonesia	94
<i>Riswan Efendi, Mazidah Mat Rejab, Nureize Arbaiy, Widya T. Yofi, Sri R. Widyawati, Izzati Rahmi, and Hazmira Yozza</i>	
Utilizing Machine Learning for Gene Expression Data: Incorporating Gene Sequencing, K-Mer Counting and Asymmetric N-Grams Features	105
<i>Chai-Wen Chuah, WanXian He, De-Shuang Huang, and Janaka Alawatugoda</i>	
Text Sentiment Analysis on VIX's Impact on Market Sentiment Dynamics	115
<i>Zhuqin Liang, Mohd Tahir Ismail, and Huimin Qu</i>	
Multilevel Monte Carlo Simulation Model for Air Pollution Index Prediction of a Smart Network	125
<i>Mustafa Hamid Hassan, Salama A. Mostafa, Rozaida Ghazali, Mohd Zainuri Saringat, Noor Aida Husaini, Aida Mustapha, Mohammed Ahmed Jubair, and Hussein Muhi Hariz</i>	
An In-Depth Strategy using Deep Generative Adversarial Networks for Addressing the Cold Start in Movie Recommendation Systems	136
<i>Muhammad Shahab, Yana Mazwin Mohmad Hassim, Rozaida Ghazali, Irfan Javid, and Nureize Arbaiy</i>	
Predicting Undergraduate Academic Success with Machine Learning Approaches	144
<i>Yuan-Zheng Li, Keng-Hoong Ng, Kok-Chin Khor, and Yu-Hsuen Lim</i>	
Comparative Assessment of Facial Expression Recognition Models for Unraveling Emotional Signals with Convolutional Neural Networks	154
<i>Afia Zafar, Nazri Mohd Nawi, Noushin Saba, Kainat Zafar, Mohsin Suleman, and Shahneer Zafar</i>	
Evaluating Path-Finding Algorithms for Real-Time Route Recommendation System Built using FreeRTOS	165
<i>Jun-Yen Liew, Keng-Hoong Ng, Kok-Chin Khor, and Kai-Yau Tee</i>	
Machine Learning-Based Phishing Website Detection: A Comparative Analysis and Web Application Development	175
<i>Jia Xin Yau and Kai Lin Chia</i>	

Comparative Performance of Multi-level Pre-trained Embeddings on CNN, LSTM and CNN-LSTM for Hate Speech and Offensive Language Detection	186
<i>Noor Azeera Abdul Aziz, Anazida Zainal, Bander Ali Saleh Al-Rimy, and Fuad Abdulgaleel Abdoh Ghaleb</i>	
Improved Classifier Chain Method Based on Particle Swarm Optimization and Genetic Algorithm for Multilabel Classification Problem	196
<i>Abdullahi O. Adeleke, Noor A. Samsudin, Shamsul Kamal A. Khalid, and Riswan Efendi</i>	
Sentiment Analysis on Umrah Packages Review in Malaysia	207
<i>Deshinta Arrova Dewi, Tri Basuki Kurniawan, Mohd Zaki Zakaria, Shahreen Kasim, and Nur Qasheeh Mustapa</i>	
Opinion Mining System for Influence Detection Using Machine Learning to Secure Business Reputation	219
<i>Shahrinaz Ismail and Kyi Lin Khant</i>	
A Presentation Mining Framework: From Text Mining to to Mind Mapping	233
<i>Vinothini Kasinathan and Aida Mustapha</i>	
Enhancing Network Intrusion Detection Systems Through Dimensionality Reduction	244
<i>Mosleh M. Abualhaj, Sumaya N. Al-Khatib, Ali Al-Allawee, Alhamza Munther, and Mohammed Anbar</i>	
Performance Evaluation of Whale and Harris Hawks Optimization Algorithms with Intrusion Prevention Systems	254
<i>Mosleh M. Abualhaj, Ahmad Adel Abu-Shareha, Ali Al-Allawee, Alhamza Munther, and Mohammed Anbar</i>	
Domestic Solid Waste Prediction with an Enhanced LSTM with SigmoReLU and RAdam Optimizer	266
<i>Abdulrahman Sharaf Mohammed Fadhel, Rozaida Ghazali, Mohd Razali Md Tomari, Yana Mazwin Mohamad Hassim, Abdullahi Abdi Abubakar Hassan, and Lokman Hakim Ismail</i>	
Sounds Prediction Instruments Based Using K-Means and Bat Algorithm	276
<i>Rozlini Mohamed, Noor Azah Samsuddin, and Munirah Mohd Yusof</i>	
A Comparative Study on Ant-Colony Algorithm and Genetic Algorithm for Mobile Robot Planning	286
<i>Piraviendran al/ Rajendran and Muhamani Othman</i>	

Enhanced Air Quality Index Prediction Using a Hybrid Convolutional Network	296
<i>Pei-Chun Lin, Nureize Arbaiy, Chen-Yu Yu, and Mohd Zaki Mohd Salikon</i>	
Filter Method Feature Selection Techniques for Solid Waste Prediction Based on GRU Deep Learning Model	307
<i>Tuba Batool, Siti Hajar Arbain, Rozaida Ghazali, Lokman Hakim Ismail, and Irfan Javid</i>	
Spiking Neural Network for Microseismic Events Detection Using Distributed Acoustic Sensing Data	317
<i>Mohd Safuan Bin Shahabudin, Nor Farisha Binti Muhamad Krishnan, and Farahida Hanim Binti Mausor</i>	
Battery Electric Vehicle Charging Load Forecasting Using LSTM on STL Trend, Seasonality, and Residual Decomposition	327
<i>Syahrizal Salleh, Roslinazairimah Zakaria, and Siti Roslindar Yaziz</i>	
Convolutional Neural Network Using Regularized Conditional Entropy Loss (CNNRCoE) for MNIST Handwritten Digits Classification	337
<i>Ashikin Ali, Norhalina Senan, and Norhanifah Murli</i>	
Optimizing Team Formation for Welfare Activities: A Study Using Four Metaheuristic Optimization Algorithms	349
<i>Muhammad Akmaluddin and Rozlina Mohamed</i>	
Detection of Paddy Plant Diseases Using Google Teachable Machine	360
<i>Nor Azuana Ramli, Agus Pratondo, Sahimel Azwal Sulaiman, Wan Nur Syahidah Wan Yusoff, and Noratikah Abu</i>	
Comparative Analysis of ResNet Models for Skin Cancer Diagnosis: Performance Evaluation and Insights	370
<i>Razan Alharith, Ashraf Osman Ibrahim, Noorhaniza Wahid, Rozaida Ghazali, and Abubakar Elsafi</i>	
The Predictive Modelling of Student Academic Performance Using Machine Learning Approaches	379
<i>Nurul Habibah Abdul Rahman, Sahimel Azwal Sulaiman, and Nor Azuana Ramli</i>	

Predictive Modeling of Gold Prices: Integrating Technical Indicators for Enhanced Accuracy	390
<i>Noor Aida Husaini, Yee Jing Gan, Rozaida Ghazali, Yana Mazwin Mohamad Hassim, Jie Shen Yeap, and Jerome Subash Joseph</i>	
Portfolio Optimization with Percentage Error-Based Fuzzy Random Data for Industrial Production	400
<i>Mohammad Haris Haikal Othman, Nureize Arbaiy, Muhammad Shukri Che Lah, and Pei-Chun Lin</i>	
The Football Matches Outcome Prediction for English Premier League (EPL): A Comparative Analysis of Multi-class Models	411
<i>Nur Amirah Adnan, Luqman Al Hakim Mohd Asri, Aida Mustapha, and Muhammad Nazim Razali</i>	
An Automated Quasi-Identification (QID) for Re-identification	421
<i>Saida Nafisah Roslan, Isredza Rahmi A Hamid, Abdulbasit A. Darem, and Nordiana Rahim</i>	
Correction to: Predicting Undergraduate Academic Success with Machine Learning Approaches	C1
<i>Yuan-Zheng Li, Keng-Hoong Ng, Kok-Chin Khor, and Yu-Hsuen Lim</i>	
Author Index	433



Prediction of OPEC Carbon Dioxide Emissions Using *K*-Means Clustering and Ensemble Algorithm

Ayodele Lasisi¹(✉) , Nur Ariffin Mohd Zin², Rozaida Ghazali², and Modupe Agagu³

¹ Department of Computer Science, College of Computer Science,
King Khalid University, Abha, Saudi Arabia
alasisi@kku.edu.sa

² Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 86400 Batu Pahat, Parit Raja, Johor, Malaysia
{ariffin,rozaida}@uthm.edu.my

³ Department of Computer Science, Olusegun Agagu University of Science and Technology, Okitipupa, Ondo, Nigeria
m.agagu@oaustech.edu.ng

Abstract. The rapid increase in the concentration of carbon dioxide (CO_2) polluting the atmosphere induces global warming and climate change. This is detrimental to human health and their natural habitat. Thus, it is imperative to proffer measures in analyzing and predicting the emissions of CO_2 . This research suggests using an ensemble approach with fuzzy nearest neighbor, sequential minimal optimization, and logistic regression to predict global CO_2 emissions. The *K*-means algorithm divides data into groups of similar and relevant patterns. Simulation findings show that the proposed model outperforms techniques such as multi-layer perceptron, fuzzy ownership nearest neighbor, and random forest. It also improves CO_2 forecast accuracy.

Keywords: Ensemble algorithm · Carbon dioxide emissions · Fuzzy nearest neighbor · Sequential minimal optimization · Logistic regression

1 Introduction

The effect of greenhouse gases serve as a major deterrent to the growth of an economy and its activities namely; agriculture yields, industrial productivity, populations, and immigration [1]. This rest solely on its most dominant constituent, carbon dioxide, (CO_2), leading to global warming. Carbon dioxide (CO_2) accommodates 90% of greenhouse effect, that affirms it as the critical to climate change [2]. In 2023, global energy-related (CO_2) emissions increased by 1.1%, totaling 410 million tonnes (Mt) and setting a new record high of 37.4 billion tonnes. [3]. Therefore, paramount attention has been directed to the reduction of greenhouse gases which ultimately reduce the emission of CO_2 .

Tapping into the advantages of ensemble methods, this paper proposes the concatenation of fuzzy nearest neighbor, sequential minimal optimization, and logistic regression in accurate prediction of CO₂ emissions. It involves two classification layers where the base classifiers occupies the first layer, and meta classifier in the second layer (this gives the final prediction). The K -means clustering approach partitions the dataset into clusters, making it suitable for training and testing algorithms. The paper's primary contributions include the following:

- A stacking ensemble learning is designed for the prediction of CO₂ emissions.
- The combination of each classifier is often advantageous. This fosters prediction accuracy.
- The clustering technique of K -means offers strategic groupings of dataset in clusters.
- The proposed method is capable of efficiently forecasting emissions released from CO₂.

The rest of this paper is organized as follows: The summary of related literature is provided in Sect. 2. The concept of fuzzy nearest neighbor is presented in Sect. 3. Embedded in Sect. 4 and 5 are the sequential minimal optimization and logistic regression. Flow process of K -means as used for clustering occupies Sect. 6. The proposed methodology emerges in Sect. 7. Experimental setup and analysis of are in Sect. 8. Conclusion is summarized in Sect. 9.

2 Related Works

The reliance on computational intelligent algorithms as used in [4–7], due to their adaptability, robustness, and superiority provide the pathway to apply these algorithm to effectively reduce CO₂ emissions. Some as found in literature include the adoption of Gaussian process regression (GPR) with modified particle swarm optimization (PSO) for forecasting CO₂ emissions [1]. The PSO is optimized with differential evolution (DE) to maintain diversity and prevent trapping in local optimum. The optimized PSO-DE combines with GPR for enhancing its hyper parameters of covariance function. Experimental results on CO₂ data from three countries shows that PSO-GPR provides better performance when compared with GPR and back propagation neural network (BPNN). The PSO also played a vital role in optimizing the weights and bias of extreme learning machine (ELM) for carbon dioxide prediction [8]. A bivariate correlation analysis was used for influential factor analysis, while the optimized PSO-ELM predicts CO₂ data in China from 1995 to 2014. Comparison analysis with ELM and BPNN revealed that PSO-ELM produced superior predictions. In continuation of their research, the authors in [8] used random forest for influential factor analysis, and improved ELM by optimizing with moth-flame optimization (MFO) algorithm [9]. Extensive experimental procedures of the proposed RF-MFO-ELM compared with MFO-ELM, RF-MFO-ELM, RF-ELM, and RF-BP concludes that RF-MFO-ELM had higher predictive CO₂ results and accuracy to others.

A hybrid cuckoo search (CS) algorithm with neural network was proposed for predicting OPEC CO₂ emissions [10]. First, an integration of CS with accelerated PSO (APSO) was performed to boost the searching capabilities of CS. This was termed HCS. Thereafter, the HCS was applied to optimize neural network called HCSNN. Performance results proved that HCSNN is accurately efficient to the compared algorithms. The Salp Swarm Algorithm (SSA) was used to optimize the hyper-parameters of least squares support sector machine to anticipate energy-related CO₂ emissions [11]. The suggested approach improves the accuracy and reliability of forecasting CO₂ emissions. Machine learning algorithm of general regression neural network (GRNN) was employed to train and estimate CO₂ emissions [12]. The CLONal selection ALGorithm (CLONALG) and Artificial Immune Recognition System (AIRS) were applied to forecast global CO₂ emissions [13]. Results were promising with the AIRS accounting for better predictive values.

3 Fuzzy Nearest Neighbor

The fuzzy K -nearest neighbor (FNN) algorithm is credited with the process of classifying a test object based on its similarity to a given K -nearest neighbor and their corresponding membership degrees [14, 15]. Since object y is a member of class C , the similarity may be expressed as (1):

$$C'(y) = \sum_{x \in N} R(x, y) C'(x) \quad (1)$$

where N is the set of y 's K -nearest neighbors. $R(x, y)$ represents the similarity of x and y and is defined as a value between 0 and 1. It is also commonly characterized as (2):

$$R(x, y) = \frac{\|y - x\|^{-2/(m-1)}}{\sum_{j \in N} \|y - j\|^{-2/(m-1)}} \quad (2)$$

where the Euclidean norm is represented by $\|\cdot\|$, whereas m represents the weight of similarity.

4 Sequential Minimal Optimization

The development of sequential minimal optimization (SMO) was conceived at alleviating the problems associated with support vector machines (SVM) [16]. When dealing with large sized problems, the SVM encounters difficulties. Thus, SMO is used in training the SVM.

The concept of SVM will be analyzed which gives a projection towards SMO's usage with SVM. If there exist collection of data points $\{(G_m, h_m)\}_m^p$; ($G_m \in S^q$ represents input values with respect to m th training data patterns; h_m is for labeling the class and lies in the region of -1 , and 1 ; p sums up all training data patterns) [17], an identical relation occurs between SVM training for classification and solving a convex quadratic programming (QP) problem in (3):

$$\begin{aligned} \text{maximize : } S(\beta_m) &= \sum_{m=1}^p \beta_m \\ &\quad - \frac{1}{2} \sum_{m=1}^p \sum_{n=1}^p \beta_m \beta_n h_m h_n T(G_m, G_n) \end{aligned} \tag{3}$$

$$\text{subject to : } \sum_{m=1}^p \beta_m h_m = 0, \quad 0 \leq \beta_m \leq r_c \quad m = 1, \dots, p \tag{4}$$

where $T(G_m, G_n)$ connotes the type of kernel function to be used, β_m for Lagrange multiplier, r_c is a user defined regularization constant. The general deployment of kernel function is the Gaussian function $e^{-\|G_m - G_n\|^2/\varphi^2}$, with φ as the kernel's breadth. Solution to the QP problem stated in (3) triggers a decision function that designates class labels for new data patterns as given by (5).

$$\text{function}(G) = \sum_{m=1}^p \beta_m h_m T(G_m, G) + a \tag{5}$$

where a is hyperplane's parameter retrieved from (3).

5 Logistic Regression

Logistic regression (LR) has its roots in statistics. LR estimates the probability of a binary outcome based on explanatory features. It describes the impact of certain factors on the analyzed dichotomous variable. However, a dependent variable consisting of unordered categories not lower than three, signals the usage of multinomial logistic regression (MLR). The MLR approach is an extension of binomial logistic regression, as it shares similar concepts and setup [18–20].

6 K-Means Clustering

The method of grouping data into clusters by partitioning is referred to as clustering. The characteristics of each cluster differs based on the homogeneous pattern of each individual data in the cluster [21]. K -means clustering is a very pronounced technique for data partitioning. It executes by pairing data consisting of k number of clusters. Notation in mathematics are elucidated as follows:

Given a set of data patterns $Z = \{z_1, \dots, z_q, \dots, z_M\}$, where $z_q = (z_{q1}, z_{q2}, \dots, z_{qd})^T \in \mathbb{R}^d$ with z_{qp} as a feature. The clustering set itself in motion to get K -partition of Z , $E = \{E_1, \dots, E_K\}$ ($K \leq M$), such that

1. $E_p \neq \emptyset, p = 1, \dots, K;$
2. $\bigcup_{p=1}^K E_p = Z;$
3. $E_p \cap E_q = \emptyset, p, q = 1, \dots, K$ and $p \neq q$.

The K -means clustering employs the use of an objective function, defined in (6), to minimize the distance between a data point and center of cluster.

$$\arg \min_G \sum_{p=1}^k \left(\sum_{z_q \in G_p} \|z_q - \lambda_p\|^2 \right) \quad (6)$$

where z_q depict a data point, G_p and λ_p are the cluster and its center respectively. The K -means clustering could be called Lloyd's algorithm because of the heuristic approach to its implementation [22,23]. To improve cluster center positions, follow steps (7) and (8):

$$G_p = \{z_q : \|z_q - \lambda_p\| \leq \|z_q - \lambda_e\| \forall 1 \leq e \leq k\} \quad (7)$$

$$\lambda_p = \frac{1}{|G_p|} \left(\sum_{z_q \in G_p} z_q \right) \quad (8)$$

The data points $z_1 \dots z_m$ gets matched to one of its closet center λ_p as given in (7). In (8), updating of the centers $\lambda_1 \dots \lambda_k$ are done through the mean of all data points in the cluster.

7 Proposed Methodology

The proposed algorithm consist of k -means clustering, fuzzy nearest neighbor, sequential minimal optimization, and logistic regression. The dataset used are unlabelled. Thus, k -means clustering is used to group data with similar patterns as clusters, which are labelled into a class attribute. The data becomes sufficient to be passed as input for the ensemble algorithm.

The ensemble algorithm starts by feeding data into base classifiers such as FNN and SMO algorithms. They are trained to anticipate both FNN and SMO. The meta-classifier uses the obtained predictive outcome to make a final prediction. The block diagram of the algorithm is shown in Fig. 1 and processes are explained in the following steps:

1. The unlabelled data is loaded to be segmented as clusters.
2. Apply K -means clustering on the unlabelled data into k clusters.
3. The labelled training data D , having m instances and n attributes is prepared for the base classifiers.
4. Algorithms of FNN and SMO that represents the base classifiers trains on D .
5. FNN and SMO predictions are integrated into a second-level dataset (D^{level2}) with m instances and M attributes.
6. The Meta-classifier (LR) trains on second-level data to provide accurate classification results.

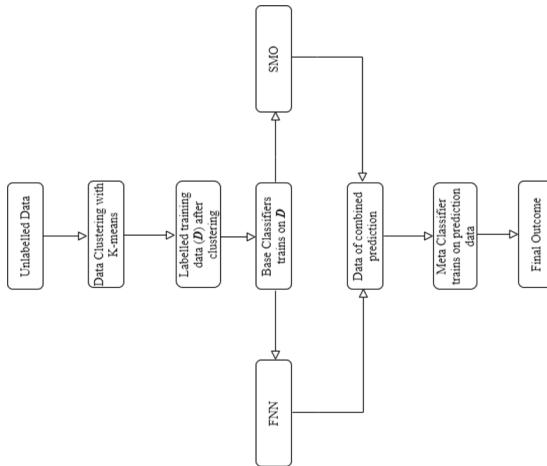


Fig. 1. Proposed algorithm of K -means clustering and Stacking ensemble.

8 Experimental Setup and Analysis

Experiments aim to anticipate OPEC CO₂ emissions from petroleum consumption using combined algorithms. The K -means clustering algorithm applied to unlabeled data. After partitioning accordingly in clusters, ensemble algorithm comprising of FNN, SMO, and LR trains on the data for efficient predictive classification. The Waikato environment for knowledge analysis (WEKA) is used for running all the experiments. In order to have a fair comparison with the proposed algorithm, multi-layer perceptron (MLP), fuzzy ownership nearest neighbor (FNN-O), random tree (RT), FNN, SMO and LR were chosen. To train and evaluate, 10-fold cross-validation is employed [24, 25]. The dataset is divided into ten equal-sized subsets: nine for training and one for testing purposes. Each result's average means are compiled.

8.1 Assessment Measures

The performance metrics to assess the effectiveness of these algorithms are the detection rate (DR) (true positive rate), false alarm rate (FAR) (false positive rate), F-measure, and Matthews correlation coefficient (MCC). The terms are defined as follows and depicted in (9) to (12):

$$DR = \frac{TP}{TP + FN} \quad (9)$$

$$FAR = \frac{FP}{FP + TN} \quad (10)$$

$$F - measure = 2 \times \frac{Positive\ Predictive\ Value \times Sensitivity}{Positive\ Predictive\ Value + Sensitivity} \quad (11)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (12)$$

where TP and FP are the true and false positives, while FN and TN are the false and true negatives.

8.2 Dataset Description

The data on OPEC Carbon Dioxide CO₂ Emissions from 1980 to 2011 was obtained from [26], a reliable source of energy data [27]. CO₂ emissions are quantified in million metric tons (mmt) and are only available annually. The data includes OPEC CO₂ emissions from 12 countries, as well as a sum of all OPEC CO₂ emissions. The dataset consists of 13 columns and 32 rows. The dependent variable is total OPEC CO₂ emissions, while the independent variables are CO₂ emissions from 12 nations.

8.3 Simulation Results

The simulations are conducted using WEKA on 3.40 GHz Intel® Core i7 Processor with 4 GB of RAM. The results after series of experiments for each dataset are tabulated and graphed. The K -means clustering is applied on the unlabelled OPEC CO₂ data with $k = 2$. The data size of 32 is correctly split into two clusters of 22 and 10 instances, as in Table 1.

Table 1. K -means clustering on OPEC CO₂ emission data

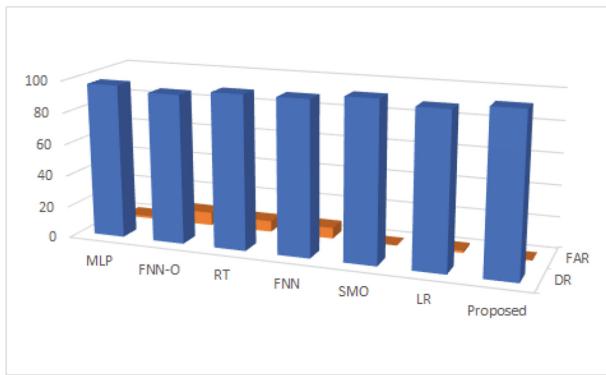
Algorithm	Data size	Cluster partition	Individual cluster %	Cluster class
K -means	32	22	69%	cluster0
		10	31%	cluster1

Upon successful clustering, the data becomes sufficient to be passed as input for the ensemble algorithm. The performance results in Table 2 with graph plots shown in Fig. 2 accommodate OPEC CO₂ emission data. With respect to detection rate, FNN, SMO and LR generated rates of 96.90%, 100%, and 96.90% respectively. Other algorithms such as the MLP, FNN-O, and random tree accounted for detection rates at 96.90%, 93.80%, and 96.90% accordingly. It can be deduced that four of the algorithms have same rate of 96.90%. The proposed model is rated first at 100% alongside SMO. Assigned with the lowest detection rate is FNN-O algorithm. Regarding false alarm rate, the algorithm shows to be better as rates decreases. The proposed model of gave the lowest and best rate at 0.00%. FNN-O also have the poorest false rate of 8.30%.

It can be revealed that in terms of F-measure, the proposed algorithm supercede all other algorithms with a rate of 100%, except for SMO. In second place is

Table 2. Results for OPEC CO₂ emission data

Algorithms	DR (%)	FAR (%)	F-measure (%)	MCC (%)
MLP	96.90	1.40	96.90	93.20
FNN-O	93.80	8.30	93.80	85.50
RT	96.90	6.90	96.80	92.80
FNN	96.90	6.90	96.80	92.80
SMO	100	0.00	100	100
LR	96.90	1.40	96.90	93.20
Proposed algorithm	100	0.00	100	100

**Fig. 2.** Graph plots of DR and FAR for OPEC CO₂ emission data.

MLP and LR at 6.90%. Also, the proposed method certifies its superiority over the compared algorithms when matthews correlation coefficient is concerned. A 100% MCC is accredited to the proposed model. Scanning through the results of OPEC CO₂ emission, FNN-O performed poorly to others overall, while the proposed model proved the best on the overall comparison.

Justification of Proposed Algorithm's Results. The justification of why the proposed algorithm gave a 100% could be attributed to distribution of CO₂ emission data. This obviously had an influence on the results generated by all algorithms with the lowest at 93.80%. The results generated by each individual algorithm making up the stacking ensemble method were 96.90%, 100%, and 96.90% for FNN, SMO, and LR respectively. Thus, it can be deduced that the result accounted for by SMO proved an enormous impact on the actualization of final outcome by the ensemble technique. A fair conduction of experimentation were executed without any biases.

9 Conclusion

The escalation of CO₂ emissions from greenhouse gases in the earth's atmosphere rises at an alarming rate. Human beings and surrounding environment falls under the danger of exposure to global warming and climate change. Preventive measures are therefore needed to control and reduce CO₂ emissions, which inspired the proposition of an ensemble algorithm with k -means clustering to improve prediction. As the base classifiers, fuzzy nearest neighbor and sequential minimal optimization trains on the CO₂ emission data to give a combined prediction. The logistic regression as the meta classifier takes as input the combined prediction and trains on it for a final output. Conclusive results after experimentation signifies that the proposed algorithmic model is potent in predicting CO₂ emissions in comparison to MLP, FNN-O, RT, FNN, SMO, and LR. Adequate policies can be formulated through prediction and forecast of CO₂ emissions.

Acknowledgements. This research was supported by Universiti Tun Hussein Onn Malaysia.

References

1. Fang, D., Zhang, X., Yu, Q., Jin, T.C., Tian, L.: A novel method for carbon dioxide emission forecasting based on improved Gaussian processes regression. *J. Clean. Prod.* **173**, 143–150 (2018)
2. Lee, S., Kim, J.-K.: Process synthesis and optimization of membrane systems with superstructure approach for the mitigation of CO₂ emissions from a coal-fired power plant. In: *Computer Aided Chemical Engineering*, vol. 43, pp. 901–902. Elsevier (2018)
3. International Energy Agency, IEA (2024), CO₂ Emissions in 2023, IEA, Paris. <https://www.iea.org/reports/co2-emissions-in-2023>. Accessed 28 Mar 2024
4. Lasisi, A., et al.: Predicting crude oil price using fuzzy rough set and bio-inspired negative selection algorithm. *Int. J. Swarm Intell. Res.* **10**(4), 25–37 (2019)
5. Mohmad Hassim, Y.M., Ghazali, R.: Using artificial bee colony to improve functional link neural network training. In: *Applied Mechanics and Materials*, vol. 263, pp. 2102–2108 (2013)
6. Wahid, F., Ghazali, R.: Hybrid of firefly algorithm and pattern search for solving optimization problems. *Evol. Intell.* **12**(1), 1–10 (2019)
7. Aseere, A.M., Lasisi, A.: A Multi-agent stacking ensemble hybridized with vaguely quantified rough set for medical diagnosis. *Intell. Autom. Soft Comput.* **27**, 683–699 (2021). <https://doi.org/10.32604/iasc.2021.014811>
8. Sun, W., Wang, C., Zhang, C.: Factor analysis and forecasting of CO₂ emissions in Hebei, using extreme learning machine based on particle swarm optimization. *J. Clean. Prod.* **162**, 1095–1101 (2017)
9. Wei, S., Yuwei, W., Chongchong, Z.: Forecasting CO₂ emissions in Hebei, China, through moth-flame optimization based on the random forest and extreme learning machine. *Environ. Sci. Pollut. Res.* **25**(29), 28985–28997 (2018)
10. Chiroma, H., et al.: Global warming: predicting OPEC carbon dioxide emissions from petroleum consumption using neural network and hybrid cuckoo search algorithm. *PLoS ONE* **10**(8), e0136140 (2015)

11. Zhao, H., Huang, G., Yan, N.: Forecasting energy-related CO₂ emissions employing a novel SSA-LSSVM model: considering structural factors in China. *Energies* **11**(4), 781 (2018)
12. Yang, S., Lei, L., Zeng, Z., He, Z., Zhong, H.: An assessment of anthropogenic CO₂ emissions by satellite-based observations in China. *Sensors* **19**(5), 1118 (2019)
13. Lasisi, A., Ghazali, R., Chiroma, H.: Utilizing clonal selection theory inspired algorithms and *K*-means clustering for predicting OPEC carbon dioxide emissions from petroleum consumption. In: Herawan, T., Ghazali, R., Nawi, N.M., Deris, M.M. (eds.) SCDM 2016. AISC, vol. 549, pp. 101–110. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-51281-5_11
14. Keller, J.M., Gray, M.R., Givens, J.A.: A fuzzy k-nearest neighbor algorithm. *IEEE Trans. Syst. Man Cybern.* **4**, 580–585 (1985)
15. Bui, D.T., Nguyen, Q.P., Hoang, N.-D., Klempe, H.: A novel fuzzy K-nearest neighbor inference model with differential evolution for spatial prediction of rainfall-induced shallow landslides in a tropical hilly area using GIS. *Landslides* **14**(1), 1–17 (2017)
16. John, C.P.: Sequential minimal optimization: a fast algorithm for training support vector machines. *MSRTR Microsoft Res.* **3**(1), 88–95 (1998)
17. Cao, L.J., et al.: Parallel sequential minimal optimization for the training of support vector machines. *IEEE Trans. Neural Netw.* **17**(4), 1039–1049 (2006)
18. Hosmer, D.W., Lemeshow, S.: Applied Logistic Regression. Wiley, New York (2000)
19. Hosmer Jr., D.W., Lemeshow, S., Sturdivant, R.X.: Applied Logistic Regression, vol. 398. Wiley, Hoboken (2013)
20. Milewska, A.J., Jankowska, D., Więsak, T., Acacio, B., Milewski, R.: The application of multinomial logistic regression models for the assessment of parameters of oocytes and embryos quality in predicting pregnancy and miscarriage. *Stud. Logic Gramm. Rhetor.* **51**(1), 7–18 (2017)
21. Capó, M., Pérez, A., Lozano, J.A.: An efficient approximation to the K-means clustering for massive data. *Knowl.-Based Syst.* **117**, 56–69 (2017)
22. Lloyd, S.: Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **28**(2), 129–137 (1982)
23. Xu, J., Lange, K.: Power k-means clustering. In: International Conference on Machine Learning, pp. 6921–6931 (2019)
24. Waheed, W., Ghazali, R., Hussain, A.J.: Dynamic ridge polynomial neural network with Lyapunov function for time series forecasting. *Appl. Intell.* **48**, 1721–1738 (2018)
25. Al-Jumeily, D., Ghazali, R., Hussain, A.: Predicting physical time series using dynamic ridge polynomial neural networks. *PLoS ONE* **9**(8), e105766 (2014)
26. Energy Information Administration of the United States Department of Energy. <http://www.eia.gov/cfapps/ipdbproject/iedindex3.cfm?tid=5&pid=5&aid=8&cid=CG9,&syid=1980&eyid=2011&unit=MMTCD>. Accessed 27 May 2014
27. Chiroma, H., Abdulkareem, S., Abubakar, A., Usman, M.J.: Computational intelligence techniques with application to crude oil price projection: a literature survey from 2001–2012. *Neural Netw. World* **23**(6), 523 (2013)



Detection of Phishing Websites from URLs Using Hybrid Ensemble-Based Machine Learning Technique

Modupe Agagu¹(✉), Ibrahim Abayomi Ogunbiyi¹, Ayodele Lasisi², and Osaremwinda Omorogiuwa³

¹ Department of Computer Science, Olusegun Agagu University of Science and Technology, Okitipupa, Ondo State, Nigeria

modupe.agagu@oaustech.edu.ng

² Department of Computer Science, College of Computer Science, King Khalid University, Abha, Saudi Arabia

alasisi@kku.edu.sa

³ Department of Computer Science and Information Technology, Igbinedion University, Okada, Edo State, Nigeria

ask4osas@iuokada.edu.ng

Abstract. With the increasing advancement in technology, people interact with the internet daily, whether it is to access their bank accounts, social media platforms, etc. Malicious actors often create clones of legitimate websites and then lure users to access these cloned sites so that they can capture user-sensitive information. This research proposes a hybrid ensemble-based machine-learning technique for predicting phishing websites from URLs to address this issue. The research employs three base classifiers: CatBoost, XGBoost, and LightGBM. The voting technique was used to combine the base classifiers to build an ensemble model. Performance metrics such as accuracy, precision, recall, and F-1 score were used to measure the performance of the model. The experimental results of the proposed model demonstrate an outstanding performance of 93.0%, 92.2%, 93.4% and 92.8% for accuracy, recall, precision and f1-score respectively on the dataset used. When compared with some existing models, the model gave an outstanding performance.

Keywords: Phishing detection · Machine learning · Ensemble Model · CatBoost · XGBoost · LightGBM

1 Introduction

Due to technological advancement, the internet has become one of the normal ways of performing daily life activities. Whether you want to connect with people, shop online, or gain access to your bank account, almost everything is done electronically. However, while this is a good thing, malicious people often exploit the vulnerabilities of the internet to manipulate people into collecting their confidential data. One of the

techniques normally used is phishing, also known as web phishing. As defined by [1], web phishing is a cyber-attack that generates a fake website to imitate a trusted website to steal sensitive information, such as usernames, passwords, and credit card information. Detecting phishing is important because it has become the norm for people to want to access services online. Therefore, there is a need to help safeguard people from unknowingly releasing their information into the hands of malicious people. Several researchers have carried out studies in detecting web phishing, including the use of machine learning such as [2, 3] where the authors used heuristic and machine-learning-based techniques to detect phishing. Out of those techniques used to detect phishing, machine learning techniques have proved to be prevalent, as supported by [4]. Machine learning techniques can be used to detect phishing from a website URL, the content of the website, etc. A typical example of using machine learning to detect phishing is evident in the work of [3], where the authors used a machine-learning technique to detect web phishing. However, low model accuracy was obtained. This research aims to build upon existing research in predicting web phishing from URLs. The major contributions of this study are as follows.

- A hybrid ensemble phishing URL-based cyberattack detection is proposed in this research to prevent crime and protect people's privacy.
- The research employs three base classifiers: CatBoost, XGBoost, and LightGBM. A voting technique was used to combine the base classifiers to form an ensemble model to classify the threats of phishing URLs and accurately carry out the prediction.
- The proposed methodology was evaluated using evaluation parameters, such as accuracy, precision, recall, and F1-score.

2 Related Work

Phishing is the most significant issue in the field of networks and the Internet. Many researchers have attempted to provide facilities to protect users from cyber-attacks by preventing the phishing of URLs using machine learning, deep learning, black lists, and white lists. Several research studies conducted previously have proposed and implemented models to predict web phishing from URLs. A study conducted by [5] developed a machine-learning model to predict web phishing using six machine-learning classifiers which are NaiveBayes, ANN, DecisionStump, KNN, J48, and RandomForest, and tested them on 3 web phishing datasets. The research contribution is to evaluate the performance of different machine learning classifiers for phishing detection using multiple datasets. However, one limitation of the research is that the authors did not compare their development with other research studies to see how their model performed. [6] developed a hybrid ensemble model to predict phishing URLs. They trained the model using 20,000 instances of legitimate and malicious URLs and then achieved an accuracy score of approximately 85% with a recall score of 84%. The hybrid model was a combination of 4 different base classifiers, including Multilayer Perception, Support Vector Machine, Decision Tree Classifier, and Random Forest. The method used to combine the model predictions to form an ensemble was the hard voting technique. The weakness of this research was that the dataset used was not representative enough. It contained 19 features. Phishing URLs are rapidly evolving, and malicious attackers are using different techniques to create phishing URLs. Also, the accuracy score of the research is

relatively low. Another research conducted by [7] involved also building an ensemble machine learning model (hybrid) to predict phishing URLs. The authors trained the model on 11,055 instances with 30 features retrieved from the University of California Irvine (UCI) ML data repository. They developed three ensemble techniques and compared the performance of each ensemble technique with different metrics, such as accuracy. The 1st ensemble involved combining Artificial Neural Network (ANN) + Random Forest Classifier (RFC), the 2nd ensemble method included K-Nearest Neighbor (KNN) + RFC, and the 3rd ensemble technique involved C4.5 decision classifier + RFC. These techniques achieved the respective accuracy scores in order: 97.16%, 97.33%, and 96.36%, with KNN + RFC achieving the highest accuracy score. The limitation of this research is that they did not test the performance of the research with different datasets to evaluate the performance of their model. [4] also developed a phishing detection system using RandomForest, SVM, and neural networks to predict phishing URLs. They achieved accuracies of 97.369%, 97.451%, and 97.259%, respectively, with SVM performing the best among all the models. This research also utilized the UCI dataset to evaluate the performance of their model. Another limitation of this research is that they did not evaluate the performance of the machine learning model with another dataset. [8] developed a machine-learning technique to predict web phishing. The research used a dataset obtained from Alexa and the Common Crawl archive, and they utilized SVM to predict phishing URLs with 95.66% accuracy. The limitation and weakness of this research are that the research is shallow. Firstly, the number of instances is 5000, and only 5 features were used to train the machine learning algorithm. Additionally, the research utilized only one classifier (SVM), and the research did not evaluate the performance of the model with another dataset. Lastly, only one metric was used to evaluate the model's performance. This research, however, proposes a hybrid-ensemble-based machine learning technique by combining the top three base ensemble classifiers trained and then utilizing a standard and more generalizable dataset, to test to achieve an improved result when compared with existing research.

3 Methodology

This section describes the method used in building the hybrid ensemble mode using the three base classifiers: CatBoost, XGBoost, and LightGBM. The whole technique employed in achieving this study is depicted in Fig. 1.

3.1 Description of the Machine Learning Classifiers

a. Catboost

CatBoost is an algorithm and machine learning package that focuses on decision tree model gradient boosting. It is renowned for its efficiency in both classification and regression problems and is made to handle categorical features. Categorical boosting is referred to as CatBoost. Order Boosting (OB) and Ordered Target Statistic (OTS) are used by CB. Because OTS and OB give odd training data, CB can systematically adjust its estimate for the atypical data, which makes it an excellent choice for datasets with categorical data. To counter the prediction shift induced by a particular kind

of target leakage present in all existing gradient-boosting algorithms, OTS and OB employed random permutations of the training data. Binary decision trees served as CatBoost's basic predictor [9].

b. XGBoost

XGboost which is short for extreme gradient boosting. It is a popular and powerful machine learning model that is also used for classification and regression tasks. It is an ensemble learning method that combines the predictions of multiple decision tree models to produce a robust and accurate final prediction. XGBoost has gained widespread popularity in data science and machine learning competitions due to its effectiveness and efficiency.

c. LightGBM

LightGBM, also known as Light Gradient Boosting Machine, is an open-source, distributed, high-performance machine learning framework developed by Microsoft. Like XGboost, LightGBM is designed for gradient boosting and is known for its efficiency and effectiveness in various supervised learning tasks such as classification and regression. LightGBM, unlike other gradient boosting algorithm, is designed to be memory-efficient and computationally efficient.

3.2 Dataset Description

In this study, the standard dataset benchmark provided by [10] was utilized. This dataset consists of a total of 11,430 records encompassing both legitimate and phishing websites. It offers a comprehensive range of features, including those derived from URLs, content-based attributes obtained through analysis of HTML content, as well as external features sourced from third-party services. Specifically, the dataset was designed to facilitate research on detecting phishing websites, with a focus on attributes relevant to URL analysis. From this dataset, a total of 58 features were extracted, encompassing various aspects such as URL structure, domain characteristics. Notably, among these features is the crucial "status" attribute, which serves as a binary indicator distinguishing phishing websites from legitimate ones. Some of other features in the dataset that serve as predictors include length_url, length_hostname, ip, nb_dots, nb_hyphens, nb_at, nb_qm, nb_and, nb_or, nb_eq, nb_underscore, nb_tilde, nb_percent, nb_slash, etc.

3.3 Model Construction

The proposed model for the web-phishing follows a sequence of stages from data collection, to feature extraction, feature selection, model selection and tuning as shown in Fig. 1:

The stages involved in the development of the proposed model include:

- Data Acquisition.
- Data Cleaning and Transformation.
- Exploratory Data Analysis.
- Splitting the Dataset into Train and Test Set.
- Base Model Selection.
- Base Model Testing and Evaluation.

- Building Proposed Model Based on Best Performing Base Model.
 - Proposed Model Evaluation and Hyperparameter Tuning.
- a. Data Acquisition: Given that the research focuses on detecting phishing websites from URLs, features related to URLs were extracted from the dataset, resulting in 58 features, including the status feature, which indicates whether a URL is a phishing website or not were extracted from the dataset described in Sect. 3.2.
 - b. Data Cleaning and Transformation: The dataset used was cleaned in the pure state, but, the status column was first in the categorical data and then transformed into numerical value by mapping “legitimate” to 0 and “phishing” to 1.
 - c. Exploratory Data Analysis: Once the dataset was cleaned and transformed into a usable form, exploratory data analysis was performed on the 58 features to uncover insights about the dataset. Figure 2 illustrates that the number of phishing and legitimate URLs in the dataset is balanced, indicating no imbalanced dataset concerns. It shows the proportion of URLs that are legitimate or phishing in the dataset. It indicates that the dataset is balanced, so there is no need to worry about the issue of imbalance when building the classifiers.
 - d. Splitting the Dataset into Train and Test Set: Since the dataset contains 11,430 records, it was divided into training and test sets. 80% of the dataset was allocated for training, and the remaining 20% for testing. The train-test split library in Scikit-learn was utilized, with the random state set to 42 for reproducibility.
 - e. Base Model Selection: At this stage, several models were chosen, including base models such as Decision Tree, and SVM, as well as ensemble models like Random Forest, AdaBoost, XGboost, CatBoost, and LightGBM. These models were arbitrarily fitted to the training dataset without hyperparameter tuning. The random state value for each model was set to 42 for reproducibility.
- f. Base Model Testing and Evaluation: After fitting the models to the training set and evaluating their performance on unseen data using accuracy score, it was observed that three boosting algorithms—CatBoost, LightGBM, and XGboost—performed better, as shown in Fig. 3. The accuracy of the base model was chosen to determine the top three models to utilize for building the hybrid ensemble.
 - g. Building the Proposed Model: Boosting algorithms are powerful and tend to perform better than other types of machine learning algorithms. Therefore, the final hybrid model was built using three boosting ensemble models, which include CatBoost, LightGBM, and XGboost. The combination strategy used for combining the models to form an ensemble is the Voting Method, and the Voting Classifier class in the Scikit-Learn library was utilized. The random state of each base classifier in the ensemble is set to 42. The number of estimator hyperparameters of each model is as follows: Catboost n_estimators:500, XGboost n_estimators: 100, LightGBM n_estimators: 100.

4 Performance Evaluation Metrics

The method used to evaluate the performance of the proposed model. The metrics include accuracy score, precision, recall, f1-score, confusion matrix, and false positive rate.

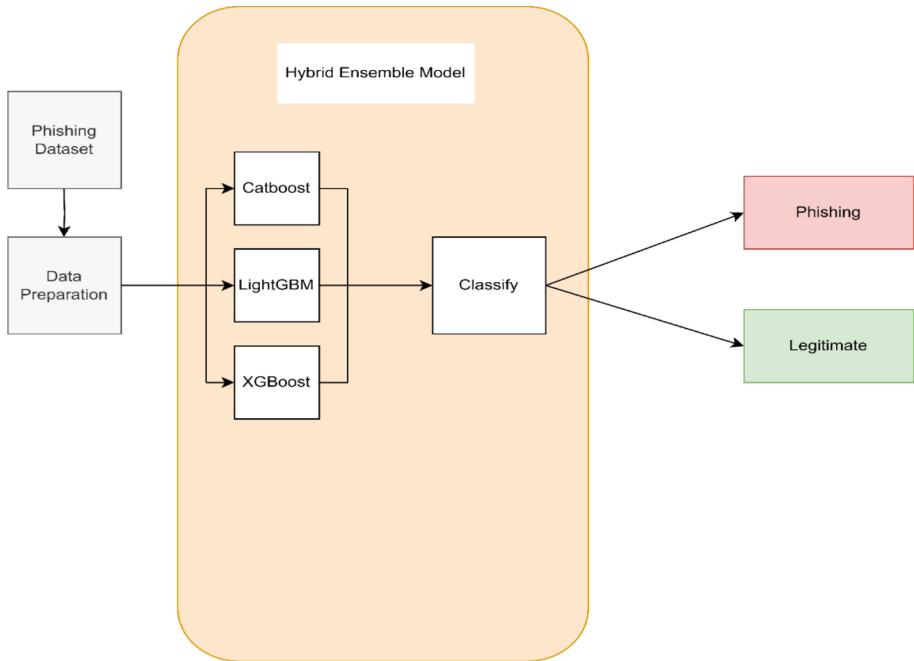


Fig. 1. Architecture of the proposed Hybrid Ensemble-based Model.

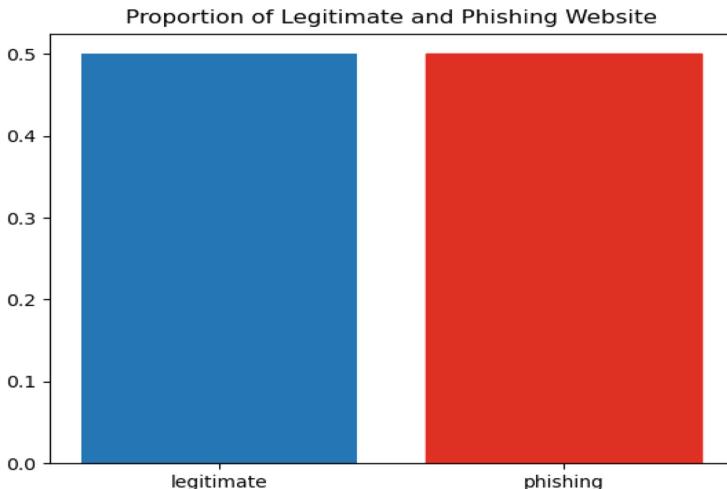


Fig. 2. Proportion of label values in the dataset.

a. Accuracy Score

Accuracy score is a common evaluation metric used in classification tasks to measure the proportion of correctly predicted instances out of the total instances in a dataset. A high accuracy score indicates that the model is making correct predictions

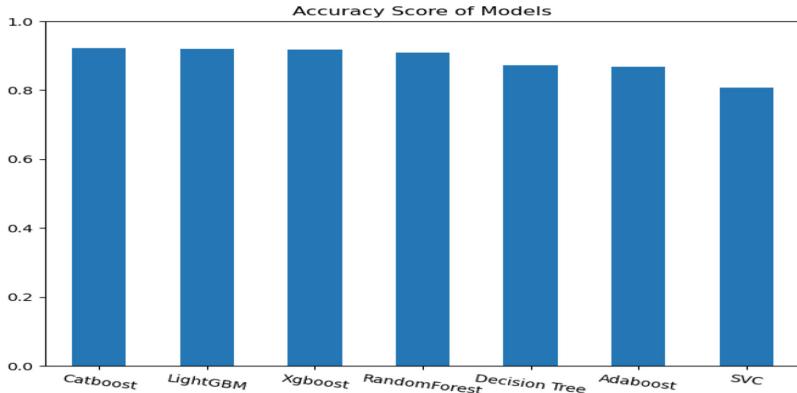


Fig. 3. Accuracy Score of each Model.

overall. It is calculated as

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad (1)$$

b. Precision

This is an evaluation metric used in classification tasks to calculate the proportion of correctly predicted instances divided by all predicted instances deemed to be true which may include both true positive and false positive. It is used to quantify how well a model identifies positive instances while minimizing the number of false positives. In other words, precision focuses on the accuracy of positive predictions made by the model.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (2)$$

c. Recall

Also known as sensitivity or true positive rate, is an evaluation metric used in classification tasks to measure the proportion of correctly predicted instances (true positive) out of all actual positive instances (true positive plus false negative – which is the number of predictions the model deemed negative but it is true). It is used to quantify how well a model identifies positive instances without missing any. In other words, recall focuses on the ability of the model to capture all positive instances.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (3)$$

d. F1-Score

The F1-score is a widely used evaluation metric in classification tasks that provides a balance between precision and recall. It's a way to assess a model's accuracy while considering both false positives and false negatives. A high F1-score indicates that the model achieves high precision and recall simultaneously, reflecting a well-performing classifier.

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

5 Result

This section presents the result obtained from the research, as stated earlier, several performance evaluation metrics were used to measure the performance of the hybrid model which include accuracy, precision, recall, and f1-score. The results of the machine learning model's performance after its deployment as a web app, as depicted in Figs. 4 and 5. When users input the website URL, the relevant features from the URL are extracted. These features are then made available for the machine learning to process and predict. Consequently, the model produces an output in the form of a message that informs the user whether the URL is either legitimate or a phishing URL. Figures 4 and 5 depict the screenshots of testing the ML Model using a Legitimate URL and testing the ML model using a Phishing URL from the OpenPhish Website respectively.



Fig. 4. Screenshot of the Result of Testing the Model using a Legitimate URL.

The hybrid ensemble model was built using different base machine learning models as stated earlier. After building the model, an accuracy score of 93% was achieved. Table 1 shows the outcome of the proposed model based on other performance evaluation metrics.

Table 1. Performance Metrics Score of the Proposed Model

Metrics	Score
Accuracy	93.0%
Recall	92.2%
Precision	93.4%
F1-Score	92.8%

A Comparison of the performance of the proposed model with different base classifiers was also carried out. Figure 6 shows the accuracy score of the proposed model compared to other classifiers with the proposed hybrid model giving the best accuracy.



Fig. 5. Screenshot of the Result of Testing the Model using Phishing URL.

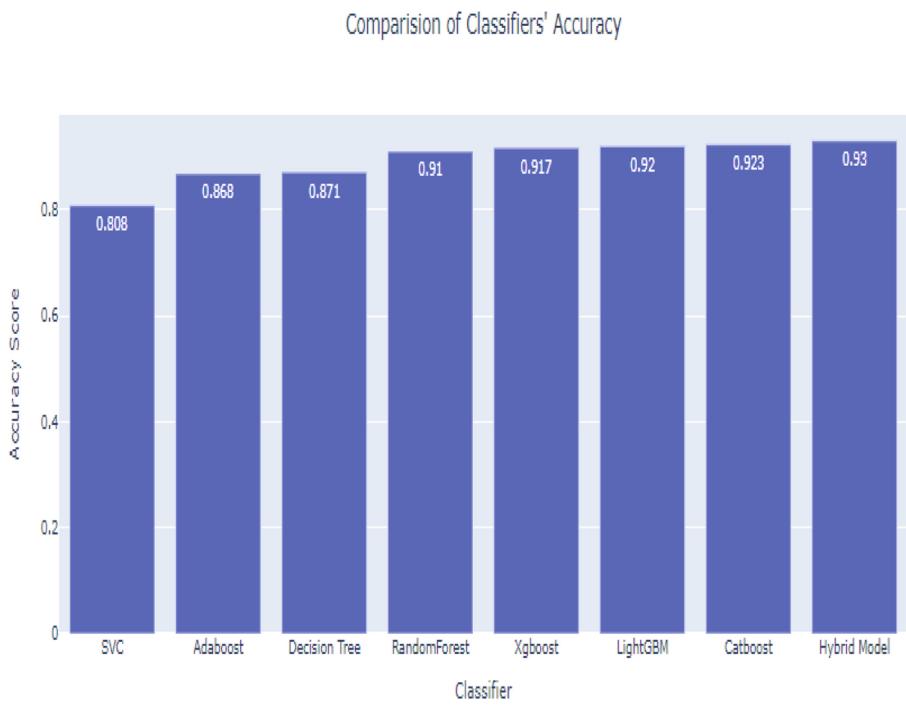


Fig. 6. Comparison of Proposed Model with different Classifier's Accuracy.

Figure 7 shows the confusion matrix of the hybrid model prediction. This illustrates how effectively the hybrid ensemble-based machine learning technique distinguishes between legitimate and phishing websites.

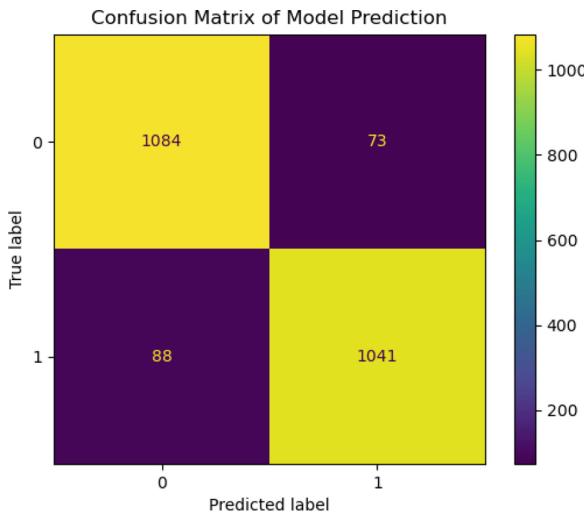


Fig. 7. Confusion matrix of model prediction.

The proposed model was also compared with previous research. The following were the results of the comparison.

a. Comparison with the proposed model in [7].

Table 2 shows the comparison of the proposed model with the previous model proposed by [7]. To evaluate the performance of this model the same workload i.e. the dataset – UCI phishing dataset used by the research was utilized. After evaluating the performance of the model, it was seen that the model proposed by this research outperformed the previous model as shown in Table 2.

Table 2. Comparison of the Performance Metrics Score of the Proposed Model with the Model in [7]

Metrics	Proposed Model Catboost + XGboost + LightGBM	Previous Model KNN + RFC
Accuracy	98.6%	97.33%
Recall	98.7%	98.3%
Precision	98.5%	97%
F1-Score	98.9%	97.6%

b. Comparison with the proposed model in [5].

Table 3 also shows the comparison of the proposed model with the previous model proposed by [5]. The dataset used in this research was published by another researcher in [11]. The dataset contained 48 features and 10,000 records, with an equal proportion of legitimate and phishing websites. After evaluating the proposed model in this research against theirs, it was observed that the proposed model showed an improvement over theirs as shown in Table 3.

Table 3. Comparison of the Performance Metrics Score of the Proposed Model with the model in [5]

Metrics	Proposed Model Catboost + XGboost + LightGBM	Previous Model Random Forest
<i>Accuracy</i>	98.6%	98%
<i>Recall</i>	98.7%	98.1%
<i>Precision</i>	98.5%	97.9%
<i>F1-Score</i>	98.6%	98%

6 Conclusion

This study has developed a hybrid machine-learning model that can accurately predict phishing URLs. The hybrid model consists of 3 base classifiers, including Catboost, LightGBM, and XGBoost. Also, the hybrid model was trained on a dataset with 11,430 instances of legitimate and phishing URLs with 58 features. Several metrics such as accuracy, precision, and recall were used to evaluate the performance of the model. Upon evaluating the model's performance, the proposed model achieved an accuracy score of 93% on the dataset used. Also, after comparing it with another dataset used by another research, the proposed system achieved a higher accuracy score than the previous research. The accuracy score achieved was 98.9%. Finally, the proposed system was deployed as a web app for testing.

7 Future Work

Phishing websites are evolving rapidly, and malicious actors are using different techniques to develop phishing URLs. This model can be developed as a browser extension so that it can detect real-time phishing URLs. Additionally, the running time of this algorithm can be evaluated and improved upon so that it can detect threats as quickly as possible. Lastly, the proposed model can be trained with a more robust dataset so that it can identify more patterns.

References

1. Ramana, A.V., Rao, K.L., Rao, R.S.: Stop-Phish: an intelligent phishing detection method using feature selection ensemble. *Soc. Netw. Anal. Min.* **11**(1), 110 (2021)
2. Kumar, S., Faizan, A., Viinikainen, A., Hamalainen, T.: MLSPD – Machine Learning Based Spam and Phishing Detection. In: Chen, X., Sen, A., Li, W.W., Thai, M.T. (eds.) *CSoNet 2018. LNCS*, vol. 11280, pp. 510–522. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04648-4_43
3. Patil, V., Thakkar, P., Shah, C., Bhat, T., Godse, S.P.: Detection and prevention of phishing websites using a machine learning approach. In: 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEAT), pp. 1–5. IEEE (2018)
4. Sindhu, S., Patil, S.P., Sreevalsan, A., Rahman, F., An, M.S.: Phishing detection using random forest, SVM, and neural network with backpropagation. In: 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), pp. 391–394. IEEE (2020)
5. Aljammal, A.H., Taamneh, S., Qawasmeh, A., Salameh, H.B.: Machine learning based phishing attacks detection using multiple datasets. *Int. J. Interact. Mob. Technol.* **17**(05), 71–83 (2023). <https://doi.org/10.3991/ijim.v17i05.37575>
6. Pandey, A., Chadawar, J.: Phishing URL Detection using hybrid ensemble model. *Int. J. Eng. Res. Technol. (Ijert)* **11**(04) (2022)
7. Basit, A., Zafar, M., Javed, A.R., Jalil, Z.: A novel ensemble machine learning method to detect phishing attack. In: 2020 IEEE 23rd International Multitopic Conference (INMIC), pp. 1–5. IEEE (2020)
8. Rashid, J., Mahmood, T., Nisar, M.W., Nazir, T.: Phishing detection using a machine learning technique. In: 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH), pp. 43–46. IEEE (2020)
9. Puri, N., Saggars, P., Kaur, A., Garg, P.: Application of ensemble Machine Learning models for phishing detection on web networks. In: 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT), pp. 296–303. IEEE (2022)
10. Hannousse, A., Yahiouche, S.: Towards benchmark datasets for machine learning-based website phishing detection: An experimental study. *Eng. Appl. Artif. Intell.* **104**, 104347 (2021)
11. Tan, C.L.: Phishing dataset for machine learning: feature evaluation. Mendeley Data (2018). <https://doi.org/10.17632/h3cgnj8hft.1>



Minimal Data for Maximum Impact: An Indonesian Part-of-Speech Tagging Case Study

Chi Log Chua^{1(✉)}, Tong Ming Lim², and Kwee Teck See¹

¹ Faculty of Computing And Information Technology, Tunku Abdul Rahman University of Management and Technology, Setapak, Kuala Lumpur, Malaysia
{chuacl,seekt}@tarc.edu.my

² Centre for Business Incubation and Entrepreneurial Ventures, Tunku Abdul Rahman University of Management and Technology, Setapak, Kuala Lumpur, Malaysia
limtm@tarc.edu.my

Abstract. Annotating low-resource languages is challenging due to its time-consuming nature and high costs. In response to these challenges, this study investigates the potential of training models with minimal annotated data (520 tagged words) and abundant unannotated data (419,055 sentences), using Indonesian Part-of-Speech (POS) tagging as a case study. For the first time, we apply Stratos and Collins' algorithm for Indonesian POS tagging, using four classifiers: Support Vector Machine, Naive Bayes, Decision Tree, and K-Nearest Neighbor. Our approach not only improved precision, recall, F1-score, and accuracy by approximately 1–5% compared to a baseline model that uses only the minimal annotated data, but also achieved a high performance for an Indonesian POS model, attaining an accuracy of 84% using this small amount of annotated data. This is a significant achievement in the context of using a small amount of annotated data to train the Indonesian POS model, as previous researchers have not achieved this level of accuracy with such a limited dataset. The method proved particularly beneficial for low-resource languages with limited high-quality annotated data but abundant unannotated data. It also reduced the workload of manual annotation as high-performance models required only a small amount of annotated data. Building on this efficiency, future work will focus on developing methods that further optimize the annotation process for low-resource language data.

Keywords: Semi-supervised learning · Part-of-Speech tagging · Indonesian language · Low-resource languages · Natural language processing

1 Introduction

Natural Language Processing (NLP) has been successfully applied in areas such as machine translation, chatbots and search engines, largely due to the availabil-

ity of large amounts of annotated and unannotated data. However, low-resource languages, which lack research and resources, face significant challenges in this regard. Due to the lack of high-quality annotated data, manual annotation has to be performed while developing NLP applications for these languages. This process is not only expensive and time-consuming, but may also require linguistic expertise in multiple languages. Moreover, the tedious nature of manual annotation adds to the complexity of the task [11]. Therefore, for future NLP practitioners working with low-resource languages, an approach that effectively eliminates the manual annotation workload while enabling high-performance NLP models would be invaluable.

Therefore, this study applies the [23]’s algorithm to Indonesian Part-of-Speech (POS) tagging, a method that has never been tested in Indonesian before. This novel application, using the Indonesian language as a case study, aims to address the challenges faced in data annotation for NLP tasks. Furthermore, unlike the studies mentioned in [23] that have predominantly used the Support Vector Machine (SVM) classifier, our experiment expands the scope by employing a variety of machine learning classifiers. This includes SVM, Naive Bayes, Decision Tree, and K-Nearest Neighbor. This diversification in classifiers not only enhances the robustness of our study but also provides a comprehensive understanding of their performance in the setting of semi-supervised learning.

The rest of the paper is structured as follows: Sect. 2 reviews related work on POS tagging for Indonesian and discusses solutions to the lack of annotated data, providing a foundation for our novel approach. Section 3 details the research methodology, explaining how we implement the[23]’s algorithm and various machine learning classifiers to address the challenges in data annotation for NLP tasks. Section 4 presents and analyses the experimental results, demonstrating the effectiveness of our approach. Finally, Sect. 5 summarises the study and suggests directions for future research, pointing towards potential improvements and applications of our work.

2 Literature Review

There are several approaches to address the lack of annotated data. These include Transfer Learning, Meta Learning, Semi-supervised Learning and Data Augmentation. Transfer Learning aims to leverage knowledge from related domains to improve learning performance or minimise the number of annotated examples required for the target domain [27]. Meta Learning also known as learning-to-learn, aims to adapt a model quickly and accurately to unknown tasks [20]. Semi-supervised Learning aims to combine supervised and unsupervised learning, and it is possible to improve classification performance through semi-supervised learning if the unannotated data is sufficiently available and the unannotated data points provide additional information relevant to prediction [12]. Data augmentation refers to methods that increase the amount of data by adding slightly modified copies of existing data or newly created synthetic data from existing data [16]. All these methods have proven effective in optimizing model performance with limited resources.

Although there are multiple methods presented above to solve the problem of lack of annotated data, this study would only focus on semi-supervised learning. This is because the languages in the field of NLP that lack annotated data are the low-resource languages such as Khmer, Malay, and Burmese [22], and since most of the high-performance pre-training models are trained with a large amount of annotated data [14], and to obtain a large amount of annotated data is something that takes a lot of manpower, money and time to achieve [8], so this leads to the fact that finding high-performance pre-trained models should be more difficult than finding their unlabelled data, making Transfer Learning and Meta Learning challenging to implement. Data Augmentation, often used in image processing, is limited in text data due to its complexity and the need for different augmentation techniques such as paraphrasing, noising and sampling [16].

Since this study is centered on Indonesian POS tagging as a case study to address the challenges faced in data annotation, particularly for low-resource languages, it is necessary to review some of the previous works on Indonesian POS. Various stochastic models such as the Conditional Random Field (CRF) and Hidden Markov Model (HMM) have been widely used in the past. For instance, one study [21] used a CRF model trained on 26,348 tagged words, achieving an accuracy of 83.72% to 91.15%. This work demonstrated the effectiveness of the CRF model for Indonesian POS tagging. Paper [26] used a HMM variant for POS tagging with 12,000 annotated words, tested on 30% of out-of-vocabulary (OOV) words, achieving 83% - 91% accuracy. This study further validates that different uses of the stochastic model, such as the introduction of affix trees and additional lexicons, can actually improve the accuracy of POS tagging, especially in predicting POS tagging of OOVs. A deep learning study by [1] used various word embeddings and a simple neural network model trained on 160,000 tagged words, achieving 86.91% to 94.78% accuracy. This work innovatively explored word embeddings in Indonesian POS tagging. However, the methods mentioned above required a fairly large amount of annotated data, which can be difficult to obtain for low-resource languages that does not have complete, public, free, and high-quality annotated data such as Malay [9]. This is where semi-supervised learning methods come into play.

However, semi-supervised learning is not a new concept in the field of NLP in Indonesian. There have been numerous applications of these techniques across various sub-fields. For instance, they have been used in Named Entity Recognition (NER) [5, 15], Sentiment Analysis [7], and text classification [13, 24]. However, to our knowledge, the semi-supervised learning method proposed by [23], which has demonstrated high model performance with as few as 200 annotated data points and achieved an accuracy of 89.34% for English POS tagging, has not yet been utilized in the context of Indonesian NLP. This observation underscores the novelty and significance of our study. Specifically, our research introduces a new approach to the field by applying the [23]'s method to Indonesian POS tagging.

3 Methodology

This study tests the hypothesis that using the semi-supervised learning method of [23]’s algorithm with various classifiers can enhance the performance of Indonesian POS tagging models trained on minimal annotated data. We compare this approach with a baseline setting that uses only the minimal annotated data. The process includes data collection, semi-supervised learning preprocessing, feature selection, model training, and evaluation. Source code for these methods is on GitHub¹, aiding further research.

3.1 Data Collection

The POS model was trained using the Indonesian GSD dataset provided by Universal Dependencies, which contains 97,000 words for training, 12,000 words for development, and 11,000 words for testing. A total of 17 POS tags were utilized [18].

The experiment aimed to train a high-performance model with a small amount of data, hence only the first 520 words of the training set were used. These words were used to train both the baseline POS setting and the semi-supervised POS setting. The full test set of the dataset was used to evaluate the performance of the models [18].

For the semi-supervised POS setting, unannotated data was also required. This data was sourced from the Indonesian unannotated data from cc-100[10], which contains about 2,270.4 million tokens. To avoid overloading computational resources, only the first 419,055 sentences from this unannotated data were used in the experiment.

3.2 Semi-supervised Learning Preprocessing

In this experiment, the unannotated data of 419,505 sentences was prepared for training the POS models in a semi-supervised setting [23]. The clustering model of [6] was employed to induce lexical representations for each word in the sentences. This model was selected based on [23]’s conjecture that its hidden states capture POS tags. Their experiment confirmed this conjecture, thereby providing a compelling reason for the use of this model in this experiment.

Before the unannotated data could be used for clustering, it needed to contain a <?> symbol, which represents unknown words. The clustering model of [6] used the implementation of [17] to derive bit-string word representations for each word in the unannotated data.

These bit-string word representations, which capture the syntactic roles of the words as indicated by the hidden states of the Brown clustering model, are then used as input features for the classifiers in the next step of our methodology. This is a crucial step as it allows us to leverage the syntactic information captured

¹ <https://github.com/chuachilog/IndoPOSTagger-SSL>.

by the Brown clustering model to improve the performance of our POS tagging model. This preprocessing stage was crucial for the successful implementation of this experiment.

3.3 Feature Selection

In this experiment, each word in the sentence was used as an input to the model. To train the model, input features were required. The feature templates used by [23], which include the base feature template and the bit feature template, were utilized in this experiment to generate features for the inputs of the baseline and semi-supervised settings respectively. A summary of the features used in the baseline and semi-supervised settings is given in Table 1.

Table 1. List of features used in experiment

Feature Template	Features
Base	The word itself Does the word's first letter is upper case Does the word just alphabet Does the word are pure numbers Prefix 1 (first letter) Prefix 2 (first two letters) Prefix 3 (first three letters) Prefix 4 (first four letters) Suffix 1 (last letter) Suffix 2 (last two letter) Suffix 3 (last three letter) Suffix 4 (last four letters) The first previous word The second previous word The first next word The second next word
Bit	Base's features Full bit-string of the current word Every prefix of the current word bit-string Full bit-string of previous word Every prefix of the previous word bit-string Full bit-string of next word Every prefix of the next word bit-string

3.4 Classification

The experiment employed four machine learning classifiers: Support Vector Machine (SVM), Naive Bayes (NB), Decision Tree (DT), and K-Nearest Neighbor (KNN), as used in [4]. These classifiers are implemented using the sklearn

library from [19], with all parameter and hyperparameter settings following the library’s defaults.

In the baseline setting, these classifiers were trained on a feature template called “Base”, which provides the underlying information as shown in Table 1. In contrast, the semi-supervised setting used a feature template called “Bit”, which includes all the features in the “Base” feature template, as well as additional features generated by Brown clustering model [17], as shown in Table 1. Specifically, these additional features are the bit-string representations of each word, derived from the clustering model, which capture the syntactic roles of the words as indicated by the hidden states of the model. By using these bit-string representations as input features, we can leverage the syntactic information captured by the clustering model to improve the performance of our POS tagging model.

Both the baseline and semi-supervised classifiers require their inputs and outputs to be numeric. Therefore, the features of each word in each sentence, as well as the POS tags, were converted to numeric representations using [19]’s OneHotEncoder and LabelEncoder functions respectively.

3.5 Evaluation

The evaluation of the four POS tagging models in both baseline and semi-supervised settings followed the methods used in [4]. Evaluation metrics such as accuracy, precision, recall and F1 score were obtained using the classification report function of [19]. Evaluation of the four POS tagging models in the baseline setting and in the semi-supervised setting was performed using the full test set of 11,000 words from the [18]. This evaluation approach ensures a comprehensive assessment of model performance.

4 Result and Discussion

This section presents the evaluation results of the Indonesian POS tagging model. The evaluation metrics used were accuracy, precision, recall, and F1 scores, calculated using the sklearn library. Four classifiers were tested in both baseline and semi-supervised settings. The detailed results are summarized in Tables 2 and 3 for the baseline and semi-supervised settings, respectively. A key finding from these evaluations is that in the semi-supervised setting, most classifiers outperform those in the baseline setting. These evaluations were performed using prepared test datasets as described earlier.

Based on the results presented in Table 2, the SVM classifier consistently outperforms the other models, achieving the highest accuracy, precision, recall, and F1 score. In contrast, the NB classifier underperforms, scoring the lowest across all metrics. This trend continues in the semi-supervised setting (Table 3), where the SVM classifier maintains its superior performance. However, in this setting, the KNN classifier that falls behind, demonstrating lower overall performance compared to the other models.

Comparing Table 2 and Table 3, it was found that the overall performance of each classifier generally improved in the semi-supervised setting. For instance, the SVM classifier improved by about 3% in terms of accuracy, precision, recall, and F1 scores compared to its performance in the baseline setting. Similarly, the NB and DT classifiers in the semi-supervised setting showed improvements of about 1% to 3% and 4% to 5% respectively. However, the KNN classifier in the semi-supervised setting did not show a significant improvement compared to its baseline performance, with an increase of 1% in accuracy, no change in precision, an increase of 1% in recall, and a decrease of 1% in F1 score.

The reason why KNN classifiers do not improve much in the semi-supervised setting is that the input data for the semi-supervised setting has a higher dimensionality compared to the input data for the baseline setting, while KNN uses a distance metric to determine the final classification output [3], the ratio of distances between nearest and farthest neighbors of a given target in the high-dimensional space is almost 1 for a wide range of data distributions and distance functions [2]. In this case, since the different distance contrasts between data points do not exist, the nearest neighbor problem becomes ill-defined [2].

Table 2. Test set performance of Baseline setting when only 520 labeled words are used

Machine Learning Classifier	Metrics			
	Precision	Recall	F1-Score	Accuracy
Support Vector Machine (SVM)	81%	81%	81%	81%
Naive Bayes (NB)	70%	69%	69%	69%
Decision Tree (DT)	77%	78%	77%	78%
K-Nearest Neighbor (KNN)	71%	71%	71%	71%

Table 3. Test set performance of Semi-supervised setting when only 520 labeled words are used

Machine Learning Classifier	Metrics			
	Precision	Recall	F1-Score	Accuracy
Support Vector Machine (SVM)	84%	84%	83%	84%
Naive Bayes (NB)	73%	71%	70%	71%
Decision Tree (DT)	82%	82%	82%	82%
K-Nearest Neighbor (KNN)	71%	72%	70%	72%

5 Conclusion

5.1 Summary

This study addresses the challenges faced by low-resource languages in obtaining large amounts of high-quality annotated data for supervised machine learning,

by applying a semi-supervised learning method to Indonesian POS tagging. In this study, we applied the method originally proposed by [23] to Indonesian POS tagging. As part of this application, we extended the scope of the method by employing not only the Support Vector Machine (SVM) classifier, which was used by [23], but also other classifiers such as Naive Bayes (NB), Decision Tree (DT), and K-Nearest Neighbor (KNN). This study found that this semi-supervised learning method, which incorporates the use of the Brown clustering model to generate bit-string representations of each word, improved the performance of SVM, NB, and DT classifiers by about 1% to 5% compared to pure supervised learning. However, KNN was less effective due to high dimensional data issues. Notably, within the context of using a small amount of annotated data, this study achieved a high accuracy that previous Indonesian POS researchers haven't achieved using this small amount of annotated data. While this POS model may not be the state-of-the-art for all time Indonesian POS models, its performance in this specific context is a significant achievement. This method proved beneficial for low-resource languages, reducing the manual annotation workload and achieving high accuracy with a small amount of annotated data.

5.2 Future Work

The semi-supervised learning approach used in this study, which incorporates the use of a clustering model to generate bit-string representations of each word, does not assist in data annotation from scratch. Its effectiveness can be influenced by the quality of annotated data. This issue may be exacerbated when data for annotation is randomly selected, leading to class imbalance. This imbalance can result in taggers that perform well overall but poorly on minority classes [25]. Therefore, future research should focus on developing data annotation methods that can identify and prioritize data for annotation, thereby addressing the class imbalance issue, saving time and resources. Predicting classification results using unannotated data could also speed up the generation of the first dataset relevant to low-resource languages.

References

1. Abka, A.F.: Evaluating the use of word embeddings for part-of-speech tagging in Bahasa Indonesia. In: 2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA), pp. 209–214 (2016). <https://doi.org/10.1109/IC3INA.2016.7863051>
2. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, pp. 420–434. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44503-x_27
3. Alfeilat, H.A.A., et al.: Effects of distance measure choice on k-nearest neighbor classifier performance: a review. Big Data **7**, 221–248 (2019). <https://doi.org/10.1089/big.2018.0175> <https://www.liebertpub.com/doi/10.1089/big.2018.0175>

4. Ariffin, S.N.A.N., Tiun, S.: Improved POS tagging model for Malay twitter data based on machine learning algorithm. *Int. J. Adv. Comput. Sci. Appl.* **13**(7) (2022). <https://doi.org/10.14569/IJACSA.2022.0130730>, <http://dx.doi.org/10.14569/IJACSA.2022.0130730>
5. Aryoyudanta, B., Adji, T.B., Hidayah, I.: Semi-supervised learning approach for Indonesian named entity recognition (NER) using co-training algorithm. In: 2016 International Seminar on Intelligent Technology and Its Applications (ISITIA), pp. 7–12 (2016). <https://doi.org/10.1109/ISITIA.2016.7828624>
6. Brown, P.F., Desouza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C.: Class-based n-gram models of natural language. *Comput. Linguist.* **18**(4), 467–479 (1992)
7. Chamid, A.A., Widowati, Kusumaningrum, R.: Graph-based semi-supervised deep learning for Indonesian aspect-based sentiment analysis. *Big Data Cogn. Comput.* **7**(1) (2023). <https://doi.org/10.3390/bdcc7010005>, <https://www.mdpi.com/2504-2289/7/1/5>
8. Chen, M.F., Cohen-Wang, B., Mussmann, S., Sala, F., Ré, C.: Comparing the value of labeled and unlabeled data in method-of-moments latent variable estimation (2021)
9. Chua, C.L., Lim, T.M., See, K.T.: An overview of part-of-speech tagging methods and datasets for Malay language. In: 2023 IEEE 8th International Conference On Software Engineering and Computer Systems (ICSECS), pp. 89–95 (2023). <https://doi.org/10.1109/ICSECS58457.2023.10256423>
10. Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8440–8451 (2019). <https://doi.org/10.18653/v1/2020.acl-main.747>, <http://arxiv.org/abs/1911.02116>
11. DRORY, A.: Individual differences in boredom proneness and task effectiveness at work. *Pers. Psychol.* **35**, 141–151 (1982). <https://doi.org/10.1111/j.1744-6570.1982.tb02190.x>, <https://onlinelibrary.wiley.com/doi/10.1111/j.1744-6570.1982.tb02190.x>
12. van Engelen, J.E., Hoos, H.H.: A survey on semi-supervised learning. *Mach. Learn.* **109**, 373–440 (2020). <https://doi.org/10.1007/s10994-019-05855-6>, <http://link.springer.com/10.1007/s10994-019-05855-6>
13. Fudholi, D.H., Juwairi, K.P.: Classifying medical document in Bahasa Indonesia using semi-supervised learning. In: IOP Conference Series: Materials Science and Engineering, p. 012015. IOP Publishing (2021)
14. Han, X., et al.: Pre-trained models: past, present and future (2021)
15. Leonandya, R.A., Distiawan, B., Praptono, N.H.: A semi-supervised algorithm for Indonesian named entity recognition. In: 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI), pp. 45–50 (2015). <https://doi.org/10.1109/ISCBI.2015.15>
16. Li, B., Hou, Y., Che, W.: Data augmentation approaches in natural language processing: a survey. *AI Open* **3**, 71–90 (2022)
17. Liang, P.: Implementation of the brown hierarchical word clustering algorithm (2012). <https://github.com/percyliang/brown-cluster>
18. Nivre, J., et al.: Universal Dependencies v2: an evergrowing multilingual treebank collection. In: Proceedings of the Twelfth Language Resources and Evaluation Conference, pp. 4034–4043. European Language Resources Association, Marseille (2020). <https://aclanthology.org/2020.lrec-1.497>
19. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)

20. Peng, H.: A comprehensive overview and survey of recent advances in meta-learning (2020)
21. Pisceldo, F., Adriani, M., Manurung, R., et al.: Probabilistic part of speech tagging for Bahasa Indonesia. In: Third international MALINDO workshop, pp. 1–6 (2009)
22. Riza, H., et al.: Introduction of the Asian language treebank. In: 2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA), pp. 1–6 (2016). <https://doi.org/10.1109/ICSDA.2016.7918974>
23. Stratos, K., Collins, M.: Simple semi-supervised POS tagging. In: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, pp. 79–87. Association for Computational Linguistics, Denver, Colorado (2015). <https://doi.org/10.3115/v1/W15-1511>, <https://aclanthology.org/W15-1511>
24. Sun, M., et al.: Semi-supervised category-specific review tagging on Indonesian E-commerce product reviews. In: Proceedings of the 3rd Workshop on e-Commerce and NLP, pp. 59–63. Association for Computational Linguistics, Seattle (2020). <https://doi.org/10.18653/v1/2020.ecnlp-1.9>, <https://aclanthology.org/2020.ecnlp-1.9>
25. Wasikowski, M., Chen, X.W.: Combating the small sample class imbalance problem using feature selection. IEEE Trans. Knowl. Data Eng. **22**(10), 1388–1400 (2010). <https://doi.org/10.1109/TKDE.2009.187>
26. Wicaksono, A.F., Purwarianti, A.: HMM based part-of-speech tagger for Bahasa Indonesia. In: Fourth International MALINDO Workshop, Jakarta (2010)
27. Zhuang, F., et al.: A comprehensive survey on transfer learning (2020)



Alleviating Sparsity to Enhance Group Recommendation with Cross-Linked Domain Model

Yui Chee Xuan¹, Rosmamalmi Mat Nawi^{1(✉)}, Nurul Aida Osman²,
and Nur Ziadah Harun¹

¹ Faculty of Computer Science and Information Technology, Tun Hussein Onn University of Malaysia, 86400 Parit Raja, Johor, Malaysia

gi220016@student.uthm.edu.my, {rosmamalmi, nurziadah}@uthm.edu.my

² Computer and Information Sciences Department and Centre for Research in Data Science, University Technology Petronas, 32610 Seri Iskandar, Perak Darul Ridzuan, Malaysia
nurulaida.osman@utp.edu.my

Abstract. As a new search paradigm emerges, users' perspectives on information searching shift from finding information to receiving it. Recommender systems (RS) are an emerging method of obtaining information. RS has succeeded in many conventional domains, such as social media and online video platforms like 'YouTube.' An insufficient number of user-item ratings triggers obstacles to individual RS and group recommender systems (GRS). Data sparsity becomes an issue as a result of this incompleteness. The accuracy of recommendations given to a group suffers when there is a lack of data within the group. It happens as a result of ineffective group formation, which usually involves individuals who possess sparse user profiles. Most of the research being done now concentrates on this problem after group formation. Assuming that handling data sparsity at the individual level would be more efficient, this study concentrated on data sparsity before the group formation process. The primary goal is to create a cross-domain technique with Linked Open Data (LOD) technology to ensure that problems with data sparsity can be addressed before the group formation process. The experiment of the proposed method leveraging LOD and cross-domain knowledge simultaneously in this study suggests that it achieved the lowest prediction errors compared to other experiments carried out in this study. A more accurate rating prediction can further alleviate data sparsity in the user-item matrix before group formation using user profiles. Hence, this research will enhance the quality of recommendations by reducing data sparsity in user profiles.

Keywords: Group recommender system · linked open data · data sparsity · cross-domain

1 Introduction

As social and group activities like vacationing, watching movies, traveling to scenic spots, and participating in sporting events produce a lot of information requirements, recommender systems (RS) for user groups are becoming increasingly popular.

In addition, according to Felfernig et al. [1], group recommender system (GRS) is a relatively new technology with few known commercially practical applications compared to conventional RS. Despite the fact that there is still little study focused on recommendations to a group of users, Xu et al. [2] report that there has been a significant rise in the demand for such recommendation applications.

GRS still has difficulties making reliable recommendations when data is limited [3]. This situation emerged due to the association between the GRS groups and insufficient data. Group formation will not be effective if user preferences are missing from the user profile. Hence, highlighting the lack of information in the group profile is crucial to offering groups high-quality and pertinent recommendations [3, 4].

2 Literature Review

Most recent studies use cross-domain [5–7] and Linked Open Data (LOD) technologies [8] for individual RS, including reducing data sparsity [8–10]. However, our literature survey indicates that no research has been done focusing on the prior group formation in GRS that simultaneously employs the cross-domain approach and LOD technology. Moreover, the GRS studies that independently concentrate on cross-domain or LOD technology did not address the group formation of GRS.

To guarantee that data sparsity concerns can be minimized before the group formation procedure, adopting the technique via cross-domain integration using LOD technology is suggested. The purpose of this study is to ascertain whether using LOD technology in conjunction with a cross-domain strategy will improve the quality of recommendations made to the group.

The data sparsity issue is addressed between groups, as evidenced by the current GRS research studies that address it after group formation. However, this study hypothesizes that it would be more efficient if the data sparsity issue were minimized before the group formation process. While many methods have been proposed to overcome data sparsity, such as recursive filtering approaches [11], artificial neural network approaches [12], and data imputation approaches [13], GRSs have yet to receive much attention.

Table 1. Related works pertaining to cross-domain in group recommender system.

Author	Method	Dataset and Domain
Liang et al. [14]	“Hierarchical attention neural network-based cross-domain group recommendation method (HANCDGR)”	1. Mafengwo (tourism website) 2. Yelp (restaurant dataset) 3. CAMRa2011 (movie rating records of group members) 4. MovieLens1M 5. MovieLens25M 6. MovieLens-Simi
Richa and Bedi [6]		Tourism and its sub-domain comprise hotels, restaurants, and others

Table 1 highlights relevant studies that use cross-domain approaches with data from other relevant domains to alleviate data sparsity for group recommendations. We also provide a brief overview concerning the study’s scope.

2.1 Group Recommender System

The emerging demand to provide group recommendations led to the establishment of GRS. GRS [3, 14] methods often adhere to a three-step procedure: (a) group formation – forming group members by identifying users with similar preferences, (b) group modeling – aggregating group members’ preferences, and (c) prediction – predicting the unrated items. Within the framework of GRS, groups can be classified in a variety of ways based on various member-related attributes, such as the types of preferences or the cause for the group’s establishment [15].

2.2 Cross-Domain Recommender System

Cross-domain methods [5–7] have largely been studied to enhance recommendations in a target domain with insufficient user preferences. The cross-domain approach is one practical solution to addressing RS’s data sparsity and cold-start issues [7]. Transferring knowledge from the source domain to enhance the knowledge in the target domain is a popular cross-domain approach to reducing data sparsity issues. By taking advantage of item attributes and user preferences in several domains that are linked to the target domain, it aims to overcome the data shortage.

2.3 Linked Open Data

LOD [3, 8] is a successful manifestation of data links on the Web. It combines disparate data from multiple sources across businesses to create new knowledge and allow sophisticated services and applications. Linked data assets in the so-called “LOD Cloud” [16] were set up to publish graph-shaped data assets in an openly accessible manner using standard Web protocols [16]. Data publishers from various fields have released many datasets based on LOD principles over the past ten years.

3 Methodology

Figure 1 introduces our proposed model in a research framework used in this study to develop a cross-linked domain model that utilizes LOD technology and cross-domain methods simultaneously, as discussed in the prior sections, respectively, to generate enhanced data in the target domain. It consists of mapping linked datasets through this model and its potential for future implementation in collaborative GRS.

As a reference shown in the top left of Fig. 1, we determined Dataset 1 from the movie domain as the target domain, intending to enrich the original sparse movie dataset into an enhanced dataset with lower sparsity through the proposed model. In contrast, Dataset 2 from the music domain will be the source domain in this study. The extracted auxiliary information from the music domain is aimed to transfer to the movie target domain. To leverage LOD technology, we used DBpedia for data extraction through the linked datasets on its platform using ‘SPARQL Protocol and RDF Query Language’ (SPARQL). DBpedia represents extracted information using the Resource Description Framework (RDF). RDF is used to combine data from several sources and domains

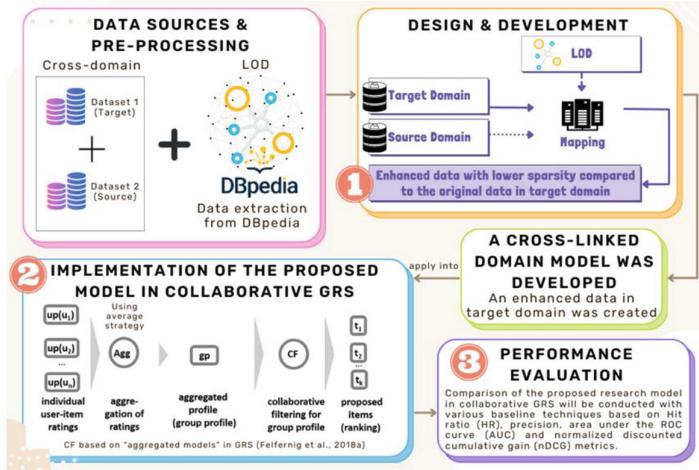


Fig. 1. Research framework.

on DBpedia. Hence, SPARQL can extract the linked data from DBpedia, one of the platforms in the LOD cloud.

We identified the mapping relation between the chosen movie dataset and the linked datasets on DBpedia to pull the auxiliary information from both the movie and music domains into the enhanced dataset in the target domain, as shown in the first phase, in the top right of Fig. 1. This is to generate an enriched dataset with more helpful information to achieve a higher prediction accuracy. Transferring extracted auxiliary data into the movie domain can alleviate the sparsity in a user-movie matrix.

This study developed a cross-linked domain model that leverages LOD technology and cross-domain simultaneously to extract auxiliary information from another domain and evaluate its performance in collaborative individual RS. This ensures that the data sparsity can be alleviated before group formation using individual user profiles.

The subsequent phases of implementation of our proposed model and its performance evaluation, as shown at the bottom of Fig. 1, will be discussed in Sect. 5: Conclusion and Recommendation. This study will end before the second phase, as shown in Fig. 1, by proving that the enhanced data can accurately predict the individual ratings in the sparse user-item matrix. The following experiment in this study will evaluate whether the proposed model can alleviate the data sparsity in individual user profiles before group formation, assuming that addressing data sparsity at the individual level would be more efficient.

3.1 Experiment Setup

As discussed, we classified music as the source domain and movies as a sparse target domain in this experiment. We retrieved the Movie Lens 1 million rating dataset¹ online to evaluate the performance of the proposed method in collaborative individual RS using

¹ Movie Lens 1 million dataset can be retrieved from <https://grouplens.org/datasets/movielens/>.

the singular value decomposition (SVD) algorithm as a matrix factorization approach to study underlying patterns and interactions in the user-movie matrix. We sampled 10% from the 1 million rows in the original dataset into 100 K rows to avoid easy memory crashes due to the limited computational access. We converted the raw data with movie items, users, and ratings into a user-movie matrix.

The sparsity of this sampled 100 K dataset is 0.9949. The sparsity divides the number of zero elements in the user-movie matrix by the total number of elements in the matrix as the formula (1) shown below. The result will be a decimal between 0 and 1, where 0 indicates no sparsity (all elements are non-zero), and 1 indicates complete sparsity (all elements are zero). The formula for sparsity is:

$$\text{Sparsity} = \frac{\text{Number of zero elements}}{\text{Total number of elements}} \quad (1)$$

We used DBpedia to extract and collect additional LOD features in this study. We utilized the linked datasets from DBpedia to extract auxiliary music and movie features. DBpedia is one of the LOD platforms that links movies and related information in the source and target domains, which are the music and movie domains in this study, respectively.

To create the model, domain item information was extracted from the linked datasets, which link to items and concepts in the source and target domains. This operation needs data mapping to obtain the data from the LOD dataset. Thus, it must recognize bridges as a strategy for collecting LOD information and transferring information across domains. The objective was to alleviate the sparse target domain by examining a prospective collaborator represented by a collection of users or items from a source domain. This study used the ‘movie title’ feature as the bridge between Movie Lens 1 million and linked datasets.

Since the movie titles were not matched between these two datasets, we used the mapping dataset by Noia et al. [17, 18] to map the ‘title’ feature in the MovieLens 1M dataset to the DBpedia for data extraction. As the data in the movie target domain is limited, we utilized LOD technology to collect and extract additional movie-related features of ‘director’ and ‘starring’ that are considered in the same movie domain. However, the available helpful feature is also limited to a specific domain.

Given the ideas, we leveraged LOD technology to collect and extract the ‘music composer’ feature from another domain. Hence, we utilized the cross-domain feature in this study of using ‘music composers’ of the movies in the music domain. We then identified the ‘music composer’ of the movies as auxiliary information in the source domain. We assumed that if someone likes music composed by their beloved composer, they might watch the movie and have a higher chance of loving it. For example, John William is one of the famous music composers who wrote the theme series for most Star Wars movies. It is a simple instance that the user has a higher chance of loving all Star Wars movies. A similar assumption applies to ‘starring’ and ‘director’ features.

Algorithm 1 in Fig. 2 forms an enhanced trainset to train our proposed model with additional LOD and cross-domain information. For this study, directors who have directed at least three movies and actors who have starred in at least three movies for all rows in the Movie Lens dataset were extracted, respectively.

As shown in Fig. 2, a sample of 100K rows from the enhanced Movie Lens 1 million dataset with three additional feature information was created as input for the below

Algorithm 1. A user-movie matrix was formed using a pivot table. The “Not a Number” (NaN) were missing values in Python and filled with 0, assuming NaN means the user has not rated the movie. The ‘Music Composer’ and ‘Director’ feature information were encoded using one-hot encoding, while ‘Starring’ was converted into a binary representation using the respective method. This is to combine the encoded LOD and cross-domain feature information with the user-movie matrix to build an enhanced trainset as the output of this algorithm.

Algorithm 1

Input: Sampled 100K dataset from enhanced Movie Lens 1 million data with three features ('director' & 'starring' \geq in 3 movies and respective 'music composer')

Output: Enhanced trainset of the extended user-movie matrix with LOD and cross-domain information

START

1. Create a pivot table (matrix) from the input dataset
2. Handle and fill NaN values with 0 in the 'Music Composer,' 'Director,' and 'Starring' feature columns
3. One-hot encode 'Music Composer' and 'Director';
`encoder_composer = OneHotEncoder(sparse=False)`
`encoder_director = OneHotEncoder(sparse=False)`
4. Convert 'Starring' to binary representation;
`mlb_starring = MultiLabelBinarizer()`
5. Combine the encoded features with the user-movie matrix
6. Build a trainset from the combined dataset;
`trainset_with_features =`
`data_with_features.build_full_trainset()`

END

Fig. 2. Algorithm 1 shows the pseudo-code we used to form an enhanced trainset to train our proposed cross-linked domain model at the initial stage.

4 Result and Discussion

The enriched data is then shown via a graph visualization using Neo4j. Program Code 1 showed how the cypher queries were used to split actor names and add ‘starring’ nodes to link to the movie nodes.

```
FOREACH (starringName IN split(row.Starring, ', ') |  
    MERGE (s:Starring {name: trim(starringName)} )  
    MERGE (s)-[:STARRED_IN] -> (m)  
)
```

Program Code 1. Cypher queries in linking the movie nodes.

The graph visualization in Fig. 3 showed the data linkage between the dataset of MovieLens 1M and DBpedia. We can see the nodes and the relationship between three additional linked attributes with cross-domain from the movie and music in this study (‘starring,’ ‘director,’ and ‘music composer’). This figure illustrated that the cross-domain feature from the music domain can link between different subsets of the movie features. It demonstrated another underlying pattern of user’s preference for the movies.

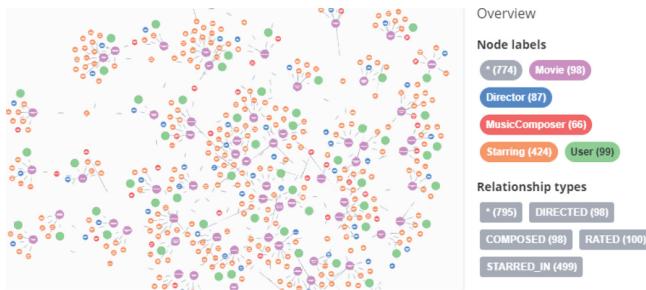


Fig. 3. Linkage of nodes in graph visualization.

This study found similar users by identifying the underlying pattern through latent factors in matrix factorization. This study used collaborative filtering of the SVD algorithm as the matrix factorization method. The user-movie matrix will then be a lower dimension, which can identify latent factors in the model training.

We adopted ‘root mean square error (RMSE),’ ‘mean square error (MSE),’ and ‘mean absolute error (MAE)’ as the fundamental evaluation metrics to ensure diversity in the evaluation process of the trained models to test the performance of our proposed method. The lower the errors, the higher the accuracy of predictions compared to the actual ratings.

We first evaluated the original user-movie matrix without additional features on the linked data using the SVD algorithm, as shown below in Row 1 in Table 2. The prediction errors were among the highest compared to the other methods that used an enriched user-movie matrix, which indicates that this accuracy is poorer in this study.

Moreover, we evaluated the enriched user-movie matrix with each auxiliary LOD and cross-domain feature using a similar SVD algorithm as matrix factorization. The linked

data lowered the error and increased the accuracy of rating predictions compared to the previously mentioned results. The result in Table 2 indicated that using the enriched matrix with only one auxiliary LOD or cross-domain feature was not significant enough to lower the prediction errors, as shown in Row 2–4. Even considering two LOD features of ‘director’ and ‘starring’ simultaneously from the same movie domain during the model training, the results were still not in favor, as the errors did not decrease significantly, as shown in Row 5.

Table 2. Experiment results of model testing using different original and enriched datasets with LOD technology and cross-domain based on RMSE, MSE, and MAE.

Method	RMSE	MSE	MAE
1. SVD by using original user-movie matrix from MovieLens 1M	0.9455	0.8939	0.7520
2. SVD-LOD by using an enhanced matrix with a ‘director’ feature	0.7016	0.4923	0.5575
3. SVD-LOD by using an enhanced matrix with a ‘starring’ feature	0.7019	0.4926	0.5587
4. SVD-CDLOD by using an enhanced matrix with a ‘music composer’ feature from the other music domain	0.7021	0.4929	0.5592
5. SVD-LOD by using an enhanced matrix with both ‘director’ and ‘starring’ features from the same movie domain	0.7009	0.4913	0.5581
6. SVD-CDLOD by using an enhanced matrix that leveraged cross-domain information with three ‘director,’ ‘starring,’ and ‘music composer’ features	0.5303	0.2813	0.4223

As shown in Table 2, this study’s proposed method of simultaneously utilizing LOD technology with cross-domain features achieved the lowest errors. It indicated the highest accuracy using the experiment’s three ‘director,’ ‘starring,’ and ‘music composer’ features, as shown in Row 6. This suggested that our proposed method of using enriched data to make rating predictions was superior to using sparse data given as the original unenriched dataset. Hence, the enhanced data can then be used to form a user profile for further group formation to provide recommendations to groups.

We experimented with some combination of the auxiliary LOD and cross-domain features related to the patterns of user-movie relationships during the model training using SVD. This aimed to lower the errors between predicted and actual ratings in the movie target domain.

The results strongly suggested that using auxiliary LOD features can lower the prediction errors compared to the actual ratings. Hence, the performance of the recommendation was better and more accurate. The errors were even lower after using the cross-domain and LOD features simultaneously rather than those from a single domain. The result suggested a more accurate prediction of the missing ratings in the original dataset to alleviate sparsity in individual user profiles before group formation. Group formation and implementation into GRS will be carried out in the future.

5 Conclusion and Recommendation

Grouping and clustering are significantly impacted by the sparsity of the data. While many efforts have been made to minimize data sparsity issues in RS for individuals, the impact on GRS remains a significant concern. GRS struggles to deliver reliable recommendations when user profile data is sparse. Our proposed method eliminated data sparsity in user profiles to provide more effective group recommendations. As shown in Fig. 1, the model related to a cross-linked domain will subsequently be implemented in the collaborative GRS with the selected aggregation strategy. Future research will build on this study's findings by including social metadata in cross-domain recommendations. Additionally, algorithms for ranking and filtering objects in the target domain by computing semantic similarities will be created. Comparison of the proposed research model in collaborative GRS will be subsequently conducted with various baseline techniques based on ‘hit ratio (HR),’ ‘precision,’ ‘area under the ROC curve (AUC),’ and ‘normalized discounted cumulative gain (nDCG)’ metrics.

Acknowledgments. This research was supported by the Ministry of Higher Education (MOHE) through the Fundamental Research Grant Scheme (FRGS/1/2022/ICT02/UTHM/03/3). The authors would like to express their gratitude to Tun Hussein Onn University of Malaysia for providing financial assistance and assistance during the literature search throughout this study.

References

1. Felfernig, A., Boratto, L., Stettinger, M., Tkalcic, Marko: Explanations for groups. In: Group Recommender Systems. SECE, pp. 105–126. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75067-5_6
2. Xu, H., Ding, Y., Sun, J., Zhao, K., Chen, Y.: Dynamic group recommendation based on the attention mechanism. Future Internet **11**, 198 (2019). <https://doi.org/10.3390/fi11090198>
3. Nawi, R.M., Noah, S.A.M., Zakaria, L.Q.: Integration of linked open data in collaborative group recommender systems. IEEE Access. **9**, 150753–150767 (2021). <https://doi.org/10.1109/ACCESS.2021.3124939>
4. Felfernig, A., Boratto, L., Stettinger, M., Tkalcic, M.: Evaluating group recommender systems. In: Group recommender systems. SECE, pp. 59–71. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75067-5_3
5. Anwar, T., Uma, V.: CD-SPM: Cross-domain book recommendation using sequential pattern mining and rule mining. J. King Saud Univ. – Comput. Inform. Sci. **34**, 793–800 (2022). <https://doi.org/10.1016/j.jksuci.2019.01.012>
6. Richa, Bedi, P.: Trust and distrust based cross-domain recommender system. Appl. Artif. Intell. **35**, 326–351 (2021). <https://doi.org/10.1080/08839514.2021.1881297>
7. Ma, M., et al.: Mixed information flow for cross-domain sequential recommendations. ACM Trans. Knowl. Discov. Data **16**, 1–32 (2022). <https://doi.org/10.1145/3487331>
8. Mahdi, A.M., Hadi, A.S.: Utilizing LOD relationships and FOAF vocabularies for top-N Recommender system. In: 2021 1st Babylon International Conference on Information Technology and Science (BICITS), pp. 98–103. IEEE (2021)
9. Behera, G., Nain, N.: Handling data sparsity via item metadata embedding into deep collaborative recommender system. J. King Saud Univ. – Comput. Inform. Sci. **34**, 9953–9963 (2022). <https://doi.org/10.1016/j.jksuci.2021.12.021>

10. Roko, A., Almu, A., Saidu, I.: An enhanced data sparsity reduction method for effective collaborative filtering recommendations. *Int. J. Educ., Manag. Eng.* **10**, 27–42 (2020). <https://doi.org/10.5815/ijeme.2020.01.04>
11. Ihm, S.-Y., Lee, S.-E., Park, Y.-H., Nasridinov, A., Kim, M., Park, S.-H.: A technique of recursive reliability-based missing data imputation for collaborative filtering. *Appl. Sci.* **11**, 3719 (2021). <https://doi.org/10.3390/app11083719>
12. Althbiti, A., Alshamrani, R., Alghamdi, T., Lee, S., Ma, X.: Addressing data sparsity in collaborative filtering based recommender systems using clustering and artificial neural network. In: 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0218–0227. IEEE (2021)
13. Inan, E., Tekbacak, F., Ozturk, C.: Moreopt: a goal programming based movie recommender system. *J. Comput. Sci.* **28**, 43–50 (2018). <https://doi.org/10.1016/j.jocs.2018.08.004>
14. Liang, R., Zhang, Q., Lu, J., Zhang, G., Wang, J.: A cross-domain group recommender system with a generalized aggregation strategy. In: Developments of Artificial Intelligence Technologies in Computation and Robotics, pp. 455–462. WORLD SCIENTIFIC (2020)
15. Valera, A., Lozano Murciego, Á., Moreno-García, M.N.: Context-aware music recommender systems for groups: a comparative study. *Information* **12**, 506 (2021). <https://doi.org/10.3390/info12120506>
16. Haller, A., Fernández, J.D., Kamdar, M.R., Polleres, A.: What are links in linked open data? a characterization and evaluation of links between knowledge graphs on the web. *J. Data Inform. Qual.* **12**, 1–34 (2020). <https://doi.org/10.1145/3369875>
17. Noia, T.D., Ostuni, V.C., Tomeo, P., Sciascio, E.D.: SPrank. *ACM Trans Intell. Syst. Technol.* **8**, 1–34 (2017). <https://doi.org/10.1145/2899005>
18. Fernández-Tobías, I., Tomeo, P., Cantador, I., Di Noia, T., Di Sciascio, E.: Accuracy and diversity in cross-domain recommendations for cold-start users with positive-only feedback. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 119–122. ACM, New York, NY, USA (2016)



Evaluating Deep Transfer Learning Models for Detecting Various Face Mask Wearings

Pei-Jin Goh, Meei-Hao Hoo, and Kok-Chin Khor^(✉)

Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, Sungai Long Campus, Jalan Sungai Long, Bandar Sungai Long, 43000 Kajang, Selangor, Malaysia
gohpeijin@gmail.com, {hoomh, kckhor}@utar.edu.my

Abstract. In Malaysia, wearing a face mask is no longer mandatory to prevent COVID-19. However, such wearing may be important in future if another outbreak occurs. Besides, wearing a face mask is also important in environments such as cleanrooms, operating theatres and crowded places. For these reasons, developing automated systems for detecting face mask-wearing is crucial. This study evaluates deep transfer learning models to detect four categories of face mask wearing: (i) not wearing masks, (ii) wearing single masks, (iii) wearing masks incorrectly, and (iv) wearing double masks. Transfer learning methods were adopted by using five pre-trained models: (i) VGG-16, (ii) MobileNetV2, (iii) ResNet-152, (iv) InceptionV3 and (v) Xception. These models were trained based on 2,000 images collected from various sources and then augmented. The results showed that ResNet-152 outperformed the others by achieving 86.67% accuracy on the testing set (120 images from the other distribution) and 84.47% accuracy on the videos captured using a smartphone.

Keywords: Face mask detection · transfer learning · images · videos · double masks

1 Introduction

Wearing face masks in public was once mandatory in many Asian countries during the pandemic. The first wave of pandemics struck Malaysia between January 25, 2020, and February 16, 2020. The second wave occurred in 2020 between February 27 and June 30, forcing the Malaysian government to take various Movement Control Order (MCO) phases to handle the spread of COVID-19. The Sabah state election on September 8, 2020, marked the start of the third wave of COVID-19 cases, which saw a sharp increase in confirmed cases and thousands of new cases reported daily [1].

The World Health Organization (WHO) states that wearing a face mask is the best way to prevent COVID-19 in public and crowded areas. Therefore, wearing a face mask is important to protect the public from airborne diseases like COVID-19. Due to the increasing public awareness of airborne diseases, it has become a customised culture for people in many Asia countries like Malaysia to wear face masks daily. Besides preventing possible public outbreaks, wearing face masks is important in places like

cleanrooms, operating theatres, etc. In Europe, the Scottish Healthcare Workers Coalition has recently criticised their government for ending universal mask-wearing in hospitals, putting medical personnel at risk [2].

It is time-consuming and ineffective to manually determine whether someone is wearing a mask, especially during an outbreak. Having a large enough workforce to be placed in many locations for monitoring is impossible. Seeing the need for face mask detection at present and possibly in the future, researchers have tried to develop detection techniques, particularly deep learning. This study contributes to understanding the effectiveness of deep learning models for automated face mask detection. We trained five pre-trained deep learning models, evaluated them and found the best performer for four categories of face mask wearing.

The remainder of this paper is organised as follows: Sect. 2 provides a comprehensive review of deep learning, an overview of current transfer learning models and recent works on face mask detection. Section 3 presents an overview of this study's methodology and data sets. Subsequently, we present and discuss the experimental results of evaluating the five models in Sect. 4. Finally, Sect. 5 concludes the study.

2 Literature Review

2.1 Deep Learning

Deep learning is getting more popular due to the big data era since it is more accurate when trained with massive data [3]. Social media, the Internet, search engines, e-commerce sites, and online movies are some possible data sources. Often, such data are unstructured and large, and extracting the features takes much time [4]. Deep learning algorithms can ingest and process unstructured data, including text and images, without the need for feature extraction or data pre-processing, which are necessary for traditional machine learning. The algorithms, i.e., Convolutional Neural Networks (CNNs), can automate feature extraction because they use hierarchical structures that enable them to adopt a non-linear approach, processing data across a series layer and gradually extracting more complex data features [5]. In image recognition, deep learning algorithms recognise the pixels first, followed by the lines and edges. In the subsequent layers, it will recognise more complex shapes and use the shapes to identify objects in images.

Deep learning algorithms contain many hidden layers. Each hidden layer has an activation function that can be utilised to send information between layers. The outputs are used as inputs to calculate the network's final output once all of the hidden layers' outputs have been produced [5, 6]. Forward propagation is the term used to describe this computation across the network. Deep learning algorithms also include backpropagation, which allows recognition errors to be propagated backwards through the layers to enhance recognition capability by adjusting the weights and biases.

2.2 Transfer Learning

In contrast to traditional learning approaches, deep learning heavily relies on large data sets. A roughly linear relationship exists between the amount of data and the size of

a deep learning model [7]. Using a small data set to train deep learning models from scratch may lead to overfitting. Transfer learning is crucial to deep learning because of the lack of data and the need for shorter training times. If a large amount of data is involved, training a model from scratch may take days or weeks. In transfer learning, an existing deep learning model for a task can be retrained using data sets from another source/domain to create a new model for another similar task [8]. The pre-trained models studied in this literature review are Visual Geometry Group (VGG), MobileNet, Residual Network (ResNet), Inception and Extreme Inception (Xception). We used these models as they are state-of-the-art deep learning models in the research domain.

Visual Geometry Group (VGG). The most common VGG networks are VGG-16 and VGG-19. The numbers 16 and 19 represent the weight layers in VGG. VGG networks use convolution layers with a 3×3 filter, a stride of 1, and equal padding. The max pooling layers were 2×2 filters with a stride of 2. These networks employ only 3×3 convolutional layers stacking on each other to increase the depth. Max pooling shall handle the volume reduction, then end with two 4,096 nodes fully connected layers and a softmax classifier. Using multiple 3×3 filters eliminates the need for large-size kernels, extracting complex features at a low cost [8]. VGG-19 performs slightly better than VGG-16 but requires more memory [9]. However, these VGG networks have the disadvantage of being very slow to train and having high architectural weights in disc or bandwidth. VGGs require more than 500 MB of memory size due to their depth and amount of fully connected nodes. Therefore, deploying a VGG takes a long time, so a smaller network architecture is typically chosen [10]. Thus, we used VGG-16 in this study.

MobileNet. A MobileNet model is intended for usage in mobile applications and devices with limited computing capacity. The model is light, as mobile devices cannot afford a large Graphics Processing Unit (GPU) to operate in the background due to space and restrictions [11]. MobileNet employs depthwise separable convolutions, including depthwise and pointwise convolutions. Compared to alternative architectures with the same depth in the network, it has substantially reduced the number of parameters. MobileNetV2 uses an inverted residual structure with depth separable convolution. It begins with a standard 3×3 convolution with 32 channels, followed by 17 bottleneck blocks and ends with a regular 1×1 convolution. Before classification, a global average pooling layer is used, followed only by the classification layer. Due to Depthwise separable Convolution, MobileNet requires less computation and parameters, allowing it to perform better in size, latency, and accuracy. We used MobileNetV2 in this study.

Residual Network (ResNet). ResNet architectures exist in many variants, each built with the same idea but a different number of layers [12]. The most popular architectures are ResNet-34, ResNet-50, ResNet-101 and ResNet-152. The digit indicates the number of neural network layers of the ResNet. When dealing with complicated problems, additional layers are often placed in deep neural networks to improve their accuracy. Adding additional layers allows the layers to learn more complex features gradually. However, He et al. found that the conventional CNN model has a depth threshold limit and shows that a deeper network will generate greater training and test errors [13]. Deep networks are extremely hard to train because of the vanishing gradient problem. When

a gradient is backpropagated to previous layers, repeated multiplication may cause the gradient to become endlessly tiny. Therefore, the network performance degrades as it becomes deeper. ResNet, or residual networks, comprised of Residual Blocks, has solved this difficulty. In the residual block, a direct connection bypasses some layers in between. This is known as a ‘skip connection’ or identity mapping. There are no parameters in the skip connection, so the output from the previous layer is just added to the next layer. The layer’s output is no longer identical due to this skip connection. We used ResNet-152 in this study as it produces the lowest error rate on the ImageNet validation set, according to He et al. [13].

Inception. Before the emergence of Inception, researchers had to figure out which filter sizes to use in deep convolutional neural networks to get the best results. Inception eliminates the necessity for such selections by using filters 1×1 , 3×3 , and 5×5 together. 1×1 convolutions will minimise the dimensions of data travelling through the network. It can also increase the network’s breadth and depth and learn patterns throughout the depth of the input. Furthermore, using 3×3 and 5×5 convolutions allows the network to learn various spatial patterns at different scales. The naive Inception model performs convolution on an input using these three filters and a max-pooling layer. Before the costly filter sizes of convolutions, the 1×1 was employed to compute reduction [14]. The outputs are then concatenated and passed to the next inception module. The pooling layer downsamples the input data by producing a smaller output with a lower height and width. Padding will be added to the pooling layer to ensure that the pooling layer’s output can be concatenated with the output of the convolution layers. We used InceptionV3 in this study as it has shown promising accuracy on the ImageNet dataset.

Extreme Inception (Xception). Xception stands for extreme Inception, which is a more advanced version of Inception. It uses the same number of parameters as Inception-V3 but performs better due to the more efficient usage of model parameters [15]. Following the Xception’s architecture, the data will move through three flows: first, the ‘input flow,’ second, the ‘middle flow,’ and it will be repeated eight times, and at the end, the ‘exit flow’. Batch normalisation is performed in all Convolution and Separable Convolution layers. A depth multiplier of 1 is used by all Separable Convolution layers. Xception outperformed Inception-V3 due to two significant changes: updated depthwise separable convolutions and non-linearity modification.

2.3 Existing Works

There are multiple ways to determine whether a face mask is present or absent using deep learning models: (i) with transfer learning – training a CNN like VGG [16], Inception [17], ResNet [18], MobileNet [19] or (ii) using a CNN backbone and accompany it with other object detection models [20–22].

In an object detection model, there are two types of frameworks: (i) region proposal-based, which is a two-stage object detector like Region-based Convolutional Neural Network (RCNN) and Faster RCNN, and (ii) regression or classification-based, which is a one-stage detector like You Only Look Once (YOLO) and Single Shot MultiBox Detector. The one-stage detector regresses the bounding boxes in a single step. However,

the two-stage detector will first produce region proposals that possibly contain objects. Then, the proposals shall be fine-tuned in the second stage. Therefore, a two-stage detector has a better performance but a slower speed than a one-stage detector.

Several problems have been resolved in recent research works, including detecting faces with/without masks [23], improper mask wearing [24], detecting occlusions on faces [25], types of masks [16], and social distancing [22]. Most of these existing methods focus on detecting no mask-wearing, single mask-wearing, and inappropriate mask-wearing. The former Director General of Health Malaysia, Tan Sri Dr. Noor Hisham Abdullah, recommended using double face masks in May 2021, citing the potential for this preventive measure to reduce COVID-19 transmissions by as much as 96.4% [26]. We thus decided to attempt detecting double masks in this study as the new challenge for this research domain.

This study employs CNN transfer learning, leveraging pre-trained models, i.e., VGG-16, MobileNetV2, ResNet-152, InceptionV3 and Xception from ImageNet dataset. This approach is advantageous when dealing with limited labelled data, reducing the need for large annotated datasets and the risk of overfitting. Fine-tuning these models on our dataset enables efficient feature extraction for object detection, resulting in robust performance with reduced computational cost and training time.

3 Methodology

The methodological flow of this study is as shown in Fig. 1. The data set for training, validation, and testing was prepared by collecting images for four different mask-wearing classes: (i) faces without a mask, (ii) faces with a single mask, (iii) incorrect mask-wearing and (iv) faces with double masks, from four online resources [27–30]. Each class contains 500 images, and in total, we collected 2,000 images. Due to a scarcity of online double-mask images, different search engines were applied to obtain them, and we also created artificially made double-mask images. Out of the four main online sources, the Face Mask Detection (FMD) data set was used as the testing data from a different distribution because its data is largely from candid photos and thus looks more real-life compared to the others. The other three datasets were treated as training and testing sets from the same distribution.



Fig. 1. The methodological flow for this study.

During data pre-processing, all images were cropped into one face per image (Fig. 2), and the data set was then converted from RGB (red, green and blue) to BGR as OpenCV read image in BGR format. Then, the images were resized according to the requirement of the pre-trained models: 224×224 for MobileNetV2, ResNet50, ResNet152 and VGG-16, and 229×229 for Inception-V3 and Xception. Subsequently, data augmentation was



Fig. 2. Sample training data for (a) no mask-wearing, (b) single mask-wearing, (c) improper mask-wearing, (d) double mask-wearing, and (e) artificially made double mask-wearing.



Fig. 3. The sample images resulted from data augmentation.

carried out with 0.1 zoom range, 25° rotation range, 0.1 width shift range, 0.1 height shift range, 0.15 shear range, flipped horizontally, and with the nearest fill mode. The augmentation was done so the models could detect face masks in different scenarios. The images resulting from the data augmentation are shown in Fig. 3.

In feature extraction, the pre-trained convolutional base is frozen to extract features, followed by replacing the top classifier for these four mask-wearing classes. This is to leverage pre-existing knowledge learned by the convolutional base to adapt to new datasets. The five pre-trained models used were MobileNetV2, VGG-16, ResNet-152, Inception-V3 and Xception. The initial stage in transfer learning is to build a base model using CNN architectures, and weights must be assigned to the model. When creating a base model, the final output layer must be removed. The layers from the pre-trained model must be frozen to prevent reinitialising the weights in those layers (Fig. 4). Their pre-trained classifier was replaced with a new one. A flattened layer was added to make the image flat by turning the multidimensional array into one-dimensional. Subsequently, a dense layer with ReLU activation was added since ReLU deals better with images. A dropout layer was added as well to avoid overfitting the model. At last, a dense layer with Softmax activation was added to the final layer. Softmax was used when there were more than two classes in the data. The neuron for the last layer was set to four as only four classes were involved.

The collected 2,000 images were divided into three sets: training (80%), validation (10%), and testing (10%). Initially, the models learned and fit the parameters using the training data set. Then, the validation set was used to fine-tune the models' hyperparameters to improve their mask detection performance. GridSearchCV from Python's Scikit-Learn was used to find the optimal hyperparameter for the models. The testing

set (200 images from the same distribution as the training set) was then used to evaluate the final models. To further evaluate the robustness of the models, we used two other testing data sets that come from other distributions: (i) 120 images from the FMD data set (serves as images from a different distribution) and (ii) 12 real-life videos captured using a smartphone (Fig. 5). To evaluate the models' performance, we used accuracy.

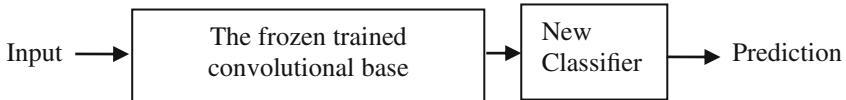


Fig. 4. Replacing the existing classifier of the pre-trained model with a new one (adopted from [31]).



Fig. 5. The test data from other distributions. The top row shows the samples of 120 static images from the FMD data set. The middle row shows the sample images captured from the frames of the videos in the last row.

4 Results and Discussion

The models were fine-tuned using GridSearchCV, and their optimal hyperparameters are shown in Table 1. The models' performance is shown in Table 2. Table 2 shows that ResNet-152 performed well with an accuracy of 0.8300, 0.8667 and 0.8447 for the test set from the same distribution as the training set, the test set from the other distribution

and video frames, respectively. Even though ResNet-152 is not the best performer for the testing set, which comes from the same distribution as the training set, it is the best model for the other two test sets, making it the most robust. ResNet-152, however, has the longest testing duration due to its model depth.

The second-best model is Inception-v3, followed by Xception. The MobileNetV2 has the shortest testing time; however, its accuracies are low. To conclude, ResNet-152 remains the best model as the testing time is so small (milliseconds) that it can be negligible.

Table 1. GridSearchCV Result Summary showing the optimised hyperparameter values and mean test scores. All the models used Adam as the optimiser.

Model	Learning Rate	Dropout Rate	Mean Test Score
MobileNetV2	0.0001	0.3	0.8025
VGG-16	0.001	0.5	0.7737
ResNet-152	0.0001	0.3	0.8512
Inception-V3	0.0001	0.3	0.8112
Xception	0.00001	0.3	0.8350

Table 2. Model Evaluation Summary. ResNet-152 is the best model for detecting face mask-wearing, regardless of images from the other distribution (the FMD data set) or real-life videos captured using a smartphone.

Model	Test Sets					
	200 images (same distribution)		120 images (the FMD data set)		12 Videos (3 videos per class)	
	Accuracy	Time (ms/step)	Accuracy	Time (ms/step)	Accuracy	Time (ms/step)
MobileNetV2	0.8250	20	0.7500	22	0.7090	22
VGG-16	0.8200	53	0.7000	57	0.6104	57
ResNet-152	0.8300	102	0.8667	101	0.8447	110
Inception-V3	0.8350	52	0.8583	58	0.7998	62
Xception	0.8450	92	0.8250	96	0.6763	97

5 Conclusions

In this study, five deep transfer learning models, MobileNetV2, VGG-16, ResNet-152, Inception-V3 and Xception, were trained and evaluated for the detection of four mask wearings: (i) no mask-wearing, (ii) single mask-wearing, (iii) improper mask-wearing and (iv) double mask wearing. ResNet-152 is the best performer, looking at the accuracies it achieved.

However, there are limitations to this study. Firstly, when gathering image data for an individual from afar to a close distance, the trained models will fluctuate in detecting the masks. Secondly, detecting double masks in video frames is difficult; all the models perform weakly with a recall rate of less than 0.30 (not tabulated in this paper). This is because the model always identifies the double masks as single masks when an individual is far from the lens. Another reason for such weak performance is the lack of real-life data for double mask-wearing used in the training.

To enhance the models' performance in detecting masks from far away, we recommend using a better camera or gathering more data sets and using them to train the models. The images of the double masks in the data set are mostly obtained from social media, and most photos collected are of faces close to the lens and come with a full-frontal face. Real-world shooting can be carried out to collect more realistic data for double mask-wearing. Several other models, such as two-stage object detectors (e.g. RCNN and Faster RCNN) and one-stage object detectors (e.g. YOLO and SSD), can be attempted and evaluated in future to improve the detection performance.

References

1. Rampal, L., Liew, B.S.: Malaysia's third COVID-19 wave-a paradigm shift required. *Med. J. Malaysia* **76**(1), 1–4 (2021)
2. Christie, B.: Covid-19: Bring back mandatory mask wearing in health settings, say Scottish workers. *BMJ* (2023). <https://doi.org/10.1136/bmj.p1648>
3. Sharma, N., Reecha, S., Jindal, N.: Machine learning and deep learning applications-a vision. *Glob. Transitions Proc.* **2**(1), 24–28 (2021)
4. Sivarajah, U., Kamal, M.M., Irani, Z., Weerakkody, V.: Critical analysis of big data challenges and analytical methods. *J. Bus. Res.* **70**, 263–286 (2017)
5. Shrestha, A., Mahmood, A.: Review of deep learning algorithms and architectures. *IEEE Access* **7**, 53040–53065 (2019)
6. Neural Networks: <https://www.ibm.com/cloud/learn/neural-networks>. Last accessed 1 Mar 2021
7. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., Liu, C.: A survey on deep transfer learning. In: Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I. (eds.) ICANN 2018. LNCS, vol. 11141, pp. 270–279. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01424-7_27
8. Krishna, S.T. and Kalluri, H.K.: Deep learning and transfer learning approaches for image classification. *International Journal of Recent Technology and Engineering (IJRTE)* **7**(5S4), 427–432 (2019)
9. Shu, M.: Deep learning for image classification on very small datasets using transfer learning (2019)
10. ImageNet: VGGNet, ResNet, Inception, and Xception with Keras. <https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/>. Last accessed 19 July 2021
11. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520 (2018)
12. Sachan, A.: Detailed Guide to Understand and Implement ResNets, <https://cv-tricks.com/keras/understand-implement-resnets/>

13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
14. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9 (2015)
15. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251–1258 (2017)
16. Hussain, S., et al.: IoT and deep learning based approach for rapid screening and face mask detection for infection spread control of COVID-19. *Appl. Sci.* **11**(8), 3495 (2021)
17. Jignesh Chowdary, G., Punn, N.S., Sonbhadra, S.K., Agarwal, S.: Face mask detection using transfer learning of inceptionv3. In: Big Data Analytics 8th International Conference, BDA 2020, pp. 81–90 (2020)
18. Sethi, S., Kathuria, M., Kaushik, T.: Face mask detection using deep learning: an approach to reduce risk of Coronavirus spread. *J. Biomed. Inform.* **120**, 103848 (2021)
19. Mercaldo, F., Santone, A.: Transfer learning for mobile real-time face mask detection and localization. *J. Am. Med. Inform. Assoc.* **28**(7), 1548–1554 (2021)
20. Loey, M., Manogaran, G., Taha, M.H.N., Khalifa, N.E.M.: A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. *Measurement* **167**, 108288 (2021)
21. Zhang, J., Han, F., Chun, Y., Chen, W.: A novel detection framework about conditions of wearing face mask for helping control the spread of COVID-19. *IEEE Access* **9**, 42975–42984 (2021)
22. Yadav, S.: Deep learning based safe social distancing and face mask detection in public areas for Covid-19 safety guidelines adherence. *Int. J. Res. Appl. Sci. Eng. Technol.* **8**(7), 1368–1375 (2020)
23. Loey, M., Manogaran, G., Taha, M.H.N., Khalifa, N.E.M.: Fighting against COVID-19: a novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection. *Sustain. Cities Soc.* **65**, 102600 (2021)
24. Jiang, X., Gao, T., Zhu, Z., Zhao, Y.: Real-time face mask detection method based on YOLOv3. *Electronics* **10**(7), 837 (2021)
25. Jiang, M., Fan, X.: Retinamask: A face mask detector, arXiv preprint [arXiv:2005.03950](https://arxiv.org/abs/2005.03950) (2020)
26. Covid-19: Wearing of double face masks recommended, says Health DG. <https://www.thestar.com.my/news/nation/2021/05/22/covid-19-wearing-of-double-facemasks-recommended-says-health-dg>. 11 May 2021
27. Medical Mask (MM) Dataset. <https://humansintheloop.org/mask-dataset-download/?submissionGuid=add801a1-b11b-4e08-8825-3f3a1d2cce2c>
28. MAFA data set. <https://www.kaggle.com/datasets/rahulmangalampalli/mafa-data>
29. MaskedFace-Net data set. <https://github.com/cabani/MaskedFace-Net>
30. FMD data set. <https://www.kaggle.com/andrewmvdf/face-mask-detection>
31. Transfer Learning in Image Classification: how much training data do we really need? <https://towardsdatascience.com/transfer-learning-in-image-classification-how-much-training-data-do-we-really-need-7fb570abe774>. Last accessed 30 Nov 2023



Classification of Stunting Events: Case Study in West Java, Indonesia

Ummi Azizah Rachmawati¹(✉), Puspa Setia Pratiwi², Yusnita³, K. Rama Abirami⁴, and Farrel Yuda Praditya¹

¹ Department of Informatics, Faculty of Information Technology, Universitas YARSI, Jakarta, Indonesia

ummi.azizah@yarsi.ac.id, farrel.yudha@student.yarsi.ac.id
² Indonesia International Institute of Life-Science (I3L), Jakarta, Indonesia

puspa.pratiwi@i3l.ac.id

³ Public Health Department, Faculty of Medicine, Universitas YARSI, Jakarta, Indonesia
yusnita@yarsi.ac.id

⁴ Department of Electrical and Computer Engineering, Faculty of Engineering and Science, Curtin University, Sarawak, Malaysia
rama.abirami@curtin.edu.my

Abstract. This study created a categorization model based on stunting episodes, separating stunting and non-stunting groups. Stunting is a persistent nutritional condition in toddlers who are shorter in height than other children their age. Data mining is one of the strategies used to extract knowledge from a large amount of big data and convert it into fresh data that can be understood as technology advances. The categorization model results provide a foundation for mapping the possibility of stunting events using the following methods: K-Nearest Neighbour, Logistic Regression, and Support Vector Machine (SVM). The proposed categorization model identifies stunting in toddlers from Pandeglang Regency, West Java, Indonesia, and Malaysia. The sample taken in this study consists of 798 children data. This study concludes that the three tests were successfully carried out, with data sharing ratios of 80%: 20%, 75%: 25%, and 70%: 30%, respectively. The model was repeated 100 times using a Support Vector Machine (SVM) and Logistic Regression (LR) and evaluated for precision, recall, f-1 score, and accuracy. The results of three experiments revealed that the best method is LR utilizing the chi-squared test feature selection, which achieved an accuracy of 85%.

Keywords: Stunting · Data Mining · Classification · SVM · Logistic Regression

1 Introduction

Stunting is a persistent dietary condition that causes toddlers to be shorter than other children their age. When determining the nutritional condition of toddlers, one of which is stunting, using an anthropometric scale, only four internal parameters are considered: gender, age, weight, and height [1]. Stunting is one of the consequences of malnutrition in terms of short body size to exceed the deficit of -2SD below standard [2]. Stunting

has interconnected factors, for example, economic, sociocultural, educational, and other factors. Socioeconomics is one of the factors of stunting that determines the amount of food available in the family, so it also determines nutritional status [3, 5]. Stunting is also associated with adverse cognitive development in children and adults, shortness of the child's school period decreased productivity, and lack of height in adults who do not reach their growth potential [5].

According to the Basic Health Research (RISKESDAS) findings [6], Indonesia has a very severe nutritional problem, with numerous cases of malnutrition among children under the age of five and those of school age. According to the survey, 30.8% of toddlers had stunting episodes, 11.5% were concise, and 19.3% were brief. The prevalence of stunting has decreased as compared to RISKESDAS 2013 results of 37.2%. Nutritional difficulties are among the most serious health issues, contributing significantly to child mortality and stunting. Stunting in children is particularly concerning because it has an impact on their physical and mental development. Stunted children have a danger of reduced intellectual ability and production, as well as an increased chance of degenerative disorders in the future. According to the World Health Organization (WHO), public health problems are considered severe if the prevalence of stunting events is 30–39% and very serious if the prevalence of stunting events is $\geq 40\%$ [7].

In Malaysia, stunting has increased. The prevalence of stunting has increased from 20.7% in 2016 [8], 16.6% in 2017, and 21.8% in 2019 [9]. According to the statistics report from the National Health Morbidity Survey (NHMS), there is a variance in the rates of stunting within Malaysia, with greater rates in Kelantan (34%), Terengganu (26.1%), and Pahang (25.7%), and the lowest rate in Kuala Lumpur (10.5%) [5]. A 40% decrease in childhood stunting is one of the many child health goals set forth by the World Health Organization (WHO) 2025, which Malaysia strives to meet. Growth is a crucial sign of a child's overall health. Healthcare practitioners need to know how to distinguish between pathological and typical childhood growth. During the fetal stage, the growth rate is at its highest at 60 cm per year. Mother and Uteroplacental health play a significant role in fetal growth. This ultimately determines the baby's birth weight and length. Nutrition plays a significant role in growth during infancy [5]. It is a case study which represents the influencing factor for stunting at various growth stages of a child in Malaysia. This analysis uses data from the 2016 National Health and Morbidity Survey (NHMS) [10]. Multiple logistic regression was used to identify the factors that contribute to malnutrition in stunted and non-stunted children [10]. This type of study combined a cross-sectional design with an analytical observational approach. The two study factors were stunting and child development. The sample size was 130 respondents. Chi-square analysis was performed to investigate the data [11].

2 Related Work

Research conducted a literature study on research related to nutritional status in children. Here are some of the results of literature studies conducted by the author and summarized in Table 1.

In this literature [20] the proposed model is divided into three stages: starting, creating linear models, and predicting outcomes using linear machine learning models.

Table 1. Summary of Literature Study

Author	Method	Accuracy
(Putra et al., 2019) [12]	Modified K-Nearest Neighbor (MKNN)	Training Data Testing with accuracy value of 97.61% with a value of k = 1
(Kaesmitan & Johannis, 2017) [13]	K-Nearest Neighbor (KNN)	100%
(Setiawan & Triayudi,, 2022) [14]	K-Nearest Neighbor (KNN)	91.79%
(Fahik et al., 2018) [15]	K-Nearest Neighbor (KNN)	85,53%
(Iriani, 2015) [16]	K-Nearest Neighbor (KNN)	Male: 78.43% Female: 87.76%
(Nugraha et al., 2017) [17]	Fuzzy K-Nearest Neighbor (FK-NN)	84,37%
(Byna & Anisa, 2018) [18]	Support Vector Machine (SVM) and Backward Elimination	SVM: 81,62% BE: 90,16%
(Wafiyah et al. 2017) [19]	Modified K-Nearest neighbor (MKNN)	K Value to Accuracy: 88.55%; Variation in the amount of training data: 92.42%; Training Data Composition for Accuracy: 87.89%; Training Data and Test Data on Accuracy: 96.35%

The polynomial regression with the pipeline model yields the best test results when the scikit-learn linear model is tested using the minimum, maximum, and average variables.

This work uses a multi-method study that employs creative statistical techniques, several literature reviews, the implementation of health and labor economic models, and multidisciplinary technical advisers. Three analytical methods were used in the comparative analyses: machine learning, Bayesian inferential statistics, and traditional frequentist statistics. The authors utilized cloud-based artificial intelligence (AI) platforms. To calculate the wage losses to private sector workers and the firm revenue losses resulting from stunting, they used labor and health, economic models. Additionally, they calculated benefit-cost ratios for nations funding interventions focused on nutrition to avert stunting [21].

3 Methods

3.1 Dataset Collection

Data for this study were collected from children in Banten, West Java Province. Banten is one of the authorized locations for stunting events. This study gathered data from ten villages that were specifically designated to deal with stunting problems. The sample

size for this research study is around 798 rows. This dataset contains many criteria or variables, including the child's identification, sensitive factors, and specific factors. Initial data is raw data that has not been processed or cleansed. This dataset consists of approximately 798 data points and 68 features. Table 2 describes the stunting dataset, including the characteristics and features which were used in this research.

Table 2. An Example of the data set

No	Column name	Data type
1	Exclusive Breastfeeding	Numerical
2	MPASI	Numerical
3	PMT	Numerical
4	PMT_Dihabiskan	Numerical
5	Child Weighed	Numerical
6	Zinc	Numerical
7	History of Deworming	Numerical
8	History of Deworming Medicines	Numerical
9	Folate Fe	Numerical
10	Breast milk counseling	Numerical
11	Vegetable Consumption	Numerical
12	Consumption of fruits	Numerical
13	YANKES	Numerical
14	How to get to Puskesmas	Numerical
15	KB	Numerical
16	Ownership of BPJS	Numerical
17	The Obligation to Educate Children	Numerical
18	Take Care of the Child	Numerical
19	Preschool	Numerical
20	Mothers Send to Preschool	Numerical

3.2 Data Pre-processing

This stage processes data, or pre-processing, by checking data whose value is empty or NaN or commonly referred to as missing value, then the row data that has a missing value will be deleted. Change the categorical data type to numeric so the computer can process it. In the pre-processing phase of trial 1, it did not use feature selection. In this second experiment, the author conducted the first experiment stage to compare or compare the KNN model obtained in the first experimental test with the Support Vector Machine (SVM) model and Logistic Regression (LR). The highest accuracy value of

the comparison results or comparisons in the experimental test can be used for the next experiment stage. The steps and stages performed in this experiment are shown in Fig. 1.

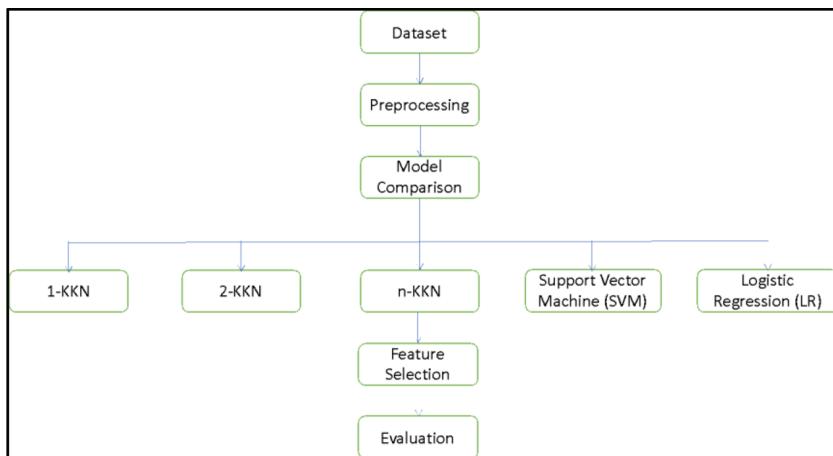


Fig. 1. Research Method Diagram

In the experiment, the researcher compared the performance results between the KNN model obtained from the results of the first experiment trial with the Support Vector Machine (SVM) model and Logistic Regression (LR). The best model from the comparison results of the four models is the one that gets the highest accuracy value in the testing data. In this trial, the researcher compared the performance results between the KNN model obtained from the results of trial 1 with the Support Vector Machine (SVM) model and Logistic Regression (LR). The model was repeated 100 times. In this second experiment stage, the researcher evaluated the model obtained from the results of trial 2 using feature selection based on precision, recall, f-1 score, and accuracy.

3.3 Model Comparison

In this trial, the researcher compared the performance results between the KNN model obtained from the results of trial 1 with the Support Vector Machine (SVM) model and Logistic Regression (LR). The model was repeated 100 times.

3.4 Model Implementation

The techniques that can be done in classifying consist of 2 (two) techniques: supervised learning and Unsupervised Learning. K-Nearest Neighbor Algorithms is one of the techniques of Supervised Learning [22]. The difference between the two techniques is that supervised learning aims to find new patterns in the data by linking pre-existing patterns from the data. In contrast, unsupervised learning data does not yet have any patterns or is commonly called not yet a label/target.

Support Vector Machine (SVM) is one of the classification algorithms using machine learning methods. It is supervised learning that predicts classes based on patterns from the training process results [23]. Based on their characteristics, SVM models can be divided into two types: Linear SVM and Non-Linear SVM. Linear SVM is a model that can separate data linearly by separating classes in the hyperplane. Meanwhile, Non-Linear SVM is a model that applies the function of the kernel trick to high-dimensional spaces [24]. The SVM Regression method is a nonparametric technique because it relies on kernel functions [25]. The following are some of the kernel types used in Non-Linear SVM and can be described in the Table 3:

Table 3. Kernel Support Vector Machine (SVM) Type

Kernel Name	Kernel Function
Linear (<i>dot</i>)	$G(x_1, x_2) = x_1' x_2$
Radial Basis Function (RBF)	$G(x_1, x_2) = \exp(- x_1 - x_2 ^2)$
Polynomial	$G(x_1, x_2) = (1 + x_1' x_2)^p$

The concept of classification with SVM is to look for the best hyperplane that serves as a separator of two data classes. SVM uses only a select few contributing data points (support vectors) to form a classification model.

K-Nearest Neighbors (KNN) is a method for classifying based on proximity or similarity of characteristics or traits of data [26]. Another explanation, K-Nearest Neighbors (KNN) is one of the classification methods that is often used. The KNN work process aims to classify new objects based on learning data that are closest to the new object [27]. The following are the steps in performing classification using the KNN algorithm:

1. Determining the value of k.
2. Calculate the distance between training data and new data inputs.
3. Sort the distance between training data and new data inputs.
4. Retrieves the value of the nearest neighbour k.
5. Get the majority value as a predicted result.

In this trial, researchers compared feature selection for each classification model. The feature selection used in this study is filter-based feature selection consisting of a chi-squared test, information gain, and Pearson correlation.

3.5 Model Evaluation

The model was evaluated using accuracy, precision, recall, f1-score, and AUC. At this stage, the model was evaluated first using the confusion matrix method. The confusion matrix is a table used to see the performance of a classification model (classifier) that has been built. The columns in each confusion matrix indicate the prediction class, while the rows in each confusion matrix indicate the actual class. In this study, there were 2 (two) classes to classify stunting events, namely stunting and non-stunting classes. Here is a table of confusion matrices for this study's 2 (two) classes.

In this trial, the researcher evaluated the model obtained from trial 2 using feature selection based on precision, recall, f-1 score, and accuracy. The best model of the feature selection comparison results is the one that gets the highest accuracy value in the testing data of the three feature selections. The number of stunting classes that are correctly classified is denoted by TP (True Positive) as summarized in Table 4. The number of stunting classes incorrectly classified into non-stunting classes is denoted by FN (False Negative). The number of non-stunting classes incorrectly classified into stunting classes is denoted by FP (False Positive). The correct number of non-stunting is classified as denoted by TN (True Negative).

Table 4. Confusion Matrix

<i>Actual Class</i>	<i>Prediction Class</i>	
	<i>Stunting</i>	<i>Not Stunting</i>
Stunting	TP	FN
Not Stunting	FP	TN

4 Results and Discussion

This section describes the experiment stages that were performed and their analysis. The first explanation is the data used, the pre-processing stage, the results of the first experiment trial, the results of the second experiment trial, and the results of the third experiment trial. The trial results include precision, recall, f1-score, and accuracy values of each test result.

The classification result of stunting events of the dataset. The classification algorithm, namely KNN, SVM, and LR. This trial 3 is to compare attribute selection and is analyzed for the best attribute selection for each classification algorithm. The dataset is divided into training and testing data with the proportion of data sharing, which is 75%: 25% and random state = 100, and the model is repeated 100 times. In this trial, the feature selection method used is filter-based feature selection consisting of a chi-squared test, information gain, and Pearson correlation. The results can be seen in the Table 5.

Table 5. Summary of Results using Chi-Squared Test

Model	Parameter	Chi-Squared Test			
		precision	recall	f1-score	accuracy
KNN26	38	0.83	0.83	0.82	0.83
LR	14	0.85	0.85	0.84	0.85

The result of the experiment using KNN and LR is summarized in Table 5 with a proportion of data sharing of 75%: 25%. In the results of this trial, it can be seen that

from the proportion of data sharing of 75%: 25% using feature selection, namely the chi-squared test on KKN26 gets a precision value = 83%, recall = 83%, f1-score = 82%, and accuracy = 83%. LR by getting precision = 85%, recall = 85%, f1-score = 84%, and accuracy = 85%.

Table 6 shows the result of trial 3 with a proportion of data sharing of 75%: 25%. In the results of this trial, it can be seen that from the proportion of data sharing of 75%: 25% using feature selection, namely Pearson correlation, the value on KNN26 gets a precision value = 83%, recall = 83%, f1-score = 82%, and accuracy = 83. While in LR get precision value = 83%, recall = 83%, f1-score = 82%, and accuracy = 83.

Table 6. Summary of Results using Chi-Squared Test

<i>Model</i>	<i>Parameter</i>	<i>Information Gain</i>			
		<i>Precision</i>	<i>recall</i>	<i>f1-score</i>	<i>accuracy</i>
KNN26	39	0.83	0.83	0.82	0.83
LR	39	0.83	0.83	0.82	0.83

Table 7 is the result of the experiment with a proportion of data sharing of 75%: 25%. In the results of this trial, it can be seen that from the proportion of data sharing of 75%: 25% using the selection feature, namely information gain, the value on KNN26 gets the value of precision = 83%, recall = 83%, f1-score = 82%, and accuracy = 83. While in LR get precision value = 83%, recall = 83%, f1-score = 82%, and accuracy = 83. It can be concluded that in trial 3, which got the highest value from the proportion of data sharing 75%: 25% by using a filter-based feature selection, namely in LR using the chi-squared test method getting precision = 85%, recall = 85%, f1-score = 84%, and accuracy = 85%. The results of this trial show that the chi-squared test method gets the best results.

Table 7. Summary of Results using Information Gain

<i>Model</i>	<i>Parameter</i>	<i>Information Gain</i>			
		<i>Precision</i>	<i>recall</i>	<i>f1-score</i>	<i>accuracy</i>
KNN26	37	0.83	0.83	0.82	0.83
LR	34	0.83	0.83	0.82	0.83

5 Conclusion

The proposed classification models the stunting in toddlers of Pandeglang Regency, West Java, Indonesia and Malaysia. The sample taken in this study consists of 798 children data. Based on the results of the author's implementation and testing, it can be concluded

that the results of the first experimental trial are to find or select the K-Nearest Neighbors (KNN) model in the stunting event classification process. The highest score, namely on KNN26 with a proportion of data sharing of 75%: 25% and gets precision value = 83%, recall = 83%, f1-score = 82%, and accuracy value = 83%.

Trial 2 to compare the results of trial 1 with other models, namely Support Vector Machine (SVM) and Logistic Regression (LR), with the proportion of data sharing which is 80%: 20%, 75%: 25%, and 70%: 30%. This study succeeded in experimenting by implementing models from the results of trial 2, namely the SVM and LR models, with a proportion of data sharing of 75%: 25%. This trial 3 is to compare attribute selection and is analyzed for the best attribute selection for each classification algorithm. The attribute selection used in trial 3 is filter-based feature selection consisting of a chi-squared test, information gain, and the Pearson correlation.

Acknowledgments. This research is part of research and community service activities by the YARSI University Team funded by the Ministry of Health of the Republic of Indonesia. The authors would like to thank the leadership and staff at the Ministry of Health, Republic of Indonesia, who has financed the implementation of this research, based on the research grant contract between LPPM Universitas YARSI and researchers for the fiscal year 2021–2022, Number: 009/INT/UM/WR III/UY/V/2021.

References

1. Rahmadhita, K.: Stunting problems and its prevention (Permasalahan Stunting dan Pencegahannya). *Jurnal Ilmiah Kesehatan Sandi Husada* **11**(1), 225–229 (2020). <https://doi.org/10.35816/jiskh.v1i1.253>
2. Dewi, I.A., Adhi, K.T.: The Impact of Protein and Zinc Consumption and a History of Infectious Diseases on the Occurrence of Short-Sleeping in Toddlers Aged 24–59 Months in the Nusa Penida III Community Health Center Working Area. In: “Pengaruh Konsumsi Protein Dan Seng Serta Riwayat Penyakit Infeksi Terhadap Kejadian Pendek Pada Anak Balita Umur 24–59 Bulan Di Wilayah Kerja Puskesmas Nusa Penida III,” *GIZI INDONESIA*, vol. 37, no. 2, (2014). <https://doi.org/10.36457/gizindo.v37i2.161>
3. Setiawan, E., Machmud, R., Masrul, M.: Factors associated with the incident of stunting in children aged 24–59 months in the working area of Andalas Health Center, East Padang District, Padang City in 2018 (“Faktor-Faktor yang Berhubungan dengan Kejadian Stunting pada Anak Usia 24–59 Bulan di Wilayah Kerja Puskesmas Andalas Kecamatan Padang Timur Kota Padang Tahun 2018”). *Jurnal Kesehatan Andalas* **7**(2), 275 (2018). <https://doi.org/10.25077/jka.v7.i2.p275-284>
4. Mavinkurve, M., Zaini, A.A., Jalaludin, M.Y.: The short child: Importance of early detection and timely referral. *Malays Family Phys.* **16**(3), 6–15 (2021). <https://doi.org/10.51866/rv1157>
5. Nurmala, Y., Anggunan, A., Febriany, T.W.: The correlation between maternal education level and family income with the incidence of stunting in children aged 6–59 months in Mataram Ilir Village, Seputih District, Surabaya, 2019 (Hubungan Hubungan Tingkat Pendidikan Ibu Dan Pendapatan Keluarga Dengan Kejadian Stunting Pada Anak Usia 6–59 Bulan Di Desa Mataram Ilir Kecamatan Seputih Surabaya Tahun 2019). *Jurnal Kebidanan Malahayati* **6**(2), 205–211 (2020). <https://doi.org/10.33024/jkm.v6i2.2409>
6. Riskesdas: ‘Hasil Utama Ringkasan Kesehatan Dasar’, Kementerian Kesehatan Badan Penelitian dan Pengembangan Kesehatan, diakses pada Juli 2020 (2018)

7. WHO: 'Childhood Stunting: Challenges and Opportunities (2013). Last accessed July 2020
8. IPH: National Health and Morbidity Survey 2016: Maternal and Child Health, vol. II: Findings. Ministry of Health Malaysia, Kuala Lumpur (2016)
9. IPH: National Health and Morbidity Survey (NHMS) 2019: Non-communicable diseases, healthcare demand, and health literacy-Technical Report Volume I. Ministry of Health Malaysia, Kuala Lumpur (2020)
10. Haron, M.Z., Jalil, R.A., Hamid, N.A.A., Omar, M.A., Abdullah, N.H.: Stunting and its associated factors among children below 5 years old on the east coast of peninsular Malaysia: evidence from the national health and morbidity survey. *Malaysian J. Med. Sci.* **30**(5), 155–168 (2023). <https://doi.org/10.21315/mjms2023.30.5.13>
11. Primasari, E.P., Sari, D.F., Syofiah, P.N., Muthia, G., Hayati, I.I.: The differences in development between stunting and normal children at the age of 3–72 months. *Med. J. Malaysia* **78**(4), 526–529 (2023)
12. Putra, M.I.P., Murdiansyah, D.T., Aditsania, A.: "Implementation of the modified k-nearest neighbor (MKNN) algorithm for classification of breast cancer (Implementasi Algoritma Modified K-Nearest Neighbor (MKNN) untuk Klasifikasi Penyakit Kanker Payudara). *e-Proc. Eng.* **6**(1) (2019)
13. Kaesmitan, Y.R., Johannis, J.A.: Classification of nutritional status of toddlers in west oesapa subdistrict using the k-nearest neighbor method (Klasifikasi Status Gizi Balita Di Kelurahan Oesapa Barat Menggunakan Metode K-Nearest Neigbor). *Multitek Indonesia* **11**(1), 42 (2017). <https://doi.org/10.24269/mtkind.v11i1.506>
14. Setiawan, R., Triayudi, A.: Classification of toddler nutritional status using web-based naïve bayes and k-nearest neighbor (Klasifikasi Status Gizi Balita Menggunakan Naïve Bayes dan K-Nearest Neighbor Berbasis Web). *Jurnal Media Informatika Budidarma* **6**(2), 777 (2022). <https://doi.org/10.30865/mib.v6i2.3566>
15. Fahik, B., Djahi, B., Rumlaklak, N.: Data Mining for Classifying Village Nutritional Status in Malacca Regency Using the K-Nearest Neighbor Method (Data Mining Untuk Klasifikasi Status Gizi Desa Di Kabupaten Malaka Menggunakan Metode K-Nearest Neighbor) *jicon* **6**(1), 1–7 (2018)
16. Iriani, Y.D.: Toddlers Nutrition Status Decision Support System Using K-Nearest Neighbor (Sistem Pendukung Keputusan Status Gizi Balita Menggunakan K-Nearest Neighbor) *Universitas. Jember* (2015)
17. Nugraha, S.D., Putri, R.G.M., Wihandika, R.C.: Application of fuzzy k-nearest neighbor (FK-NN) to determine the nutritional status of toddlers (Penerapan Fuzzy K-Nearest Neighbor (FK-NN) Dalam Menentukan Status Gizi Balita). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* **1**(9), 925–932 (2017)
18. Byna, A., Anisa, F.N.: Backward elimination to increase the accuracy of stunting events using support vector machine algorithm analysis (Backward Elimination Untuk Meningkatkan Akurasi Kejadian Stunting Dengan Analisis Algoritma Support Vector Machine). *Dinamika Kesehatan* **9**(2), 217–225 (2018)
19. Wafiyah, F., Hidayat, N., Perdana, R.S.: Implementation of the modified k-nearest neighbor (MKNN) algorithm for classification of febrile diseases (Implementasi Algoritma Modified K- Nearest Neighbor (MKNN) untuk Klasifikasi Penyakit Demam). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* **1**(10), 1210–1219 (2017)
20. Mambang, M., Marleny, F.D., Zulfadhilah, M.: Prediction of linear model on stunting prevalence with machine learning approach. *Bull. Electr. Eng. Inform.* **12**(1), 483–492 (2023). <https://doi.org/10.11591/eei.v12i1.4028>
21. Akseer, N., et al.: Economic costs of childhood stunting to the private sector in low- and middle-income countries. *eClinicalMedicine* **45**, 101320 (2022). <https://doi.org/10.1016/j.eclim.2022.101320>

22. Kurniawan, D., Saputra, A.: Application of k-nearest neighbor in admission of students with a zoning system (Penerapan K-Nearest Neighbour dalam Penerimaan Peserta Didik dengan Sistem Zonasi). *Jurnal Sistem Informasi Bisnis* **9**(2), 212 (2019). <https://doi.org/10.21456/vol9iss2pp212-219>
23. Santoso, I., Gata, W., Paryanti, A.B.: The use of feature selection in support vector machine algorithm for general election commission sentiment analysis (Penggunaan Feature Selection di Algoritma Support Vector Machine untuk Sentimen Analisis Komisi Pemilihan Umum). *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)* **3**(3), 364–3700 (2019). <https://doi.org/10.29207/resti.v3i3.1084>
24. Santoso, V.I., Virginia, G., Lukito, Y.: Application of sentiment analysis to lecturer evaluation results using the support vector machine method (Penerapan Sentiment Analysis Pada Hasil Evaluasi Dosen Dengan Metode Support Vector Machine). *Jurnal Transformatika* **14**(2), 72 (2017). <https://doi.org/10.26623/transformatika.v14i2.439>
25. Nurmasani, A., Utami, E., Fatta, H.A.: Support vector machine analysis in predicting rice commodity production (Analisis Support Vector Machine Pada Prediksi Produksi Komoditi Padi). *Jurnal Informasi Interaktif* **2**(1), 39–46 (2017)
26. Sandi, D.: Opinion classification using the k-nearest neighbor algorithm on vaccination news on Twitter (Klasifikasi Opini Dengan Menggunakan Algoritma K-Nearest Neighbor Pada Berita Vaksinasi Di Twitter). *Nuansa Informatika* **16**(1), 156–160 (2022). <https://doi.org/10.25134/nuansa.v16i1.5343>
27. Anshori, L., Putri, R.R.M., Tibyani. Implementation of the K-Nearest Neighbor Method for Recommending Study Interests (Case Study: Department of Informatics Engineering, Brawijaya University). Implementasi Metode K-Nearest Neighbor Untuk Rekomendasi Keminatan Studi (Studi Kasus: Jurusan Teknik Informatika Universitas Brawijaya) *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* **2**(7), 2745–2753 (2018)



The Effects of Data Reduction Using Rough Set Theory on Logistic Regression Model

Izzati Rahmi^{1,2(✉)}, Riswan Efendi¹, Nor Azah Samat¹, Hazmira Yozza²,
and Muhammad Wahyudi³

¹ Mathematics Department, Faculty of Science and Mathematics, Universiti Pendidikan Sultan Idris, 35900 Tanjung Malim, Perak, Malaysia
izzatirahmihg@sci.unand.ac.id, {riswanefendi, norazah}@fsmt.upsi.edu.my

² Department of Mathematics and Data Science, Faculty of Mathematics and Natural Science, Andalas University, Padang 25163, Indonesia
hazmirayozza@sci.unand.ac.id

³ Department of Information Technology Politeknik Caltex Riau, Pekanbaru, Indonesia
wahyudi@pcr.ac.id

Abstract. Logistic regression is a statistical technique for estimating the probability of a binary outcome, such as the existence or absence of a disease or a particular event. In this paper, a new integrated classification approach based on binary logistic regression analysis and rough set theory is presented. This new method is applied to two types of data sets, namely, anemia data and diabetes data. The results of the data analysis show that this method can improve the logistic regression model's performance by removing inconsistent samples using the Rough Set Theory technique. There is a tendency that the increased model performance with this hybrid method is more visible on data that has many inconsistent samples.

Keywords: logistic regression · rough set theory · hybrid method · inconsistent sample

1 Introduction

Logistic regression (LR) is one of the most commonly used statistical procedures in research. It is regarded as one of the most important statistical routines in fields such as medical statistics [1], educational research [2], risk management [3], dentistry [4], and other similar areas. LR has also been considered by many analysts to be an important procedure in predictive analytics [5].

Another method that can be used in analyzing categorical responses is rough set theory (RST). RST was developed by Pawlak et al. in the early 1970s [6] and has received wider attention in many research fields as a means of data analysis. Several studies using RST include those in the health sector [3, 7], customer behavior [8], and education [9].

Some previous studies discussed integrating and comparing logistic regression and RST for classification. Liu et al. incorporated RST and logistic regression into the integrated new classification [10]. Li et al. proposed an approach termed RST-LR in electronic commerce [11]. Vasist and Garg compare these two tools on a common dataset [12]. Kilinc used a hybrid model of RST and multinomial logistic regression by using RST on the attribute selection [13]. Nuraeni et al. [14] used RST for dimension reduction on ML algorithms (SVM, LR, KNN). Ka-Khan et al. used rough set-based feature selection for predicting diabetes using logistic regression [15]. Furthermore, Bhukya & Manchala used rough set-based feature selection for the prediction of breast cancer by using a few machines learning methods, including logistic regression [16].

This study will examine the capability of RST to LR data reduction in cleaning inconsistent samples. This study will also examine the effect of outlier elimination on the performance of the LR model before and after data reduction. The primary goal of this research is to assess the impact of the RST data reduction strategy and outlier removal on various aspects, including accuracy, sensitivity, specificity, and the F1-measure. Additionally, we will conduct a comparative analysis to determine the significance of the impact of independent variables on the dependent variable.

2 The Basic Theories and Methodology

2.1 Rough Set Theory (RST)

The Rough Set Theory (RST) is an important mathematical tool for dealing with imprecise, inconsistent, incomplete information and knowledge primarily in the context of data classification [17]. Over the past 20 years, RST and its applications have drawn a lot of attention [18]. RST can estimate the decision rules for categorizing items shown in a decision tab.

The decision table is a table with the research objects represented by each row and the conditional and decision attributes of the research represented by the columns. The decision table can express as

$$IS = (U, A) = \left(U, At = C \cup D, \{V_a | a \in At\}, I_a | a \in At \right) \quad (1)$$

where U denote a non-empty set of n objects $\{x_1, x_2, \dots, x_n\}$; At represents a finite set of non-empty collection of attributes which include a set of condition attributes C , describing the objects, and a decision attribute D , determining the class of the object; V_a is a non-empty set of values $a \in At$; $I_a : U \rightarrow V_a$ is a function that maps objects from U to a single value in V_a [19].

If each pair of items in C has the same conditional value as corresponding to D , then the decision table is said to be consistent. Conversely, the decision table is deemed inconsistent if a pair of objects in C has the same conditional value, but a different decision value in D . Building the decision table using measurement or observation generally leads to inconsistent data [11].

Supposed that IS is a decision table and A is a subset of conditional attributes, $A \subseteq C$. The indiscernibility relation $IND(A) \subseteq U \times U$ is described as

$$IND(A) = \{ (x, y) \in U \times U | \forall a \in A, I_a(x) = I_a(y) \}. \quad (2)$$

A discernibility matrix may be obtained by using the equivalent class, which is the indiscernibility relation of the set of all condition characteristics [20, 21].

A crucial concept in RST is approximation. Let $S = (U, A)$ be an information system with $B \subseteq A$, and $X \subseteq U$. A set X can be approximated by building its *B-lower* and *B-upper approximations*, which are represented as $\underline{B} X$ and $\bar{B} X$ respectively, where:

$$\underline{B} X = \{x \in U | [x]_B \subseteq X\}. \quad (3)$$

$$\bar{B} X = \{x \in U | [x]_B \cap X \neq \emptyset\}. \quad (4)$$

It is certain that the objects in $\underline{B} X$ belong to X , but the objects in $\bar{B} X$ can only be classified as potential members of X . The B-boundary region of X is represented by the set $BN_B(X) = \bar{B} X - \underline{B} X$, which includes items that cannot be conclusively categorized into X using the information in B . The definition of set approximations as previously discussed is shown graphically in Fig. 1.

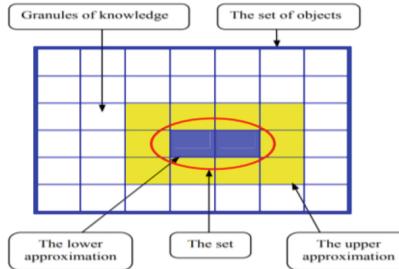


Fig. 1. All equivalency classes that the ellipse fully contains are included in the lower approximation, while those that the ellipse only partially contains are included in the upper approximation. Based on the known attributes, the figure suggests that a rough collection has all the information.

According to IS, if the values of two items differ in at least one attribute, then those objects are considered discernible. The discernibility matrix $M(x,y)$ is a matrix whose members are a collection of properties that differentiate object x from object y than can be defined as

$$M(x,y) = \{c \in C | [I_c(x) \neq I_c(y)] \wedge [I_D(x) \neq I_D(y)]\}. \quad (5)$$

for $I_D(x) = I_D(y)$, then $M(x,y) = \emptyset$. As $M(x,y) = M(y,x)$, the discernibility matrix is symmetric [22].

Every column of the discernibility matrix is used to create the boolean discernibility function. Denoted by f_{IS} , this function is defined as

$$f_{IS} = \wedge \{\vee(M(x,y)) | \forall x, y \in U, M(x,y) \neq \emptyset\}. \quad (6)$$

Using Boolean algebraic techniques, the discernibility functions at each column in the discernibility matrix are improved. Finding a comparable function that needs fewer

operations is one method to make the discernibility function simpler. The formation is known as reduct.

The final step in the RST process is creating decision rules, which are written as “if f then g” or $f \rightarrow g$. The decision attribute’s value is represented by the g component, whereas the conditional attributes’ value is represented by the f component. Examining the table of created equivalent classes yields the decision criteria from the resultant reduct [23].

2.2 Logistic Regression Model

Logistic regression is appropriate for modelling a binary variable that only has two values 0 and 1 [24]. There are numerous study questions where the conclusion can only be “yes” or “no,” for example, whether the patient has an illness or not. These ideals might be viewed as a dichotomy of any kind between “positive” and “non-positive,” or as “success” and “failure.” Logistic regression (LR), which is based on responses that have been altered by the model’s predictor, attempts to estimate the true parameter(s) of the underlying function of probability density [25].

Let x be a specific event. The formula for the probability of occurrence is $\pi(x) = \text{Pr}((Y = 1)|x)$. Non-occurrence is thus represented by $1 - \pi(x) = \text{Pr}((Y = 0)|x) = 1 - P(Y = 1)|x$. The odds ratio’s natural logarithm (\ln) is used to model the expected probabilities and it can be written as

$$\ln\left(\frac{\pi(x)}{1 - \pi(x)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k. \quad (7)$$

and

$$\pi(x) = \frac{e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k}}{1 + e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k}}. \quad (8)$$

where β_0 is the intercept, while β_1, β_2, \dots , and β_k represent the regression coefficients associated with x_1, x_2, \dots, x_k , respectively [5].

The logistic regression model of Eq. (8) shows a direct relationship between the predictor variables and the probability of $Y = 1$. Maximum likelihood estimation will be used to estimate the unknown parameters in Eq. (7). The regression coefficients demonstrate the strength of correlation between each independent variable and the outcome. Each coefficient shows the expected change in the response variable if the predictor variable changed by one the following discriminant rules are used to decide how to access a new observation:

$$y = \begin{cases} 1, & \pi(x) \geq 0.5; \\ 0, & \pi(x) < 0.5 \end{cases}. \quad (9)$$

3 Implementation Hybrid Classification Approach with LR Analysis and RST

This research proposes a new integrated classification strategy based on RST and binary logistic regression analysis. Furthermore, the Logistic Regression and Reduction Rough Set (LR3S) is the name given to the suggested model. To assess the evolution of the final

model, a comparison was done between the performance of the model generated using the original model (before to data reduction), the LR model following data reduction with the rough set, and modeling data after outliers have been removed. The following procedures will be taken in order to undertake this research:

1. Create an Information System (IS).
2. Create an Original Logistic Regression (OLR) model
3. Detect and remove outliers in OLR Model based on observations that have $|t_{studentized}\text{pearson residuall}| > 2$.
4. Create a RO2LR (Remove Outlier Original Logistic Regression) model.
5. Find and eliminate inconsistent sample bases on RST with probability ≤ 0.5
6. Create the Logistic Regression Reduction Rough Set (LR3S) Model
7. Evaluate OLR, RO2LR, and LR3S model performance.

This procedure will be applied on two data set namely Diabetes Data and Anemia Data. Diabetes Data were obtained from the UCI Machine Learning Repository, and Anemia Data were obtained from Padmi's (2018) research on factors influencing the incidence of anemia in pregnant women at Tegal Rejo Health Centre, Yogyakarta [26].

3.1 Implementation Hybrid Model on Anemia Data Set

In this part, the hybrid LR and RST models will be applied to the anemia data set. Rough set theory refers to dependent variables as decision variables and independent factors as condition variables. The decision variable is the occurrence of anemia, which consists of two categories, namely: (1) yes and (2) no. The occurrence of anemia is assumed to be influenced by five conditional attributes: gestational age (X_1), age of pregnant women (X_2), parity (X_3), employment status (X_4), chronic energy deficiency (CED) status (X_5), and education level (X_6). There are two categories in each of the X_1 , X_2 , and X_3 : (1) at-risk and (2) non-risk. X_4 consists of two categories: (1) do not work and (2) work; X_5 is divided into two categories: (1) CED and (2) not CED. Additionally, X_6 is divided into two categories: (1) do not work, and (2) work. In this instance, we observed 172 pregnant women, resulting in 172 rows and 7 columns in the information table.

Table 1 presents the results of parameter estimation and p-value for the three models used, based on the previously described stages of the analysis. According to Table 1, the OLR, RO2LR, and LR3S models have 2, 3, and 4 significant variables ($p\text{-value} \leq 0.5$), respectively. The significant variables are highlighted in Table 1. The LR model indicates that the number of significant variables increases as the data is reduced. Moreover, it is evident that each variable has a positive slope for all models. Therefore, for all three models, the relationship between the incidence of anemia and each independent variable is the same.

Table 1. The results of LR analysis for OLR, RO2LR, and LR3S models on anemia data

	OLR		RO2LR		LR3S	
	β	p-value	B	p-value	β	p-value
X ₁	0.968	0.013	0.979	0.016	2.021	0.008
X ₂	0.455	0.385	0.715	0.188	1.007	0.202
X ₃	1.315	0.114	2.323	0.049	4.502	0.002
X ₄	0.070	0.835	0.095	0.786	-0.793	0.201
X ₅	1.262	0.002	1.562	0.000	4.487	0.000
X ₆	0.248	0.556	0.665	0.141	3.003	0.000
Constant	-7.363	0.000	-11.238	0.000	-24.767	0.000

Next, Table 2 shows a comparison of the accuracy, precision, sensitivity, and F1-score values of the three models' performances.

Table 2. Performances OLR, RO2LR and LR3S Models on anemia data

Model	Accuracy	Precision	Sensitivity	F1-Score
OLR	66.3	57.0	70	67.5
RO2LR	69.8	65.1	72.7	68.7
LR3S	94.4	94.4	92.7	93.5

Table 2 demonstrates that the LR model performs better when outlier or inconsistent samples are eliminated from the data. Eliminating inconsistent samples (LR3S model) significantly improves model performance; however, eliminating outliers only slightly improves the RO2LR model. Stated differently, in this case, when it comes to enhancing the performance of the LR model, the LR3S model represents the most noteworthy upgrade choice.

3.2 Implementation Hybrid Model on Diabetes Data Set

This section presents the diabetes data set classification using the hybrid LR and RST model. This information is drawn from the diabetes data in the UC Learning Repository, with a sample size of 1000. The decision variable is the incidence of diabetes, which consists of two categories, namely: (1) yes and (2) no. Ten conditional attribute are thought to influence the incidence of diabetes: high blood pressure (Z_1), high cholesterol (Z_2), high cholesterol check(Z_3), body mass index (Z_4), consume Fruit 1 or more times per day (Z_5), consume vegetables 1 or more times per day (Z_6). Heavy drinkers (Z_7), general health (Z_8), have serious difficulty walking or climbing stairs (Z_9) and sex (Z_{10}). There are two categories in each of the $Z_1, Z_2, Z_3, Z_5, Z_6, Z_7, Z_9$: (1) No and (2) Yes. Z_4

consist of four categories: (1) underweight, (2) normal, (3) overweight and (4) obesity. Z_5 is divided in five categories: (1) excellent, (2) very good, (3) good, (4) fair and (5) poor. Additionally, Z_{10} is divided into two categories: (1) female and (2) male.

Table 3 shows the results of parameter estimation and p-value for the three models used, based on the previously described stages of the analysis. According to Table 3, the OLR and LR3S models have the same number of significant variables, and different with RO2LR but all off the three models have the same relationship between each independent variable and the incidence of diabetes. The significant variables are highlight in Table 3. Therefore, data reduction does not significantly change the calculated LR model parameters. The small percentage of outliers and inconsistent samples in this data may contribute to this condition, meaning that removing them through data reduction does not significantly alter the LR model.

Table 3. The results of LR analysis for OLR, RO2LR, and LR3S models on diabetes data

	OLR		RO2LR		LR3S	
	β	p-value	β	p-value	β	p-value
$Z_{1(1)}$	-1.064	.000	-1.051	.000	-1.102	.000
$Z_{2(1)}$	-.893	.000	-.881	.000	-.933	.000
$Z_{3(1)}$	-1.725	.107	-1.736	.104	-1.683	.116
Z_4		.000		.000		.000
$Z_{4(1)}$	-2.982	.010	-2.998	.009	-3.032	.009
$Z_{4(2)}$	-1.254	.000	-1.256	.000	-1.254	.000
$Z_{4(3)}$	-.399	.027	-.422	.020	-.420	.024
$Z_{5(1)}$	-.038	.817	-.032	.844	-.032	.848
$Z_{6(1)}$	-.047	.818	-.050	.808	-.053	.799
$Z_{7(1)}$.845	.043	.734	.085	.845	.045
Z_8		.000		.000		.000
$Z_{8(1)}$	-3.025	.000	-3.018	.000	-3.073	.000
$Z_{8(2)}$	-2.075	.000	-2.094	.000	-2.107	.000
$Z_{8(3)}$	-1.380	.001	-1.362	.002	-1.362	.002
$Z_{8(4)}$	-.766	.081	-.731	.096	-.733	.097
$Z_{9(1)}$	-.304	.149	-.286	.177	-.322	.131
$Z_{10(1)}$	-.076	.635	-.085	.594	-.075	.647
Constant	2.216	.000	2.300	.000	2.257	.000

Next, Table 4 displays the LR model performances for the diabetes data based on the confusion matrix.

Table 4 demonstrates that removing outlier or inconsistent samples from the data improves the performance of the LR model. In contrast to the case of data anemia, the

Table 4. Performances LR Models Logistic Regression on diabetes data

Model	Accuracy	Precision	Sensitivity	F1-Score
OLR	74.9	75.4	77.6	76.5
RO2LR	75.0	75.7	77.6	76.7
LR3S	76.0	76.1	78.8	77.4

performance improvement of the RO2LR and LR3S models on diabetes data is nearly same, and there is slighty improvement in the model's performance. However, the LR3S is still the model with the best performance.

3.3 Discussion

In this study, there are two data sets used to implement data reduction integration with RST and LR models. These two data are assumed to have different characters. Anemia data that still contains quite a lot of inconsistent data. Meanwhile, diabetes data is data that contains little inconsistent data. Based on the percentages of sample inconsistencies and outliers, Table 5 compares the features of the two sets of data.

Table 5. Distribution of inconsistent samples and outliers in anemia and diabetes data

Data	n	Inconsistent sample (%)	Inconsistent sample discarded (%)	Outlier (%)
Anemia	172	34.9	27.9	1.7
Diabetes	1000	3.4	2.8	1.2

Based on Tables 2, 4 and 5, there is a tendency that the more inconsistent samples, the greater the boost in the LR model's performance that results from eliminating the inconsistent samples. Meanwhile, based on outliers, both data contain a very small percentage of outliers, so that in both data sets, eliminating outliers only slightly improves the performance of the LR model.

Furthermore, the LR3S model has a considerably higher number of significant variables than the other two models based on the model coefficient value obtained on anemia data, but in diabetes data the number of significant variables in LR3S is more than RO2LR but the same as in OLR model.

However, for the two sets of data analyzed, the regression coefficient of the logistic regression model remained unchanged for all three models employed, suggesting that the direction of the independent variable's influence on the incidence of anemia or diabetes remained same.

4 Conclusion

This paper presents a new method for enhancing the performance of the logistic regression (LR) model using the Rough Set Theory (RST) technique to eliminate inconsistent data. This hybrid model applies to two types of data sets: data with a high number of inconsistent samples, like anemia data, and data with a low number of inconsistent samples, like diabetes data. The Logistic Regression Reduction Rough Set (LR3S) is the name given to the hybrid model. This model will be compared with two other models that have been often used in previous studies: the OLR (Original Logistic Regression) model and the RO2LR (Removing Outlier Original Logistic Regression) model, using a variety of model performance criteria. The growing LR model performance on anemia data is substantially higher than on diabetes data due to the tendencies that many inconsistent samples cause. But LR3S performs better as a model than OLR and RO2LR in both data sets. Further simulation experiments are necessary to validate the results of this preliminary study.

Acknowledgments. The authors would like to express gratitude to Andalas University, for supporting this research.

References

1. Boateng, E.Y., Abaye, D.A.: A review of the logistic regression model with emphasis on medical research. *J. Data Anal. Inform. Process.* **7**, 190–207 (2019)
2. Niu, L.: A Review of The Adoption of Logistic Regression In Educational Research: Common Issues, Implications, and Suggestions. *Educational Review. Advance online publication* (2018)
3. Velu, A.: Application of logistic regression models in risk management. *Int. J. Innov. Eng. Res. Technol.* **8**, 251–260 (2021)
4. Srimaneekarn, N., Hayter, A., Liu, W., Tantipoj, C.: Binary response analysis using logistic regression in dentistry. *Int. J. Dent.* **2022**, 1–7 (2022)
5. Hilbe, J.M.: Practical Guide to Logistic Regression. Chapman and Hall/CRC (2016)
6. Pawlak, Z.: Rough sets. *Int. J. Comput. Inform. Sci.* **11**(5), 341–356 (1982)
7. Burney, S.M.A., Abbas, Z.: Applications of rough sets in health sciences and disease diagnosis. *Recent Res. Appl. Comput. Sci.* **8**, 153–161 (2015)
8. Yekkala, I., Dixit, S.: Prediction of heart disease using random forest and rough set based feature selection. *Int. Jo. Big Data Anal. Healthcare* **3**, 1–12 (2018). <https://doi.org/10.4018/ijbdah.2018010101>
9. Forghani, E., Sheikh, R., Sana, S.S.: Extraction of rules related to marketing mix on customers' buying behavior using rough set theory and fuzzy 2-tuple approach. *Int. J. Manag. Sci. Eng. Manag.* **18**, 16–25 (2023)
10. Liu, D., Li, T., Liang, D.: Incorporating logistic regression to decision-theoretic rough sets for classifications. *Int. J. Approximate Reason.* **55**, 197–210 (2014)
11. Li, X.: Attribute Selection Methods in Rough Set Theory. San Jose State University, San Jose, CA, USA (2014)
12. Vashit, K., Garg, M.L.: Comparing and contrasting rough set with logistic regression for a dataset. *Int. J. Rough Sets Data Anal.* **1**, 81–98 (2014)

13. Kan-Kilinç, B., Yazırı, Y.: Performance of the hybrid approach using three machine learning algorithms. *Pak. J. Stat. Oper. Res.* **16**, 217–224 (2020)
14. Nuraeni, R., Surono, S.: Rough set theory for dimension reduction on machine learning algorithm. *Jurnal Fourier* **10**, 29–37 (2021)
15. Kaka-Khan, K.M., Mahmud, H., Ali, A.A.: Rough set-based feature selection for predicting diabetes using logistic regression with stochastic gradient decent algorithm. *UHD J. Sci. Technol.* 85–93 (2022)
16. Bhukya, H., Manchala, S.: Rough Set based Feature Selection for Prediction of Breast Cancer. Research Square (2022)
17. Pawlak, Z.: Rough set theory and its applications to data analysis. *Cybern. Syst.* **29**, 661–688 (1998)
18. Cao, H.: The Utilization of rough set theory and data reduction based on artificial intelligence in recommendation system. *Soft. Comput.* **25**(3), 2153–2164 (2020)
19. Komorowski, J., Polkowski, L., Skowron, A.: Rough sets: a tutorial. In: *Rough Fuzzy Hybridization: a new trend in decision-making*. In *Rough Fuzzy Hybridization: A New Trend in Decision-Making*, pp. 3–98. Springer-Verlag, Singapore (1999)
20. Abbas, S.M., Alam, K.A., Shamshirband, S.: A soft-rough set based approach for handling contextual sparsity in context-aware video recommender systems. *Mathematics* **7**(8), 740 (2019)
21. Yao, Y., Zhao, Y.: Discernibility matrix simplification for constructing attribute reducts. *Inform. Sci.* **179**(7), 867–882 (2009). <https://doi.org/10.1016/j.ins.2008.11.020>
22. Skowron A., Rauszer, C.: The discernibility matrices and functions in information systems. In: *Intelligent Decision Support*. Springer Netherlands, Dordrecht, pp. 331–362 (1992). https://doi.org/10.1007/978-94-015-7975-9_21
23. Andersson, R.: Implementation of a Rough Knowledge Base System Supporting Quantitative Measures. Master's thesis, Linköping University (2004)
24. Hosmer, D.W., Lemeshow, S.: *Applied Logistic Regression*, Second. John Wiley and Sons, New York (2000)
25. Pal, A.: Logistic regression: a simple primer. *Cancer Res. Stat. Treat.* **4**, 551–554 (2021)
26. Padmi, D.R.K.N., Setyawati, N.: Faktor-Faktor yang Mempengaruhi Kejadian Anemia pada Ibu Hamil di Puskesmas Tegalrejo Tahun 2017, Skripsi, Politeknik Kesehatan Kementerian Kesehatan, Kota Yogyakarta (2018)



Robust Heart Disease Prognosis: Integrating Extended Isolation Forest Outlier Detection with Advanced Prediction Models

Irfan Javid^{1,2(✉)}, Norlida Hassan¹, Rozaida Ghazali¹,
Yana Mazwin Mohmad Hassim¹, Tuba Batool¹, Noor Aida Husaini¹,
and Syed Irteza Hussain Jafri¹

¹ Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia

{norlida, rozaida, yana}@uthm.edu.my, irtezasyed@upr.edu.pk

² Department of Computer Science and IT, University of Poonch Rawalakot, Rawalakot, Pakistan

irfanjavid@upr.edu.pk

Abstract. Heart disease can be prevented with an accurate prognosis, but it can also be catastrophic if the forecast is erroneous. This paper presents an innovative approach for the prediction of heart disease by using machine learning and deep learning techniques with Extended Isolation Forest outlier detection. The proposed system leverages a dataset comprising various clinical features and diagnostic parameters associated with heart disease. Initially, the extended isolation forest algorithm is employed to identify outliers and mitigate their influence on subsequent analyses. This preprocessing step enhances the overall robustness of the prediction system. Next, machine learning and deep learning models are utilized for the prediction of heart disease. Multiple classification algorithms, including logistic regression, support vector machine, random forest and light gradient boosting are trained on the preprocessed dataset to identify patterns and relationships between the features and disease outcomes. Furthermore, a deep learning model, such as gated recurrent unit is employed to extract intricate patterns from the input data and capture temporal dependencies. The accuracy and confusion matrix is used to validate several promising outcomes. The integration of outlier detection techniques further enhances the system's performance by minimizing the impact of erroneous data, and also the data has been standardized to achieve optimal results. The accuracy of 93.4% was achieved using a deep learning method.

Keywords: Machine Learning · Deep Learning · Heart Disease · Extended Isolation Forest · Confusion Matrix

1 Introduction

Heart disease is a serious and potentially fatal condition that should be treated with utmost seriousness. According to a publication by Harvard Health, there is a higher likelihood of men developing heart disease compared to women [1]. Researchers have

discovered that men are having approximately twice chance as women to experience a heart attack during their lifetime. This increased risk remains even after considering established factors such as diabetes, high blood pressure, body mass index, physical activity, and high cholesterol. The dataset being studied by researchers is of significant importance as it contains crucial information, including records dating back to 1998. It is widely recognized as a benchmark dataset for predicting heart disease. Created in 1988, the dataset comprises four databases: Cleveland, Hungary, Switzerland, and Long Beach V, and has yielded promising findings in research [12]. A wide range of disorders are encompassed by heart disease that influence the functioning of the heart. With the passage of time, an increasing number of research findings and medical data are becoming accessible. Patient information can be accessed through multiple open sources and studies can be conducted to explore the potential of utilizing computer technologies to accurately diagnose individuals and pinpoint this condition before it reaches a severe stage. The machine learning and artificial intelligence role in the healthcare system is widely recognized. These technologies can be leveraged to diagnose the condition, classify data, and predict outcomes. For these purposes, a range of machine learning and deep learning models can be utilized [2, 11, 13].

Machine learning algorithms have the capability to efficiently perform comprehensive analysis of genetic data. By leveraging these algorithms, patient history can be extensively studied and modified to improve predictive capabilities. Additionally, algorithms can be trained to enhance the accuracy of outbreak forecasts, enabling better preparedness and response strategies [3, 14]. Several researchers have undertaken various studies with the purpose of investigating the classification and prediction of heart disease diagnosis using various machine learning techniques. Table 1 summarizes the associated work, which includes numerous research, algorithms/methodologies, and performance measures.

Table 1. Summary of Related works for Heart Disease Prediction

Study	Algorithm/Methodology	Performance/Accuracy
Melillo et al. [4]	CART (Classification and Regression Tree)	Sensitivity: 93.3% Specificity: 63.5%
Rahhal et al. [5]	ECG strategy with deep neural networks	Performance improvement
Parthiban and Srivatsa [6]	SVM approaches	Accuracy: 94.60%
Dun et al. [7]	Machine learning and deep learning algorithms	Neural Networks: 78.3%
Singh et al. [8]	Extended discriminant analysis	Nonlinear attributes extraction

The use of feature extraction and feature selection techniques has shown improved outcomes in both prediction and classification tasks [10]. Dun et al. [7] conducted a study where they utilized a range of machine learning and deep learning algorithms for

the purpose of diagnosing cardiovascular disease. Hyperparameter tuning was applied to enhance the accuracy of the outcomes.

The paper is divided into four distinct sections. Section 1 comprises the introduction, Sect. 2 details the methodology, Sect. 3 encompasses the discussion and results analysis, and finally, Sect. 4 covers the conclusion and future scope of the study.

2 Methodology

2.1 Summary of Dataset

This study utilized the Cleveland heart disease dataset, an online dataset, available on the UCI machine learning repository [9]. Despite containing a total of 76 features, including the predicted attribute, previous studies have solely focused on a subset of 14 features. The “target” attribute is used to indicate the existence or nonexistence of cardiac disease, with a value of 0 representing no disease and 1 representing the presence of disease.

2.2 Data Preprocessing

The dataset doesn't contain any missing values, although it does have a significant number of outliers that needed to be addressed. Additionally, the dataset was not evenly distributed. Two techniques were applied to tackle these challenges. The first technique, which involved directly feeding the data into machine learning algorithms without outlier removal or feature selection, yielded unfavorable results. However, promising findings were obtained by first normalizing the dataset to address overfitting and then employing the Extended Isolation Forest (EIF) algorithm for outlier detection. Various visualization methods were utilized to examine the skewness, identify outliers, and assess data distribution. These preprocessing approaches are crucial when submitting the data for prediction or classification tasks.

2.2.1 Data Distribution

When it comes to predicting or classifying an issue, data distribution is crucial. Heart disease occurred in 56.26% of cases in the dataset, whereas no heart disease occurred in 43.74% of cases. To prevent overfitting the model, it is essential to balance the dataset. By achieving a balanced dataset, the model will be better equipped to detect patterns related to heart disease.

2.2.2 Data Skewness

Multiple distribution sets are examined to assess the input parameters and identify any skewness in the data, allowing for better interpretation. Various plots are generated to provide a comprehensive overview of the data. This includes analyzing distributions related to age and sex, resting blood pressure and chest pain, abstaining blood levels and cholesterol, ECG resting electrode and maximum heart rate distributions, presence of exercise-induced angina and ST depression induced by exercise, slope and number of major vessels (ca), as well as thalassemia type and the target variable.

2.2.3 Gaussian Distribution

Figures 1 and 2 illustrate the attributes that have a significant impact on heart disease and those that are deemed irrelevant in relation to heart disease. The essential elements here exhibit a distinct variance, indicating that they are significant. Based on all these numerical facts, it could be concluded that a Gaussian distribution plays a significant role in heart disease, while the absence of a Gaussian distribution does not have a substantial impact on heart disease.

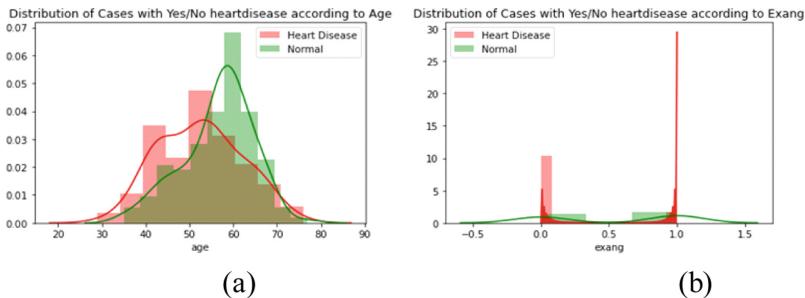


Fig. 1. Important Attributes for Heart disease prediction

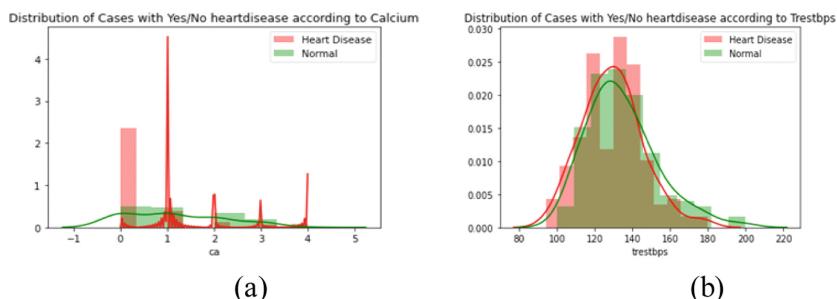


Fig. 2. Attributes not significant for heart disease prediction.

2.2.4 Algorithm for Feature Selection

For feature selection, the Lasso algorithm, a type of embedded technique, is utilized to identify and select the most significant features. This algorithm surpasses filter approaches in terms of predictive accuracy and effectively generates high-quality feature subsets for the selected algorithm. To make the final feature selection, the model available in the sci-kit-learn library, which is specifically designed for feature selection, is employed.

2.2.5 Examining Dataset for Duplicate Values

It is essential to carefully eliminate duplicates to ensure the model's generalization remains intact. Mishandling duplicates can lead to their presence in both datasets i.e. the training datasets and the testing datasets, potentially affecting the performance of the model. Figure 3 depicts the duplicate values observed.

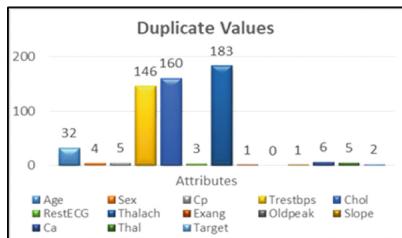


Fig. 3. Duplicate values

2.3 Machine Learning Techniques

The recommended technique was applied to the dataset, which included initially a thorough analysis of the data, followed by the application of various machine learning algorithms, including Logistic Regression, followed by a tree-based approach like Decision Tree Classifier, and a most common technique Random Forest Classifier. Support Vector Machine was also utilized to examine and handle the high dimensionality of the data. The Light GBM classifier is another strategy that uses a combination of ensemble and Decision Tree methods. The proposed model is illustrated in Fig. 4.

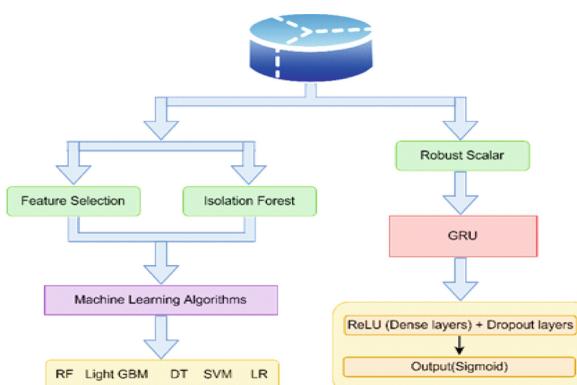


Fig. 4. Proposed Methodology for heart disease prediction

2.4 Deep Learning Algorithm

In this distinct study, a sequential deep learning model known as gated recurrent unit with its deeply connected dense layer is used. To prevent over-fitting, and flattening, dropout are also used. The results of machine learning and deep learning approaches are compared based on accuracy, learning, and computational time as shown in the figures discussed further in the result section.

2.5 Evaluation Parameters

The evaluation procedure incorporates various metrics such as the confusion matrix, accuracy score, recall, precision, sensitivity, and F1 score. The confusion matrix contains true positive and true negative values. It is divided into four sections: the first section represents true positives (TP), where the values are correctly labeled as true; the second section represents false positives (FP), indicating values that are reported as true but are actually false; the third section represents false negatives (FN), signifying cases where a correct outcome has been classified as negative; and the fourth section represents true negatives (TN), indicating values that are correctly identified as negative.

For model's performance assessment, an accuracy value is employed. This score is calculated by summing the true positive and true negative values and dividing it by the sum of true positive, true negative, false positive, and false negative. The formula for accuracy can be expressed as follows:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Also, there is a specificity, that could be calculated by taking the fraction of real negative instances labeled as negative; therefore, it is a measure of how successfully a classifier recognizes negative situations. It is often referred to as the genuine negative rate. The formula is as follows:

$$\text{specificity} = \frac{TN}{TN + FP}$$

After specificity, there is sensitivity, which refers to the fraction of actual positive instances that were anticipated to be positive (or true positive). The recall is an additional synonym for sensitivity. In other words, an unhealthy individual was anticipated to be sick. The formula of the sensitivity is given as follows:

$$\text{sensitivity} = \frac{TP}{TP + FN}$$

3 Results Evaluation

The study employed three techniques to analyze the data. Firstly, different machine learning techniques were utilized to perform classification. Secondly, the deep learning algorithm GRU was applied to investigate its impact on the data. In the first technique,

the original unprocesses dataset has been utilized directly for the classification purpose. In the subsequent technique, the dataset went through feature selection method and outlier detection. The results obtained from this approach were highly promising. Lastly, the dataset has been normalized, considering both outliers and feature selection. The outcomes achieved using this approach were significantly better than traditional methods, and when results were compared to other accuracies of data analysis, our outcomes were highly compelling.

3.1 Implementing the First Strategy, Which Involves Neither Feature Selection nor Outlier Detection

Initially, the dataset lacks normalization, lacks an equal distribution of the target class, and exhibits numerous negative values when a correlation heatmap is analyzed, as depicted in Fig. 5. Even after applying feature selection techniques, outliers are still present, as indicated by Fig. 6.

Random Forest technique achieved an accuracy of 81.2%, Logistic Regression technique achieved an accuracy of 80.32%, Support Vector Machine technique achieved an accuracy of 83.09%, KNN technique achieved an accuracy of 79.38%, Decision Tree technique achieved an accuracy of 72.0%, and Light GBM technique achieved an accuracy of 84.09% when using the first approach. Light GBM has the maximum accuracy value, which has been obtained by employing grid search and cross-validation to discover the optimim parameters, or conducting hyperparameter optimization in other words. Then, using the sequential model technique, the deep learning algorithm GRU is deployed after machine learning. The activation function utilized in the model is ReLU, and in the output layer, the sigmoid activation function has been utilized which is a single class prediction problem, with loss as binary cross-entropy and gradient descent optimizer as Adam. 85.7% accuracy was attained.

3.2 Implementing the 2nd Strategy: Feature Selection Without Outlier Detection

After performing feature selection and scaling the data, the robust standard scalar is applied due to the presence of outliers in the dataset. This method is specifically employed when dealing with datasets that contain outliers. Among the different classification algorithms used, SVM achieved an accuracy of 86.2%, Logistic Regression sion has achieved an accuracy value of 83.7%, KNN has achieved an acc uracy value of 77.47%, Random Forest has achieved an accuracy value of 82.04%, Decision Tree has achieved an accuracy value of 74.13%, and Light GBM has achieved an accuracy value of 84.16%. Notably, SVM emerges as the clear winner with an accuracy value of 86.2% and F1 score of 86.5%.

Subsequently, employing identical settings as before, the deep learning algorithm GRU is utilized, resulting in an accuracy of 86.8% and an evaluation accuracy of 89.7%. This demonstrates a notable improvement compared to the initial strategy.

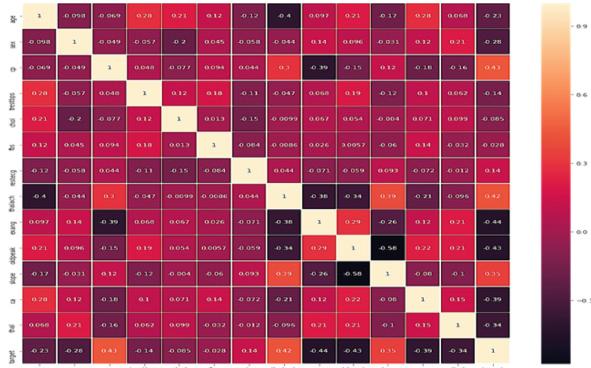


Fig. 5. Correlation Heatmap

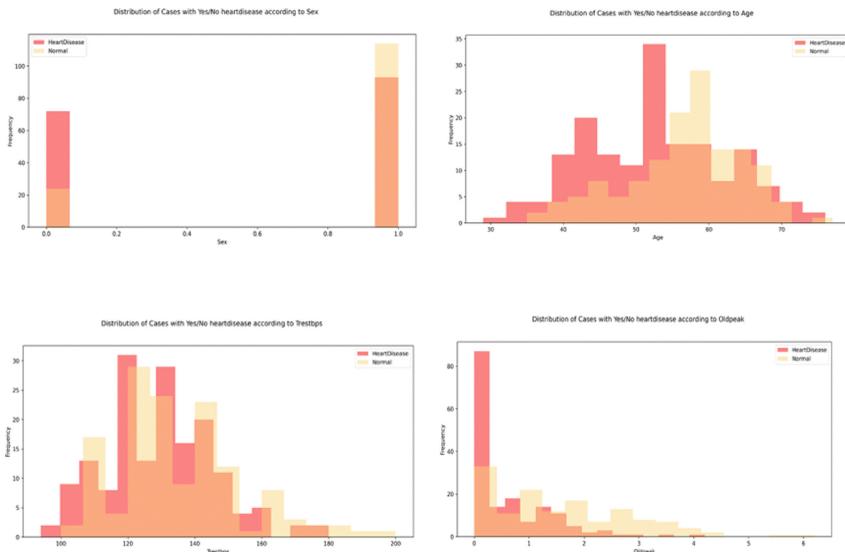


Fig. 6. Attribute Selection based on heatmap

3.3 Employing the 3rd Strategy (Feature Selection and Detection of Outliers)

In this method, the dataset undergoes normalization, feature selection, and outlier handling using the iForest algorithm. Among the machine learning classifiers, Random Forest achieves the highest accuracy of 86.5%, followed by Logistic Regression (84.53%), KNN (82.90%), Support Vector Machine (85.47%), Decision Tree (84.55%), and Light GBM (85.6%). Random Forest also exhibits the highest precision of 81.7% and a specificity of 80%. However, in the third technique involving deep learning, an accuracy of 93.4% is achieved. Among the machine learning models, Random Forest obtains the highest accuracy of 86.51%, while deep learning achieves a maximum accuracy of 93.4%. Thus, the deep learning algorithm, specifically the GRU model, demonstrates a

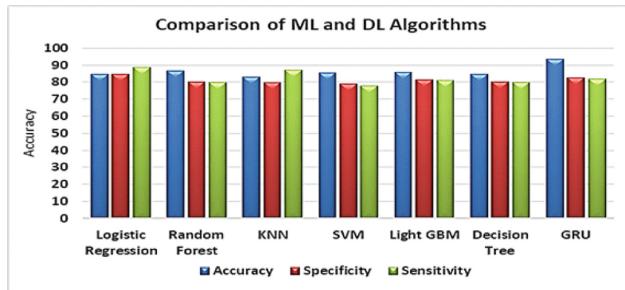


Fig. 7. Comparison of ML and DL Algorithms

higher accuracy of 93.4% as compared to the machine learning models. Consequently, the deep learning algorithm proves to be more accurate and promising. Figure 7 displays the comparative results of various ML and DL classifiers.

4 Conclusion

Within this study, we propose three strategies to conduct comparative analysis, leading to promising outcomes. Our findings indicate that deep learning models outperformed other approaches. It aligns with the notion previously suggested by many scholars, advocating for the utilization of machine learning in scenarios involving small datasets, which is demonstrated in this research. Comparative methodologies, including confusion matrix, precision, specificity, sensitivity, and F1 score, were employed to evaluate the performance. Notably, when data pre-processing was implemented, the Random Forest classifier exhibited superior performance among the machine learning techniques, utilizing the 13 attributes present in the dataset.

Furthermore, there was a reduction in calculation time, which is advantageous for model deployment. Another crucial finding was that normalizing the dataset is necessary to avoid overfitting the training model. This is particularly important for achieving accurate results when evaluating the model on real-world data that significantly differs from the training dataset. Additionally, it was determined that statistical analysis plays a vital role in dataset analysis, specifically in assessing the suitability of a Gaussian distribution. Furthermore, the detection of outliers was identified as a crucial step, which was effectively handled using the Extended Isolation Forest approach.

In the future, there is potential to increase the size of the dataset and explore the utilization of deep learning techniques along with various optimizations to yield more promising results. By employing machine learning algorithms and incorporating a range of optimization strategies, further improvements can be achieved in evaluating the findings. The data can be normalized using different approaches, and the results can be compared to enhance the analysis.

Acknowledgments. This research was supported by Universiti Tun Hussein Onn Malaysia (UTHM).

References

1. Harvard Medical School. *roughout life, heart attacks are twice more common in men than women (2020). <https://www.health.harvard.edu/heart-health/throughout-life-heartattacks> are twice as common in men than women
2. Wahid, F., Ismail, L.H., Ghazali, R., Aamir, M.: An efficient artificial intelligence hybrid approach for energy management in intelligent buildings. *KSII Trans. Internet Inf. Syst.* **13**(12), 5904–5927 (2019)
3. Shalev-Shwartz, S., Ben-David, S.: *Understanding machine learning, From Theory to Algorithms*, Cambridge University Press, Cambridge, UK (2020)
4. Melillo, P., De Luca, N., Bracale, M., Pecchia, L.: Classification tree for risk assessment in patients suffering from congestive heart failure via long-term heart rate variability. *IEEE J. Biomed. Health Inform.* **17**(3), 727–733 (2013)
5. Al Rahhal, M.M., Bazi, Y., AlHichri, H., Alajlan, N., Melgani, F., Yager, R.R.: Deep learning approach for active classification of electrocardiogram signals. *Inform. Sci.* **345**, 340–354 (2016). <https://doi.org/10.1016/j.ins.2016.01.082>
6. Parthiban, G., Srivatsa, S.K.: Applying machine learning methods in diagnosing heart disease for diabetic patients. *Int. J. Appl. Inform. Syst.* **3**(7), 25–30 (2012)
7. Dun, B., Wang, E., Majumder, S.: Heart disease diagnosis on medical data using ensemble learning (2016)
8. Singh, R.S., Saini, B.S., Sunkaria, R.K.: Detection of coronary artery disease by reduced features and extreme learning machine. *Med. Pharm. Rep.* **91**(2), 166–175 (2018). <https://doi.org/10.15386/cjmed-882>
9. UCI Machine Learning Repository. <https://archive.ics.uci.edu/ml/machine-learning-databases/heartdisease/heart-disease.names>
10. Javid, I., Alsaedi, A.K.Z., Ghazali, R., Hassim, Y.M.M., Zulqarnain, M.: Optimally organized GRU-deep learning model with Chi 2 feature selection for heart disease prediction. *J. Intell. Fuzzy Syst.* **42**(4), 4083–4094 (2022)
11. Javid, I., Ghazali, R., Zulqarnain, M., Hassan, N.: Data pre-processing for cardiovascular disease classification: a systematic literature review. *J. Intell. Fuzzy Syst.* **44**(1), 1525–1545 (2023)
12. Javid, I., Ghazali, R., Zulqarnain, M., Husaini, N.A.: Deep learning GRU model and random forest for screening out key attributes of cardiovascular disease. In: Ghazali, R., Nawi, N.M., Deris, M.M., Abawajy, J.H., Arbaiy, N. (eds.) *Recent Advances in Soft Computing and Data Mining: Proceedings of the Fifth International Conference on Soft Computing and Data Mining (SCDM 2022)*, May 30–31, 2022, pp. 160–170. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-00828-3_16
13. Hassim, M., Mazwin, Y., Ghazali, R.: Using artificial bee colony to improve functional link neural network training. *Appl. Mech. Mater.* **263**, 2102–2108 (2013)
14. Ghazali, R., Al-Jumeily, D.: Application of pi-sigma neural networks and ridge polynomial neural networks to financial time series prediction. In: Zhang, M. (ed.) *Artificial Higher Order Neural Networks for Economics and Business*, pp. 271–293. IGI Global (2009). <https://doi.org/10.4018/978-1-59904-897-0.ch012>



Overlapping Granular Clustering: Application in Fuzzy Rule-Based Classification

Muhammad Zaiyad Muda¹(✉) and George Panoutsos²

¹ College of Engineering, Universiti Teknologi MARA, Selangor, Malaysia
zaiyad@uitm.edu.my

² Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, UK

Abstract. A clustering technique often aims to create a number of disjoint clusters or granules, in which an element or instance is only permitted to belong to one cluster. However, the majority of real-world data sets have information that overlaps, causing specific data objects or patterns to belong to multiple clusters. For instance, an individual may concurrently be a member of more than one social group, such as a family group and a friend group. Therefore, the purpose of this study is to use a parameter called R-value to allow and provide parametric control for cluster overlaps. In this research, it is demonstrated that the inclusion of R-value in the Granular Clustering (GrC) enables GrC to control the amount of overlapping between clusters. Datasets from the UCI Machine Learning Repository are used to illustrate the new GrC algorithm with overlapping measure. Results reveal that the GrC with overlapping measure surpasses the traditional GrC in terms of classification accuracy, highlighting the possible application of the overlapping GrC for creating Fuzzy Logic rule bases.

Keywords: Granular Clustering · Iterative Data Granulation · Fuzzy Logic · overlapping

1 Introduction

Clustering involves organising a group of things into classes, with related objects belonging to the same cluster and dissimilar objects belonging to different clusters. Most clustering algorithms that employ hard clustering techniques assign each instance to a single cluster. Nonetheless, the structure provided by fuzzy clustering methods allows each object to contribute to the definition of each cluster [1]. Clustering algorithms can be classified in a number of ways, one of which is based on the types of clusters they produce [2]:

- Disjoint, where an element only belongs to one cluster.
- Fuzzy, when one element is included in all clusters but only to a certain extent.
- Overlapping, when a component may be a part of multiple clusters.

The majority of clustering methods produce exclusive clusters, meaning that each sample can only belong to one cluster [3]. Traditional methods like the k-means may be effective at accurately splitting instances into clusters, especially when the boundaries are defined and the data is free of outliers [4]. However, most real-world data sets include overlapping information, thus some data objects or patterns might belong to several clusters [5]. In a simple case where a person may belong to more than one group, or organisation in the community, the use of hard or disjoint clustering might not be appropriate, therefore, the need for an overlapping clustering becomes inevitable.

Overlapping clusters can happen for a variety of reasons, including noisy data, attributes that don't fully capture the information needed to separate the clusters, or overlap that naturally arises from the processes that produce the data [6]. Numerous studies employing the overlapping clustering method have been carried out to tackle this problem. Recent methods that have been proposed are theoretical rather than heuristic in nature, and as a result, they take overlaps in their ideal criteria into consideration [7].

In Granular Computing, the development of a new information granule results in a lack of distinguishability due to the fact that the overlapping is not taken into account in the compatibility metric. This will also affect the distinguishability in the Fuzzy Logic (FL) rules, hence deteriorating the predictive performance of the FL models. The work on overlapping in GrC is, however, scarce. In order to have an additional parameter that enables the compatibility metric to assess and quantify the overlapping behaviour, Solis et al. [8] devised the neutrosophic technique. This strategy convinces the compatibility search to cease looking for potential granules that can result in granular overlapping, which would lessen model transparency and undermine the consistency of the rules. This method, however, only aims to attenuate the overlapping behaviour, even though it is recognised that some overlapping is necessary in building FL models.

In this paper, GrC uses the R-value to describe the overlapping among granules and a novel compatibility equation that integrates the quantification of granules overlapping is proposed. The suggested method might be seen as a middle ground between hard and fuzzy clustering techniques, in which an object or instance is allowed to belong to one or more granules rather than simply one. In Fuzzy Logic, the overlap between the membership functions (as the results of overlap clusters) reflects the imprecise nature of the underlying concept [9]. It has been used as a combination in various applications (e.g., image processing and decision making), and to create overlap indices between two fuzzy sets in fuzzy rule-based classification systems [10]. This signifies the importance of overlapping clusters in representing the real-world data in the classification tasks.

We examine and assess how enhanced GrC with overlapping measure is used in Fuzzy Logic systems. Popular datasets including Iris, Wine, and Glass are used to verify the new framework. The results suggest that the overlapping GrC outperforms the classic GrC in terms of classification accuracy, indicating the overlapping GrC's potential for building Fuzzy Logic rule bases.

The rest of the paper is organized as follows. The GrC algorithm and the formation of fuzzy rulebase is discussed in Sect. 2. The proposed overlapping GrC is described in Sect. 3. Section 4 presents the experiment results based on three benchmark datasets and Sect. 5 presents the impact of the proposed overlapping clustering on the system's interpretability.

2 GrC-Fuzzy Logic Models

This section provides the details of the algorithm under study (Granular Clustering) and the elicitation of Fuzzy rule-bases from the final clusters (or granules).

2.1 Granular Clustering

The basis for information granulation in this work is a clustering method described in [11], known as Granular Clustering, which organises data in the form of hyper boxes. This methodology's objective is to collect data through granular data structure processes, which is then suppressed depending on specific similarities [12, 13]. By splitting up the original data into smaller pieces, an abstraction level is intended to be reached. The following iterative method carries out the core concept of the granulation approach suggested in [11]:

- Finding and combining the two information granules that are most compatible (based on the highest compatibility measure). The data set is smaller since the new information granule now contains both of the original information granules.
- Repeating the previous step until an acceptable level of granulation has been achieved.

In this paper, the compatibility measure is given as:

$$\text{Compat}(A, B) = \text{Distance}_{MAX} - \text{Distance}_{A,B} \cdot \exp(-\alpha \times CF) \quad (1)$$

in which

$$\text{Compactness Factor } CF = \frac{C_{A,B}/\text{Cardinality}_{MAX}}{L_{A,B}/\text{Length}_{MAX}} \quad (2)$$

here, Distance_{MAX} is given by the sum of maximum distance in every dimension:

$$\text{Distance}_{max} = \sum_{n=1}^d (\text{distance}) \quad (3)$$

and $\text{Distance}_{A,B}$ denotes the distance between granule A and B:

$$\text{Distance}_{A,B} = \frac{\sum_{v=1}^d w_v (D_1 - D_2)}{d} \quad (4)$$

where

$$D_1 = \max(\text{max}_{AV}, \text{max}_{BV}) \quad (5)$$

$$D_2 = \min(\text{min}_{AV}, \text{min}_{BV}) \quad (6)$$

With w_v is the weight for feature v , d is the number of input features; α balances the requirement between distance and density; Cardinality_{MAX} represents the maximum number of objects in the data space; Length_{MAX} is the maximum possible length of a granule in the data set; $C_{A,B}$ is the sum of cardinality of A and B ; and $L_{A,B}$ is the multi-dimensional length of the new granule, given by:

$$L_{A,B} = \sum_{v=1}^d (\text{max}_{Xv} - \text{min}_{Xv}) \quad (7)$$

2.2 Formation of Fuzzy Logic Rule Base

From the information granules obtained in Section A, it is possible to find relational information (rules) equivalent to a Mamdani FIS rule-base, where the IF part is called antecedent of the rule, and the THEN part is called the consequent of the rule [14].

$$\begin{aligned} \text{Rule 1 : IF}(\text{input } A = A_1 \text{ and } \text{input } B = B_1 \text{ and } \dots) \text{THEN}(\text{output} = O_1) \\ \text{Rule 2 : IF}(\text{input } A = A_2 \text{ and } \text{input } B = B_2 \text{ and } \dots) \text{THEN} (\text{output} = O_2) \end{aligned} \quad (8)$$

A Gaussian Fuzzy Logic membership function (MF) is dependent on two important parameters, namely c (the centre of a fuzzy set) and the σ , which stands for the width, or standard deviation. This information is extracted from the final information granules obtained from the iterative data granulation, in which one fuzzy rule is characterised by each information granule.

3 Overlapping GrC

The overlapping between the granules is numerically measured in this research using the R-value proposed in [15]. R-value, which may be thought of as the ratio of samples in the overlapping area, is a metric that can quantify how much category overlap there is in a dataset [16]. Many studies in literature have chosen to use R-value in their research because of its potential, particularly when dealing with overlapping data for example in [7] and [17].

3.1 R-Value

R-value between the two categories C_i and C_j is represented by $R(C_i, C_j)$ and it signifies the ratio of instances of C_i and C_j which are positioned in the overlapping area of C_i and C_j . R-value uses normalised values, meaning that R-value ranges between 0 and 1. The overlapping measure R-value in this paper is defined as:

$$R(C_i, C_j) = \frac{1}{|C_i| + |C_j|} [r(C_i, C_j) + r(C_j, C_i)] \quad (9)$$

where

$$r(C_i, C_j) = \sum_{m=1}^{|C_i|} \lambda(|kNN(p_{im}, c_j)| - \theta) \quad (10)$$

where $\lambda(x) = 1$ if $x > 0$, else $\lambda(x) = 0$, p_{im} is the m -th instance of category C_i , $kNN(p_{im})$ is the set of k -nearest neighbour instances for an instance p_{im} , $kNN(p_{im}, C_j)$ is the set of instances in $kNN(p_{im})$ that belong to the different category C_j , can be defined as:

$$kNN(p_{im} C_{ij}) = \{x | x \in kNN(p_{im}) \wedge x \in C_j\} \quad (11)$$

where k is the amount of instances that a specific instance has as nearest neighbours. The parameter θ (ranging from 0 to $k/2$) is the cut-off value on the number of different class neighbours for determining if an instance falls within an overlap region. Figure 1 depicts the k -nearest neighbour for two points with $k = 3$ and $\theta = 1$, as well as an illustration of evaluating overlapping using R-value based on a synthetic dataset.

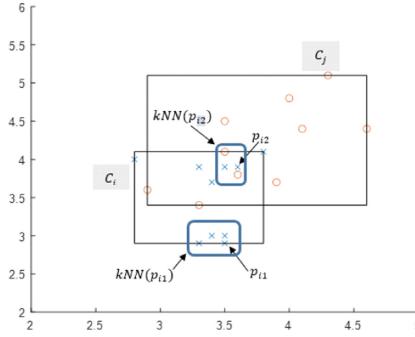


Fig. 1. K-nearest neighbour for 2 points.

3.2 A New Overlapping Measure During the Iterative Data Granulation

In this proposed algorithm, the overlapping is not only allowed to occur, but also can be controlled by means of two parameters, k and θ . k is the number of nearest neighbour instances for a given instance, and θ sets the maximum number of instances allowed to belong in class neighbours. For consistency, in this research the values of k and θ are set to be 5 and 1, respectively.

The overlapping of the granules is taken into consideration in a modification of the compatibility equation. The R-value of 1, indicates that the granules are within same categories, hence are likely to merge. Low overlapping suggests that the categories are clearly distinct and well-separated, thus they shouldn't be combined. Therefore, the proposed compatibility measure includes the overlapping measure, or R-value, in the compatibility equation as follows:

$$\text{Compat}(A, B) = \text{Distance}_{\text{MAX}} - \text{Distance}_{A,B} \cdot \exp(-\alpha(CF \times Rvalue)) \quad (12)$$

R-value's prominence in (12) encourages overlapping to a greater extent throughout the iterative data granulation. As a result, overlapping between the granules in GrC is permitted, and the degree of overlapping can be managed using the parameters k and θ . However, the compactness factor still has an impact on this equation's R-value. Therefore, five distinct values of, ranging from 0.2 to 1, are employed in the experiment.

Generally, the algorithm starts with treating all instances as a granule, followed by computing the R-value and hence the compatibility metrics. The two most compatible granules will be merged to form a new information granule, and the iteration continues an acceptable level of granulation has been achieved. Figure 2 shows the framework for overlapping GrC based FL modelling.

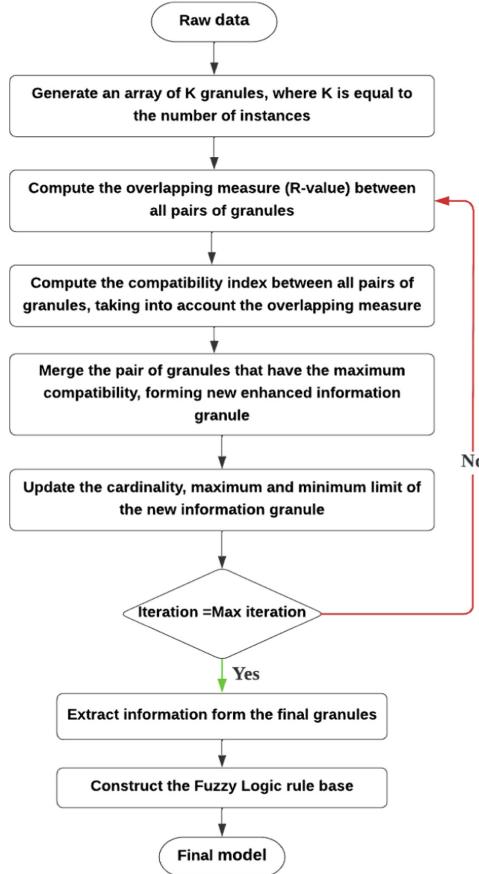


Fig. 2. Overlapping GrC based Fuzzy Logic modelling framework.

4 Case Study and Simulation Results

This section presents the improvement that the GrC with overlapping measures showed. Iris, Wine, and Glass datasets are utilised to verify the effectiveness of the suggested approach. The results are derived from 10 trials in each experiment, with data divided as follows: 80% for training and 20% for testing.

For consistency, the values of k and θ are kept constant at 5 and 1, respectively, throughout the experiment. The compactness factor value of the experiment ranged from 0.2 to 1.0, with a 0.2 increment. Table 1 to Table 3 contain the results for Iris, Wine, and Glass.

Table 1 displays the accuracy and RMSE of the experiment results for Iris (with 4 input variables, 150 instances). Three simulations of GrC with overlapping measure out of five (with varying values of α) indicate greater performance in terms of classification accuracy (i.e. when $\alpha = \{0.2, 0.4, 0.8\}$). The same trend is shown when RMSE for GrC with overlapping measure and GrC without overlapping measure are compared.

The improvement achieved by the proposed method can be seen more clearly in the Wine dataset (with 13 input variables, 178 instances), where the accuracy of the GrC with overlapping measure consistently surpasses the standard GrC. This is illustrated in Table 2. When $\alpha = 0.6$, the GrC with overlapping measure achieves the best accuracy, scoring 96.67%. Additionally, it has been found that the standard GrC can only maintain low values of α ($\alpha \leq 0.6$) in order to produce satisfactory results (i.e. accuracy 90%). Higher α (0.8 and 1.0) results in conventional GrC performing worse, with scores of 76.67% and 52.67%, respectively, whereas overlapping GrC performs better, with scores of 96% and 95.33%. This further emphasises the reliability of the overlapping GrC in FL model construction.

The performance of overlapping GrC in the Glass dataset (with 9 input variables, 214 instances) is shown in Table 3. In the Glass dataset, GrC with overlapping measure regularly outperforms the traditional GrC. The performance level with the highest accuracy is at $\alpha = 0.4$ (71.16%). For all values of α , the standard GrC produces good results (above 60%), with the exception of when $\alpha = 1.0$. The results show that GrC with overlapping measure may still deliver good results even for more complex datasets (like Glass), even when high values of were employed. In Glass, the distribution of the data per class is uneven, ranging from 9 to 76 samples per class. Hence, bootstrapping method is required in the pre-processing step. All the results are benchmarked with other related works, for example 97.33% (Iris) and 95% (Wine) in [18, 19], and 75.45% (Glass) in [19].

Table 1. Experiment result for Iris in terms of accuracy and RMSE.

α	GrC with overlapping			Standard GrC		
	RMSE	Acc.	SD	RMSE	Acc.	SD
0.2	0.1164	97.14	2.3	0.1261	96.67	3.00
0.4	0.1170	97.62	2.52	0.1298	95.71	2.52
0.6	0.1227	96.67	3.33	0.1224	97.14	2.30
0.8	0.1178	97.62	3.17	0.1325	96.19	1.26
1.0	0.1421	93.33	3.85	0.1249	97.62	3.17

Table 2. Experiment result for Wine in terms of accuracy and RMSE.

α	GrC with overlapping			Standard GrC		
	RMSE	Acc.	SD	RMSE	Acc.	SD
0.2	0.1134	94.67	2.98	0.092	93.33	4.08
0.4	0.1251	92.67	4.35	0.1386	91.33	4.47
0.6	0.0846	96.67	2.36	0.1272	92.00	3.80
0.8	0.0945	96.00	4.35	0.2416	76.67	9.13
1.0	0.1037	95.33	2.98	0.3082	52.67	29.48

Table 3. Experiment result for Glass in terms of accuracy and RMSE.

α	GrC with overlapping			Standard GrC		
	RMSE	Acc.	SD	RMSE	Acc.	SD
0.2	0.2252	66.05	5.35	0.2285	65.58	4.47
0.4	0.1974	71.16	2.08	0.1995	66.05	4.53
0.6	0.2142	70.23	5.04	0.2166	65.11	4.65
0.8	0.2185	68.37	6.28	0.1630	67.91	5.55
1.0	0.2239	63.26	6.24	0.2326	59.53	9.95

From the results presented in Tables 1, 2, 3, it is shown that the GrC with overlapping measure outperforms the conventional GrC in terms of classification accuracy and RMSE, suggesting that the overlapping GrC can be used to provide rule bases for Fuzzy Logic. The robustness of the proposed method is also demonstrated where it achieves good results using all compactness factors ($\alpha = [0.2, 1]$).

5 Interpretability Index

Apart from the accuracy, the proposed model is analysed in terms of the system's interpretability. The interpretability of a fuzzy model depends on several elements such as rules, antecedents and/or consequent numbers, and the semantics at the fuzzy partitioning level [20]. In this paper, the interpretability is assessed using Nauck's index (NI) [21] which has been used in many research related to Fuzzy Logic's interpretability [22, 23]. The NI shown in Table 4 indicates that the impact of overlapping clustering on the system's interpretability is minimum.

The Nauck's index is the product of three elements; complexity, coverage and partition of FL system and defined as:

$$\text{Nauck index} = \text{comp} \times \overline{\text{cov}} \times \overline{\text{part}} \quad (13)$$

(the details of NI can be referred to [21]).

Table 4. Comparison of the interpretability index.

Dataset	Nauck's Index	
	GrC	Overlapping GrC
Iris	0.0317	0.0319
Wine	0.0094	0.0092
Glass	7.2487 e-04	7.786 e-04

6 Conclusion

This work proposes a novel GrC algorithm with an overlapping measure. In order to accommodate for granule overlap during iterative data granulation, the compatibility equation includes a parameter known as R-value that is used to model the overlapping between the granules. By enabling an object to belong to one or more granules rather than just one, this strategy leads to the overlap of the final granules.

Utilising the Iris, Wine, and Glass datasets from the UCI Machine Learning Repository, the proposed overlapping GrC algorithm is validated. The suggested algorithm performs better in Iris, and the benefit is more pronounced in datasets with more intricate structures (Wine and Glass). The results show that, although having a larger compactness factor ($\alpha \geq 0.6$), the GrC with the overlapping measure beats the traditional GrC in terms of classification accuracy. This demonstrated the strength of the overlapping GrC and its potential for creating a FL rule bases.

However, this work is still limited to available datasets in UCI Repository. With good results achieved in the experiments, the overlapping GrC has the potential to be applied in more complex applications, such as additive manufacturing.

References

1. Cleuziou, G., Martin, L., Vrain, C.: PoBOC: an overlapping clustering algorithm, application to rule-based classification and textual data. In: Proceedings of the 16th European Conference on Artificial Intelligence, pp. 440–444 (2004)
2. Án, B.B., Nõ, D.V.: Survey of overlapping clustering algorithms. Comput. Sist. **24**(2), 575–581 (2020)
3. Khanmohammadi, S., Adibeig, N., Shafehbandy, S.: An improved overlapping k-means clustering method for medical applications. Expert Syst. Appl. **67**, 12–18 (2017)
4. Whang, J.J., Hou, Y., Gleich, D.F., Dhillon, I.S.: Non-exhaustive, overlapping clustering. IEEE Trans. Pattern Anal. Mach. Intell. **41**(11), 2644–2659 (2019)
5. Danganan, A.E., Los Reyes, E.: Ehmcoke: an enhanced overlapping clustering algorithm for data analysis. Bull. Electr. Eng. Inf. **10**(4), 2212–2222 (2021)
6. Adam, A., Kulevien, H.B.: Dealing with overlapping clustering: a constraint-based approach to algorithm selection. CEUR Workshop Proc. **1455**, 43–54 (2015)
7. Ben N'cir, C.E., Essoussi, N.: On the extension of k-means for overlapping clustering average or sum of clusters' representatives? In: IC3K 2013; KDIR 2013 - 5th International Conference on Knowledge Discovery and Information Retrieval and KMIS 2013 – 5th International Conference on Knowledge Management and Information Sharing, Proc., pp. 208–213(2013)

8. Solis, A.R., Panoutsos, G.: Granular computing neural-fuzzy modelling: a neutrosophic approach. *Appl. Soft Comput. J.* **13**(9), 4010–4021 (2013)
9. Berthold, M.R.: Fuzzy logic. In: Berthold, M., Hand, D.J. (eds.) *Intelligent Data Analysis*, pp. 321–350. Springer, Berlin, Heidelberg (2007)
10. Asmus, T.C., Dimuro, G.P., Bedregal, B., Sanz, J.A., Pereira, S., Bustince, H.: General interval-valued overlap functions and interval-valued overlap indices. *Inf. Sci.* **527**, 27–50 (2020)
11. Pedrycz, W., Bargiela, A.: Granular clustering: a granular signature of data. *IEEE Trans. Syst. Man Cybernet. Part B: Cybernet.* **32**(2), 212–224 (2002)
12. Solis, A.R.: Uncertainty and Interpretability Studies in Soft Computing with an application to Complex Manufacturing Systems. PhD Thesis, Department of Automatic Control and Systems Engineering, The University of Sheffield, UK (2014)
13. Muda, M.Z., Panoutsos, G.: An entropy-based uncertainty measure for developing granular models. In: 2020 7th International Conference on Soft Computing & Machine Intelligence (ISCFMI), Stockholm, Sweden, pp. 73–77 (2020)
14. Izquierdo, S.S., Izquierdo, L.R.: Mamdani fuzzy systems for modelling and simulation: a critical assessment. *J. Artif. Soc. Soc. Simul.* **21**(3), 1–18 (2018)
15. Oh, S.: A new dataset evaluation method based on category overlap. *Comput. Biol. Med.* **41**(2), 115–122 (2011)
16. Li, Z., Qin, J., Zhang, X., Wan, Y.: Addressing class overlap under imbalanced distribution: an improved method and two metrics. *Symmetry (Basel)* **13**(9), 1–16 (2021)
17. Fatima, E.B., Boutkhoum, O., Abdelmajid, E.M., Rustam, F., Mehmood, A., Choi, G.S.: Minimizing the overlapping degree to improve class-imbalanced learning under sparse feature selection: application to fraud detection. *IEEE Access* **9**, 28101–28110 (2021)
18. Riid, A., Preden, J.S.: Design of fuzzy rule-based classifiers through granulation and consolidation. *J. Artif. Intell. Soft Comput. Res.* **7**(2), 137–147 (2017)
19. Niu, J., Chen, D., Li, J., Wang, H.: Fuzzy rule based classification method for incremental rule learning. *IEEE Trans. Fuzzy Syst.* **30**(9), 3748–3761 (2021)
20. Pekaslan, D., Chen, C., Wagner, C., Garibaldi, J.: Performance and interpretability in fuzzy logic systems – can we have both? *Information processing and management of uncertainty. Knowl. Based Syst.* **1237**, 571–584 (2020)
21. Nauck, D.D.: Measuring interpretability in rule-based classification systems. In: The 12th IEEE International Conference on Fuzzy Systems, vol. 1, pp. 196–201 (2003)
22. Muda, M.Z., Solis, A.R., Panoutsos, G.: An evolving feature weighting framework for radial basis function neural network models. *Expert. Syst.* **40**(5), 1–14 (2023)
23. Razak, T.R., Garibaldi, J.M., Wagner, C., Pourabdollah, A., Soria, D.: Toward a framework for capturing interpretability of hierarchical fuzzy systems—a participatory design approach. *IEEE Trans. Fuzzy Syst.* **29**(5), 1160–1172 (2021)



Improved Rough-Multiple Regression for Unemployment Rate Model in Indonesia

Riswan Efendi^{1,3(✉)}, Mazidah Mat Rejab², Nureize Arbaiy², Widya T. Yofi³, Sri R. Widayati³, Izzati Rahmi^{1,4}, and Hazmira Yozza^{1,4}

¹ Mathematics Department, Faculty of Science and Mathematics, Universiti Pendidikan Sultan Idris, 35900 Tanjung Malim, Malaysia

riswanefendi@fsmt.upsi.edu.my

² Faculty of Computer Sciences and Information Technology, Universiti Tun Hussein Onn, 86400 Batu Pahat, Malaysia

³ Mathematics Department, Faculty of Science and Technology, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia

⁴ Department Mathematics and Data Science, Faculty of Mathematics and Natural Science, Andalas University, Padang 25163, Indonesia

Abstract. Many statistical approaches have been employed to model unemployment data such multiple regression and time series approaches. However, most researchers are not really concerned with inconsistent information in the data set before going to model building. While the accuracy of model prediction depends on data quality or data input. In this paper, we adopt rough set theory into multiple regression model for investigating the inconsistent sample and variables, respectively. This adoption is also considered to reduce a large number of independent variables. Some modifiable variables are related to unemployment rate, including labor force participation rate, gross domestic product, human development index, number of population and province in Indonesia. Interestingly, there are 34 provinces as dummy variables in representing of the geographical variable. The main objective is to measure and compare relationships between unemployment rate with these modifiable variables before (2019) and during (2020) in the pandemic era based on conventional multiple regression and proposed rough-multiple regression models. All information regarding variables above were gathered from Statistics Indonesia. The results showed the proposed rough-multiple regression able to improve the coefficient determination significantly from 67.6% to 85.9% for 2019 and 67.3% to 84.5% for 2020, respectively. While no significance different of this coefficient if derived with conventional multiple regression model. The proposed model better than conventional regression in explaining of the variation of the unemployment rate in Indonesia.

Keywords: Rough sets · multiple regression · dummy variable · unemployment rate · geographical factor · pandemic era

1 Introduction

One repercussion of the COVID-19 pandemic is the upsurge in unemployment figures. According to [1], one out of every four workers in Indonesia is at risk of unemployment due to the impact of COVID-19. The secondary data was obtained from the Statistics Indonesia survey further substantiates this claim, revealing that Indonesia faced a 7.07% open unemployment rate (OUR) in August 2020, exhibiting a rise of 1.84% compared to August 2019.

Dealing with the substantial unemployment rate poses a genuine concern for the government. The dynamics of unemployment involve a multitude of interrelated factors that mutually influence each other, forming intricate patterns that are not easily discernible [2]. Without immediate intervention, the growing unemployment rate will trigger new and multifaceted complications. Through the analysis of existing data, numerous researchers aim to uncover the factors influencing the unemployment rate. One of the widely used methods is linear regression. In the research conducted by [3], by using the linear regression method, it was found that economic growth and population growth did not affect the unemployment rate in Timika Regency. Similarly, in the studies conducted by [4] and [5], regression analysis was selected as their research methodology. However, linear regression has several limitations, including the non-occurrence of multicollinearity between independent variables, normally distributed data, and the difficulty of interpreting the intercept coefficient in real life [6, 20]. To address these constraints, researchers express interest in employing the Rough Sets method. Rough sets offer a method to manage information characterised by uncertainty [7]. Rough sets will evaluate important attributes, construct a minimal subset of independent attributes, and ensure the same classification quality as the entire set of attributes [8].

To discern the contrast between these two methods, the researcher applied both Linear Regression and Rough-Regression. The purpose of this observation is to see the factors that affect unemployment in Indonesia before and during the outbreak of COVID-19. Consequently, the dissimilarities in the factors influencing unemployment before and during the COVID-19 outbreak can be observed using both methods. Researchers hope that these results can be additional information in setting policies to reduce Indonesia's unemployment rate in the pandemic crisis era.

2 Variable Framework and Methods

In this research, the method employed is a quantitative method with a secondary data approach where the researchers only process the data that is already available. The secondary data was collected from the official website of Statistics Indonesia from each province. The sampling technique used is total population sampling or saturated sampling, which is a sampling technique that makes all objects or observations of the population a sample [9]. The population in this research is all regencies/cities in Indonesia with a total of 514, consisting of 416 regencies and 98 cities. The selection of 2019 and 2020 data in the research was based on the assumption that they represent the period before and during the pandemic of COVID-19.

In the previous studies [2, 10–14, 21–24], some modifiable variables have been associated to unemployment rate (OUR) such as human development index (HDI), population growth rate and labor force participation rate (LFPR). In addition, economic and geographical factors are represented by gross regional domestic product (GRDP) [2] and regional minimum wage (RMW) [13]. Meanwhile, geographical factors include the classification of provinces and districts/cities [14]. Based on these studies above, we considered variable framework is presented in Fig. 1.

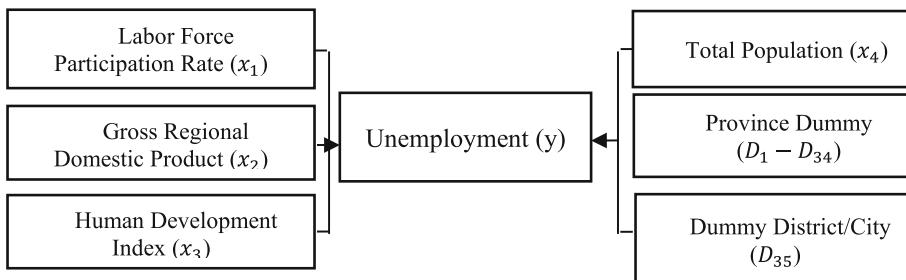


Fig. 1. Variable framework for unemployment rate

Based on source of the secondary data, there was missing data on the variable open unemployment rate in 2020 from the following 11 districts: Pacitan, Paniai, Nduga, Lanny Jaya, Mamberamo Tengah, Yalimo, Puncak, Dogiyai, Intan Jaya and Deiyai. Over recent years, districts with missing data tended to exhibit relatively low unemployment rate. As a result, the expected figure for the regency's unemployment rate is likely to be low, mirroring the trend in past data. In handling of missing data, SPSS software (Social Science Statistical Package) is used to estimate the values of this data.

2.1 Multiple Linear Regression

Linear regression is an analysis of the dependence of one or more independent variables on one dependent variable with the aim of predicting the average value of the population based on the values of the independent variables [15, 25]. The equation of multiple linear regression with the addition of dummy variables can be written as follows.

$$y = a + b_1x_1 + b_2x_2 + \dots + b_nx_n + c_1d_1 + c_2d_2 + \dots + c_nd_n + \varepsilon. \quad (1)$$

In Eq. (1), y is a dependent variable, x is an independent variable, d is a dummy variable, a is a constant value, b is the value of the regression coefficient of variable, x , c is the value of the regression coefficient of dummy variable d and ε is an error.

2.2 Rough Sets Theory

Rough sets theory is a mathematical tool to deal with ambiguity and uncertainty introduced to process uncertainty and inaccurate information [6]. The steps of employing the rough sets procedure are briefly outlined in Fig. 2 [16, 17, 26].

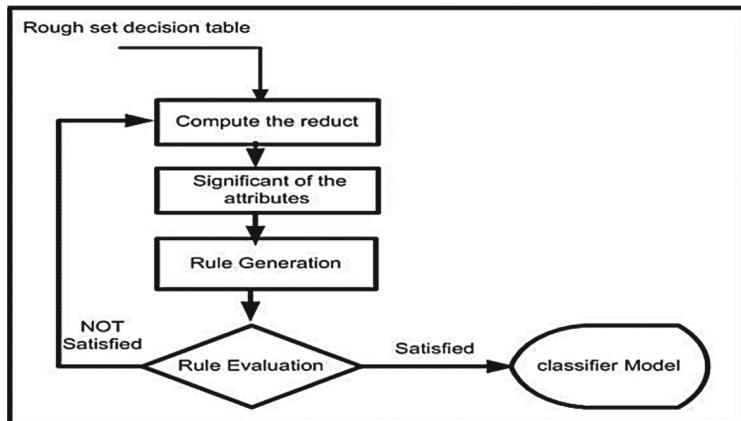


Fig. 2. Flowchart of Rough Sets for Decision Making

Based on Fig. 2, the inconsistent samples or objects or variables can be reduced using this theory. The reduction step is addressed to obtain the better quality of data sets before going to further analysis.

3 Results and Discussion

3.1 Descriptive Statistics for Unemployment Rate and Its Variables

We describe a general overview of the unemployment rate and its variables in Indonesia before and during pandemic era in Table 1.

Table 1. Descriptive Statistics of Unemployment Variables in Indonesia 2019 and 2020

Variables	2019		2020	
	Mean	Std. Deviation	Mean	Std. Deviation
OUR (y)	4.4253	2.27345	5.5780	2.68707
LFPR (x_1)	68.9761	6.13259	69.1786	6.33376
Real GRDP (x_2)	2160	47469.70048	21194	46225.32468
HDI (x_3)	69.5272	6.54558	69.6347	6.51408
Total Population (x_4)	521600	646818	535890	677278

Table 1 shows a general overview of each variable. The variable representing real GDRP exhibits a standard deviation larger than the mean, highlighting differences in GRDP among provinces. This finding is due to the varied patterns of economic growth observed across different regions [18]. Similarly, when considering the population variable, distinct factors impacting each region's population may lead to notable differences in values. An evaluation was carried out using a difference test (t-test) to assess potential discrepancies within the open unemployment rate (OUR) data between 2019 and 2020 is presented in Table 2.

Table 2. t-Test for unemployment rate between 2019 and 2020

		Paired Differences			t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean			
Pair	$y_{2019} - y_{2020}$	-1.15270	1.34939	.05952	-19.367	513	.000

The significant value presented in Table 2 may serve as a pivotal factor in decision-making. A significant value below 0.05 indicates variance within the open unemployment rate (OUR) population between 2019 and 2020. Hence, the researchers are keen on modelling the unemployment rate to explore the disparity between the two years.

3.2 Multiple Linear Regression Model for Unemployment Rate

In this section, the intercept and slopes of multiple linear regression model for the unemployment rate is presented in Table 3.

Based on Table 3, the grey cells indicate that the variable is not related to the open unemployment rate (OUR). The regression equation includes solely the variables that exhibit a significant correlation with the unemployment rate, leading to the formulation of the model in Eq. (2) and Eq. (3), respectively.

$$\begin{aligned} OUR \text{ (2019)} = & 11.352 - 0.185 LFPR + \dots + 1.244 KK \\ & + 1.531 AC + 1.759 SU + 1.265 SB + 1.486 RI + 0.992 KR \\ & + 2.748 BT + 2.718 JB - 1.162 BA + 0.964 KB + 1.725 KI \\ & + 0.861 SA + 1.529 MA + 2.293 PB + 2.355 PA + 1.244 KK \end{aligned} \quad (2)$$

$$\begin{aligned} OUR \text{ (2020)} = & 1.126 - 0.099 LFPR + \dots + 1.681 KK \\ & + 1.219 AC + 1.075 SB + 1.422 KR + 3.327 BT + 3.028 JB \\ & + 0.786 JT + 1.063 KB + 1.446 MA + 1.993 PB + 2.042 PA \\ & + 1.681 KK \end{aligned} \quad (3)$$

Table 3. Multiple Regression Model for unemployment rate

Variables	2019		2020	
	Parameter	Sig.	Parameter	Sig.
(Constant)	11.35224	1.3E-10	1.125757	0.529286
LFPR (x_1)	-0.18451	2.54E-36	-0.09978	4.86E-13
Real GRDP (x_2)	1.96E-06	0.434514	5.47E-06	0.060864
HDI (x_3)	0.045784	0.008997	0.117577	3.72E-09
Total Population (x_4)	4.14E-07	0.019811	7.31E-07	0.000105
AC-Aceh (D_1)	1.530634	6.4E-05	1.219023	0.00633
SU-North Sumatera (D_2)	1.759806	1.87E-07	0.547615	0.162128
SB-West Sumatera (D_3)	1.265929	0.00132	1.075533	0.020871
RI-Riau (D_4)	1.486261	0.001225	0.757652	0.164043
KR-Riau Islands (D_5)	0.992451	0.082315	1.421829	0.035092
JA-Jambi (D_6)	-0.28177	0.553458	-0.20526	0.7141
BE-Bengkulu (D_7)	0.281021	0.568154	-0.29768	0.610302
SS-South Sumatera (D_8)	0.575709	0.151379	0.18696	0.694669
BB-Bangka Belitung Islands (D_9)	-0.20043	0.723792	0.245338	0.715494
LA-Lampung (D_{10})	0.684202	0.101143	-0.19057	0.699755
BT-Banten (D_{11})	2.748284	5.39E-07	3.326734	2.5E-07
JB-West Java (D_{12})	2.71819	5.07E-13	3.027957	6.56E-12
JK-DKI Jakarta (D_{13})	-0.58996	0.471441	0.158726	0.869086
JT-Central Java (D_{14})	0.594133	0.062192	0.786285	0.037806
YO-DI Yogyakarta (D_{15})	-0.18458	0.78054	-1.15236	0.1394
BA-Bali (D_{17})	-1.16187	0.025575	0.392119	0.521497
NB-West Nusa Tenggara (D_{18})	-0.06414	0.914335	-0.9849	0.169662
NT-East Nusa Tenggara (D_{19})	0.436784	0.231995	0.451284	0.29666
KB-West Kalimantan (D_{20})	0.960433	0.027085	1.063435	0.038941
KS-South Kalimantan (D_{21})	0.398625	0.369333	-0.3053	0.562073
KT-Central Kalimantan (D_{22})	0.703518	0.109033	-0.34439	0.508509
KI-East Kalimantan (D_{23})	1.724837	0.000595	0.583858	0.321451
KU-North Kalimantan (D_{24})	0.384077	0.55771	0.062957	0.935711
GO-Gorontalo (D_{25})	0.359936	0.551624	-0.22069	0.758741
SR-West Sulawesi (D_{26})	0.317633	0.599092	-0.38727	0.588969
ST-Central Sulawesi (D_{27})	-0.43645	0.331774	-0.55463	0.297054
SG-Southeast Sulawesi (D_{28})	0.042191	0.918505	-0.01782	0.970818
SN-South Sulawesi (D_{29})	-0.12216	0.741779	-0.2224	0.613677
SA-North Sulawesi (D_{30})	0.861117	0.048065	0.994125	0.053074
MA-Maluku (D_{31})	1.529596	0.001704	1.446117	0.01146
MU-North Maluku (D_{32})	0.771753	0.126677	0.551095	0.357025
PB-West Papua (D_{33})	2.293227	8.67E-07	1.993446	0.000271
PA-Papua (D_{34})	2.355291	4.82E-09	2.042131	1.83E-05
Regency/City (D_{35})	1.24422	1.63E-08	1.681014	1.18E-10

Table 3 illustrates shifts occurring between 2019 and 2020 in seven provinces, specifically North Sumatra, Riau, Riau Islands, Central Java, Bali, East Kalimantan, and North Sulawesi. North Sumatra, Riau, Bali, East Kalimantan, and North Sulawesi underwent a

transition from having a contribution to the OUR variable to lacking a correlation. Meanwhile, the Riau Islands and Central Java shifted from lacking a association to having a correlation with the OUR variable.

3.3 Rough-Multiple Regression Model for Unemployment Rate

In this section, the inconsistent samples and variables are removed from the data set by using data reduction of rough sets, the proposed model is presented in Table 4.

Table 4. Rough-Multiple Regression Model for OUR

Variable	2019		2020	
	Parameter	Sig.	Parameter	Sig.
(Constant)	10.627	.000	6.533	.013
LFPR (x_1)	-.162	.000	-.132	.000
Real GRDP (x_2)	3.517E-7	.947	7.971E-6	.036
HDI (x_3)	.020	.409	.118	.000
Total Population (x_4)	9.500E-7	.002	6.325E-7	.004
AC-Aceh (D_1)	2.108	.000	-1.220	.077
SU-North Sumatra (D_2)	1.738	.000	-2.408	.000
SB-North Sumatra (D_3)	2.220	.000	-1.832	.004
RI-Riau (D_4)	1.585	.006	-2.503	.003
KR-Riau Islands (D_5)	.757	.268	-1.654	.040
JA-Jambi (D_6)	-.377	.511	-3.474	.000
BE-Bengkulu (D_7)	.507	.292	-3.337	.000
SS-South Sumatra (D_8)	.563	.196	-3.115	.000
BB-Bangka Belitung Islands (D_9)	.018	.977	-3.000	.001
LA-Lampung (D_{10})	.838	.113	-3.071	.000
BT-Banten (D_{11})	3.551	.000	.313	.657
JB-West Java (D_{12})	2.581	.000	-3.384	.004
JK-DKI Jakarta (D_{13})	1.071	.401	-1.622	.019
JT-Central Java (D_{14})	.311	.413	-3.991	.000
YO-DI Yogyakarta (D_{15})	-.094	.879	-2.394	.000
BA-Bali (D_{17})	-1.024	.039	-2.471	.003
NB-West Nusa Tenggara (D_{18})	-.067	.905	-3.803	.000
NT-East Nusa Tenggara (D_{19})	.499	.257	-2.378	.001
KB-West Kalimantan (D_{20})	1.062	.014	-1.846	.005
KS-South Kalimantan (D_{21})	.600	.171	-3.113	.000
KT-Central Kalimantan (D_{22})	.935	.042	-3.246	.000
KI-East Kalimantan (D_{23})	2.093	.000	-2.175	.006
KU-North Kalimantan (D_{24})	.744	.238	-2.971	.001
GO-Gorontalo (D_{25})	.281	.677	-3.221	.000
SR-West Sulawesi (D_{26})	.500	.387	-3.316	.055
ST-Central Sulawesi (D_{27})	-.167	.712	-3.839	.000
SG-Southeast Sulawesi (D_{28})	.654	.176	-2.959	.000
SN-South Sulawesi (D_{29})	.324	.464	-2.960	.000
SA-North Sulawesi (D_{30})	1.333	.006	-1.911	.007
MA-Maluku (D_{31})	1.898	.001	-1.304	.105
MU-North Maluku (D_{32})	1.154	.053	-3.126	.001
PB-West Papua (D_{33})	3.400	.000	.302	.736
PA-Papua (D_{34})	3.829	.000	-.926	.184
Regency/City (D_{35})	1.781	.000	1.434	.000

From the results in Table 4, the grey cells indicate that the variable is not related to the OUR. The rough-regression equation encompasses solely the variables that exhibit a noteworthy relationship with the unemployment rate, resulting in the formulation of the model as written in Eq. (4) and Eq. (5), respectively.

$$\begin{aligned} OUR(2019) = & 11.352 - 0.162 LFPR + \dots + 1.781 KK \\ & + 1.738SU + 2.220SB + 1.585RI + 3.551BT \\ & + 2.581JB - 1.024BA + 1.062KB + 0.935KT + 2.093KI \\ & + 1.333SN + 1.898SA + 3.400PB + 3.829PA + 1.781KK \end{aligned} \quad (4)$$

$$\begin{aligned} OUR(2020) = & 6.533 - 0.132 LFPR + \dots + 1.434 KK \\ & + 0.118HDI + 0.0000006325TP - 2.408SU - 1.832SB - 2.503RI \\ & - 3.384JB - 1.622JK - 3.991JT - 2.394YO - 2.471BA - 3.803NB \\ & - 2.378NT - 1.846KB - 3.113KS - 3.246KT - 2.175KI - 2.971KU \\ & - 3.221GO - 3.839ST - 2.959SG - 2.960SN - 1.911SA \\ & - 3.126MU + 1.434KK \end{aligned} \quad (5)$$

Upon observation, a notable shift becomes apparent in the variables correlated with the OUR's variable. A total of 18 provinces experienced a change from non-correlated to being correlated, namely the provinces of Riau Islands, Jambi, Bengkulu, ..., South Sulawesi, and North Maluku. On the other hand, the provinces of Aceh, Banten, Maluku, West Papua, and Papua experienced a change from being correlated to not being correlated.

3.4 Comparison Multiple Regression and Rough-Multiple Regression

This section presents a comparison between multiple linear regression and rough-multiple linear regression models for modelling OUR before and during pandemic as can be seen in Table 5.

Table 5. Comparison of Multiple and Rough-Multiple Regression

	Multiple Regression		Rough-Multiple Regression	
	2019	2020	2019	2020
Related Variables	17 variables	14 variables	17 variables	32 variables
R Square	0.676	0.673	0.859	0.845
Mean Error	0.260	28.835	0.866	0.016

Based on Table 5, it is found that the *R*-Square value of the proposed rough-multiple regression model is higher than the multiple regression model. The *R*-Square value of 0.85 denotes that the model elucidates 85% of the variability observed in OUR. The improvement of this value is also stated in previous study [27]. Furthermore, within

the results of both models, there is a discernible average error value for the year 2020, notably accentuated in the multiple regression approach. As explained in Sect. 2, in 2020, missing data occurred in several districts. Despite employing the method deemed optimal by the researchers for estimating missing data, the resulting value still fails to completely mirror the actual value. When multiple regression is used to model the unemployment rate, the estimation data will contribute to providing an error value. On the contrary, the proposed rough multiple regression model eliminates the inconsistent data, including estimations of missing data, which aims to minimize errors. Upon reviewing the results of the related variables, the researchers' preference seems inclined to the outcomes of the COVID-19 pandemic that changed all walks of life [1], including the unemployment rate. As evidence, Pearson's test was performed to see if there is a relationship between the number of cases of COVID-19 and OUR. r_{cal} value of 0.563 was obtained. The value is greater than the value of r_{tab} , concluding that there is a relationship between the number of cases of COVID-19 and OUR. This result provides an explanation that each related provincial dummy variable indicates the contribution of regional characteristic factors such as the number of COVID-19 cases to the magnitude of the OUR value.

For the total population variable, the result showed that there is a positive relationship between the total population and the open unemployment rate (OUR). Aligned with Malthus's theory [5], this finding implies that in contemporary society, an increasing population leads to a surplus of the workforce, yet without corresponding job opportunities, resulting in high unemployment. Theoretically, the relationship between the workforce and OUR is described as negative, implying that an increase in workforce participation typically corresponds to a decrease in TPT, and conversely, an increase in OUR aligns with a decrease in workforce participation [19].

4 Conclusion

Based on the computed results, distinctions between the model's characteristics before and during the pandemic crisis are identified. This crisis showcases a noticeable variation in geographical factors. Across 2020, almost all province-specific attributes had a bearing on the unemployment rate. Contrarily, the unemployment rate calculations for 2019 involved the participation of only 14 provinces. The rough sets method generates a regression model with a lower mean error compared to multiple regression and accounts for 85% of the variation in OUR. In addition, the authors are inclined toward the findings of the rough multiple regression method because it produces a model that better fits the existing context. The proposed rough multiple regression results highlight a disparity between the unemployment rate model before and during the pandemic crisis. Based on these findings, the rough-regression method could be a viable option for future researchers exploring unemployment rate modelling. Furthermore, it is encouraged that those utilising data from the Statistics Indonesia website exercise greater attention to the acquired data. This situation arises as authors commonly discover discrepancies between web-based data and materials released by this website.

Acknowledgement. This research was supported by Universiti Tun Hussein Onn Malaysia.

References

1. Samudra, R.R., Setyonaluri, A.: Inequitable impact of COVID-19 In Indonesia: evidence and policy response. Lembaga Demografi, Universitas Indonesia, Jakarta (2020)
2. Amrullah, W.A., Istiyani, N., Muslihatinningsih, F.: Analisis Determinan Tingkat Pengangguran Terbuka di Pulau Jawa Tahun 2007–2016. e-Journal Ekon. Bisnis dan Akunt. **6**(1), 43–49 (2019)
3. Taime, H., Djaelani, P.N.: Pengaruh Pertumbuhan Ekonomi dan Pertumbuhan Penduduk terhadap Tingkat Pengangguran di Kabupaten Mimika. J. Econ. Reg. Sci. **1**(1), 54–66 (2021)
4. Rayhan, A.A.M., dan Yanto, H.: Factors influencing unemployment rate: a comparison among five Asian countries. J. Econ. Educ. **9**(1), 37–45 (2020)
5. Handayani: Analisis Pengaruh Jumlah Penduduk, Pendidikan, Upah Minimum, dan PDRB terhadap Tingkat Pengangguran Terbuka di Provinsi Jawa Tengah. Diponegoro J. Econ. **1**(1), 159–169 (2019)
6. Efendi, R., Dewi, V.A., Rahmadeni, Basriati, S.: Pengaruh Pengangguran dan PDRB Terhadap Tingkat Kemiskinan Menggunakan Regresi Linier Berganda dan Rough Sets. Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIKI-10) (2018)
7. Skowron, A., Dutta, S.: rough sets: past, present, and future. Nat. Comput. **17**, 855–876 (2018)
8. Walczak, B., Massar, D.L.: Tutorial Rough sets theory. Chemom. Intell. Lab. Syst. **47**, 1–16 (1999)
9. Sugiyono: Metode Penelitian Kuantitatif Kualitatif dan R & D. Bandung: Alfabeta (2017)
10. Garnella, R., Wahid, N.A., Yulindawat: Pengaruh Pertumbuhan Ekonomi, Indeks Pembangunan Manusia (IPM) Dan Kemiskinan terhadap Tingkat Pengangguran Terbuka di Provinsi Aceh. J. Ilm. Mhs. Ekon. dan Bisnis Islam **1**(1), 21–35 (2020)
11. Bakce, R.: Analisis Perkembangan Jumlah Penduduk dan Tingkat Pengangguran Terbuka di Kota Pekanbaru. Menara Ilmu **14**(2), 139–149 (2020)
12. Mughni, M., Fadly, F., Adnan, A., Harison: “Pemodelan Tingkat Pengangguran Terbuka di Pulau Sumatera Dengan Menggunakan Regresi Nonparametrik Spline. J. Sains Mat. dan Stat. **6**(1), 133–144 (2020)
13. Fachrudin, F.A., Rahmanta & Rujiman: Analysis of factors affecting unemployment rate in Sumatera Utara Province. Int. J. Res. Rev. **7**(9), 117–122 (2020)
14. Weerasiri, A.R.P., Samaraweera, G.R.S.R.C.: Factors influencing youth unemployment in Sri Lanka. Asian J. Manag. Stud. **1**(1), 49–72 (2021)
15. Suliyanto: Ekometrika Terapan: Teori & Aplikasi dengan SPSS. Yogyakarta: CV. Andi, (2011)
16. Ramadan, N., Abdelaziz, A., Salah, A.: A hybrid machine learning model for selecting suitable requirements elicitation techniques. Int. J. Comput. Sci. Inf. Secur. **14**(6), 380–391 (2016)
17. Abdallah, S.R., Hassan, Y.F.: Using Gene Expression Programming In Learning Process Of Rough Neural Networks (2015)
18. Watrianthos, R.: Sistem Perekonomian Indonesia. Yayasan Kita Menulis, Medan (2021)
19. Wijaya, A.F.H.: “Analisis Faktor-Faktor yang Mempengaruhi Tingkat Pengangguran Terbuka (TPT) di Provinsi Aceh dengan Regresi Nonparametrik Spline Truncated,” Institut Teknologi Sepuluh Nopember Surabaya (2018)
20. Montgomery, D.C.: Introduction to linear regression analysis. In: Montgomery, D.C., Peck, E.A., Vining, G.G. (eds.), 5th edn. John Wiley & Sons, Inc., Hoboken, New Jersey (2012)
21. Syera, I.A., Tanjung, A.A., Triana, W.: The effect of human development index, inflation and economic growth on unemployment in Medan City. IJEC **2**(2), 410–422 (2023)
22. Putra, A.A.H., Hasmarin, M.I.: Analysis of the effect of education, gross regional domestic product, district minimum wage and population on open unemployment rates in central java in 2020–202. In: Proceedings of the International Conference on Economics and Business Studies (ICOEBS-22-2), pp. 207–216. Atlantis Press (2023). https://doi.org/10.2991/978-94-6463-204-0_18

23. Apergis, N., Arisoy, I.: Unemployment and labor force participation across the US States: new evidence from panel data. *J. Econ. Bus.* **67**(4), 45–84 (2017). ISSN 2241-424X
24. Pratiwi, F.R.: The effect of population growth and gross regional domestic product (Grdp) on the level of unemployment In The City of Makassar. *Econ. Resour.* **3**(2), 13–21 (2020)
25. Tranmer, M., Murphy, J., Elliot, M., Pampaka, M.: Multiple Linear Regression (2nd Edition); Cathie Marsh Institute Working Paper 2020-01 (2020). <https://hummedia.manchester.ac.uk/institutes/cmist/archive-publications/working-papers/2020/2020-1-multiple-linear-regression.pdf>
26. Cao, H.: The utilization of rough set theory and data reduction based on artificial intelligence in recommendation system. *Soft Comput.* **25**(3), 2153–2164 (2021)
27. Rasyidah, R.E., Nawi, N.M., Deris, M.M., Aqil Burney, S.M.: Cleansing of inconsistent sample in linear regression model based on rough sets theory. *Syst. Soft Comput.* **5**, 200046 (2023)



Utilizing Machine Learning for Gene Expression Data: Incorporating Gene Sequencing, K-Mer Counting and Asymmetric N-Grams Features

Chai-Wen Chuah^{1,2}(✉) WanXian He¹, De-Shuang Huang¹,
and Janaka Alawatugoda^{3,4}

¹ Guangxi Academy of Sciences, Nanning, Guangxi, China
aiwenchuah@gmail.com, dshuang@tongji.edu.cn

² Guangdong University of Science and Technology, Dongguan, Guangdong, China

³ Research and Innovation Centers Division, Rabdan Academy, Abu Dhabi, UAE

⁴ Institute for Integrated and Intelligent Systems, Griffith University, Nathan, QLD, Australia
jalawatugoda@ra.ac.au

Abstract. Comprehending the sequence of deoxyribonucleic acid sequence is a pivotal aspect of bio-informatics research. With the substantial surge in biological data, there arises a necessity of effective methodologies to address the crucial challenge within the overarching computation framework of deoxyribonucleic acid sequence classification. Numerous machine learning's can be used to complete these tasks compared to a manual technique that is followed for ages. The aim of this project is to perform effective approaches for pre-processing deoxyribonucleic acid sequences (K562 ChIP-seq) and use machine learning's to train the model by using deoxyribonucleic acid sequences to create judgments, predictions and classifications deoxyribonucleic acid sequences into known categories. As the machine learning may analyze large and complex datasets to identify patterns and trends of deoxyribonucleic acid sequences that may not be apparent to humans. In this study, the pre-processing methods are k -mers and N -grams. The machine learning classifiers, we employ Naïve Bayes classifier, K-nearest neighbors classifier and Random Forest classifier to train model and to evaluate the accuracy of predicting the K562 ChIP-seq. All the classifiers achieve overall accuracy over 80%. We show that Naïve Bayes classifier based on k -mers encoding which gives good result for both symmetric and asymmetric N -grams. The Naïve Bayes classifier based on 3-mers until 6-mers encoding have the highest accuracy of all the classifiers tested at 89.8% for symmetric N -grams and 90% for asymmetric N -grams. The Random Forest based on 3-mers and (2,3)-grams achieves the second highest accuracy in predicting the K562 ChIP-seq with 87.1%. The highest accuracy for K-nearest neighbors classifier based on 2-mers until 6-mers encoding for asymmetric N -grams tested at 86.9%. The results also reveal that the performance of the novelty classifiers' finding based on asymmetric N -grams are better if compared with symmetric N -grams. Naïve Bayes classifiers based on k -mers encoding and N -grams features shows significant performance in term of accuracy, precision, recall and F1-score.

Keywords: DNA sequencing · Machine learning · Naïve Bayes Classifiers · K-nearest neighbors Classifiers · Random Forest Classifiers · k -mers · N -grams

1 Introduction

Deoxyribonucleic acid (DNA) is distinctive in nature. The DNA sequence is a genetic codes that appears as seemingly random letters. The letters consist of adenine (A), cytosine (C), guanine (G), and thymine (T). The DNA sequence carry instructions for constructing and maintaining the body known as genes. These genes, responsible for the uniqueness of physical features, exhibit minor variations among individuals. Genes play a crucial role in guiding cells on protein production, essential for cell repair, growth, and tissue maintenance. The body's proteins are in a perpetual state of turnover. However, errors occurring in the translation process from genes to proteins may lead to the development of cancer genes, causing uncontrolled cell growth [25]. Inherited gene errors from parents or ancestors may also elevate the risk of cancer.

This study aims to identify Chromatin Immunoprecipitation Sequencing (ChIP-Seq) data for K562 chronic myelogenous leukemia (CML) sourced from the Encyclopedia of DNA Elements (ENCODE). K562 is an immortalized myelogenous leukemia cell associated with blood cells. The precise identifying of ChIP-Seq data for the cell line can significantly enhance our comprehension of the genetic mechanism underlying the disease, thereby advancing precision in drug discovery [1,2]. Current methods for discovering ChIP-Seq typically rely on experimental analysis of DNA-binding sequence structures to identify functional elements within the human genome sequence [3,4]. However, these experimental analyses are often time-consuming [5,6]. Given the rapid expansion of genomics DNA data, there is a growing need for predicting ChIP-Seq to identify transcription factor binding sites. Machine learning techniques have demonstrated exceptional results in classifying these tasks. Numerous studies [7–13,15] primarily propose the utilization of machine learning classification algorithms to analyze and predict genome DNA. Machine learning has the potential to autonomously predict and continuously improve decision-making processes by identifying specific trends and patterns based on the provided data. Noted that, the provided data which is the DNA sequences that may not be apparent to humans.

Mohamed [10] analyses coronavirus disease using language modeling (N -grams) together with machine learning language such that Naïve Bayes (NB), K-nearest neighbors (KNN), Artificial Neural Networks, Decision tree and Support Vector Machine. The range N -grams modeling is 2, 6 and 9. The research claims his models provide same global results but failed to show any accuracy prediction result. In related work of [11], their models applies N -grams modeling range of 2, 3 and 4. The experiments show high accuracy prediction results that using Random Forest (RF) but low accuracy prediction for NB which only 0.4 for 2-grams and 0.7 for 3-grams. In the work of [15], the researchers perform 4-mers towards the DNA sequences then apply machine learning's to classify the finding. The results show that low accuracy by using KNN that is 0.761 and RF accuracy is 0.915. Ravikumar *et al.* simulate their model using 4-mers and 5 mers respectively [13]. The accuracy by using 4-mers for NB is 0.854, KNN is 0.813 and RF is 0.337. The another model that using 5-mers has poor result for RF is 0.309 but good result for NB is 0.823. These common data pre-processing methods are either using N -grams or k -mers. Having the N -grams during data pre-processing assuming that the genome patterns are independent with each other. The experiments that only use k -mers in data pre-processing ignoring the positional context and dependency between

multiple k -mers that appear nearby in the underlying genome [14]. Hence, this paper considering and the contribution is to figure out the dependency between the genome patterns and possibility of independence for the underlying genome incorporating K -mer counting and asymmetric N -grams features. While the asymmetric N -grams features selection is the novelty finding in this paper. The genome is K562 ChIP-seq. The research experimental study is performed thought three steps that work successively: The first step is use k -mers; its role is to divide the sequence into k sequences. The second step applies the k sequences into N -grams; its role is to extract relevant information from a given sequences and to present it in a numeric forms. The third step is use machine learning techniques, NB, KNN and Random Forest (RF) to classify the K562 ChIP-seq.

The remainder of this paper is organized as follows: Sect. 2 presents the experimental processes, which include data preprocessing, the classification models and performance metrics. Section 3 contains the experimental results and discussions. Finally, Sect. 4 concludes the paper.

2 Materials and Methods

The 11033 labeled Chip-seq data K562 chronic myelogenous leukemia (CML) cell, 2340 labeled Chip-seq data GM12878 lymphoblastoid cell and H1 human embryonic stem cells database were obtained from Encode which had been processed and provided in [16]. There are 13373 DNA cells in total and there are no missing labelled. All the DNA sequences are labeled as 0 or 1. Non-K562 sequences are labelled as 0 while K562 sequences are labelled as 1. Then, these data are transformed into a uniform format that can be understood by the learning algorithms, which include tokenization using k -mers and N -gram modelling. Next, the models are classified using NB , KNN and RF. Each experiment is undergo two phases: training and validation. For the training phase, the experiment will randomly train 80% of the dataset. Then, the remaining dataset is validated at validation phase. Lastly, the performance for the models are evaluated. These steps are shown in Fig. 1.

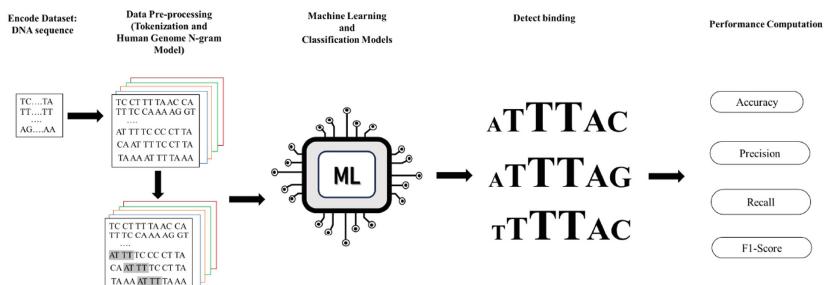


Fig. 1. Research methodology

2.1 Data Pre-processing

The data pre-processing steps include tokenization using k -mers and human genome language model using N -grams.

k -Mers. k -mers is the common method for tokenizing the genome that splitting the long DNA sequence into k length biological sub-sequences [17,18]. As shown in Table 1, there are five k -mers where we can tokenize the sequence “TCTAGTGGCTCA”. The five different k -mers will result different tokens and hence affect the performance of the language models. The k -mers range is between two until six are chosen as 1-mer will not provide any useful DNA sequence relation and accuracy prediction after 7-mers is decreased.

Table 1. Biological sub-sequences generated by k -mers

k -mers	Biological sub-sequences
2	TC CT TA AG GT TG GG GC CT TC CA
3	TCT CTA TAG AGT GTG TGG GGC GCT
4	TCTA CTAG TAGT AGTG GTGG TGGC
5	TCTAG CTAGT TAGTG AGTGG GTGGC
6	TCTAGT CTAGTG TAGTG AGTGG AGTGGC

N -Grams. This research applied N -grams model to assign a probability score to predict the DNA-binding sequences from different k -mers tokens. Equation 1 is a relative frequency is used to estimate the N -grams probability [19,20]. The probability is completed by dividing the observed frequency of a known or specific DNA-binding sequences by the observed frequency of a prefix or word of DNA sequences. There are nine N -grams models which consists of five symmetric and four asymmetric N -grams models respectively. Each models are trained and classified using NB, KNN and RF respectively. Again, the higher N -grams models are not consider due to low accuracy and long time in model training.

$$P(w_n|w_{n-N+1:n-1}) = \frac{C(w_{n-N+1:n-1}w_n)}{C(w_{n-N+1:n-1})} \quad (1)$$

2.2 Classification Model

Naïve Bayes (NB), k-nearest neighbors (KNN) and Random forest (RF) are supervised learning algorithms which are used in this research to evaluate which N -grams models will provide better accuracy in predicting the K562 sequences.

Naïve Bayes. The Naïve Bayes (NB) algorithm is adept at swiftly making predictions, particular in the context of addressing text classification problems that includes a high-dimensional training dataset. It operates based on probability Bayes theorem as shown in Eq. 2 [21]. Given an input $X = x_1x_2, \dots x_n$, the NB classifier assigns it a binary classes

such that ($c \in \{0, 1\}$), this is equivalent to determining c by comparing the likelihood ratio with a parameter δ . If the likelihood ratio surpasses δ , the classifier predicts c is 1. Otherwise, the classifier predicts c is 0. In this simulation, the smoothing parameter varies across the range of 0.1, 1, 10, 100 and 1000.

$$\frac{P(c = 1|X = x_1 x_2, \dots x_n) P(A|B)}{P(c = 0|X = x_1 x_2, \dots x_n) P(A|B)} = \frac{P(c = 1) \prod_{i=1}^n P(x_i|c = 2)}{P(c = 0) \prod_{i=1}^n P(x_i|c = 1)} > \delta \quad (2)$$

K-Nearest Neighbors. The Euclidean distance between $X_1 = (x_{11}, x_{12}, \dots, x_{1n})$ and $X_2 = (x_{21}, x_{22}, \dots, x_{2n})$ is expressed by Eq. 3 [22].

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2} \quad (3)$$

K-nearest neighbors (KNN) identify the closest classification point by considering the known distances to the classification of other points. This can be formally expressed as Equation 4, where C represents the predicted class, which is the most prevalent class label [22]. In this simulation, the KNN classification, the number of neighbors varies across the range of 2, 5, 8, 10 and 15.

$$C_i = \{x \in C_p; d(x, x_i) \leq d(x, x_m), i \# m\} \quad (4)$$

Random Forest. Random forests (RF) create diverse decision trees, each sharing the same nodes but employing different data, leading to distinct leaves. Each decision tree produces an output, and these outputs are combined, with the average classification result being taken. RF utilizes the Gini Index (GI) for measurement, defined in Eq. 5 by [23]. In this equation, P_i represents the probability of an element being classified for a particular class. The Gini Index calculates the probability within the range of 0 to 1. A GI value of 1 indicates a random distribution of elements across different classes, while a value of 0 suggests that all elements belong to a specific class. The GI value of 0.5 indicates an equal distribution of elements across various classes. The configuration parameters for the RF model include the number of trees in the forest, set within the range of 10, 25, 30, 50, 100, and 200. Additionally, the maximum depth of the tree varies in the range of 2, 3, 5, 10, and 20, and the minimum number of samples required to be at a leaf node spans 5, 10, 20, 50, 100, and 200.

$$GI = 1 - \sum_{i=1}^n (P_i)^2 \quad (5)$$

2.3 Performance Metrics

The model performance evaluation is based on accuracy (A), precision (P), recall (R), and F1-score ($F1$). These performance are evaluated using a confusion metric as shown in Table 2. The are four elements for the confusion matrix: True positives, true negatives, false positives, and false negatives which we denoted them as TP, TN, FP and FN

respectively. TN corresponds to instances when both predicted and actual events pertain to non-K562 ChIP-seq, indicating the sequence is not K562 ChIP-seq. FP denotes cases when the model incorrectly classifies non-K562 ChIP-seq as K562 ChIP-seq. FN occurs where the prediction is non-K562 ChIP-seq, but the actual data is K562 ChIP-seq. TP represents situations the model correctly predict the data is K562 ChIP-seq.

Table 2. Confusion Matrix

ChIP-seq	Non-K562 seq	K562 seq
Non-K562 seq	TN	FP
K562 seq	FN	TP

Accuracy. Accuracy refers to the proximity of a measurement to the accepted value. As defined in Eq. 6 [24], accuracy is the ratio of correct predictions for both TP and TN. Achieving high accuracy necessitates maintaining a combination of high precision and high trueness.

$$\text{Accuracy}(A) = \frac{TP + TN}{TP + FP + FN + TN} \quad (6)$$

Precision. Precision refers to positive predictive value. As defined in Eq. 7 [24], the precision is represents the ratio of correct predictions among both TP and FP, indicating instances where the model correctly predicts the DNA sequence as K562 ChIP-seq. Achieving high precision requires high trueness.

$$\text{Precision}(P) = \frac{TP}{TP + FP} \quad (7)$$

Recall. Recall refers to the sensitivity of the model in capturing TP value is expressed in Eq. 8 [24]. The recall value represents the proportion of correct predictions among the actual positive value.

$$\text{Recall}(R) = \frac{TP}{TP + FN} \quad (8)$$

F1-Score. F1-score refers to achieve a balance between the precision and recall is expressed in Eq. 9 [24]. F1-score assesses whether there is any uneven class distribution.

$$F1 - Score(F1) = \frac{2 * P * R}{P + R} \quad (9)$$

3 Result and Discussion

The results and discussion consist tables of performance metric that include accuracy, precision, recall, F1-score and graph of learning curve and scalability. The approaches we are using is combination of k -mers encoding and N -grams feature. There are five k -mers encoding and nine types of N -grams. We have three machine learning algorithms

to classify the DNA requesting. The total experiments are 135. The resulting classifier achieves over 0.80 overall accuracy.

Table 3 displays the accuracy, precision, recall and F1-score results for symmetric N -grams. It is being proved that the NB approaches outperform the others with a precision score is 0.897 as well as accuracy, recall and F1-score are 0.898. The parameter for this outperform is smoothing value equal to 1. The second most accuracy is performed by KNN at 0.868. The parameter for KNN classifier (number of neighbors) is 8. The lowest accuracy performs by RF at 0.862. The parameters for RF to achieve this accuracy are the number of trees, the maximum depth and the minimum number of samples, the values are 30, 20 and 5 respectively.

Table 3. Symmetric N -gram performance matrices for k -mers

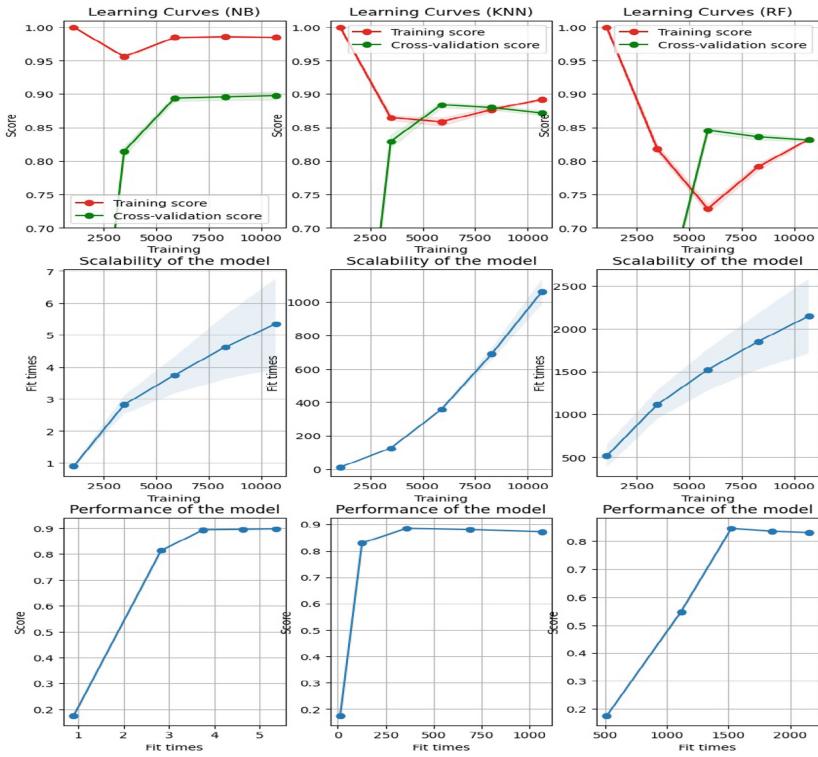
N -Grams	ML	2-mers				3-mers				4-mers				5-mers				6-mers				
		A	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1	A	P	R	F1	
2,2	NB	0.819	0.807	0.809	0.808	0.831	0.853	0.831	0.839	0.852	0.854	0.852	0.853	0.87	0.884	0.87	0.875	0.884	0.893	0.884	0.888	
	KNN	0.856	0.856	0.84	0.831	0.859	0.859	0.855	0.825	0.857	0.857	0.858	0.82	0.863	0.863	0.864	0.831	0.867	0.867	0.867	0.838	
	RF	0.847	0.847	0.838	0.805	0.856	0.856	0.848	0.822	0.86	0.86	0.859	0.827	0.86	0.86	0.867	0.822	0.841	0.841	0.846	0.787	
3,3	NB	0.831	0.853	0.831	0.839	0.852	0.854	0.852	0.853	0.87	0.884	0.87	0.875	0.884	0.893	0.884	0.888	0.898	0.897	0.898	0.898	
	KNN	0.859	0.859	0.855	0.825	0.857	0.857	0.858	0.82	0.863	0.863	0.864	0.831	0.867	0.867	0.867	0.838	0.868	0.868	0.865	0.841	
	RF	0.855	0.855	0.854	0.818	0.862	0.862	0.867	0.827	0.859	0.859	0.868	0.821	0.838	0.838	0.853	0.778	0.827	0.827	0.857	0.752	
4,4	NB	0.852	0.854	0.852	0.853	0.87	0.884	0.87	0.875	0.884	0.893	0.884	0.888	0.897	0.898	0.896	0.898	0.881	0.873	0.881	0.869	
	KNN	0.857	0.857	0.858	0.82	0.863	0.863	0.864	0.831	0.867	0.867	0.867	0.838	0.868	0.868	0.865	0.865	0.865	0.865	0.865	0.837	
	RF	0.859	0.859	0.858	0.825	0.861	0.861	0.862	0.826	0.853	0.853	0.855	0.811	0.831	0.831	0.851	0.761	0.826	0.826	0.832	0.75	
5,5	NB	0.87	0.884	0.87	0.875	0.884	0.893	0.884	0.888	0.898	0.897	0.898	0.898	0.881	0.873	0.881	0.869	0.882	0.873	0.882	0.874	
	KNN	0.863	0.863	0.864	0.831	0.867	0.867	0.867	0.838	0.868	0.868	0.868	0.865	0.841	0.865	0.865	0.86	0.837	0.859	0.859	0.853	0.828
	RF	0.855	0.855	0.858	0.816	0.849	0.849	0.856	0.802	0.831	0.831	0.86	0.761	0.824	0.824	0.679	0.745	0.824	0.824	0.679	0.745	
6,6	NB	0.884	0.893	0.884	0.888	0.898	0.897	0.898	0.898	0.881	0.873	0.881	0.869	0.882	0.873	0.882	0.874	0.84	0.875	0.84	0.851	
	KNN	0.867	0.867	0.867	0.838	0.868	0.868	0.865	0.841	0.865	0.865	0.86	0.837	0.859	0.859	0.853	0.828	0.857	0.857	0.846	0.826	
	RF	0.838	0.838	0.842	0.78	0.832	0.832	0.86	0.762	0.827	0.827	0.835	0.751	0.824	0.824	0.679	0.745	0.824	0.844	0.679	0.745	

Similarly, Table 4 shows the accuracy, precision, recall and F1-score results for asymmetric N -grams. The NB classifier with smoothing parameter equal to 1 has the better performance than other model with a accuracy and recall are 0.9, precision score is 0.896 and F1-score 0.898. The second most accuracy is performed by RF at 0.871. The parameters for RF to achieve this accuracy are the number of trees, the maximum depth and the minimum number of samples, the values are 30, 20 and 10 respectively. The lowest accuracy performs by KNN at 0.869. The KNN performs well for the number of neighbors are 8. Noted that, even though asymmetric N -gram outperforms compare with symmetric N -gram, the running time for asymmetric N -gram is longer compare with symmetric N -gram.

Figure 2 shows a comparative analysis in terms of the accuracy and scalability of NB, KNN and RF classifiers for (4-mers, {5,6}-grams), (5-mers, {4,5}-grams) and (6-mers, {3,4}-grams). From the investigation of the learning curves, it is notified that the KNN classifiers training and testing validation curves are almost closer to each other than other models do not have. The learning curve for NB classifiers training curve is high compares with the testing validation curve. The learning curve for RF classifiers is started high, then drop in the middle of training process and arise at the end. Next, when the scalability for all the models increase, the run-time increase.

Table 4. Asymmetric N -gram performance matrices for k -mers

N-Grams	ML	2-mers				3-mers				4-mers				5-mers				6-mers			
		A	P	R	F1																
2,3	NB	0.826	0.788	0.826	0.791	0.856	0.859	0.856	0.857	0.853	0.879	0.853	0.862	0.892	0.889	0.892	0.89	0.894	0.897	0.894	0.896
	KNN	0.855	0.855	0.841	0.828	0.86	0.86	0.858	0.827	0.861	0.861	0.859	0.829	0.864	0.864	0.863	0.832	0.869	0.869	0.868	0.841
	RF	0.855	0.855	0.853	0.818	0.871	0.871	0.869	0.847	0.868	0.868	0.87	0.838	0.853	0.853	0.858	0.811	0.838	0.838	0.84	0.781
3,4	NB	0.856	0.859	0.856	0.857	0.853	0.879	0.853	0.862	0.892	0.889	0.892	0.89	0.894	0.897	0.894	0.896	0.9	0.896	0.9	0.898
	KNN	0.86	0.86	0.858	0.827	0.86	0.861	0.859	0.829	0.864	0.864	0.863	0.832	0.869	0.869	0.858	0.841	0.869	0.869	0.865	0.842
	RF	0.861	0.861	0.856	0.829	0.865	0.865	0.866	0.833	0.856	0.856	0.862	0.817	0.836	0.836	0.85	0.773	0.833	0.833	0.854	0.765
4,5	NB	0.853	0.879	0.853	0.862	0.892	0.889	0.892	0.89	0.894	0.897	0.894	0.896	0.9	0.896	0.9	0.898	0.878	0.875	0.878	0.877
	KNN	0.861	0.861	0.859	0.829	0.864	0.864	0.863	0.832	0.869	0.869	0.868	0.841	0.869	0.869	0.868	0.842	0.862	0.862	0.854	0.834
	RF	0.86	0.86	0.863	0.824	0.855	0.855	0.857	0.816	0.844	0.844	0.845	0.793	0.831	0.831	0.845	0.763	0.824	0.824	0.679	0.745
5,6	NB	0.892	0.889	0.892	0.89	0.894	0.897	0.894	0.896	0.9	0.896	0.9	0.898	0.878	0.875	0.878	0.877	0.878	0.877	0.878	0.877
	KNN	0.864	0.864	0.863	0.832	0.869	0.869	0.868	0.841	0.869	0.869	0.868	0.842	0.862	0.862	0.854	0.834	0.865	0.865	0.861	0.836
	RF	0.858	0.858	0.868	0.817	0.839	0.839	0.844	0.782	0.827	0.827	0.807	0.753	0.827	0.827	0.835	0.751	0.825	0.825	0.855	0.746

**Fig. 2.** Performance comparison for NB, KNN and RF

4 Conclusion and Future Work

In this paper, the k -mers encoding with symmetric and k -mers encoding with asymmetric N -grams have been introduced for the data pre-processing phase. The novelty finding is the k -mers encoding with asymmetric N -grams. In the experiments, the DNA sequences are sized from 2-mers up to 6 mers are considered. There are five symmetric

N -grams and four asymmetric N -grams. The machine learning algorithms (NB, KNN and RF) are analysed in terms of accuracy, precision, recall and F1-score. The results reveal that the proposed k -mers encoding together with asymmetric N -grams gives better results when compared with the common k -mers encoding with symmetric N -grams in classifying the DNA sequence, specifically for K562 CML cell. That is NB classifier has an accuracy of 90%. One limitation for the machine learning models are the run-time increase when the data size increase. There is possible to design the models in parallel computing which might overcome the run-time versus data size problem and yet preserve the accuracy of learning and prediction.

Acknowledgment. This work is supported by Guangdong University of Science and Technology, Program ASEAN Talented Young Scientists, Guangxi Academy of Sciences and Radban Academy.

References

1. Yan, H., Tian, S., Slager, S.L., Sun, Z.: ChIP-seq in studying epigenetic mechanisms of disease and promoting precision medicine: progresses and future directions. *Epigenomics* **8**(9), 1239–1258 (2016)
2. Zou, Z., Iwata, M., Yamanishi, Y., Oki, S.: Epigenetic landscape of drug responses revealed through large-scale ChIP-seq data analyses. *BMC Bioinform.* **23**(1), 1–20 (2022)
3. Aimone, C.D., et al.: An experimental strategy for preparing circular ssDNA virus genomes for next-generation sequencing. *J. Virol. Methods* **300**, 114405 (2022)
4. Sontakke, V.A., Yokobayashi, Y.: Programmable macroscopic self-assembly of DNA-decorated hydrogels. *J. Am. Chem. Soc.* **144**(5), 2149–2155 (2022)
5. Roth, S., Ideses, D., Juven-Gershon, T., Danielli, A.: Rapid biosensing method for detecting protein-DNA interactions. *ACS sensors* **7**(1), 60–70 (2022)
6. Scaglione, E., et al.: An experimental analysis of five household equipment-based methods for decontamination and reuse of surgical masks. *Int. J. Environ. Res. Public Health* **19**(6), 3296 (2022)
7. Ali, F., Kumar, H., Patil, S., Ahmed, A., Banjar, A., Daud, A.: DBP-DeepCNN: prediction of DNA-binding proteins using wavelet-based denoising and deep learning. *Chemom. Intell. Lab. Syst.* **229**, 104639 (2022)
8. Liu, T., Wang, Z.: DeepChIA-PET: accurately predicting ChIA-PET from Hi-C and ChIP-seq with deep dilated networks. *bioRxiv* (2022)
9. Urda, D., Montes-Torres, J., Moreno, F., Franco, L., Jerez, J.M.: Deep learning to analyze RNA-Seq gene expression data. In: Rojas, I., Joya, G., Catala, A. (eds.) *IWANN 2017. LNCS*, vol. 10306, pp. 50–59. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59147-6_5
10. El Boujnouni, M.: A study and identification of COVID-19 viruses using N-grams with Naïve Bayes, K-nearest neighbors, artificial neural networks, decision tree and support vector machine. In: 2022 International Conference on Intelligent Systems and Computer Vision (ISCV), pp. 1–7 (2022)
11. Pandya, D.D., Jadeja, A., Degadwala, S., Vyas, D.: Ensemble learning based enzyme family classification using n-gram feature In: 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1386–1392 (2022)
12. Aswath, S., Kumar, C.H.M.S., Deepthi, V.H., Javeed, S.I., Rupesh, S.V.N.: DNA sequence classification with improved performance of supervised classifiers using nature inspired algorithms. In: 2022 2nd International Conference on Intelligent Technologies (CONIT), pp. 1–7 (2022)

13. Ravikumar, M., Prashanth, M.C., Guru, D.S.: Matching pattern in DNA sequences using machine learning approach based on K-Mer function. In: Gunjan, V.K., Zurada, J.M. (eds.) *Modern Approaches in Machine Learning & Cognitive Science: A Walkthrough*. SCI, vol. 1027, pp. 159–171. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-96634-8_14
14. Suzuki, Y., Myers, G.: Accurate k-mer classification using read profiles. In: 22nd International Workshop on Algorithms in Bioinformatics (WABI 2022) (2022)
15. Sarkar, S., Mridha, K., Ghosh, A., Shaw, R.N.: Machine learning in bioinformatics: new technique for DNA sequencing classification. In: Shaw, R.N., Das, S., Piuri, V., Bianchini, M. (eds.) *Advanced Computing and Intelligent Technologies*. LNEE, vol. 914, pp. 335–355. Springer, Singapore (2022). https://doi.org/10.1007/978-981-19-2980-9_27
16. Alipanahi, B., Delong, A., Weirauch, M.T., Frey, B.J.: Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nat. Biotechnol.* **33**(8), 831–838 (2015)
17. Compeau, P.E.C., Pevzner, P.A., Tesler, G.: How to apply de Bruijn graphs to genome assembly. *Nat. Biotechnol.* **29**(11), 987–991 (2011)
18. Chor, B., Horn, D., Goldman, N., Levy, Y., Massingham, T.: Genomic DNA k -mer spectra: models and modalities. In: Berger, B. (ed.) *RECOMB 2010*. LNCS, vol. 6044, pp. 571–571. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12683-3_37
19. Broder, A.Z., Glassman, S.C., Manasse, M.S., Zweig, G.: Syntactic clustering of the web. *Comput. Netw. ISDN Syst.* **29**(8–13), 1157–1166 (1997)
20. Dunning, T.: Statistical identification of language. Computing Research Laboratory, New Mexico State University Las Cruces (1994)
21. Murty, M.N., Devi, V.S.: *Pattern Recognition: An Algorithmic Approach*. Springer, Heidelberg (2011). <https://doi.org/10.1007/978-0-85729-495-1>
22. Cover, T., Hart, P.: Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **13**(1), 21–27 (1967)
23. Ho, T.K.: Random decision forests. In: Proceedings of 3rd International Conference on Document Analysis and Recognition, vol. 1, pp. 278–282 (2015)
24. Powers, D.M.W.: Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint [arXiv:2010.16061](https://arxiv.org/abs/2010.16061) (2020)
25. Ruggero, R., Pier Paolo, P.: Does the ribosome translate cancer? *Nat. Rev. Cancer* **3**(3), 179–192 (2003)



Text Sentiment Analysis on VIX's Impact on Market Sentiment Dynamics

Zhuqin Liang¹ , Mohd Tahir Ismail¹ (✉) , and Huimin Qu²

¹ School of Mathematical Sciences, Universiti Sains Malaysia,
11800 Gelugor, Penang, Malaysia
m.tahir@usm.my

² Centre for Sustainable Urban Planning and Real Estate (SUPRE), Faculty of Built Environment, Universiti Malaya, 50603 Kuala Lumpur, Malaysia

Abstract. This paper examines the impact of the US VIX (Volatility Index) on the sentiment of Chinese investors. We collected comments related to stocks on Sina Weibo from 2019 to 2023, and classified all comments using sentiment analysis with SnowNLP. Based on this, we constructed the Chinese Investor Sentiment Index (CISI). We then studied the relationship between VIX and CISI using Pearson correlation, linear tests, and Granger causality tests, and found that the influence of VIX on Chinese investor sentiment is not significant. However, we observed a positive correlation between the Chinese stock index and sentiment in the comments. Therefore, we conclude that the sentiment in the US stock market is relatively independent from the sentiment of Chinese investors.

Keywords: VIX · Sentiment Analysis · SnowNLP · Correlation · Granger Causality Tests

1 Introduction

In the era of globalization, the interconnectedness of global financial markets has become increasingly evident. The fluctuations in the U.S. stock market, being a major player in the global financial market, have significant implications for stock markets worldwide. One key indicator reflecting market sentiment and volatility in the United States is the VIX (Volatility Index). The VIX measures the market's expectations for future volatility and is often considered the sentiment of investors.

The present study focuses on exploring the potential impact of the U.S. VIX on the sentiment of the stock market in China. The inquiry is prompted by the fact that the volatility of the stock market in the US may trigger the volatility of the global market, including China. The surge in the VIX could potentially heighten global market sentiment, prompting us to investigate its specific influence on the sentiment of the Chinese stock market.

The sentiment of the Chinese stock market can reflect irrational behavior in the market, which behavioral finance posits that investors sometimes exhibit,

causing market prices to deviate from their intrinsic values. Moreover, the sentiment of the Chinese stock market can also serve as a forward-looking signal, a large volume of social media data can be utilized as a leading indicator for the future trend of the market. For instance, when negative sentiment reaches its peak, it may signal that the market is poised for a bottom-out and subsequent rebound. Conversely, excessive optimism might indicate an impending market top.

To quantify the sentiment of the Chinese stock market, we employ a unique approach—using sentiment analysis on stock-related comments from the Chinese microblogging platform Weibo. By analyzing the sentiment expressed in Weibo comments related to stocks, we aim to construct an index that reflects the sentiment level of the Chinese market.

The ultimate goal of this study is to employ a series of statistical tests to determine whether the fluctuations in the VIX impact the sentiment of the Chinese stock market. In recent research, the utilization of deep learning has become increasingly widespread; however, when it comes to characterizing temporal correlations in time series data, deep learning exhibits a complex and unstable nature. In this study, we chose Pearson correlation analysis, linear regression testing, and Granger causality tests to investigate the relationship between the US VIX Index and Chinese investor sentiment (CISI). This selection is based on their unique academic merits: Pearson correlation for assessing linear associations, linear regression for quantifying the impact of VIX changes on CISI, and Granger causality test for determining if VIX can predict changes in CISI. These methods collectively provide a comprehensive and rigorous examination of both the correlation and potential causal link between the volatility in the US stock market and Chinese investor sentiment, offering robust empirical evidence for global financial market dynamics and their influence on local investor behavior.

2 Literature Review

The sentiment of investors has become an important factor in the financial market, and it can also be an input variable when we forecast the market. Investor sentiment has recently been integrated into deep learning models for stock market prediction in research, effectively enhancing both accuracy and efficiency [5]. Moreover, sentiment can also affect the Bitcoin market, Georgoula et al. [2] studied the comments for Bitcoin using sentiment analysis, and then forecasted the Bitcoin's price using a Support Vector machine with the variable of comment sentiment.

In the course of textual comment analysis, it has been determined that sentiment analysis constitutes a suitable technique within natural language processing (NLP). To analyze the Chinese text, Xu et al. [11] presented a method that enhances text sentiment analysis accuracy by leveraging an extended sentiment dictionary and Naive Bayes classifier to determine the polarity of polysemous sentiment words, with experimental validation of its feasibility and effectiveness in improving sentiment recognition for comment texts. Furthermore, for

the implementation of sentiment analysis on Chinese text, SnowNLP stands out as a professional Python library in fulfilling this function. Researchers [10] have conducted a comparative analysis of the effectiveness of classifiers, contrasting those that neglect time series factors with those that incorporate such considerations. The findings indicate that accounting for time series factors leads to superior outcomes.

To examine the impact of VIX on the sentiment of Chinese investors, we conducted a series of correlation tests. Specifically, our study used the bootstrap technique to estimate the confidence interval of the Pearson correlation coefficient [1]. In addition, we also employed linear regression to assess the significance of the slope of the regression equation, to determine the correlation between these 2 time series [4]. Furthermore, the Granger causality test is also applied to examine whether VIX can influence the sentiment of Chinese investors. It is worth noting that the Granger causality test has been used to examine whether the composite stock indexes of the 4 countries effectively predict declines in their respective GDPs and verifies the causal relationship between stock market indices and GDP [7].

3 Methodology

Our study will follow the flow chart shown in Fig 1.

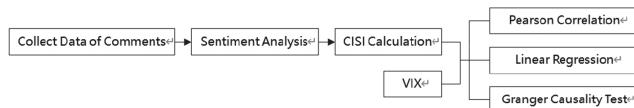


Fig. 1. Flow Chart of Methodology

3.1 Data Collection

Utilizing the Octopus software, we can systematically extract popular comments from Sina Weibo, wherein ‘popularity’ is defined by a substantial volume of engagements, including comments, likes, and reposts. The Chinese keywords associated with these comments encompass ‘stock’, ‘stock market’, and ‘A-shares’. Notably, ‘A-shares’ represents shares of Chinese companies traded on the Shanghai and Shenzhen stock exchanges in yuan, predominantly accessible to mainland Chinese investors. Alternatively, we have the option to directly acquire daily data for the US Volatility Index (VIX) from Yahoo Finance.

Given the real-time dynamics of global stock markets, our analysis with lagged VIX and CISI data inherently lacks immediacy. However, we contend that the impact of this lag is mitigated because VIX’s closing corresponds to a period of low commentary activity in Chinese media due to time differences.

Moreover, during the comment collection window, US market activity is also relatively subdued. Thus, we believe that lagged data has a limited effect on our results.

Furthermore, the choice to build CISI using daily data balances the transient nature of investor sentiment—longer periods can dilute sentiment signals, while shorter intervals suffer from data scarcity. This temporal resolution optimizes capturing significant shifts in sentiment while ensuring adequate data for robust analyses.

3.2 Sentiment Analysis of SnowNLP

Sentiment analysis [8], a natural language processing technique, is aimed at determining the emotional orientation within a given text, reflecting the author’s sentiments, typically categorized as positive, negative, or neutral. Positive sentiment indicates a favorable emotional expression, negative sentiment signifies an unfavorable emotional expression, and neutral implies a lack of discernible positive or negative emotions in the text.

SnowNLP, a natural language processing library in Python, employs Bayesian methodology to implement sentiment analysis through its sentiments attribute [3]. This attribute returns a value ranging from 0 to 1, representing the emotional polarity of the comment. Values nearing 1 signify positive sentiment, while values approaching 0 signify negative sentiment. Consequently, a threshold, such as 0.5, can be employed to ascertain whether the sentiment is positive sentiment or negative sentiment.

3.3 Sentiment Index

After we yield the proportions of positive sentiment comments (POS) and negative sentiment comments (NEG) for each day, we can obtain the time series data for the proportions of POS and NEG, respectively. Then, we can compute the Chinese Investor Sentiment Index (CISI) for each day by

$$CISI(t) = \begin{cases} \frac{POS(t)}{NEG(t)}, & \text{if } POS(t) > NEG(t) \\ \frac{NEG(t)}{POS(t)}, & \text{otherwise} \end{cases} \quad (1)$$

where

t represented the t th day,

$POS(t)$ represented the proportion of POS for the t th day,

$NEG(t)$ represented the proportion of NEG for the t th day.

In this formula, when the ratio between POS and NEG is approximately balanced, the CISI tends to approach 1, indicative of a relatively stable market sentiment. If there is a significant imbalance, with a noticeably higher number of positive or negative sentiment comments, the CISI will show a substantial increase, signifying high positive or negative market sentiment and suggesting market mood is volatile.

The principle of this formula is based on the meaning of the VIX index, which serves as an indicator of market sentiment, a low VIX corresponds to stable market sentiment, while a high VIX is associated with high positive or negative sentiment in the market. Therefore, the CISI formula is designed to capture analogous dynamics in the context of sentiment expressed in comments, aligning with the principles reflected in the VIX index as a measure of market sentiment.

3.4 Pearson's Correlation

Pearson's correlation coefficient [6], often denoted as “ r ”, is a statistical metric that measures the strength and direction of a linear relationship between two continuous variables. The result is between -1 to $+1$, where a value of $+1$ signifies a perfect positive linear relationship, -1 signifies a perfect negative linear relationship, and 0 signifies no linear relationship. Pearson's correlation is extensively employed in research and data analysis to evaluate the extent of association between variables.

3.5 Linear Correlation

Linear correlation refers to the statistical assessment to gauge the linearity of the relationship between two variables, often involving the examination of the null hypothesis that the slope is equal to zero after we build the linear regression model [4]. The process typically involves calculating a t-statistic based on the estimated slope, standard error, and degrees of freedom, and then comparing it to a critical value, or using the p-value to establish the statistical significance of the linear relationship.

3.6 Granger Causality Test

Granger causality testing [9] is a statistical method employed in time series analysis to ascertain if one variable is considered a cause of another. Named after Clive Granger, this test evaluates whether the historical values of one variable offer information about the future values of another beyond what is already known. The test involves estimating autoregressive models for each variable and examining whether the inclusion of lagged values of one variable significantly improves the prediction of the other.

4 Empirical Result and Analysis

4.1 Data Description and Cleaning

Due to the trading hours of the Volatility Index (VIX) in the United States, which operate between 9:30 am and 4:00 pm Eastern Time, accounting for changes in daylight saving time and winter time, the trading window for the

VIX in Beijing time must be between 9:30 pm and 5:00 am. Consequently, we exclusively collected Weibo comments between 5:00 am and 9:30 pm Beijing time to avoid market conditions during trading hours affecting comment sentiment. The data set comprises 189708 comments, gathered daily, commencing from January 1, 2019, to November 15, 2023.

Alternatively, the daily dataset of VIX closing prices obtained from Yahoo Finance encompasses 1228 data points, spanning from December 31, 2018, to November 14, 2023. It is noteworthy that the comments dataset lags behind the VIX dataset by one day due to the existence of a time difference between the two countries. The data from China requires a one-day adjustment to synchronize with the VIX data. The plot of the VIX value is shown in Fig. 2.

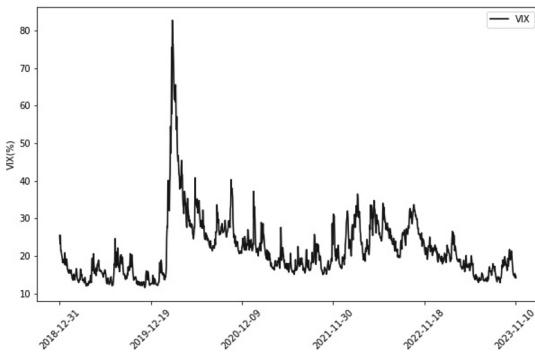


Fig. 2. Plot of VIX

4.2 Text Sentiment Index

Considering that the number of comments on certain dates falls below 20, indicating relatively low discussion regarding the stocks on those days and low volatility of sentiment, the CISI for these dates is set to 1. For all other dates, the CISI is computed according to Formula 1 in Sect. 3.3. In cases where the count of positive or negative comments is zero, the CISI for these dates is set to the maximum value among the valid CISI values on other dates. Then, the plot of the CISI is shown in Fig. 3.

4.3 Pearson's Correlation

We conducted a Pearson's correlation analysis between the data of VIX and the next-day CISI for days following VIX trades. Days without VIX trading were excluded from the analysis. The resulting correlation coefficient was 0.021. To assess whether the correlation coefficient significantly differed from zero, the

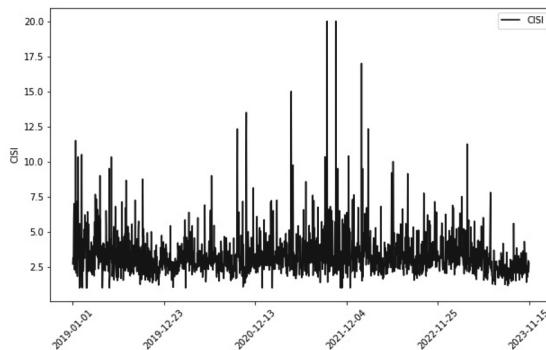


Fig. 3. Plot of CISI

bootstrap technique is employed to evaluate the confidence interval of the correlation coefficient. Considering our data sample is relatively small, the use of bootstrap methods for calculating correlation coefficients tends to rely less on stringent assumptions and is more robust, potentially yielding results that are closer to the true distribution.

We randomly sampled with replacement from the original dataset 1000 times, generating 1000 bootstrap samples. Each bootstrap sample was of the same size as the original sample, and due to the nature of the replacement sampling, each sample might contain repeated observations.

For each bootstrap sample, we recalculated the Pearson correlation coefficient. Based on the Pearson correlation coefficients obtained from the bootstrap samples, we computed the confidence interval. Using a 95% confidence level, we took the 2.5th and 97.5th percentiles of the sorted bootstrap sample correlation coefficients as the lower and upper bounds of the confidence interval. Consequently, the confidence interval for the Pearson correlation between the two sets of data is $(-0.025, 0.066)$.

We observed that zero lies within the confidence interval. Therefore, the Pearson correlation between the 2 variables is not significantly different from zero.

4.4 Linear Correlation

To assess the linear relationship between VIX and CISI, we employed the slope test in regression analysis. Initially, CISI was designated as the dependent variable, with VIX as the independent variable, and a linear regression equation was established: $CISI = 3.1585 + 0.003652 * VIX + \epsilon$. In this equation, 3.1585 represents the intercept, 0.003652 denotes the slope, and ϵ represents the residual term. The standard error of the slope was determined to be 0.005081. Consequently, employing the formula, the t-statistic for the slope was calculated to be 0.7187. In the context of the slope test, the null hypothesis is that the slope of the model is equal to zero, while the alternative hypothesis is that the slope of

the model is not equal to zero. Given that our calculated t-statistic is less than 1.96, where 1.96 is the threshold for a 95% confidence level, we fail to reject the null hypothesis. Consequently, we are unable to draw a conclusive inference regarding the existence of a linear relationship.

4.5 Granger Causality Test

The Granger causality testing is employed to check whether the time series X exerts an influence on the time series Y. Typically, this involves testing X against lagged Y. To facilitate alignment of the time series for the VIX, a zero value is appended at the end of its time series, while for the CISI, a zero value is added at the beginning of its time series (Table 1).

Table 1. Granger Causality Test for Lag 1 and Lag 2

Test	Lag	Statistic	p-value
SSR-based F test	1	0.0038	0.9508
SSR-based χ^2 test	1	0.0039	0.9500
Likelihood ratio test	1	0.0039	0.9500
Parameter F test	1	0.0038	0.9508
SSR-based F test	2	0.7468	0.4767
SSR-based χ^2 test	2	1.5731	0.4554
Likelihood ratio test	2	1.5608	0.4582
Parameter F test	2	0.7468	0.4767

For lag 1, the F-statistic is very small, and the p-value is high. Generally, this suggests that we failed to reject the null hypothesis. For lag 2, although the F-statistic is slightly higher, the p-value is still relatively high. This indicates that, at the given lag orders, the first time series is not Granger causal to the second time series.

4.6 Test for Chinese Market

In the above sections, we observed that the US VIX index does not have a significant impact on CISI. We believe this could be due to the relatively independent nature of the US and Chinese stock markets, where changes in the Chinese stock market may have a greater influence on stock market sentiment. Therefore, we utilized the daily returns of the Shanghai Composite Index (a major stock index in China) from January 1, 2019, to November 15, 2023, as a new time series and conducted another correlation analysis with CISI. The following conclusions were obtained.

- Pearson’s Correlation: The coefficient is 0.1734, while the confidence interval is (0.115, 0.232).

- Linear Test: The coefficient of the slope is 16.4, the standard error of the slope is 2.7, and the t-statistic for the slope is 6.08.
- Granger Causality Test (Table 2):

Table 2. Granger Causality for 1 and 2 Lags in Chinese Market

Test	Lag	Statistic	p-value
SSR-based F test	1	0.0876	0.7673
SSR-based χ^2 test	1	0.0878	0.7670
Likelihood ratio test	1	0.0878	0.7670
Parameter F test	1	0.0876	0.7673
SSR-based F teste	2	1.9517	0.1425
SSR-based χ^2 test	2	3.9201	0.1409
Likelihood ratio test	2	3.9136	0.1413
Parameter F test	2	1.9517	0.1425

We found from the results that the confidence interval of the Pearson correlation is significantly greater than 0, and the t-statistic of the slope in a linear regression model is also greater than 1.96, where 1.96 represents a 95% confidence level. Therefore, the slope is also significantly greater than 0. However, the Granger causality test still shows that the Shanghai Composite Index does not have a significant causal effect on the sentiment of Chinese stock market commentary. In conclusion, through the Pearson correlation and linear regression tests, we can determine that there is a positive correlation between the daily change rate in the Shanghai Composite Index and the sentiment of the Chinese stock market commentary.

5 Conclusion

In summary, our research employed the Pearson correlation, linear testing, and Granger causality testing to investigate the impact of the VIX index on Chinese media sentiment towards the stock market. However, all three methods showed the correlation is insignificant. However, when we conducted a study using the Chinese stock market index, we found a positive correlation between the index and Chinese stock market comment sentiment.

From the experimental results, we observe a positive correlation between the Chinese stock market index and investor sentiment on social media, indicating that stock market performance directly impacts investor sentiment. However, the influence of the VIX Index on investor sentiment is not statistically significant. This could potentially be attributed to the issue of market transmission; for the VIX to affect the sentiment in Chinese social media towards the stock market, it would first need to impact the performance of the Chinese market. If the

VIX and the Chinese market index are indeed independent from one another, then the conclusion about their mutual independence with regard to investor sentiment becomes more firmly established, which we will further investigate in our subsequent research by analyzing the interdependence between the two markets.

On the other hand, this paper has only employed three methods—Pearson correlation, linear regression, and Granger causality tests—all of which are univariate time series analysis techniques. In our next phase of analysis, we plan to consider using more advanced models such as vector autoregression (VAR) or multivariate generalized autoregressive conditional heteroskedasticity (multivariate GARCH) models. These methods allow for a multidimensional approach to examine spillover effects among time series data, thereby enabling us to determine whether there exists any mutual influence among these time series variables.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Dufera, A.G., Liu, T., Xu, J.: Regression models of Pearson correlation coefficient. *Stat. Theory Relat. Fields* **7**(2), 97–106 (2023)
2. Georgoula, I., Pournarakis, D., Bilanakos, C., Sotiropoulos, D., Giaglis, G.M.: Using time-series and sentiment analysis to detect the determinants of bitcoin prices. Available at SSRN 2607167 (2015)
3. Hu, N.: Sentiment analysis of texts on public health emergencies based on social media data mining. *Comput. Math. Methods Med.* **2022** (2022)
4. James, G., Witten, D., Hastie, T., Tibshirani, R., Taylor, J.: Linear regression. In: James, G., Witten, D., Hastie, T., Tibshirani, R., Taylor, J. (eds.) *An Introduction to Statistical Learning: With Applications in Python*. Springer Texts in Statistics, pp. 69–134. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-38747-0_3
5. Jin, Z., Yang, Y., Liu, Y.: Stock closing price prediction based on sentiment analysis and LSTM. *Neural Comput. Appl.* **32**, 9713–9729 (2020)
6. Liu, Y., Mu, Y., Chen, K., Li, Y., Guo, J.: Daily activity feature selection in smart homes based on Pearson correlation coefficient. *Neural Process. Lett.* **51**, 1771–1787 (2020)
7. Molnár, A., Csiszárík-Kocsir, Á.: Forecasting economic growth with V4 countries' composite stock market indexes—a granger causality test. *Acta Polytech. Hung.* **20**(3), 135–154 (2023)
8. Taboada, M.: Sentiment analysis: an overview from linguistics. *Annu. Rev. Linguist.* **2**, 325–347 (2016)
9. Troster, V.: Testing for granger-causality in quantiles. *Economet. Rev.* **37**(8), 850–866 (2018)
10. Tseng, C.W., Chou, J.J., Tsai, Y.C.: Text mining analysis of teaching evaluation questionnaires for the selection of outstanding teaching faculty members. *IEEE Access* **6**, 72870–72879 (2018)
11. Xu, G., Yu, Z., Yao, H., Li, F., Meng, Y., Wu, X.: Chinese text sentiment analysis based on extended sentiment dictionary. *IEEE Access* **7**, 43749–43762 (2019)



Multilevel Monte Carlo Simulation Model for Air Pollution Index Prediction of a Smart Network

Mustafa Hamid Hassan¹, Salama A. Mostafa^{2(✉)}, Rozaida Ghazali²,
Mohd Zainuri Saringat², Noor Aida Husaini³, Aida Mustapha⁴,
Mohammed Ahmed Jubair¹, and Hussein Muhi Hariz⁵

¹ College of Information Technology, Imam Ja'afar Al-Sadiq University, 66002 Al-Muthanna, Iraq

{mustafa.hamed, mohammed.a}@sadiq.edu.iq

² Faculty of Computer Science and Information Technology, Universiti Tun Hussin Onn Malaysia, 84600 Parit Raja, Johor, Malaysia

{salama, rozaida, zainuri}@uthm.edu.my

³ Faculty of Computing and Information Technology, Tunku Abdul Rahman University of Management and Technology, Jalan Genting Kelang, 53300 Kuala Lumpur, Malaysia
nooraida@tarc.edu.my

⁴ Faculty of Applied Sciences and Technology, Universiti Tun Hussein Onn Malaysia, 84500 Johor, Malaysia
aidam@uthm.edu.my

⁵ Department of Computer Techniques Engineering, Mazaya University College, 6400 Dhiqar, Iraq
huhraiz22@mpu.edu.iq

Abstract. Air pollution has become a significant environmental challenge in the 21st century due to widespread industrialization and urbanization worldwide. Effectively reducing it requires precise predictions of air quality. However, existing methods for predicting the Air Pollution Index (API) fail to effectively model short-term variables' dependencies and mostly neglect spatial correlations. Given these limitations, the statistical method emerges as the most suitable choice. Monte Carlo Simulation (MCS) is one of the best short-term time series prediction statistical approaches. This paper proposes a Multilevel Monte Carlo Simulation (MLMCS) model based on the MCS model for forecasting the API. The study covers the analysis of an air pollution dataset that includes the API and ambient air quality of ten locations in Beijing, China. The results show that the MLMCS improves the performance of API prediction compared to the MCS. The MLMCS model has the highest accuracy of 86.45% and the lowest computational time of 3.43 s compared to the MCS model's accuracy of 82.90% and computational time of 7.5 s.

Keywords: Air quality prediction · Air Pollution Index (API) · Suspended Particulate Matter (SPM) · Monte Carlo Simulation (MCS)

1 Introduction

An air pollutant is a term applied to any substance in the atmosphere that is considered suitable for contaminating it. These pollutants are sometimes highly concentrated and can persist in the atmosphere for quite a long time. That is dangerous to the health of people, animals, and plants [1, 2]. The harmful Suspended Particulate Matter (SPM) could be more detrimental to our health than other pollutants. The SPM is a collection of solid particles (either liquid or solid) found in a specific environment. These dust particles come from different sources, various compositions, and sizes, demonstrating a complex mixture of solid e and organic materials [3, 4]. The dust particles are classified according to their aerodynamic properties, which include conveying and eliminating the particles from the air. They are deposited in the respiratory system and associated with chemical sources and composition [5].

Many complexities are associated with assessing the risk of air pollutants through estimation or probabilistic methods. Researchers have used many algorithms and techniques to propose and develop methods for predicting air quality [4, 6]. In their studies, attention is given to some of the statistical and artificial intelligence approaches that achieved a promised result in several prediction domains because they can handle uncertainties associated with input data. The statistical methods include stochastic and probabilistic analysis [3]. Statistical algorithms such as Monte Carlo Simulation (MCS) are utilized to enhance the accuracy of air pollution prediction, which is the most promising direction in smart environment development [4]. Meanwhile, machine learning algorithms, such as artificial neural networks, involve processes that imitate human intelligence using instrument machines, especially for prediction tasks [5].

Forecasting air pollutants is crucial in safeguarding public health by providing early alerts about dangerous air quality levels. Yet, making these predictions is difficult due to various intricate factors, including varying weather conditions, the complex and nonlinear progression of air quality metrics, and the widespread distribution of air pollution. Many approaches have been proposed to tackle the problem of predicting short-term air quality. Monte Carlo Simulation (MCS) has previously been applied to perform estimation and forecasting in different domains, including air quality, gas, groundwater, ground motion, electricity, and road networks. This work proposes a Multilevel Monte Carlo Simulation (MLMCS) model for forecasting the Air Pollution Index (API). It then compares the MLMCS and the MCS models using the air pollution dataset to check the best forecasting method for the API concentrations.

2 Related Work

The MCS-based methods have experienced a rise in the horizon of short-term weather prediction, and they aim to consider the complexities and uncertainties in weather processes. The combination of MCS techniques and atmospheric dispersion models allows the simulation of the spreading of pollutants by considering emission rates, meteorological conditions, and terrain characteristics [4]. Moreover, MCS combines historical pollutant concentration data and temperature and humidity information to render forecasts. Ensemble forecasting methods, in which MCS simulations present the user with

real-life outcomes, are highly suited to air quality management and climate models that estimate the probabilities effect of various scenarios [6]. The decision-making is based on the relative likelihood of the selected scenario. Although this approach may be perceived as a suitable tool to improve the precision and applicability of short-term air quality forecasts, some challenges are yet to be addressed. For example, refinement of methodologies, incorporating various data sources, and validating model outputs are areas of improvement.

The work of Tchórzews et al. [7] concerned that a gas pipeline had been proven to fail based on the field test device materials gathered over a decade (2004–2014). Many gas network systems are used to test the work. The failure rate of the gas network is indicated as a function of the pressure and diameters of the pipeline and pressure. Ren et al. [8] take advantage of MCA's interval transformation analysis (ITA), which is based on MCS, to manage groundwater treatment by naphthalene contamination. To conduct the analysis, key input data such as health risks, total costs, and levels of contaminants are presented in the form of ranges. This approach helps reduce computation time using the MCS sampling method. This method is effective in pinpointing the optimal management strategies amidst data uncertainties. The effectiveness of this approach is validated through a real-world case study. The findings from this study suggest that the best management strategy is action 15 for a 5-year plan, action 8 for a 10-year plan, action 12 for a 15-year plan, and action 2 for a 20-year plan.

In work done by Heck et al. [9], an MCS analysis of the leveled cost of electricity is used to obtain the probability distributions for the costs of main generation technologies instead of the normal point values. The MCS approach is accompanied by a degree of complexity that differs from point values. However, it offers more realistic details regarding uncertainties and risk, allowing a more useful analysis of potential investments in the generation of analysis. With MCS, a quantitative evaluation of uncertain investment risk can be done based on an individual's risk aversion.

In Akkar et al. [10], MCS is used to compute the yearly exceedance rates of dynamic ground-motion intensity measures (GMIMs), such as spectral acceleration and peak ground acceleration. Their work integrated spatial relationships and near-fault directivity using the multi-scale random field technique. Using the multi-scale random, they are able to generate MCS to assess the probabilistic seismic hazard of dynamic GIMMs. A conditional hazard curve can be produced utilizing the proposed approach.

A hybrid method is proposed by Gehl et al. [11] from the Bayesian modeling for real-world seismic hazard networks. The modeling is carried out based on a preliminary MCS. Although computational challenges restrict the use of Bayesian Networks to more basic infrastructure systems, MCS has successfully pinpointed the most frequent and typical damage scenarios. This feature has enabled the creation of a more streamlined Bayesian Network model. The MCS applicability is showcased through its use on an intricate road network in the French Pyrenees. The developed Bayesian Network from this application can forecast different system performance indicators and refine these predictions with the help of on-the-ground observations.

In the work of Rezvani et al. [12], the techno-economic reliabilities and benefits of solar-powered heaters are estimated using MCS. The study focuses on a product range produced by a local company in Australia. The company's historical data is represented

in the model using the inverse Weibull distribution function. The aim is to predict the number of failures per operating time for each component.

In an attempt to assess the risk posed by groundwater that petroleum PAHs contaminate the health of humans, Rajasekhar et al. [13] employ the MCS technique. In this work, the authors used one of the metropolitan cities in India as the case study. The assessment is conducted among children aged 0 to 5 years and adults aged 21 to 70, using probabilistic and deterministic approaches. The children and adults used for the assessment are exposed to PAHs. One of the aims of their study is to address the disparities and uncertainties associated with using MCS in the two categories. Also, they aimed to determine the preliminary remediation goals and afterward proposed corrective measures as a strategy for risk management.

3 Methods

The main methods used in this work are the MCS, MLMCS, and the testing dataset of air pollution. Subsequently, the performance evaluation metrics of Mean Absolute Error (MAE), Root-Mean-Square Error (RMSE), and Coefficient of Determination (R^2) are presented. They are used to assess and predict air quality. These methods are described in detail in this section.

3.1 Monte Carlo Simulation

MCS is a technique that performs statistical procedures that mainly depend on random numbers. This technique can also provide approximate remedies to various classes of mathematical problems [3, 7]. This problem can be solved in an MCS by carrying out the samplings using statistical computer experiments. Then, the problem is solved using different methods for issues where probability is not inherent and in issues with the natural framework of probability. The MCS method accommodates the prediction model's uncertainty. The idea behind using MCS is to leverage the repetitive sampling of the predictive model's functioning to depict the forecast accurately. This model uses the prediction values as deterministic input variables to the MCS algorithm. MCS has the relatively lowest error errors compared to other time-series forecasting algorithms tested. The rationale is to use the most accurate predicted inputs to achieve the most reliable simulated outcomes for the MCS. This method is an integral of numerical algorithms based on the repeated use of random sampling with the central fact that the user's ability to create random and ideally random numbers is necessary for that method. The algorithm for MCS has the following algorithm as its core algorithmic steps.

- Step 1: define the range of possible input parameters;
- Step 2: generate a set of random values for each parameter;
- Step 3: perform a deterministic computation for each set of random inputs;
- Step 4: aggregate the results;
- Step 5: analyze the results to know the range of possible outcomes;
- Step 6: refine the simulation parameters based on the analysis;
- Step 7: repeat the process until obtaining consistent and reliable results;

MCS stands for a model that is based on simulating a real system with mathematical models. The simulation of MCS is at the core of the method because of the use of random numbers and repetitive calculations, which make the method very suitable for computational calculations. This approach offers an efficient, useful, and reasonably accurate method, particularly when finding an exact solution with a deterministic algorithm is impractical [8, 9]. When MCS is applied to risk assessment, the risk appears as a frequency distribution graph similar to the familiar bell-shaped curve, which non-statisticians cannot intuitively understand.

3.2 Multilevel Monte Carlo Simulation

MCS has become one of the driving computing tools in several domains, such as the finance industry and air quality prediction. One of the main reasons this method is becoming increasingly important in the industry is the need to simulate high-dimensional stochastic models. This requirement often arises because the complexity of these models tends to increase linearly with the size of the problem at hand. In computational terms, the primary goal of this approach is to achieve the required level of precision, although this often comes with a significant computational expense.

The key issue of the Multilevel Monte Carlo Simulation (MLMCS) model is providing an architecture that performs simulations at multiple levels. MLMCS employs independent standard MCS on various resolution levels and uses the differences as the control variables for the MCS at its most granular level, which, in mathematical terms, is given by Eq. 1.

$$E[Y_L] = E[Y_1] + \sum_{i=2}^L E[Y_i - Y_{i-1}] \text{ where } Y_l = G(X_l) \quad (1)$$

Based on Eq. 1, we intend to approximate $E[Y]$ where $Y = G(X)$ is functional of the random variable X . The traditional MCS approach requires a high computational complexity to attain the Root Mean Square Error (RMSE) of $O(\epsilon)$ in a biased context. A multilevel approach addresses this issue and reduces the computational complexity in the biased framework. The addition of the MLMCS over the MCS includes:

- Step 1: determine the number of levels;
- Step 2: define the range of possible input parameters for each level;
- Step 3: generate a set of random inputs for each level;
- Step 4: perform the simulation using the specific discretization for level;
- Step 5: compute the differences in results between successive levels;

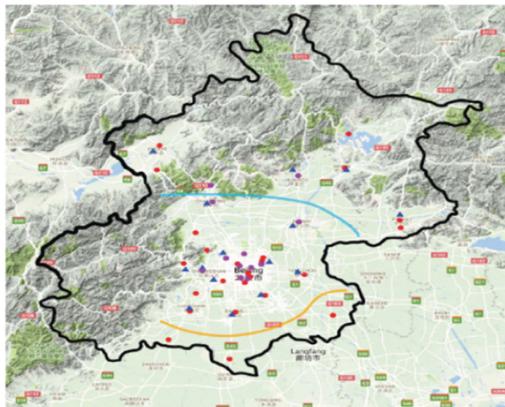
3.3 Air Pollution Dataset

The air pollution dataset comprises air quality attributes data collected hourly from 10 API monitoring locations in Beijing from March 2013 to February 2017, as shown in Table 1 and Fig. 1. The 24-h care center extracts the data from these public environmental locations [14]. The climatological data in the API site are coordinated with the adjacent climate place from the climatological management center.

Table 1. The characteristics of the air pollution dataset.

Characteristics	Particular
<i>Dataset type:</i>	<i>Multivariate, Time-Series</i>
<i>Number of Instances:</i>	420,768
<i>Area:</i>	Physical
<i>Number of Attributes:</i>	18
<i>Attribute Characteristics:</i>	Integer, Real
<i>Missing Values?</i>	Yes
<i>Associated Tasks:</i>	Regression
<i>The number of Web Hits:</i>	4246

The geographic positions of the air pollution dataset are shown in Fig. 1. The response API is classified into four categories: $\text{API} \leq 35 \mu\text{g m}^{-3}$ (green), $35 \mu\text{g m}^{-3} < \text{PM2.5} \leq 75 \mu\text{g m}^{-3}$ (yellow), $75 \mu\text{g m}^{-3} < \text{API} \leq 150 \mu\text{g m}^{-3}$ (orange) and $\text{API} > 150 \mu\text{g m}^{-3}$ (red). Each colored node contains four numbers showing the distribution of API categories at that branch layer. Additionally, the percentage displayed represents the sample's marginal proportion at that node.

**Fig. 1.** The geographic position of the China air pollution case study [14].

3.4 Performance Evaluation Metrics

In this study, the comparative study of statistical models is evaluated and compared from Four perspectives, which are Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Coefficient of Determination (R^2), and computational time. Statistically speaking, MAE gauges the difference between two continuous variables. Consider S

and H as corresponding observations representing similar phenomena; comparing H vs. S might include predicted versus actual values, measurements taken at different times, or results obtained using different measurement methods. The MAE is essentially the average distance, vertically and horizontally, between each point in the dataset and the line of identity (i.e., perfect agreement). The formula for MAE is shown in Eq. 2.

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_{i-x}| \quad (2)$$

RMSE determines the difference between sample values a model anticipates, as shown in Eq. 3. These variances are represented as residuals to approximate anticipating errors when calculated from the sample. RMSE accumulates the size of errors numerous times into a single prediction procedure. It measures accuracy in forecasting errors of different dataset models.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (Y_i - y_i)^2}{n}} \quad (3)$$

The R-squared (R^2) is a key statistical metric in regression models, representing the proportion of variance for a dependent variable that is explained by one or more independent variables. In simpler terms, R^2 measures how closely the data fits the regression model or how well the modeled data matches the observed data. It defines the ability of the prediction model to predict the independent variable. R^2 can be calculated by using Eq. 4.

$$R^2 = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}} \quad (4)$$

where n is the number in the given dataset, x is the first variable in the context (or observation data), and y is the second variable (or modeled data).

4 Results and Discussion

This section compares the results of implementing the proposed MLMCS model with the basic MCS. It starts with performing correlation analysis for the various features of the air pollution dataset.

4.1 Correlation Analysis

Correlation analysis measures the strength of the relationship between two continuous variables. This analysis could be between an independent and a dependent variable or amongst two independent variables. Figure 2 (a) to (e) highlights the correlation between API reading and all parameters that affect air quality in the performed study, which are Ozone (O_3), particulate matter $< 10\mu m$ (PM10), Nitrogen Dioxide (NO_2), Sulphur Oxides (SO_2), and one additional parameter particulate matter $< 2.5 \mu m$ (PM2.5).

In this case study that focused on predicting air quality, the correlation analysis reveals a strong relationship between the microscopic solid or liquid particles suspended in the air and the API. This finding underscores the need to monitor these particles closely. In general, particle matter less than 10 μm in diameter can get deep into the lungs and, in some cases, into the bloodstream. This issue implies that PM2.5, tiny particles or droplets in the air that are in microns or less in width, pose a greater health risk than PM10. In Beijing data, a significant association is found between ambient concentrations of PM2.5 (particulate matter with a diameter of less than 2.5 μm) and the risk of influenza-like illness (ILI) [15].

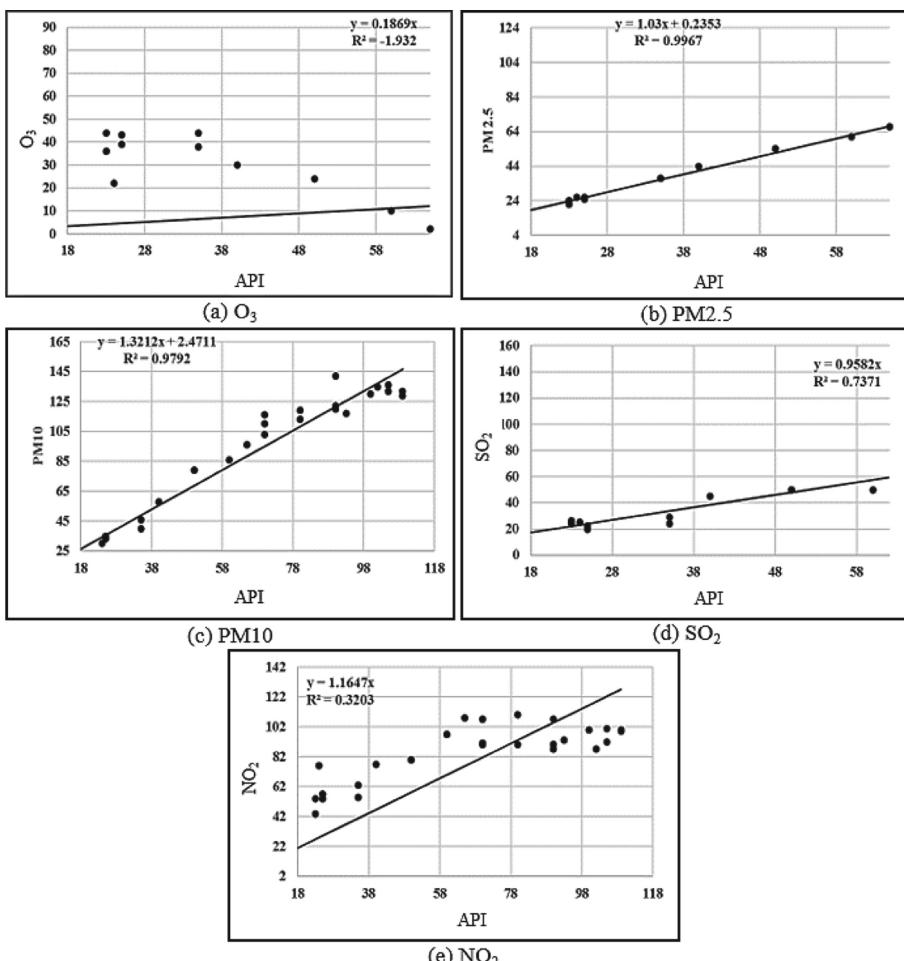


Fig. 2. Correlation between API and features.

The overall results of 1-day and 2-day prediction in advance for both the MCS and MLMCS are shown in Table 2. The best R^2 is achieved by the MLMCS model for

both 1-day and 2-day predictions, which indicates that the MLMCS model objectively increases the relative prediction accuracy. The MLMCS model has the highest accuracy of 86.45% compared to MCS, which has an accuracy of 82.90%. The most accurate predictions occur for the 1-day forecast in location 1 and the 2-day forecast in location 3. The error rates for the 1-day predictions are lower than those for the second day. This outcome aligns with the concept of error accumulation, which suggests that forecasting errors made one day in advance carry over and impact the accuracy of the following day's prediction. The MLMCS method outperforms the standard MCS for both 1-day and 2-day predictions. The prediction model significantly reduces the MLMCS time from 7.5 s to 3.43 s compared with the conventional MCS model. The results show that the MLMCS model obtained higher prediction results improvement on RMSE. For the MAE, the best prediction is achieved in 1-day by the MLMCS.

Table 2. The results of the MCS and MLMC models.

API 1-day advance prediction					MLMCS			
MCS					MLMCS			
Location	R ²	RMSE	MAE	Time	R ²	RMSE	MAE	Time
1	0.98	2.70	1.80	3.10	0.98	2.73	1.73	1.7
2	0.95	4.64	3.75	3.90	0.95	4.63	3.77	1.2
3	0.83	2.95	2.40	3.10	0.84	3.06	2.50	1.1
4	0.91	5.70	4.54	3.90	0.93	6.85	5.35	1.2
5	0.73	2.30	2.00	3.10	0.84	2.53	2.20	1.2
6	0.81	16.7	11.70	3.10	0.83	16.98	12.00	1.2
7	0.93	9.80	7.10	3.90	0.95	13.18	9.62	1.2
8	0.50	11.00	9.80	3.10	0.60	19.16	15.40	1.7
9	0.91	12.9	10.70	3.10	0.94	11.02	8.91	1.9
10	0.97	6.40	5.30	3.10	0.98	6.04	4.79	1.2
average	0.852	7.509	5.91	3.34	0.88	8.61	6.62	1.3
API 2-day advance prediction								
1	0.94	11.6	6.80	4.10	0.97	11.67	6.82	2.2
2	0.87	15.7	10.60	4.90	0.89	15.74	10.62	2.1
3	0.76	7.20	4.70	4.80	0.82	7.22	4.71	2.1
4	0.75	11.20	7.40	4.90	0.79	11.59	7.87	2.1
5	0.96	9.70	6.00	4.80	0.98	9.78	6.08	2.1
6	0.85	21.90	17.90	4.80	0.89	22.00	18.06	2.1
7	0.82	15.00	11.80	4.80	0.86	16.24	13.08	2.2
8	0.40	27.00	17.00	4.10	0.50	29.32	20.25	2.2
9	0.81	13.70	12.10	4.80	0.86	12.94	11.17	2.1
10	0.90	11.50	9.00	4.10	0.93	11.43	8.69	2.1
average	0.806	14.45	10.33	4.61	0.849	14.793	10.735	2.13

5 Conclusion

This work focuses on deploying an MCS model for API prediction of smart network air pollution attributes data in Beijing, China. It first visualizes the air pollution dataset's parameters and API reading. It is followed by correlation analysis to compute different effects between the locations of the study and the relationship between these locations using correlation analysis. Then, it performs the API prediction of MCS and Multilevel Monte Carlo Simulation (MLMC) models based on the air pollution dataset. It reports and discusses the experimental results for the MCS and MLMCS models and compares the results regarding prediction ability by using MSE, RMSE, R², and processing time. The results show that the MLMCS model produces better API prediction results than the MCS model. The MLMCS model produces an average accuracy of 86.45% and consumes a processing time of 3.43 s, while the conventional MCS model produces an accuracy of 82.90% and consumes a processing time of 7.5. Further research is needed to identify the integration of machine learning algorithms that can eventually improve the accuracy and efficiency of the models.

Acknowledgments. This research was supported by Universiti Tun Hussein Onn Malaysia.

References

1. Ly, H.B., et al.: Development of an AI model to measure traffic air pollution from multisensor and weather data. *Sensors* **19**(22), 4941 (2019)
2. Bakhtavar, E., Hosseini, S., Hewage, K., Sadiq, R.: Air pollution risk assessment using a hybrid fuzzy intelligent probability-based approach: mine blasting dust impacts. *Nat. Resour. Res.* **30**, 2607–2627 (2021)
3. Hamid Hassan, M., Mostafa, S.A., Baharum, Z., Mustapha, A., Saringat, M.Z., Afyenni, R.: A nested monte carlo simulation model for enhancing dynamic air pollution risk assessment. *Int. J. Inform. Visual.* **6**(4), 1–7 (2022)
4. Hassan, M.H., et al.: A new collaborative multi-agent Monte Carlo Simulation model for spatial correlation of air pollution global risk assessment. *Sustainability* **14**(1), 510 (2022)
5. Cabaneros, S.M., Hughes, B.: Methods used for handling and quantifying model uncertainty of artificial neural network models for air pollution forecasting. *Env. Modell. Softw.* **158**, 105529 (2022). <https://doi.org/10.1016/j.envsoft.2022.105529>
6. Hassan, M.H., Mostafa, S.A., Mustapha, A.: A statistical risk assessment method of dynamic environments: a case study of air pollution. *AUS J.* **26**, 1 (2020)
7. Tchórzewska-Cieślak, B., Pietrucha-Urbanik, K., Urbanik, M.: Analysis of the gas network failure and failure prediction using the Monte Carlo simulation method. *Eksplotacja i Niezawodność – Maintenance and Reliability* **18**(2), 254–259 (2016). <https://doi.org/10.17531/ein.2016.2.13>
8. Ren, L., He, L., Lu, H., Chen, Y.: Monte Carlo-based interval transformation analysis for multi-criteria decision analysis of groundwater management strategies under uncertain naphthalene concentrations and health risks. *J. Hydrol.* **539**, 468–477 (2016)
9. Heck, N., Smith, C., Hittinger, E.: A Monte Carlo approach to integrating uncertainty into the leveled cost of electricity. *Electr. J.* **29**(3), 21–30 (2016)
10. Akkar, S.: Probabilistic permanent fault displacement hazard via Monte Carlo simulation and its consideration for the probabilistic risk assessment of buried continuous steel pipelines. *Earthq. Eng. Struct. Dynam.* **46**(4), 605–620 (2017)

11. Gehl, P., Cavalieri, F., Franchin, P., Negulescu, C.: Robustness of a hybrid simulation-based/Bayesian approach for the risk assessment of a real-world road network. In: Proceedings of the 12th International Conference on Structural Safety and Reliability (2017)
12. Rezvani, S., Bahri, P.A., Urmee, T., Baverstock, G.F., Moore, A.D.: Techno-economic and reliability assessment of solar water heaters in Australia based on Monte Carlo analysis. *Renew. Energy* **105**, 774–785 (2017)
13. Rajasekhar, B., Nambi, I.M., Govindarajan, S.K.: Human health risk assessment of ground water contaminated with petroleum PAHs using Monte Carlo simulations: a case study of an Indian metropolitan city. *J. Environ. Manage.* **205**, 183–191 (2018)
14. Zhang, S., Guo, B., Dong, A., He, J., Xu, Z., Chen, S.X.: Cautionary tales on air-quality improvement in Beijing. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **473**, 20170457 (2017)
15. Feng, X., Li, Q., Zhu, Y., Hou, J., Jin, L., Wang, J.: Artificial neural networks forecasting of PM2. 5 pollution using air mass trajectory based geographic model and wavelet transformation. *Atmos. Environ.* **107**, 118–128 (2015)



An In-Depth Strategy using Deep Generative Adversarial Networks for Addressing the Cold Start in Movie Recommendation Systems

Muhammad Shahab¹, Yana Mazwin Mohamad Hassim^{1(✉)}, Rozaida Ghazali¹, Irfan Javid², and Nureize Arbaiy¹

¹ Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia

hi20005@student.uthm.edu.my, {yana, rozaida, nureize}@uthm.edu.my

² Department of Computer Science and IT, University of Poonch, Rawalakot, AJK, Pakistan
irfanjavid@upr.edu.pk

Abstract. This study addresses the intricate cold start challenge in movie recommendation systems (RSs) by integrating Collaborative Filtering (CF) and Singular Value Decomposition (SVD), complemented by Generative Adversarial Networks (GAN). The cold start problem arises when the system is unable to provide accurate recommendations for new users or items with limited interaction history. The addition of Content-Based (CB) filtering enhances outcome depth, while thorough data exploration and meticulous feature engineering, including extracting details like release years, enrich the dataset for a nuanced understanding of user preferences. The unique strength lies in the synergistic application of CF-SVD, GAN, and CB filtering, providing a multifaceted strategy to overcome the cold start challenge. Results, guided by strategic CB filtering, offer comprehensive insights, emphasizing the pivotal role of content in crafting accurate and personalized movie recommendations. The innovative integration establishes our approach as a robust solution, proficient in providing nuanced suggestions, especially in scenarios with limited user interaction data, thereby advancing the efficacy of movie RSs. GAN, a state-of-the-art framework, contributes significantly to our integrated RSs by leveraging a generator model to create synthetic data, refining CF outcomes. This contribution is evaluated using the MovieLens dataset, encompassing ratings and movie information.

Keywords: Recommendation System · Generative Adversarial Networks · Content Based Filtering · Collaborative Filtering · Scalability · Cold Start

1 Introduction

In the era of information overload on the internet, people often struggle to find what they are looking for. RSs step in to help by suggesting items based on user preferences, incorporating item features and user context. Their aim is to improve the quality of recommendation, making it easier for users to navigate the vast amount of data and simplify

decision-making [1]. RSs are pivotal in numerous sectors, including e-learning, retail, and sharing economy platforms, and are utilized by companies like Coursera, Amazon, Airbnb, and Netflix. They play a crucial role in helping users navigate the overwhelming amount of content on the internet and apps by automatically recommending the most suitable items based on their needs [2].

A variety of algorithms have been designed to elevate the effectiveness, efficiency, and precision of personalized recommendations for consumers by considering their preferences. These recommendation methodologies are then categorized based on the nature of collected data and their utilization in RSs, leading to distinctions such as content-based (CB), collaborative filtering (CF), and hybrid approaches (HA) [3]. HA-RSs leverage a combination of CB and CF algorithms to enhance the depth and accuracy of their RSs [4].

RSs aim to provide users with timely and suitable items. Ongoing research explores various methods to tackle challenges such as scalability and the Cold Start (CS) problem, which arises when dealing with new users or items lacking preference information [5]. The CS issue occurs when an RSs cannot predict ratings for new items in advance. It manifests when a new product is introduced, and there are no prior ratings, or when a new user joins without a rating history [6]. In addressing data sparsity and CS in RSs, the proposed solution is Deep Transfer Learning with Multimodal Embedding (DTLME), executed in two phases. The offline phase involves extracting features from images for new items, creating dense user and item feature vectors to handle user cold starts. Subsequently, in the online phase, these features are utilized to significantly enhance the effectiveness of personalized product recommendations [7].

The structure of our paper is as follows: Sect. 2, Author review related works to provide context and build upon existing knowledge. Section 3 details our proposed work, offering insights into its components. The experimental results and discussions follow in Sect. 4, shedding light on our findings. Lastly, Section 5 concludes the paper and outlines potential avenues for future research.

2 Related Works

The One significant challenge in RSs is the CS Problem. It involves accurately recommending items to new users or items with lower profiles due to insufficient information. Studies are currently investigating innovative machine learning strategies to tackle this obstacle and improve the overall performance of the system.

The GAN, introduced in 2014, is a deep neural network comprising a Generator and a Discriminator. It generates new samples resembling the training set and distinguishes between real and synthetic samples. Originally designed for unsupervised learning, GAN has applications in reinforced learning [8] by employing deep learning techniques using brain-inspired networks to understand big data and make predictions [9, 10].

In the realm of tourist recommendation enhancement, researchers have devised a pioneering approach utilizing GAN and a hybrid collaborative filtering strategy, integrating contextual data for personalized recommendations [11]. Addressing data imbalance within recommendation systems, another study introduces CWGAN-GP-PacGAN, a hybrid GAN-based method combining conditioning on the minority class with auxiliary classifier loss [12]. Authors have tackled the significant challenge posed by the

pure new user cold-start problem in recommender systems using a variety of innovative approaches. One such method involves harnessing Linked Open Data, collaborative features, and social network-based attributes to construct user profiles, resulting in the development of a hybrid recommendation system that seamlessly integrates ontology, Linked Open Data, and collaborative filtering techniques. This system has demonstrated notable effectiveness in addressing the cold-start issue [9].

The primary objective of this study is to solve the Cold Start Problem that is inherent in RSs. This goal is achieved through the integration of various methodologies, such as combining CF with SVD, incorporating GAN, and integrating CB approach. The overarching aim is to enhance the accuracy of recommendations for new or limited rated items by leveraging deep learning through GAN.

3 Research Methodology

In this section, the author undertakes a detailed data preparation process and proposes a novel framework for RSs as depicted in Fig. 1. The developed framework includes the CF utilizing SVD, the integration of CF-SVD with GANs, and lastly CB filtering.

3.1 Data Preparation

To begin, the author initiated the loading and examination of MovieLens datasets from Kaggle [15], focusing on movies and ratings. This study employed data visualization techniques, including histograms, to comprehend the distribution of ratings and identify popular movies and active users. Missing values were addressed through fundamental validation procedures. In the feature engineering process, the author extracted release years from movie titles to enhance the overall dataset. In the subsequent stage, the author partitioned the data into training (80%) and testing (20%) sets using, as shown in Table 1. This step is crucial for the training and assessment of models. The partitioning was achieved using 'scikit-learn,' for splitting datasets into subsets for training and testing purposes. The author saved the resulting subsets for potential use in future analyses, experiments, or evaluations.

Table 1. MovieLens Dataset.

Datasets	Size	Attribute Names
Movies	9742 rows and 3 columns	movieId, title and genres
Ratings	100836 rows and 4 columns	movieId, userId, rating and timestamp

3.2 Collaborative Filtering with Singular Value Decomposition (CF-SVD)

During this stage, which involves the implementation of CF using SVD, our initial step is to load preprocessed training data (T_d) containing user-item ratings shown in (Eq. 1)

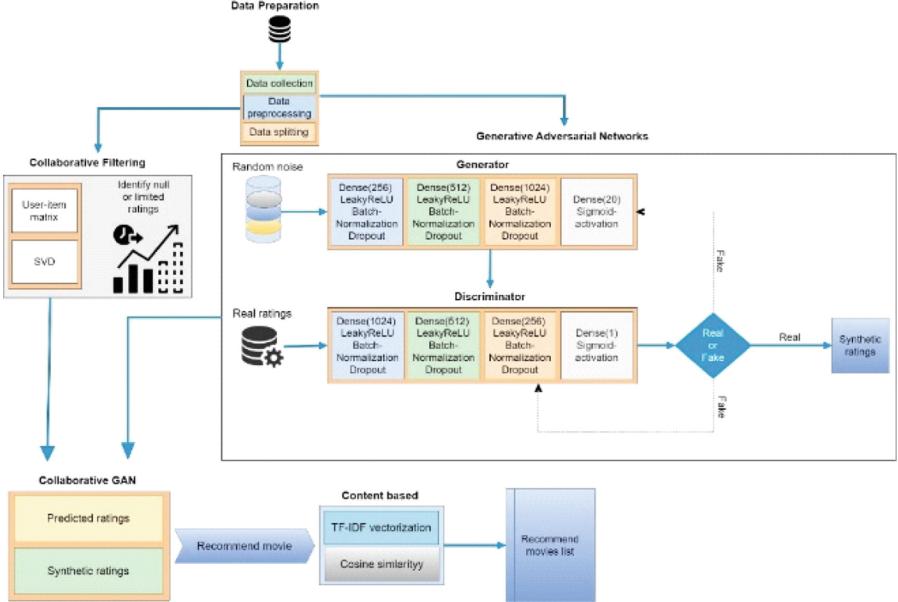


Fig. 1. The Proposed GAN framework

Subsequently, author imports the essential libraries required for the task. SVD proves instrumental in identifying latent factors (u , σ and vT) while constructing a user-item matrix. This matrix, in turn, facilitates the prediction of ratings for the test set, effectively tackling the CS issue. The utilization of this method results in accurate predictions of user preferences for items, thereby mitigating challenges commonly encountered in RSs.

$$Td = u \Sigma v^T \quad (1)$$

3.3 Generative Adversarial Networks (GANs)

The process of generating synthetic ratings begins by importing necessary libraries and loading preprocessed movies and ratings data. The training sets are then merged to create a user-item matrix. The GAN architecture is introduced, comprising a generator and a discriminator, both utilizing Leaky ReLU activation functions. The generator consists of four layers with 256, 512, and 1024 nodes, respectively, and a final layer with 20 nodes and Sigmoid activation to produce synthetic ratings based on random noise. Simultaneously, the discriminator consists of four layers with 1024, 512, 256, and 1 node(s), each layer incorporating Leaky ReLU activation, Dropout, and BatchNormalization, undergoing training utilizing the GAN loss function (Eq. 2) throughout the adversarial training process. Upon completion of training, the GANs are utilized to generate fake ratings, effectively addressing the cold start problem within the RSs.

$$\text{GAN loss} = -\log(d(\text{generated ratings})) - \log(1 - d(\text{real ratings})) \quad (2)$$

3.4 Collaborative Filtering (CF) with SVD and GANs

During this stage, Utilizing CF-SVD and GANs, this study aims to explore user preferences by analyzing patterns in their interactions with movies. Additionally, GANs play a crucial role in mitigating the Cold Start problem by generating ratings for movies that have limited or no existing ratings.

The integration of CF-SVD and GANs results in a holistic understanding of user preferences, by merging observations from actual user behavior with creatively generated synthetic ratings. This integrated approach ensures a more comprehensive and diverse set of movie recommendations. It caters to established user preferences while also suggesting movies that lack substantial historical ratings but have been creatively rated by GAN. This synergistic strategy enhances the effectiveness and adaptability of the RS.

3.5 Content Based Filtering (CB)

In the final stage, content-based (CB) features, specifically TF-IDF (Term Frequency-Inverse Document Frequency) and cosine similarity, serve as valuable augmentations to the collaborative filtering with singular value decomposition (CF-SVD) and generative adversarial networks (GANs) approaches within our comprehensive recommendation system (RS). While CF-SVD and GANs address the cold start problem by focusing on user interactions and generating synthetic ratings, CB, through TF-IDF and cosine similarity, considers intrinsic movie characteristics.

By incorporating CB with these features, our system gains the capability to recommend movies based on their content similarity to user usage patterns. This integration significantly enhances the recommendation experience, particularly for new items with limited interaction history. Leveraging the features of CF-SVD, GANs, and CB with TF-IDF and cosine similarity, this combined approach creates a robust RS that effectively considers both user behavior and movie content, thereby improving accuracy and coverage in recommendations.

4 Results

In this section, the study undertakes a thorough evaluation of our RSs model, examining critical facets of its performance. Essential performance metrics, including Precision, Recall, and F1 Score, are employed as standardized benchmarks for assessing the accuracy and efficacy of RSs. The respective mathematical expressions for these metrics are expressed as follows:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (3)$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (4)$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

After this, the author conducts a meticulous analysis of our proposed recommendation model using these metrics, and the outcomes are succinctly presented in Figs. 2 and 3. Remarkably, our model consistently surpasses existing methodologies, including (ColdGAN, Simple GAN, CoFiGAN) [13], CDAE [14] and MF [3], as evidenced by the lower values across Precision, Recall and F1 Score. These outcomes underscore the superior performance of our model in furnishing accurate and dependable recommendations.

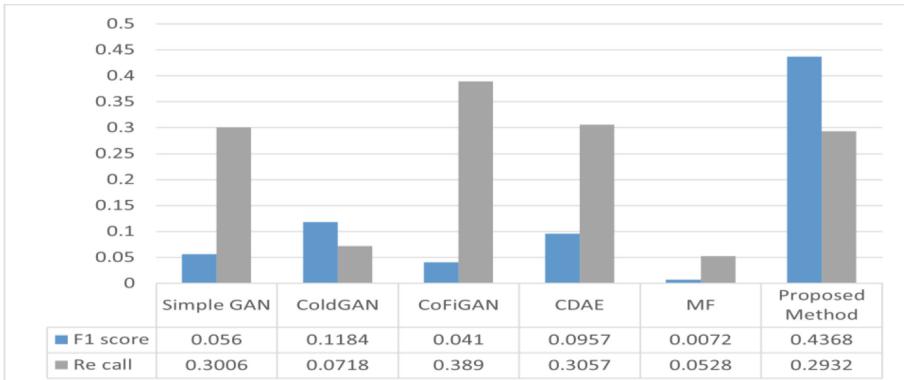


Fig. 2. Performance comparisons on MovieLens datasets

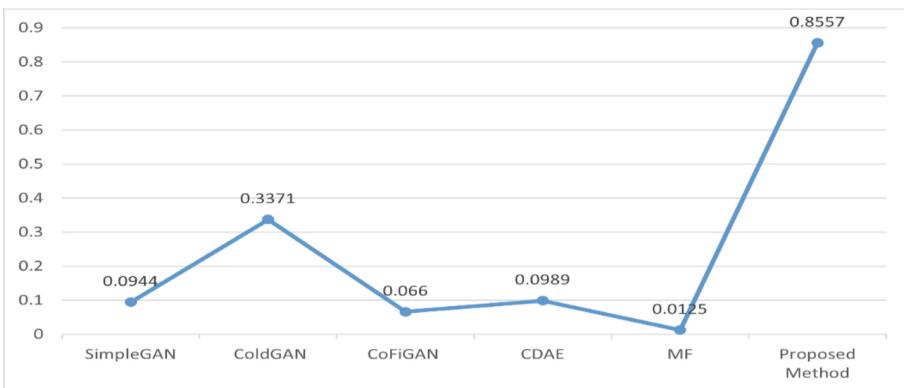


Fig. 3. Comparisons on precision using MovieLens datasets.

Furthermore, study delves into the intricacies of the training dynamics within the GAN architecture, with a specific focus on the losses incurred by the generator and discriminator components in Fig. 4. The generator loss, denoting the dissimilarity between generated and actual recommendations, exhibits a discernible declining trajectory, indicative of an enhanced capacity to generate high-quality recommendations. Conversely, the discriminator loss, measuring the effectiveness of distinguishing between real and synthetic recommendations, these encouraging developments highlight the GAN's

ability generating synthetic recommendations that possess both depth and variety. This contributes significantly to enhancing overall efficiency and elevating user satisfaction levels within the RSs.

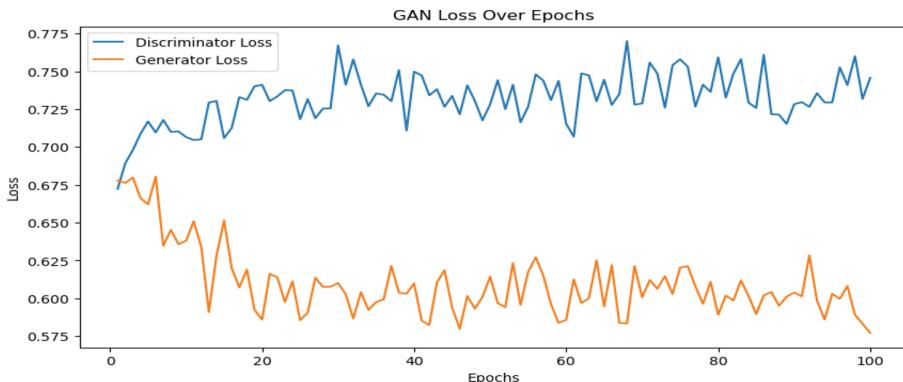


Fig. 4. Generator and discriminator losses

5 Conclusion

In conclusion, our study has addressed the challenging cold start issue in movie recommendation systems by integrating collaborative filtering (CF) with singular value decomposition (SVD) and incorporating generative adversarial networks (GAN), along with strategically using content-based (CB) filtering. Through thorough data exploration and careful feature engineering, such as extracting release years from movie titles, we gained a deeper understanding of user preferences. What makes our approach unique is how we combine CF, SVD, GAN, and CB filtering, making it robust enough to provide personalized suggestions even with limited user data.

Looking ahead, future research could explore hybrid models that blend different recommendation techniques to improve accuracy and personalization. There is also potential in incorporating time dynamics into recommendation models and creating specialized evaluation metrics for cold start scenarios. Expanding the range of data sources beyond standard metadata could further enhance recommendation systems. Additionally, a focus on understanding user behavior and feedback, while addressing ethical concerns, is crucial for the responsible advancement of recommendation systems.

Acknowledgments. This research was supported by the Ministry of Higher Education (MOHE) through the Fundamental Research Grant Scheme (FRGS/1/2020/ICT02/UTHM/02/1).

References

1. Ricci, F., Rokach, L., Shapira, B.: Recommender systems: techniques, applications, and challenges. In: Ricci, F., Rokach, L., Shapira, B. (eds.) *Recommender Systems Handbook*, pp. 1–35. Springer US, New York, NY (2022). https://doi.org/10.1007/978-1-0716-2197-4_1

2. Abdar, M., Yen, N.Y.: Analysis of user preference and expectation on shared economy platform: an examination of correlation between points of interest on Airbnb. *Comput. Hum. Behav.* **107**, 105730 (2020). <https://doi.org/10.1016/j.chb.2018.09.039>
3. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Comput. (Long Beach Calif.)* **42**(8), 30–37 (2009). <https://doi.org/10.1109/MC.2009.263>
4. Hussien, F.T.A., Rahma, A.M.S., Wahab, H.B.A.: Recommendation systems for e-commerce systems: an overview. *J. Phys.: Conf. Ser.* **897**(1), 12024 (2021). <https://doi.org/10.1088/1742-6596/1897/1/012024>
5. Yadav, U., Duhan, N., Bhatia, K.: Dealing with pure new user cold-start problem in recommendation system based on linked open data and social network features. *Mobile Inform. Syst.* **2020**, 1–20 (2020). <https://doi.org/10.1155/2020/8912065>
6. Jindal, H., Singh, S.K.: A Hybrid Recommendation System for Cold-Start Problem Using Online Commercial Dataset. 2014. <https://api.semanticscholar.org/CorpusID:212474568>
7. Jafri, S.I.H., Ghazali, R., Javid, I., Mahmood, Z., Hassan, A.A.A.: Deep transfer learning with multimodal embedding to tackle cold-start and sparsity issues in recommendation system. *PLOS ONE* **17**(8), e0273486 (2022). <https://doi.org/10.1371/journal.pone.0273486>
8. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., et al. Generative Adversarial Networks. Published online (2014)
9. Shafqat, W., Byun, Y.-C.: A hybrid gan-based approach to solve imbalanced data problem in recommendation systems. *IEEE Access* **10**, 11036–11047 (2022). <https://doi.org/10.1109/ACCESS.2022.3141776>
10. Waheed, W., Ghazali, R., Hussain, A.: Dynamic ridge polynomial neural network with Lyapunov function for time series forecasting. *Appl. Intell.* **48**(7), 1721–1738 (2017). <https://doi.org/10.1007/s10489-017-1036-7>
11. Saeed, W., Ghazali, R.: A novel error-output recurrent neural network model for time series forecasting. *Neural Comput. Appl.* **32**, 9621–9647 (2019). <https://api.semanticscholar.org/CorpusID:203071703>
12. Stephy, E.E., Rajeswari, M.: Empowering tourists with context-aware recommendations using GAN. In: 2023 Second International Conference on Electronics and Renewable Systems (ICEARS), pp. 1444–1449 (2023). <https://doi.org/10.1109/ICEARS56392.2023.10085604>.
13. Chen, C.C., Lai, P.L., Chen, C.Y.: ColdGAN: an effective cold-start recommendation system for new users based on generative adversarial networks. *Appl. Intell.* **53**(7), 8302–8317 (2022). <https://doi.org/10.1007/s10489-022-04005-1>
14. Wu, Y., DuBois, C., Zheng, A.X., Ester, M.: Collaborative denoising auto-encoders for Top-N recommender systems. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining. WSDM '16. Association for Computing Machinery; pp. 153–162 (2016). <https://doi.org/10.1145/2835776.2835837>.
15. Mishra, A.: (29 C.E., March 29). MovieLens. Kaggle. https://www.kaggle.com/datasets/ayu_shimishra2809/movielens-dataset. Retrieved 29 March 2020



Predicting Undergraduate Academic Success with Machine Learning Approaches

Yuan-Zheng Li¹, Keng-Hoong Ng^{2(✉)} , Kok-Chin Khor² , and Yu-Hsuen Lim¹

¹ School of Computing, Asia Pacific University of Technology and Innovation, Jalan Teknologi 5, Taman Teknologi Malaysia, 57000 Kuala Lumpur, Malaysia
`{tp074580, tp075445}@mail.apu.edu.my`

² Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, Sungai Long Campus, Jalan Sungai Long, Bandar Sungai Long, 43000 Kajang, Selangor, Malaysia
`{nkhoong, kckhor}@utar.edu.my`

Abstract. The opportunity to pursue tertiary education has increased in recent years, attributed to the initiatives and efforts made by governments, industry players, and educational institutions to make education more affordable and accessible. Hence, predicting undergraduate academic performance plays a crucial role in higher learning institutions' success. This is because the predicted outcomes could provide valuable insights and benefits such as early intervention and support for students at risk of academic struggles, graduating on time, personalized learning, etc. This study selected two machine learning algorithms, i.e., Light GBM and Random Forest, to predict undergraduate academic success in a higher-learning institution. The preliminary result indicates that Light GBM is the best performer, obtaining the highest accuracy of 79.3% after hyperparameter tuning. The result is marginally better than the recently published works that used identical or highly similar datasets.

Keywords: Academic Success · Random Forest · Student Performance · Student Dropout · Machine Learning

1 Introduction

The success or failure of a student's academic journey has implications that extend beyond the individual, impacting broader social, economic, and psychological landscapes [1, 2]. Higher education institutions worldwide have consistently shown interest in understanding student persistence toward achieving educational goals. However, student dropout remains a pressing issue globally, particularly in developing countries [3, 4]. Recent advancements in data science and analytics have enabled researchers to construct robust predictive models for investigating student retention [2, 5]. These models employ sophisticated data mining, machine learning, and recommendation systems to process information and predict students' performance, grades, or dropouts [6]. The timely identification of students at risk of dropping out and the implementation of appropriate intervention measures can reduce the wastage of valuable time and resources of the

The original version of the chapter has been revised. Author name has been corrected. A correction to this chapter can be found at

https://doi.org/10.1007/978-3-031-66965-1_42

faculty and the institution while optimizing the benefits for students, higher education institutions, and society [7]. Furthermore, due to the competitive nature of institutions within the education industry, analyzing student performance has become crucial to ensure graduation on time and reduce dropout rates that reflect the quality of the respective institution [8]. Educational data mining (EDM) is thus key to discovering, exploring, and analyzing hidden insights and knowledge in educational data [9].

Hence, this study aims to employ machine learning algorithms to predict the academic completion rates among undergraduate students. Specifically, we focus on key aspects such as (1) exploring the efficacy of various machine learning algorithms in predicting student's academic success and (2) optimizing the parameters of the predictive model to enhance its accuracy and reliability.

The remainder of the paper is structured as follows. The next section highlights the recent research literature related to the research topic. It is followed by the material and methods section, which adequately describes the experiment design and the flow of primary processes to arrive at the results discussed in the fourth section. The last section concludes the research findings and future enhancement.

2 Related Work

Yağcı [6] proposed a model for predicting final exam grades using limited parameters. These included the midterm exam grades of students, department, and faculty data. The researcher aimed to contribute to developing effective intervention plans that could improve students' academic performance. The research dataset comprised 1,854 instances, each representing an individual student enrolled in the Turkish language course at a Turkish state university. Results revealed that Random Forest (RF), Neural Network (NN), and Support Vector Machine (SVM) obtained good accuracy.

The study by [10] attempted to predict the CGPA of students upon graduation. The collected dataset was the scores of students under the Information System (IS) programs of Hawassa University. The dataset was cleaned and normalized for effective prediction. Root Mean Square Error (RMSE) was used as the evaluation metric. Results revealed that Support Vector Regression (SVR) and Linear Regression (LR) performed relatively better than the NN.

Research by Altabrawee et al. [11] aimed to identify students who require academic support to improve their grades. The four algorithms used were Decision Tree (DT), Naïve Bayes (NB), LR, and NN. Data were collected from two humanities departments at an Iraqi university. The dataset consists of 161 instances and 20 attributes; it was preprocessed to ensure normalization and proper labelling. Results indicated that ANN performed best with the highest accuracy and ROC index of 77.04% and 0.807, respectively. Tang et al. [2] conducted a comparative study using LR and RF models to predict student persistence. They used an undergraduate dataset from the Polytechnic Institute of Port Alegre in Portugal. RF model outperformed LR in both raw and SMOTE-enhanced data.

Research by [12] employed data mining techniques to identify students at risk of dropping out at a specific educational institution in Malaysia and the factors contributing to it. The dataset consists of 64 instances and 27 attributes comprising the target variable, gender, final CGPA, and grades for 24 courses. Data profiling was performed, and

visual exploration was provided to display the attribution distribution. Five classifiers, including KNN, DT, NN, LR, and RF, were adopted to predict student dropout rates. The outcome reveals that LR was most accurate at predicting dropout, obtaining the highest classification accuracy and AUC of 0.908 and 0.876, respectively.

Marbouti et al. [13] identified a gap in the current literature whereby existing models cannot address the complexities associated with all courses. To overcome the shortcoming, they utilized an ensemble model combining NB, SVM, and KNN after feature selection. The ensemble model outperformed the other models with 85% accuracy. Nevertheless, the study is constrained by the small volume of data.

Tan and Shao [14] centred on predicting dropout rates among students enrolled in e-learning programs. They developed machine learning models incorporating ANN, DT, and Bayesian Network (BN) algorithms. The dataset has 62,375 instances and was derived from China's largest online educational institution. All 26 attributes in the dataset are related to the individual characteristics of students and their respective academic performance. The issue of class imbalance was tackled by duplicating the minority dropout class. Results revealed that all models effectively predicted student dropout with high accuracy rates. Nevertheless, DT was the most effective overall at 94.63%.

3 Research Design and Methodology

3.1 Dataset Source

The dataset was collected from Portugal's Polytechnic Institute of Port Alegre, encompassing student records from the academic year 2008/2009 to 2018/2019 [15]. This comprehensive dataset covers multiple undergraduate degrees spanning diverse fields of study, including agronomy, design, education, nursing, journalism, management, etc. [7]. The dataset has 4,424 entries with 37 distinct attributes and is free from missing values [4]. The attributes fall under six main categories: demographic data, socio-economic data, macroeconomic data, academic data upon enrollment, academic data at the end of the first semester, and academic data at the end of the second semester. The details of each attribute are shown in Table 1.

Table 1. Data type for each attribute in the dataset.

Demographic Attributes	Data Type	Academic data upon enrollment	Data Type
Marital status	Discrete	Application mode	Discrete
Nationality	Discrete	Application order	Discrete
Displaced	Binary	Course	Discrete
Gender	Binary	Daytime/evening attendance	Binary
Age of enrollment	Discrete	Previous qualification	Discrete
International	Binary	Previous qualification (grade)	Numeric
		Admission grade	Numeric

(continued)

Table 1. (*continued*)

Demographic Attributes	Data Type	Academic data upon enrollment	Data Type
Socio-economic Attributes		Academic Data at the End of the First Semester	
Mother qualification	Discrete	Curricular units 1st sem (credited)	Discrete
Father qualification	Discrete	Curricular units 1st sem (enrolled)	Discrete
Mother occupation	Discrete	Curricular units 1st sem (evaluations)	Discrete
Father occupation	Discrete	Curricular units 1st sem (approved)	Discrete
Educational special needs	Binary	Curricular units 1st sem (grade)	Numeric
Debtor	Binary	Curricular units 1st sem (without evaluations)	Discrete
Tuition fees up to date	Binary		
Scholarship holder	Binary		
Macroeconomic Features		Academic Data at the End of the Second Semester	
Unemployment rate	Numeric	Curricular units 2nd sem (credited)	Discrete
Inflation rate	Numeric	Curricular units 2nd sem (enrolled)	Discrete
GDP	Numeric	Curricular units 2nd sem (evaluations)	Discrete
		Curricular units 2nd sem (approved)	Discrete
		Curricular units 2nd sem (grade)	Numeric
		Curricular units 2nd sem (without evaluations)	Discrete
Target	Categorical		

3.2 Exploratory Data Analysis

Upon conducting EDA, we examined the statistical properties of the dataset to gain insights into the distribution of student characteristics and potential factors related to academic success. Table 2 summarizes the statistical findings of several attributes in the dataset.

Table 2. Statistical finding of the attributes in the dataset.

Attribute(s)	Statistical Finding	Statistical Details
Marital status	The majority of students are single	Single – 3919, married – 379, divorced – 91, facto union – 25, legally separated – 6, widower – 4
Application mode	A wide distribution in the modes of application used by students. This might reflect the diverse backgrounds of students. Different application modes might relate to the student's academic preparedness and motivation	1 st phase general contingent – 1708, 2 nd phase general contingent – 872, over 23 years old – 785, change of course – 312, technological specialization diploma holders – 213, etc. (only showed the top five)
Previous Qualification	Majority of students enrolling to the institution via secondary education. It is followed by technological specialization course, basic education 3 rd cycle, and etc	Secondary education – 3717, technological specialization course – 219, basic education 3 rd cycle or equivalent – 162, higher education degree – 126, Other: 11year of schooling – 45, etc. (only showed the top five)

(*continued*)

Table 2. (*continued*)

Attribute(s)	Statistical Finding	Statistical Details
Curricular Unit Grade (1st sem & 2 nd sem)	The average grades (between 0 – 20) for the first and second semesters are similar, indicating a consistent academic performance within a year for students. However, the distribution and standard deviation of the grades warrants further observation to identify key factors that might influence academic success	Refer to Fig. 2 for the visualization of 1st and 2 nd semesters' average grade distributions
Target variable	The category of 'Graduate' is predominant, accounting for almost 50%. This suggests that most students have successfully completed their studies within the stipulated duration. Following this, the 'Dropout' category represents 32%, signifying a crucial segment for this research. Lastly, the 'Enrolled' category comprises 18%, indicating students who haven't concluded their studies by the end of the specified period but continue their education. This analysis shows that the study deals with an imbalanced dataset	Graduate – 2209 Dropout – 1421 Enrolled – 794

3.3 Data Preprocessing

Data preprocessing is crucial in building a successful predictive model, encompassing key processes like converting numerical features to categorical, outlier detection and handling, missing value handling, normalization, imbalanced data handling, etc. The dataset was curated and used in other research work earlier in this study.

We applied only outlier detection and normalization processes for the dataset. Interquartile range (IQR) [16] was utilized to identify possible outliers in the dataset. Outliers are observations that significantly deviate from the majority of the data. Their presence can have detrimental effects on the performance of machine learning algorithms. Data points below $Q1 - 3 \times IQR$ or above $Q3 + 3 \times IQR$ were considered outliers in the IQR method. In this study, IQR has identified three outliers in admission grade and previous qualification (grade), 718 outliers in curricular unit 1st sem (grade), and 870 outliers in curricular unit 2nd sem (grade).

Considering the nature of the detected outliers, appropriate treatments were performed. The three outliers detected in the admission grade and previous qualification (grade) were treated by replacing the outliers with the mean values of the features. On the other hand, outliers found in the curricular unit 1st sem (grade) and 2nd sem (grade) were untreated. This is because an investigation into these outliers revealed that all of them have a grade of '0', and this is likely because it may genuinely represent students who did not attend the exam or did not earn credits.

For this student dataset, considering the nature of the data and the requirements of our analytical methodologies, min-max normalization was applied. The primary rationale

for this decision is the presence of multiple continuous attributes in the dataset, such as ‘Previous qualification (grade)’ and ‘Admission grade’. Normalization ensures these attributes are transformed onto a uniform scale, making the dataset more amenable to machine learning models, especially those that rely on distance measures such as K-Nearest Neighbors and Support Vector Machines. Furthermore, this normalization technique offers a consistent scaling approach without excessively distorting the inherent relationships in the data.

3.4 Classification Algorithms

This study selected two classification algorithms, Random Forest (RF) and LightGBM, to build effective prediction models for undergraduate student academic success. The selection of the RF model was based on the following justifications: (1) **Robust to overfitting** – due to its ensemble nature and the randomness introduced during training, RF is less prone to overfitting, which is essential for imbalanced datasets commonly found in educational settings [2, 15]. (2) **High recall rate for minority class** – RF has shown high recall rates for minority classes, especially when a technique like SMOTE is employed [2]. (3) **Feature importance** – RF offers built-in methods for feature importance estimation, which can be invaluable for interpreting the factors contributing to student performance.

Besides the RF model, the LightGBM model is also considered a suitable candidate for this study and the following factors support it. Firstly, the model is designed to be highly efficient, making it suitable for large-scale educational datasets. In addition, LightGBM often outperforms standard methods like LR or SVM, particularly in complex and imbalanced datasets [15]. The last supporting factor is that it offers much flexibility in handling different types of data and optimization objectives, making it adaptable to the specific needs of educational data. In short, we could summarize that both models offer high performance, robustness, and interpretability. Furthermore, their effectiveness in handling imbalanced datasets and their feature importance estimation capabilities align with this study’s objectives and challenges.

4 Results and Discussion

LightGBM and Random Forest algorithms were trained and tested in this study with 10-fold cross-validation. The details of the model training, together with hyperparameter tuning and feature selection, are briefly described in this section. Firstly, both classifiers were initialized with a random state, with the stratified cross-validation applied to the dataset. This ensures that each fold has the same proportion of target labels as the complete dataset. Both classifiers were evaluated using the default parameters. The second evaluation involved both classifiers with the best hyperparameters. In the hyperparameter tuning, each classifier defined a hyperparameter grid, including the number of estimators, learning rate, number of leaves, and other relevant parameters. The *GridSearchCV* was applied to find the best hyperparameters for the classifier.

4.1 Evaluations of Classifiers Using Default Parameters

The prediction results obtained from the two machine learning models, i.e., LightGBM and RF, are shown in Table 3. The evaluation metrics included accuracy, precision, recall, Kappa statistic, MAE, and RMSE. Based on the results, both models exhibit similar performance concerning most metrics. The accuracies for the LightGBM and RF are approximately 78.6% and 78.7%, respectively. The Kappa statistics, a measure of the agreement between observed and expected accuracies, also showed comparable values for both models (LightGBM: 0.896, RF: 0.892). This suggests that both models are relatively consistent in their predictions.

Table 3. Performance Summary of Default LightGBM and RF Models.

Metric	LightGBM	RF
Accuracy	0.786	0.787
Kappa	0.896	0.892
Mean Absolute Error (MAE)	0.382	0.379
Root Mean Square Error (RMSE)	0.797	0.794

The accuracy obtained by each class in the dataset (Table 4) reveals that the “Graduate” class (majority class) showed the highest recall values in both models, suggesting that both LightGBM and RF are proficient in identifying students who will graduate [17]. The “Dropout” class has higher precision, indicating fewer false positives in this category. However, the “Enrolled” class demonstrated low precision and recall rates in both models, indicating room for improvement in identifying ongoing enrollments. In general, both models exhibit good overall accuracy. However, there is a discrepancy in performance across different classes. Both models excel in identifying “Graduate” and “Dropout” categories, but they were unable to perform well in the “Enrolled” category. This suggests a need for further tuning or model-specific adjustments to enhance their predictive accuracy.

4.2 Model Parameter Optimization by Hyperparameter Tuning

The selection of appropriate hyperparameters can significantly impact the performance of machine learning models. This study focused on optimizing hyperparameters for the LightGBM and RF models using the Grid Search, a computational method that systematically works through multiple combinations of parameter tunes to determine the best set for the given scoring function. After running the Grid Search, the optimal set of hyperparameters was obtained and recorded as follows.

Best parameters of LightGBM:

`'boosting_type': 'gbdt', 'colsample_bytree': 0.78, 'learning_rate': 0.13, 'min_child_samples': 33, 'n_estimators': 80, 'num_leaves': 25, 'subsample': 0.73`

Table 4. Comparison of Precision and Recall Rates of Each Target Class Obtained by LightGBM and RF Models.

Class	Metric	Hyperparameter Tuning	LightGBM		RF	
			Before	After	Before	After
Graduate	Precision Recall		0.812 0.930	0.820 0.928	0.800 0.941	0.803 0.937
Dropout	Precision Recall		0.836 0.778	0.841 0.787	0.843 0.784	0.851 0.789
Enrolled	Precision Recall		0.559 0.400	0.577 0.429	0.575 0.362	0.579 0.387

Best parameters of RF:

'bootstrap': False, 'max_depth': 20, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200

Both models' performance after tuning hyperparameters is summarized in Table 5, and they are compared with the best result from other similar works of literature. It could be observed that the tuning process led to marginal improvements in both models' performance. LightGBM slightly increased its prediction accuracy from 78.6% to 79.3%, whereas RF improved from 78.7% to 79.0%. The precision and recall rates for the Enrolled class have also been improved, particularly for the LightGBM with more than 2% of advancement (Table 4). The Kappa statistics remained relatively stable, indicating that the models maintained their robustness even after the tuning. To ensure fair and justifiable comparison, we only searched published works that used the same or highly similar dataset to ours for comparison (Table 5). The LightGBM and RF models performed slightly better than the accuracy results reported in [7] and [15].

Table 5. Performance Summary of LightGBM and RF Models After Hyperparameter Tuning.

Metric	After Hyperparameter Tuning		Martins et al., 2023 [7]	Martins et al., 2021 [15]
	LightGBM	RF	SVMSMOTE + RF	Extreme Gradient Boosting
Accuracy	0.793	0.790	0.748	0.730
Kappa	0.898	0.896	-	-
MAE	0.390	0.378	-	-
RMSE	0.807	0.794	-	-

5 Conclusion

The primary objective of this study is to develop a robust predictive model for student academic success. Two machine learning algorithms, LightGBM (LGB) and Random Forest (RF), were utilized and underwent a rigorous methodology involving 10-fold stratified cross-validation, and hyperparameter tuning. Both ML algorithms demonstrated comparable performance metrics, with an accuracy rate of more than 79%, and a Kappa statistic close to 0.9. The obtained results have strongly indicated that the models are consistent and reliable in their predictions. Furthermore, the achieved accuracy rates have outperformed several similar published works by a margin of 4%–5%. In conclusion, the study provides valuable insights into predicting student academic success, which could be beneficial to educational institutions for early intervention and resource allocation. Both LGB and RF algorithms have proven to be effective in this context, even though each algorithm has its limitations that need to be addressed for more accurate predictions. In the future work, we may consider to include other factors such as nutrition, health, and cultural data into this study.

References

1. Nouri, J., Larsson, K., Saqr, M.: Bachelor thesis analytics: using machine learning to predict dropout and identify performance factors. *Int. J. Learn. Analytics Artif. Intell. Educ.* **1**(1), 116–131 (2019)
2. Tang, Z., Chen, L., Jain, A. (2023). Exploring Individual Feature Importance in Student Persistence Prediction. *Journal of Higher Education Theory & Practice*, 23(6)
3. Mduma, N., Kalegele, K., Machuve, D.: A survey of machine learning approaches and techniques for student dropout prediction. *Data Sci. J.* (2019). <https://doi.org/10.5334/dsj-2019-014>
4. Realinho, V., Machado, J., Baptista, L., Martins, M.V.: Predicting student dropout and academic success. *Data* **7**(11), 146 (2022)
5. Rastrollo-Guerrero, J.L., Gómez-Pulido, J.A., Durán-Domínguez, A.: Analyzing and predicting students' performance by means of machine learning: a review. *Appl. Sci.* **10**(3), 1042 (2020)
6. Yağcı, M.: Educational data mining: prediction of students' academic performance using machine learning algorithms. *Smart Learn. Env.* **9**(1), 11 (2022)
7. Martins, M.V., Baptista, L., Machado, J., Realinho, V.: Multi-class phased prediction of academic performance and dropout in higher education. *Appl. Sci.* **13**(8), 4702 (2023)
8. Albreiki, B., Zaki, N., Alashwal, H.: A systematic literature review of student' performance prediction using machine learning techniques. *Educ. Sci.* **11**(9), 552 (2021)
9. Alyahyan, E., Düztegör, D.: Predicting academic success in higher education: literature review and best practices. *Int. J. Educ. Technol. High. Educ.* **17**, 1–21 (2020)
10. Obsie, E.Y., Adem, S.A.: Prediction of student academic performance using neural network, linear regression and support vector regression: a case study. *Int. J. Comput. Appl.* **180**(40), 39–47 (2018)
11. Altabrawee, H., Ali, O.A.J., Ajmi, S.Q.: Predicting students' performance using machine learning techniques. *J. Univ. Babylon Pure Appl. Sci.* **27**(1), 194–205 (2019)
12. Yaacob, W.W., Sobri, N.M., Nasir, S.M., Norshahidi, N.D., Husin, W.W.: Predicting student dropout in higher institution using data mining techniques. *J. Phys.: Conf. Ser.* **1496**(1), 012005 (2020)

13. Marbouti, F., Diefes-Dux, H.A., Madhavan, K.: Models for early prediction of at-risk students in a course using standards-based grading. *Comput. Educ.* **103**, 1–15 (2016)
14. Tan, M., Shao, P.: Prediction of student dropout in e-Learning program through the use of machine learning method. *Int. J. Emerg. Technol. Learn.* **10**(1), 11 (2015). <https://doi.org/10.3991/ijet.v10i1.4189>
15. Martins, M.V., Tolledo, D., Machado, J., Baptista, L.M., Realinho, V.: Early prediction of student's performance in higher education: a case study. *Trends Appl. Inform. Syst. Technol.* **1**(9), 166–175 (2021)
16. Barbato, G., Barini, E.M., Genta, G., Levi, R.: Features and performance of some outlier detection methods. *J. Appl. Stat.* **38**(10), 2133–2149 (2011)
17. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. *Inf. Process. Manage.* **45**(4), 427–437 (2009)



Comparative Assessment of Facial Expression Recognition Models for Unraveling Emotional Signals with Convolutional Neural Networks

Afia Zafar¹(✉), Nazri Mohd Nawi², Noushin Saba¹, Kainat Zafar¹, Mohsin Suleman¹, and Shahneer Zafar¹

¹ Department of Computer Science, National University of Technology, Islamabad, Pakistan
{afiazafar, kainatz}@nutech.edu.pk

² Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia

Abstract. Facial expressions are vital in conveying human emotions, forming a foundation of non-verbal communication. Recognizing these expressions computationally known as Facial Expression Recognition (FER) holds great significance in Artificial Intelligence (AI) research. This study focuses on employing Convolutional Neural Networks (CNNs) for FER classification using static images, omitting pre-processing or feature extraction. Our approach integrates pre-processing steps like face detection and illumination correction to bolster future accuracy. Through feature extraction, we pinpoint critical facial areas jaw, mouth, eyes, nose, and eyebrows enhancing computational modelling. Additionally, we survey existing literature to inform our CNN architecture, addressing challenges posed by components like max-pooling and dropout layers, culminating in improved performance. Our experiments attained a 78.9% test accuracy in the seven-class FER2013 dataset classification and 82.03% in FER plus dataset. This work not only advances technology but also lays the groundwork for further decoding the nuanced language of facial expressions.

Keywords: Facial Expression Recognition (FER) · Convolutional Neural Networks (CNNs) · Artificial Intelligence (AI) · Pre-Processing · Feature Extraction · Facial Action Coding Scheme (FACS)

1 Introduction

Facial expressions constitute a vital channel for human emotional expression serving as a fundamental medium in social communication [1]. The deciphering of these expressions has long intrigued researchers due to its significance in understanding human behavior, cognition, and interpersonal dynamics. In recent years, the integration of advanced computational techniques, particularly Convolutional Neural Networks (CNNs), has revolutionized the realm of Facial Expression Recognition (FER), offering promising avenues for decoding emotional cues [2]. The emergence of deep learning paradigms, notably CNNs, has redefined the landscape of FER, enabling more nuanced and accurate analysis

of facial expressions from static images or videos. These advancements have propelled the field forward, fostering applications across domains encompassing human-computer interaction, psychology, healthcare, and social robotics [3].

Facial Expression Recognition (FER) stands as a dynamic focal point within the realm of artificial intelligence, finding application across diverse domains such as surveillance, security and law enforcement, marketing and entertainment and social humanoid robots [4]. The capacity to automatically discern facial expressions holds substantial promise for various fields, including data analytics, psychological research, social gaming, and other realms encompassing human-computer interactions. The mission for enhanced accuracy, real-time processing, and generalizability remains a focal point for researchers, driving ongoing investigations into refining CNN-based FER models. Studies have revealed specific facial expressions hold universally recognized meanings [5]. In 1978, Ekman and Friesen introduced the Facial Action Coding System (FACS), identifying 6 facial expressions happiness, sadness, surprise, fear, anger, and disgust that seem to exceed cultural boundaries [6]. The challenge of classifying these emotions, including the addition of a seventh emotion, neutrality, is evident in research challenges like Kaggle's Facial Expression Recognition challenge. However, recognizing these common human expressions in natural conditions proves challenging for computers due to disparities in lighting and diverse head poses.

This paper contributes to this evolving landscape by presenting a comprehensive exploration of Facial Expression Recognition for unravelling emotional signals. Leveraging the advancements in CNNs and considering the latest insights available. This study pioneers a novel Convolutional Neural Network (CNN) architecture tailored for the precise classification of human facial expressions. By emphasizing from-scratch design principles, coupled with advanced pre-processing and feature extraction techniques, we aim to elevate the accuracy of emotion classification within the FER2013 dataset. This research marks a significant leap in decoding nuanced emotions encoded in facial images, reflecting the evolving synergy between deep learning architectures and refined pre-processing methodologies.

2 Related Work

Traditional methods for facial feature extraction, such as geometric features and texture features e.g., Local Binary Patterns (LBP), Facial Action Units (FAC), Local Directional Patterns (LDA), and Gabor wavelets have been employed [7]. In recent years the success of deep learning, particularly with architecture like Convolutional Neural Networks (CNNs) has led researchers to adopt this approach for automatic feature extraction and classification [8]. Therefore, significant efforts have been invested in developing deep neural network architectures that yield highly satisfactory results in the realm of recognition of facial expressions.

In a recent study showed by Kim et al. [9] the integration of both unregistered and registered versions of face images is highlighted as advantageous for training and testing in Facial Expression Recognition (FER). To mitigate the impact of registration errors on FER performance, the registration process is selectively applied based on the outcomes of facial landmark detection. The research demonstrates that deep networks can effectively

perform registration, and the incorporation of pose information from these networks yields a modest improvement of approximately 0.4% in FER performance.

Jin et al. [10] present a region-based CNN architecture that focuses on capturing discriminative features from specific facial regions demonstrating enhanced accuracy in facial expression recognition tasks. One limitation of this paper is the potential dependency on accurate facial region localization, as errors in this process could compromise the effectiveness of the region-based Convolutional Neural Network (CNN) architecture proposed for facial expression recognition.

Zhu et al. [11] proposes an attention-based CNN model for facial expression recognition, emphasizing the importance of selectively attending to informative facial regions, resulting in improved performance. As this proposed model improved performance but one factor to keep in mind is the heightened computational complexity of the attention-based CNN model, which may pose challenges for real-time applications and demand additional computational resources.

Liu et al. [12] proposed dual-stream CNN architecture presented here achieves state-of-the-art results in facial expression recognition nevertheless, it may require additional computational resources due to its dual-stream design. Nguyen et al. [13] explores the effectiveness of temporal CNNs for facial expression recognition, showing promising results in capturing temporal dynamics. However, the limitation includes potential challenges in handling long-range dependencies.

Zhao et al. [14] introduces an efficient and robust CNN model for facial expression recognition, demonstrating enhanced accuracy across diverse datasets. However, the model's sensitivity to variations in lighting conditions poses a limitation, suggesting the need for further robustness in real-world, variable lighting scenarios.

3 Dataset Description

The study evaluates its proposed method using two selected datasets named FER203[15] and FER plus [16] (Face Expression Recognition Plus dataset). FER2013 is a facial expression dataset consisting of over 35,000 labelled images categorized into seven emotional expressions, including happiness, sadness, surprise, fear, anger, disgust, and neutrality. The dataset was created for training and testing facial expression recognition models, providing a diverse set of images capturing various emotional states in natural conditions. It serves as a widely used benchmark for evaluating facial expression recognition algorithms and models. FER + (Face Expression Recognition Plus) is an extended version of the FER2013 dataset, comprising over 100,000 facial images annotated with seven emotional labels. With enhanced diversity and improved quality, FER + addresses some limitations of FER2013, making it a valuable resource for advancing facial expression recognition research and the evaluation of machine learning models in diverse real-world scenarios.

4 Methodology

In this section, we will delve into our CNN architecture and techniques aimed at boosting accuracy on the FER2013 and FER+ datasets. The work is centered on three key components: data pre-processing, feature extraction and the design of the CNN architecture. The proposed architecture is presented in Fig. 1.

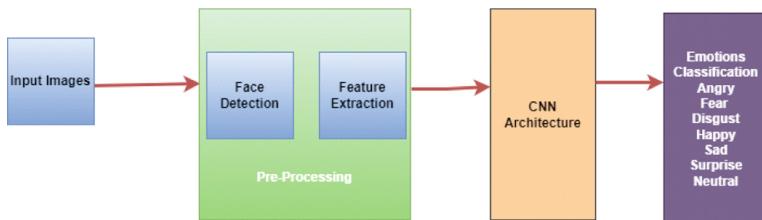


Fig. 1. The proposed design for facial expression recognition.

4.1 Pre-processing

Pre-processing is performed before feature extraction to ensure that input data is uniform, enhancing model efficiency and generalization. By standardizing dimensions, converting to grayscale, and normalizing pixel values, pre-processing optimizes the dataset for subsequent feature extraction, allowing the model to focus on relevant facial features without being hindered by variations in colour, size, or irrelevant background information.

In both the FER2013 and FER + datasets there is a meticulous approach to face registration, with an automated process that standardizes spatial requirements and centres the faces within the images. This uniformity facilitates reliable analysis and model training. Utilizing a Haar Cascade classifier, as depicted in Fig. 2, exemplifies the effective face detection mechanism employed in FER + to maintain consistency across the datasets.



Fig. 2. Depicts the stages of face detection, highlighted by the green square. The intermediary step involves applying illumination correction, while the subsequent image showcases the extraction of features, including the right eye, left eye, nose, and mouth, indicated by the use of an orange color.

4.2 Feature Extraction

Feature extraction after pre-processing in facial expression recognition with CNNs is crucial for reducing dimensionality, capturing relevant patterns, enhancing generalization, and ensuring robustness to variability, ultimately improving computational efficiency. We used the dlib facial landmark detector [17] and pre-trained on the iBUG 300-W dataset to extract features. The choice of the dlib facial landmark detector for feature extraction is based on its pre-trained model's effectiveness in accurately identifying key facial landmarks providing robust and reliable feature representations for facial expression recognition. The key facial components, comprising both eyebrows, eyes, nose, inner and outer outlines of the mouth, and the jaw, were extracted. In Fig. 3 the final image illustrates the extraction of the right and left eyes, nose, and inner and outer outlines of the mouth, highlighted in orange.

4.3 CNN Architecture

Convolutional Neural Networks (CNNs) have garnered extensive application across diverse computer vision domains, prominently in the realm of Facial Expression Recognition (FER). In the early 21st century, a thorough exploration of FER literature revealed the efficacy of CNNs in handling alterations in facial location and scale variations. Notably, these studies demonstrated that CNNs outperformed multilayer perceptron (MLP) models, particularly when confronted with previously unseen face pose variations.

Researchers strategically employed CNNs to address a spectrum of challenges in facial expression recognition, encompassing translation, rotation, subject independence, and scale invariance. The versatility of CNNs in navigating these complexities underscores their significance in advancing the capabilities of FER systems. This not only underscores the effectiveness of CNNs in handling nuanced facial features but also highlights their superiority over alternative methodologies, contributing to the continual evolution of facial expression recognition technology. Our model was trained by incorporating the following features:

- i. A CNN architecture featuring five convolutional layers has been employed, each using the “RELU” activation function. The final output layer is equipped with the SoftMax activation function. The filter specifications for each convolutional layer are as follows: 32 for the first layer, 64 for the second layer, 128 for the third and fourth layers, and 64 for the fifth layer.
- ii. The architecture incorporates four max-pooling layers with a stride of (3*3) and a pool size of (2*2). Max pooling was applied after the initial two convolutional layers, followed by an additional layer after the third convolutional layer, yet another after the fourth convolutional layer, and finally, the fourth pooling layer was applied after the fifth convolutional layer.
- iii. Five dropout layers have been integrated into the model, each with dropout rates of 0.3, 0.4, and 0.5.
- iv. The model architecture includes one flattened layer and one dense layer with 128 units.
- v. The total parameters and trainable parameters amount to 2 million each.

- vi. A split of 75% training and 25% testing data with FER 203 and FER plus datasets to facilitate accurate evaluation.

The detail of the structure is presented in Fig. 3 for a more in-depth understanding.

Layer (type)	Output Shape	Param #
<hr/>		
conv2d (Conv2D)	(None, 46, 46, 32)	320
conv2d_1 (Conv2D)	(None, 44, 44, 64)	18496
max_pooling2d (MaxPooling2D)	(None, 22, 22, 64)	0
dropout (Dropout)	(None, 22, 22, 64)	0
conv2d_2 (Conv2D)	(None, 20, 20, 128)	73856
max_pooling2d_1 (MaxPooling2D)	(None, 10, 10, 128)	0
dropout_1 (Dropout)	(None, 10, 10, 128)	0
conv2d_3 (Conv2D)	(None, 8, 8, 128)	147584
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 128)	0
dropout_2 (Dropout)	(None, 4, 4, 128)	0
conv2d_4 (Conv2D)	(None, 2, 2, 64)	73792
max_pooling2d_3 (MaxPooling2D)	(None, 1, 1, 64)	0
dropout_3 (Dropout)	(None, 1, 1, 64)	0
flatten (Flatten)	(None, 64)	0
dense (Dense)	(None, 128)	8320
dropout_4 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 7)	903
<hr/>		
Total params: 323271 (1.23 MB)		
Trainable params: 323271 (1.23 MB)		
Non-trainable params: 0 (0.00 Byte)		

Fig. 3. Demonstration of CNN architecture

5 Results

The proposed study used a five-layer convolutional neural network (CNN) architecture, achieving a competitive FER2013 test accuracy of 78.9% and 80.23% in FER plus dataset respectively. Optimal results were obtained with a batch size of 32 and 50 epochs, highlighting the effectiveness of the proposed CNN architecture and strategic parameter choices in enhancing emotion recognition accuracy. The confusion matrix illustrates the classification performance, with rows indicating true labels and columns representing predicted labels. Our model accurately predicted 8 angry, 30 sad and 17 fears respectively

of them. The confusion matrix and ROC curve of FER2013 is represented in Figs. 4 and 5 respectively.

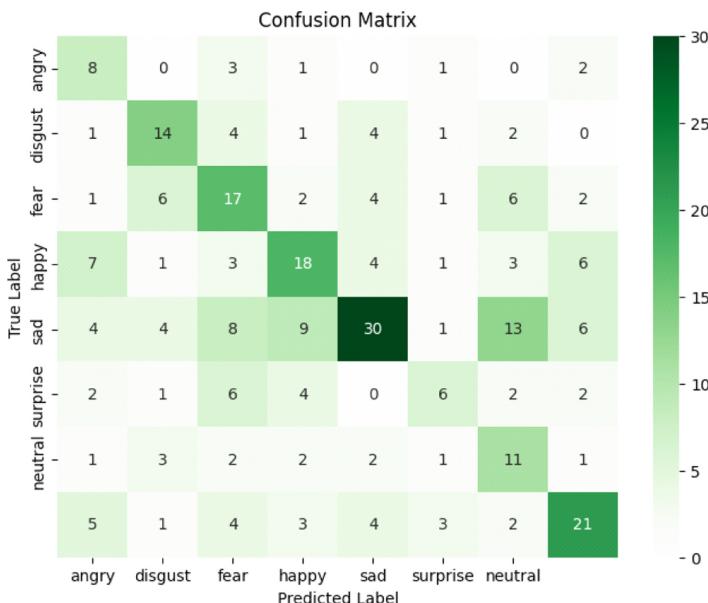


Fig. 4. Confusion matrix on the FER 2013 dataset.

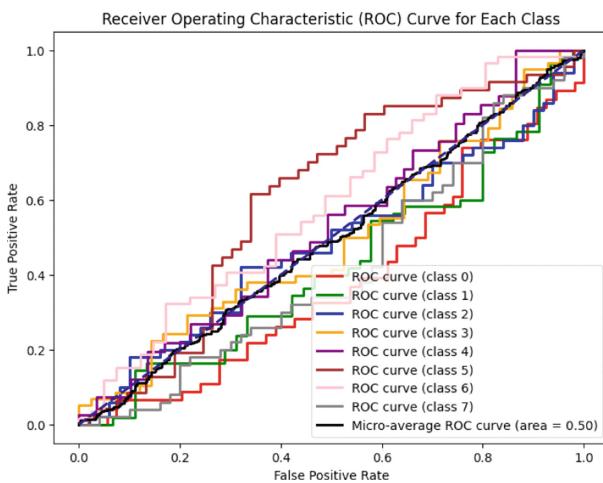


Fig. 5. ROC curve on the FER 2013 dataset

Figures 6 and 7 representing confusion matrix and ROC curve for FER plus data respectively.

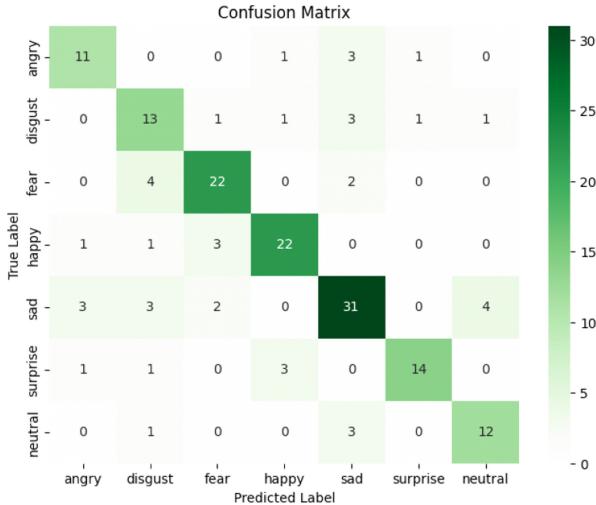


Fig. 6. Confusion matrix on the FER plus dataset

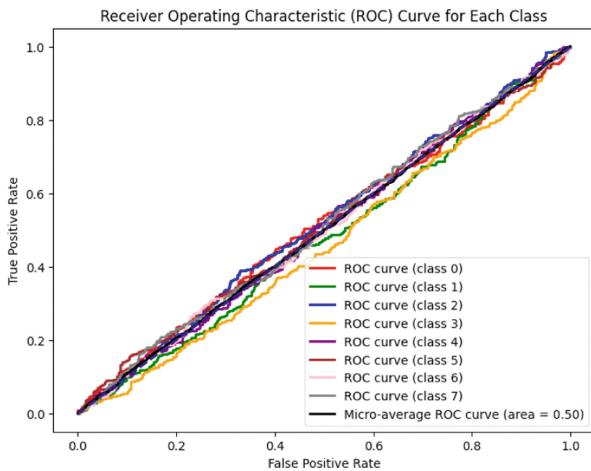


Fig. 7. ROC curve on the FER plus dataset

Upon analysing the confusion matrix, it became evident that the emotions of fear and sadness were the most prone to misclassification, with accuracies of 41.65% and 48.46%, respectively. In Fig. 8, misclassified images of FER2013 datasets are presented alongside their actual and predicted labels. For instance, the third image serves as an illustration, originally labelled as ‘sad.’ However, the model assigned probabilities of 39%, 46%, and 18% for predictions of fear, sadness, and anger, respectively. Additionally, there were marginal accuracies of 2.5% collectively for predictions of disgust, surprise, and neutrality.



Fig. 8. Presentation of misclassified images in the FER2013 test dataset.

In Fig. 9, misclassified images of FER plus datasets are presented alongside their actual and predicted labels. For instance, the fourth image serves as an illustration, originally labelled as ‘surprise.’ However, the model assigned probabilities of 25%, 34%, and 20% for predictions of fear, sadness, and anger, respectively. Additionally, there were marginal accuracies of 4% collectively for predictions of contempt, surprise, and neutrality.

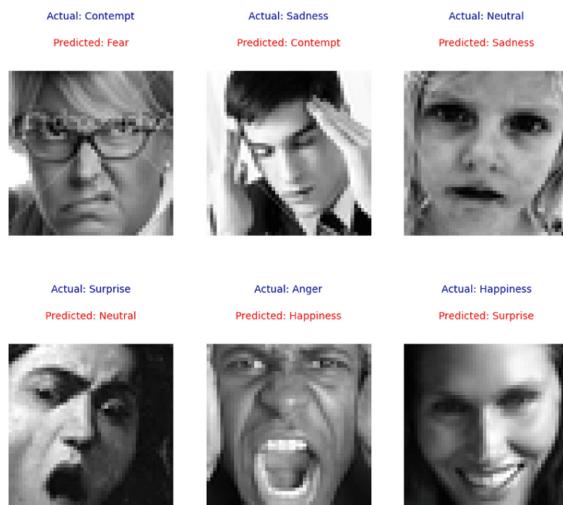


Fig. 9. Presentation of misclassified images in the FER plus test dataset.

6 Discussions and Future Work

The proposed study has been evaluated on two standard datasets named as FER2013 and FER plus. The proposed CNN consisted a five layer of architecture and obtained 78.09% and 80.23% accuracy respectively. Our research on Facial Expression Recognition using Convolutional Neural Networks (CNNs) employed a heuristic approach for optimal network architecture selection. Moving forward, we recognize the imperative task of refining our network design, exploring innovative architectures beyond the heuristic method employed. Our future endeavours will extend to the addressing the over fitting challenge through data augmentation techniques. Moreover limited dataset diversity and cross-cultural variations challenge model generalization, while real-time processing constraints hinder efficient deployment in practical applications. We envision exploring novel regularization techniques and leveraging transfer learning to extract richer facial features. By embracing these advancements, our aim is to not only elevate test accuracy but also contribute to the ongoing evolution of state-of-the-art methods in facial expression recognition. Moreover, in future to enhance network architecture, integrating CNN with optimization algorithms like NSGA promises to yield optimized hyper parameter values, ensuring superior performance in Facial Expression Recognition.

References

1. Cherbonnier, A., Michinov, N.: The recognition of emotions beyond facial expressions: comparing emoticons specifically designed to convey basic emotions with other modes of expression. *Comput. Hum. Behav.* **118**, 106689 (2021)
2. Ghazouani, H.: Challenges and emerging trends for machine reading of the mind from facial expressions. *SN Computer Science* **5**(1), 103 (2023)
3. Ozmen Garibay, O., et al.: Six human-centered artificial intelligence grand challenges. *Int. J. Human-Comput. Interact.* **39**(3), 391–437 (2023)
4. Al Qassab, I.L.A: Applying a facial emotion prediction approach based on algorithms of artificial intelligence (Master's thesis, Altınbaş Üniversitesi/Lisansüstü Eğitim Enstitüsü) (2023)
5. Cowen, A.S., Keltner, D., Schroff, F., Jou, B., Adam, H., Prasad, G.: Sixteen facial expressions occur in similar contexts worldwide. *Nature* **589**(7841), 251–257 (2021)
6. Clark, E.A., et al.: The facial action coding system for characterization of human affective response to consumer product-based stimuli: a systematic review. *Front. Psychol.* **11**, 920 (2020)
7. Choudhary, D., Shukla, J.: Feature extraction and feature selection for emotion recognition using facial expression. In: 2020 IEEE sixth international conference on multimedia big data (BigMM), pp. 125–133. IEEE (2020)
8. Singh, N., Sabrol, H.: Convolutional neural networks—an extensive arena of deep learning. a comprehensive study. *Arch. Comput. Methods Eng.* **28**(7), 4755–4780 (2021). <https://doi.org/10.1007/s11831-021-09551-4>
9. Kim, B.K., Dong, S.Y., Roh, J., Kim, G., Lee, S.Y.: Fusing aligned and non-aligned face information for automatic affect recognition in the wild: a deep learning approach. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 48–57 (2019)
10. Jin, X., Jin, Z.: MiniExpNet: a small and effective facial expression recognition network based on facial local regions. *Neurocomputing* **462**, 353–364 (2021)

11. Zhu, K., Du, Z., Li, W., Huang, D., Wang, Y., Chen, L.: Discriminative attention-based convolutional neural network for 3D facial expression recognition. In: 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), pp. 1–8. IEEE (2019)
12. Liu, Y., Zhang, X., Zhou, J., Fu, L.: SG-DSN: A semantic graph-based dual-stream network for facial expression recognition. Neurocomputing **462**, 320–330 (2021)
13. Nguyen, H.D., Kim, S.H., Lee, G.S., Yang, H.J., Na, I.S., Kim, S.H.: Facial expression recognition using a temporal ensemble of multi-level convolutional neural networks. IEEE Trans. Affect. Comput. **13**(1), 226–237 (2019)
14. Zhao, Z., Liu, Q., Zhou, F.: Robust lightweight facial expression recognition network with label distribution training. Proc. AAAI Conf. Artif. Intell. **35**(4), 3510–3519 (2021). <https://doi.org/10.1609/aaai.v35i4.16465>
15. Kusuma, G.P., Jonathan, J., Lim, A.P.: Emotion recognition on fer-2013 face images using fine-tuned vgg-16. Adv. Sci., Technol. Eng. Syst. J. **5**(6), 315–322 (2020)
16. Amal, V.S., Suresh, S., Deepa, G.: Real-time emotion recognition from facial expressions using convolutional neural network with Fer2013 dataset. In: Karuppusamy, P., Perikos, I., García Márquez, F.P. (eds.) Ubiquitous Intelligent Systems. SIST, vol. 243, pp. 541–551. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-3675-2_41
17. Keong, J., Dong, X., Jin, Z., Mallat, K., Dugelay, J.L.: Multi-spectral facial landmark detection. In: 2020 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6. IEEE (2020)



Evaluating Path-Finding Algorithms for Real-Time Route Recommendation System Built using FreeRTOS

Jun-Yen Liew¹, Keng-Hoong Ng² , Kok-Chin Khor² , and Kai-Yau Tee¹

¹ School of Computing, Asia Pacific University of Technology and Innovation, Taman Teknologi Malaysia, Jalan Teknologi 5, 57000 Kuala Lumpur, Malaysia
`{tp064175, tp059063}@mail.apu.edu.my`

² Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, Sungai Long Campus, Jalan Sungai Long, 43000 Bandar Sungai LongKajang, Selangor, Malaysia
`{nkhoong, kckhor}@utar.edu.my`

Abstract. The severity of traffic congestion in Klang Valley, Malaysia, has recently worsened by the rising number of private automobiles and commercial vehicles. In this study, we developed a real-time route recommendation system using FreeRTOS and simulated an environment that included 44 shopping malls in Klang Valley. We evaluated the system using three shortest-path finding algorithms: the standard Dijkstra's, Dijkstra's with binary heap, and A*. The results indicated that the standard Dijkstra's algorithm achieves the best performance as it can handle up to a million queries with an average response time of between 0.02 to 0.03 milliseconds per query. This research demonstrates the efficacy of the standard Dijkstra's algorithm in processing massive amounts of user queries that fulfil the real-time requirements.

Keywords: Route Recommendation · Real-Time · Shortest Path · Dijkstra · Binary Heap · A*

1 Introduction

Technology is rapidly progressing in this modern globalisation, and the worldwide population is experiencing exponential growth. As the population increases, the number of vehicles also rises. As a result, the traffic congestion problem has been intensified in Klang Valley, Malaysia [1]. This problem has consequences that will impact many urban motorists daily lives as well as their overall well-being. These include direct increased of travel time, fuel consumption and cost. The prolonged period of traffic congestion is also one of the main causes to the environmental pollution (especially urban areas), motorist's mental health (stress and frustration), and reduced productivity (late arrivals at work). One of the solutions is the utilisation of route recommendation applications. Such applications are widely used nowadays, particularly in a big city like Kuala Lumpur.

A recent study [2] on the mobile travel apps (MTAs) which analysed 193 Apple apps and 250 Google apps, and finally selected 36 applications that met the specific

criteria for evaluation purposes. The study reveals that MTAs are widely used and also accepted in the tourism industry. Real-time personalization or recommendation is one of the major features that preferred to be included in MTAs. The study also highlights several challenges of developing MTAs with smart features, they included significant technology investment cost, location accuracy and privacy concerns. Hence, a mobile travel or traffic application that comes with real-time feature and reasonable technology cost is highly desired nowadays.

In the context of increasing global urbanization and the corresponding rise in traffic congestion, particularly in the Klang Valley, Malaysia. The importance of route recommendation applications is becoming more pronounced. These apps, popular in urban centres like Kuala Lumpur have faced challenges such as the accuracy of the route recommended. Recent research [3] suggests integrating user preferences with geographical data and using the historical travel patterns for Points of Interest (POIs) interest determination to improve route suggestions when navigating between shopping malls. This strategy aims to ease transportation concerns by providing more efficient travel options, indirectly increasing user satisfaction as well as reducing the current urban congestion.

Route recommendation applications may fail to suggest optimal routes due to the lack of real-time data [4]. Thus, this study aims to develop a simulated real-time route recommendation system using FreeRTOS, which is a secure, free and light-weight real-time operating system. The system shall recommend the best routes between the 44 shopping malls in Klang Valley for urban motorists. This paper is organised as follows. Section 2 reviews the related research to such systems. The methodology of the study is then presented in Sect. 3. In Sect. 4, we discuss the evaluation results of the system along with the findings. Finally, Sect. 5 concludes the study.

2 Related Work

A study by [5] developed a travel route recommendation system for tourists based on their time, distance, and popularity preferences. The researchers utilised the Wi-Fi routers at 149 unique locations in Jeju Island to capture the tourists' movement patterns. The data structure of the adjacency matrix was utilised to store the distance transition and the connections between the locations. Besides that, an optimisation objective function was applied to recommend the optimal route to tourists. The generated optimal routes are varied based on user preference, weather, traffic conditions, etc. The Haversine formula was also applied to determine the distance between the two locations.

The researchers in [6] conducted a comparative study on the efficiency of Dijkstra, A*, Contraction Hierarchies (CH), and Floyd-Warshall. An open-source microscopic road simulator (SUMO) was utilised to simulate the urban mobility traffic systems, focusing on a 10 km map of Los Angeles. The algorithms were experimented with using the parameters of vehicles, including acceleration, deceleration, minimum gap, length of vehicle, emission class, and maximum speed, and were evaluated using average route length, average travel time, average waiting time, and average speed of vehicles. The results showed that the CH algorithm performed the best in average route length, travel time, and vehicle speed. A* achieved the best average waiting time. However, the implementation of CH cannot be applied to the real world. This is because its concept

of creating new “shortcuts” does not apply to real-world path-finding as the real roads and paths cannot be changed dynamically.

Wang et al. [7] utilised the popular A* algorithm together with the hyperparameter tuning in its heuristic function’s weights to come up with a better algorithm named “Improved A*”. It is then compared with A* and Dijkstra through an 80×80 grid simulation map and evaluated using the number of nodes traversed, path cost, and path length. The improved A* algorithm was found to perform better than the other two algorithms. However, factors important for path planning, including the path’s congestion, tourism emergencies, and tourists’ moving speed, were excluded. Thus, the research may not provide a realistic solution.

Ravish et al. [8] proposed an intelligent route recommendation system that considers travelling time, flexibility, cost, and traffic intensity. They used a predetermined road network graph in Bengaluru city. Two algorithms were evaluated: State-Action-Reward-State-Action (SARSA) and Dynamic Programming (DP). The findings indicated that DP is effective when the transition probabilities of the road network are known, as it uses that to find the optimal path. The SARSA algorithm used an epsilon-greedy method to find an optimal route and may recommend sub-optimal routes.

Research by [9] addressed the inefficiency of taxi routing systems in urban areas by proposing a new hybrid method comprising Artificial Potential Field (APF) and Dijkstra’s algorithms. APF removes unnecessary network nodes that are unlikely to be part of the recommended route and is followed by Dijkstra’s algorithm to recommend the optimal route. This method improves the route recommendation, ranging from 1% to 55% compared to other algorithms. This method also has a computation speed up to $28\times$ faster than the traditional Dijkstra algorithm in simulated networks and $9\times$ faster than real-world road networks.

Wang et al. [10] explored the application of personalised route recommendations in Cooperation Vehicle-Infrastructure Systems (CVIS) and its impact on network traffic flow. They proposed a novel recommendation system based on Pearson’s correlation coefficient and four identified key travel factors, i.e., distance, grade, time, and toll. A search algorithm was designed to compute all feasible routes from the origin to the destination by considering both time and space complexity.

In summary, various algorithms have been adopted for route recommendation and path planning. We investigated algorithms that balance computational efficiency by considering static factors such as distances and the dynamic factors of traffic and weather conditions. The combination of algorithmic robustness with these factors is advantageous to our study.

3 Research Design and Methodology

3.1 Adjacency Matrix

We chose the weighted directed adjacency matrix [11] to store the data of shopping malls (nodes) and routes between shopping malls (edges) because it could easily accommodate weighted dense graphs, allowing algorithms to have computational efficiency in checking the connectivity of the graph within $O(1)$ constant-time. Several adjacency matrices will be created to represent the different dimensions of the graph, including the distance

matrix, average speed matrix, and average travelling time matrix. The distance matrix represents the direct distance between each pair of nodes in the graph. The average speed matrix shows the average travel speed between a node pair, and the average travelling time represents the average time taken to travel between a pair of nodes.

Distance Adjacency Matrix

The distance adjacency matrix represents the connections and distances between each pair of the 44 shopping malls in Klang Valley of Malaysia. The distances were manually collected through Google Maps. Fig.1 shows the distance matrix table that represents the existence of connections between a pair of nodes and is stored asymmetrically. For instance, the values of an edge from node AMTM to BSC and BSC to AMTM reflect the asymmetrical nature of the graph, representing the distances of 13 km and 11.2 km.

	AEON Mall Taman Maluri (AMTM)	Aurora Place Bukit Jalil (APBJ)	Avenue K Shopping Mall (AKSM)	Bangsar Shopping Centre (BSC)	Berjaya Times Square (BTS)	The Weld Menara Weld (TW)
AEON Mall Taman Maluri (AMTM)	INF	14.5	7.6	13	4
Aurora Place Bukit Jalil (APBJ)	14.6	INF	17.4	13.6	14.4
Avenue K Shopping Mall (AKSM)	7.3	22.1	INF	9.6	INF
Bangsar Shopping Centre (BSC)	11.2	13.1	INF	INF	INF
Berjaya Times Square (BTS)	3.4	INF	INF	9.1	INF
.....
The Weld Menara Weld (TW)	1.6	7.6	3.3	9	INF	INF

Fig. 1. Distance adjacency matrix of 44 x 44 shopping malls. The “INF” value represents unreachable or no edge between the pair of nodes.

Average Travelling Speed Adjacency Matrix

The average travelling speed adjacency matrix is mainly implemented to represent the average travelling speed between each pair of nodes. Each cell in this average travelling speed matrix will hold the structure of the “speedRoute_t”, which comprises three pieces of data, including the pathCount, accSpeed, and avgSpeed. The pathCount represents the number of motorists on the current path. The accSpeed indicates the accumulated speed of motorists on the current path. The avgSpeed (AS) signifies the average travelling speed on the current path, which is calculated by implementing the formula shown in Formula 1, where S and C are the accumulated speed and the number of motorists on a single path, respectively.

$$AS = \frac{S}{C} \quad (1)$$

Average Travelling Time Adjacency Matrix

The average travelling time adjacency matrix, denoted by T_i , represents the average time taken to travel between each pair of nodes. It acts as the cost metric in the graph, enabling

algorithms to find the shortest route. T_i is calculated by applying the distance (D) divided by the average travelling speed (AS) (Formula 2). Each element represents the time in minutes.

$$T_i = \frac{D}{AS} \quad (2)$$

3.2 Path Finding Algorithms

Dijkstra Algorithm

Dijkstra's algorithm is widely used for solving single-source shortest path problems in graphs with non-negative edge weights. It is versatile and applicable to various graphs, including implementing weighted directed adjacency matrices [12]. The algorithm has several variants, including the standard type and priority queues. The binary heap is commonly used as the priority queue, facilitating the algorithm in finding the shortest path within an optimal time. This study evaluated the naive implementation of Dijkstra's and Dijkstra's with binary heap.

A* Algorithm

A* algorithm is particularly useful in graph traversal problems and path-finding [13, 14]. It excels in finding the most efficient route connecting two locations within a network of roadways. The algorithm works by using the standard cost metrics like what is being used in Dijkstra's algorithm and another function called "heuristic" to estimate the cost (using Euclidean distance) of the cheapest path from a source to a destination. The algorithm efficiently navigates through the nodes, balancing the path traversed and the estimated journey ahead.

3.3 Real-Time Operating System (RTOS)

FreeRTOS was selected to develop this study's real-time route recommendation system for the following reasons. **Simplicity:** FreeRTOS functions as a thread scheduler and TCP/IP stack. The kernel showcases efficient programming, with each line meticulously crafted for essential functionality, thereby reducing the likelihood of errors [15]. **Cost Effective:** The compact nature of FreeRTOS is highly beneficial for microprocessors with limited memory and storage. It significantly reduces the equipment costs [16]. **Open-Source:** It was developed using an open-source platform. Its universal design promotes hardware independence, boosting portability and lessening reliance on specific microprocessor manufacturers [17]. **Time-sensitive:** FreeRTOS can manage time-sensitive events efficiently. It is well-suited for various small devices, whether or not a real-time operating system is required [18].

3.4 Functional Diagram of the Simulated System using FreeRTOS

Fig. 2 represents the functional diagram for the simulated route recommendation system. Four distinct tasks are created. The **User Query Generation Task** generates a random number of user queries with varying sources and destinations. This task shall send multiple source and destination data to “queryQueue” to the **Path Calculation Task**. Upon receiving the data, the Path Calculation Task employs Dijkstra’s or A* to recommend the shortest route. These algorithms rely on the initialised or updated cost metric in the average travelling time matrix table.

The **Motorist Data Generation Task** is used to simulate motorists’ speed data. It generates motorists’ random paths and speed data into batches. Every batch holds up to 5,000 motorist data, and the data is sent to “mtSpeedqueue” to be processed by the **Matrices Update Periodic Task** for updating the matrix tables periodically. However, the Matrices Update Periodic Task will only perform the updating action when it is activated at predetermined intervals and acquires the mutex. Mutex is used for mutual exclusion, preventing multiple tasks from accessing matrices that store traffic data simultaneously. This mutex is shared among the two tasks: Path Calculation Task and Matrices Update Periodic Task.

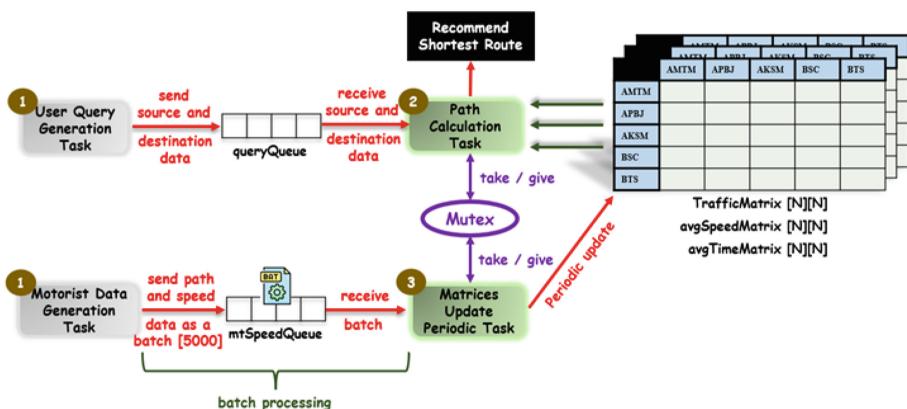


Fig. 2. Overview of the System Functional Diagram.

The Matrices Update Periodic Task holds a higher priority than the Path Calculation Task. Therefore, when activated, the Matrices Update Periodic Task will always be prioritised in acquiring the mutex. In summary, the User Query Generation Task and Motorist Data Generation Task are the senders that send data to both queues, namely “mtSpeedQueue” and “queryQueue”. The Path Calculation Task and Matrices Update Periodic Task are the receivers that read data from the queues. The latter is responsible for updating the traffic data, whilst the former is responsible for utilising the latest traffic data to find and suggest the shortest route to the users.

4 Results and Discussion

To evaluate the simulated system, we focused on the accuracy of the recommended route and the speed performance of the system response to user queries. In this study, the elapsed time of a user query refers to the time taken for the user query to be sent to the system, and then the system responds with a recommended route. In the speed performance evaluation, a statistical analysis was performed using metrics such as total elapsed time and average elapsed time across different query numbers ranging from 1,000 to 100,000 queries.

4.1 Validate the accuracy of the recommended route

The system's output precision was examined across three critical dimensions: (1) the distance measured in kilometres, (2) the efficiency of the proposed route, and (3) the estimated time of arrival (ETA). This analysis highlights the system's adeptness in navigating these parameters and underscores the algorithm's role in tailoring accurate solutions.

Accuracy Checking	
Source	: LK
Destination	: IKEC
Direct Path	: LK → IKEC
Total Distance	: 18.30 km
ETA	: 33.27 minute[s]
Recommended Path	: LK → GDSM → IKEC
Total Distance	: 19.10 km
ETA	: 12.54 minute[s]

Fig. 3. Recommended route for LK -> IKEC.

We take one example from the system for discussion. Fig.3 illustrates the recommended route from Lotus's Kepong (LK) to Inc, KL Eco City (IKEC). The suggested path includes passing by the Glo Damansara Shopping Mall (GDSM), resulting in a shorter ETA (Estimated Time of Arrival) than the direct LK to IKEC route. An investigation into the adjacency distance matrix table and average travelling speed matrix table revealed that the recommended route is accurate. The direct distance from LK to IKEC is 18.3 km, and its average speed is 33 km/h. This means that the ETA can be calculated by dividing 18.3 (distance) with 33 (average speed) and then multiplying that by 60 (minutes). Hence, the calculated ETA is 33.27 min. The adjacency distance matrix also showed that the distance from LK to GDSM is 10.6 km and 8.5 km from GDSM to IKEC. Hence, the total distance for LK → GDSM → IKEC is 19.1 km. The average travelling speed matrix table shows that the current value for LK → GDSM is 90 km/h and GDSM → IKEC is 93 km/h. As such, the ETA is 12.54 minutes, the average travelling time of LK → GDSM and GDSM → IKEC. This validation has provided evidence that the recommended route is working accurately.

4.2 Performance Evaluation

Table 1 shows the time performance of the simulated system. Using the standard Dijkstra algorithm, the system responded quickly to the 100,000 user queries, with the lowest average elapsed time (per user query) of just 0.026 ms. Based on the results, it could also be observed that the average elapsed time for the three algorithms did not escalate significantly with the increment of user queries, and all responses can be delivered in real time. The average elapsed time was quite stable, along with the growth of user query size. We examined this phenomenon and discovered that it was attributed to the parallel processing in the FreeRTOS platform. The real-time operating system (RTOS) can multitask and quickly switch among tasks.

Table 1. Performance Evaluation of the Path Finding Algorithms with Different User Queries Numbers.

User Query	Total Elapsed Time (ms)			Average Elapsed Time (ms)		
	Dijkstra (standard)	Dijkstra (PQ)	A*	Dijkstra (standard)	Dijkstra (PQ)	A*
1,000	26	61	45	0.0260	0.0610	0.0450
5,000	162	319	236	0.0324	0.0638	0.0472
10,000	306	613	484	0.0306	0.0613	0.0484
20,000	556	1,103	904	0.0278	0.0552	0.0452
50,000	1,353	2,603	2,149	0.0271	0.0521	0.0430
100,000	2,737	5,081	4,007	0.0260	0.0508	0.0401

The evaluation continued by testing the path-finding algorithm's limit on the number of user queries it can handle concurrently (Fig. 4). The number of user queries that could be handled by Dijkstra's algorithm (used priority queue) and the A* algorithm peaks at 828,000 and 827,000, respectively. This limitation arises as the memory allocation, often called the heap, hits its maximum threshold. Consequently, any additional processing attempts surpassing this limit lead to a crash. On the contrary, the standard Dijkstra's algorithm (array-based) approach demonstrated consistent memory usage during its entire runtime. This observation strongly suggests applying the fixed memory allocation (array) strategy instead of dynamic memory allocation (A* or priority queue) for any user queries above 825,000.

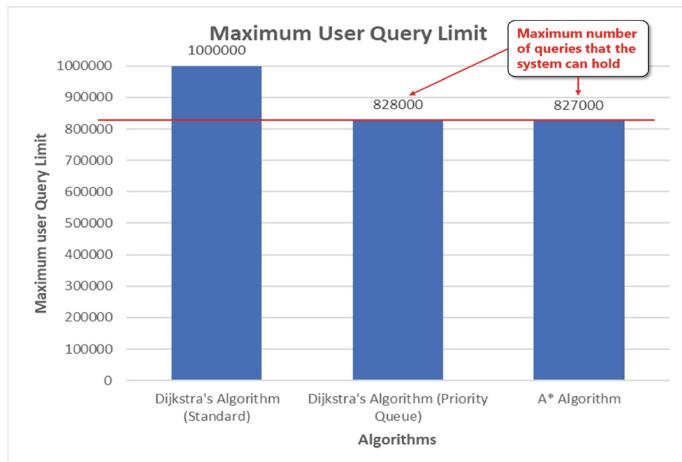


Fig. 4. Maximum number of user queries handled by the Dijkstra (standard), Dijkstra with priority queue, and A*.

5 Conclusion

This study developed a real-time route recommendation system for shopping malls in Klang Valley of Malaysia. It is implemented using FreeRTOS in simulated mode. The real-time system recommends optimal routes for motorists travelling between shopping malls daily using the traffic conditions. Matrix tables are used as the data structure to store data such as distances between shopping malls, average travelling speed per route, average travelling time per route, etc. The implemented Dijkstra's variants (standard and priority queue) and A* algorithm were evaluated. The results show that the system can process large amounts of user queries in real-time (based on 100,000 user queries, fastest elapsed time per query is 0.026 ms) with the standard Dijkstra's algorithm that can handle the maximum. The system's accuracy was also validated, where recommended routes are based on the shortest travelling time rather than the shortest distance. In conclusion, this study demonstrates a simple, intelligent transportation system that can be implemented using FreeRTOS with limited resources. It showcases the considerable benefits of instantaneously processing data to improve urban transportation. Future enhancements of this study include (1) real-world system implementation, which factors in the communication network delay and latency, and (2) optimisation of system resources to support more concurrent use queries/ requests.

References

1. Irtema, H.I., Ismail, A., Borhan, M.N., Das, A.M., Alshetwi, A.B.: Case study of the behavioural intentions of public transportation passengers in Kuala Lumpur. *Case Stud. Transport Policy*. **6**(4), 462–474 (2018)
2. Sia, P.Y.H., Saidin, S.S., Iskandar, Y.H.P.: Systematic review of mobile travel apps and their smart features and challenges. *J. Hospitality Tourism Insights* **6**(5), 2115–2138 (2023). <https://doi.org/10.1108/jhti-02-2022-0087>

3. Cheng, X.: A travel route recommendation algorithm based on interest theme and distance matching. *EURASIP J. Adv. Signal Process.* (2021). <https://doi.org/10.1186/s13634-021-00759-x>
4. Falek, A.M., Gallais, A., Pelsser, C., Julien, S., Theoleyre, F.: To re-route, or not to re-route: Impact of real-time re-routing in urban road networks. *J. Intell. Transport. Syst.* **26**(2), 198–212 (2022)
5. Mehmood, F., Ahmad, S., Kim, D.: Design and development of a real-time optimal route recommendation system using big data for tourists in Jeju Island. *Electronics* **8**(5), 506 (2019). <https://doi.org/10.3390/electronics8050506>
6. Shahi, G.S., Bath, R.S., Egerton, S.: A comparative study on efficient path finding algorithms for route planning in smart vehicular networks. *Int. J. Comput. Netw. Appl.* **7**(5), 157–166 (2020)
7. Wang, X., Zhang, H., Liu, S., Wang, J., Wang, Y., Shangguan, D.: Path planning of scenic spots based on improved A* algorithm. *Sci. Reports* **12**(1), 1320 (2022). <https://doi.org/10.1038/s41598-022-05386-6>
8. Ravish, R., Rangaswamy, S., Arpitha, V., Vasuprada, U.: User preference-based intelligent road route recommendation using SARSA and dynamic programming. *J. Control Decis.* **10**(3), 443–453 (2023). <https://doi.org/10.1080/23307706.2022.2096705>
9. Zhang, W., et al.: APFD: an effective approach to taxi route recommendation with mobile trajectory big data. *Front. Inform. Technol. Electron. Eng.* **23**(10), 1494–1510 (2022). <https://doi.org/10.1631/fitee.2100530>
10. Wang, J., Zhou, W., Li, S., Shan, D.: Impact of personalised route recommendation in the cooperation vehicle-infrastructure systems on the network traffic flow evolution. *J. Simulation* **13**(4), 239–253 (2019). <https://doi.org/10.1080/1747778.2018.1515579>
11. Bodwin, G., Vempala, S.: A unified view of graph regularity via matrix decompositions. *Random Struct. Algorithms* **61**(1), 62–83 (2022)
12. Chen, Y.Z., Shen, S.F., Chen, T., Yang, R.: Path optimisation study for vehicles evacuation based on Dijkstra algorithm. *Procedia Eng.* **71**, 159–165 (2014)
13. Candra, A., Budiman, M.A., Hartanto, K.: Dijkstra's and a-star in finding the shortest path: A tutorial. In: 2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics, pp. 28–32 (2020).
14. Rios, L.H.O., Chaimowicz, L.: A survey and classification of A* based best-first heuristic search algorithms. In: Brazilian Symposium on Artificial Intelligence, pp. 253–262. Springer, Berlin Heidelberg (2010)
15. Saramud, M.V., Kovalev, I.V., Losev, V.V., Petrosyan, M.O.: Application of FREERTOS for implementation of the execution environment of real-time multi-version software. *Int. J. Inform. Technol. Secur.* **10**(3) (2018)
16. Venkataraman, A., Chitra, P.: Real-time implementation of RTOS based vehicle tracking system. *Biosci. Biotechnol. Res. Asia* **12**(1), 237–241 (2015)
17. Guan, F., Peng, L., Perneel, L., Timmerman, M.: Open source FreeRTOS as a case study in real-time operating system evolution. *J. Syst. Softw.* **118**, 19–35 (2016)
18. Akgün, G., Görhringer, D.: Power-aware real-time operating systems on reconfigurable architectures. In: 2021 31st International Conference on Field-Programmable Logic and Applications (FPL), pp. 402–403. IEEE (2021).



Machine Learning-Based Phishing Website Detection: A Comparative Analysis and Web Application Development

Jia Xin Yau and Kai Lin Chia^(✉)

Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, Kajang, Selangor, Malaysia
yaujiaxin@utar.my, klchia@utar.edu.my

Abstract. Phishing, a cybercrime that uses sociotechnical and technical deception, targets identifiable information and financial credentials and poses a high risk according to the IBM Cost of a Data Breach Report 2022 which shows that on average precisely, the cost per transaction is \$4.91 million, phishing attacks are on the rise, challenging the ability of traditional scanning systems to adapt to trends. This study examines and it compares the effectiveness of three anti-phishing methods: Autoencoder, Extreme Gradient Boost (XGBoost), and Random Forest (RF). Through feature selection and robust machine learning (ML) algorithms, including Random Forest achieving a remarkable 97.03% accuracy, the proposed solution integrates list-based methods with ML models for two-tier security. The wrapper method is employed to extract crucial features, facilitating precise phishing detection. Specific algorithms such as Random Forest and XGBoost are chosen for their proven effectiveness in handling complex data and class imbalances. However, potential limitations include the need for continuous adaptation to new phishing methods and exploring ensemble techniques for enhanced model robustness. Benchmarking against existing methods highlights the superiority of Random Forest in achieving balanced recall and precision. This study contributes to advancing phishing detection systems by leveraging machine learning and proposing strategies for improved performance and accuracy, which are then applied to a web application for countering phishing attacks.

Keywords: Phishing detection · Machine learning · Internet security · Google Safe Browsing · Web application

1 Introduction

Phishing, as defined by the Anti-Phishing Working Group (APWG), is a crime with significant threat that uses social engineering and technical deception to steal personal and financial information [1, 2]. The IBM Cost of a Data Breach Report 2022 highlights phishing as the second most common and costly attack vector, with an average cost of \$4.91 million per incident and over 1.27 million attacks in 2022 [3]. These attacks not only lead to financial losses but also damage reputations and have long-term economic impacts. Given the increasing sophistication of phishing attacks, which challenge

traditional detection systems, there is an increasing need to evaluate and enhance anti-phishing techniques. This study compares three anti-phishing methods—autoencoder, Extreme Gradient Boost (XGBoost), and Random Forest (RF)—based on performance metrics such as accuracy, precision, and recall. The goal is to identify the most effective approach and integrate it into a user-friendly web application, leveraging Google Safe Browsing List for a comprehensive defense against phishing, including zero-day attacks.

2 Literature Review

Research on cybercriminal activities reveals a need for accurate and timely information. Machine learning and deep learning models have emerged as pivotal tools to address this gap. Various studies show in Table 1. Employ list-based, heuristic, machine learning, and deep learning techniques for phishing website detection.

Table 1. Summary of existing phishing websites detection research.

Techniques	Explanation
Deep learning & heuristic, Machine learning	Feng & Yue used RNN and bi-directional RNN with LSTM and GRU for phishing detection, achieving 99.50% accuracy from 17 characteristics in 1.5 million URLs. Seok & Sung [4] improved sensitivity by 3.98% with a convolutional autoencoder, focusing on character-level URL properties. Gupta et al. [5] found Random Forest most effective among four classifiers, with 99.57% accuracy. Yang et al. [6] combined CNN and LSTM, using XGBoost for classification, resulting in 98.99% accuracy and a 0.59% false positive rate. Saha et al. reported 95% training and 93% testing accuracy with a Multilayer Perceptron Neural Network. Maci et al. [7] demonstrated a DDQN-based classifier's superiority in web phishing detection over DNN, CNN, LSTM, and BiLSTM
Machine learning	Study [8] combined DT, SVM, and Random Forest, reaching 98.52% accuracy but increased model complexity and cost. Butnaru et al. [9] used five machine learning methods, with Random Forest topping at 99.29% accuracy, outperforming Google Safe Browsing. Rao et al. [10] employed domain-specific HTML and text embeddings, with a focus on text-based detection, noting limitations with graphic content. Another study achieved 97.7% phishing and 89.2% spam detection accuracy using Random Forest and Multilayer Perceptron, utilizing the same dataset for training and testing
Visual similarity & machine learning	A study [11] used web page text and screenshots for identifying similar websites, achieving 99.20% target and 99.66% phishing detection with logistic regression, noting dependency on search engine results. Another research [12] evaluated LBET, RoFBET, ABET, and BET models, with LBET detecting over 97.5% of phishing using 11,055 websites

(continued)

Table 1. (*continued*)

Techniques	Explanation
List based, visual similarity, heuristic & machine learning	Logistic Regression and Random Forest (RF) were used to achieve 97% accuracy in malware and phishing detection, with frequent updates from URL blacklists slowing the process [13]. Rao & Pais [14] developed an ensemble model combining RF, Extra-Tree, and XGBoost, achieving 98.72% accuracy and a 97.39% Mathews Correlation Coefficient (MCC), though with high response time. Nathezhtha et al. [15] introduced a three-phase phishing detection using DNS blacklists, heuristics, and web crawlers, focusing on web content, URL, and traffic analysis
White-list-based & visual similarity	The study [16] achieved 95% accuracy in phishing detection by verifying webpages using hyperlink/URL properties, testing on 200 sites (140 phishing, 60 real). Cao et al. [17] created an automatic allowlist updated with login-page IPs, attaining 100% true positives and 0% false negatives using a Naive Bayesian classifier, but was limited to frequently used login sites and required user participation, unable to detect new phishing sites
Machine learning, heuristic & list based	Stobbs et al. [18] achieved 99.33% accuracy using Random Forest with Particle Swarm Optimization for feature selection and Tree-based Parzen Estimator for hyperparameter tuning, focusing on accuracy and recall without detailing the training/testing split. Jain & Gupta [19] improved phishing detection efficiency and reduced false negatives using a URL and DNS matching module with a whitelist, achieving an 86.02% true positive rate and less than 1.48% false negative rate, utilizing Jsoup for hyperlink extraction and Guava libraries for domain identification

3 Research Design

The research design, shown in Fig. 1, covers critical components, including data overview, feature selection, machine learning (ML) techniques implementation, performance evaluation, and web application development. Traditional list-based methods, such as Google Safe Browsing, have long been foundational for identifying established phishing websites. However, they have some drawbacks. One notable challenge is the detection lag, where new phishing threats may not be promptly identified. Additionally, the coverage of these lists might be limited, leaving users vulnerable to zero-day assaults. Heuristic-based detection can also incorrectly flag legitimate websites as phishing threats [20].

To address the drawbacks associated with list-based approaches, the integration of ML models offers a two-tier security approach. Dynamic adaptation, inherent in ML models, allows for the continuous evolution of the system to counter new phishing methods and threats [21]. ML models leverage trends and anomalies to prevent zero-day phishing attacks, showcasing adaptability to evolving threats. ML algorithms, on the other hand, can recognize common traits and behaviours of phishing sites despite their deceitful appearance. Their continuous learning capabilities ensure they adapt to new

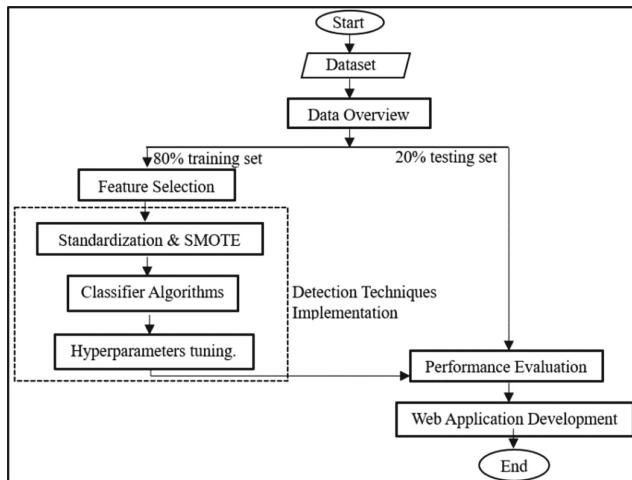


Fig. 1. The research design of this study.

threats efficiently, a feature lacking in list-based solutions that may experience delays in reflecting the evolving threat landscape.

3.1 Dataset Overview

The ‘dataset_full.csv’ from Mendeley Data comprises 88,647 instances, split into 58,000 legitimate (65.43%) and 30,647 phishing (34.57%) websites, reflecting real-world distribution (Fig. 2) [22]. This dataset, essential for developing cybersecurity tools like malware detection systems and phishing filters, has been widely used in research for machine learning-based phishing detection. Notably, Bahaghigat et al. [23] and Alani and Tawfik [24] employed it to enhance anti-phishing models and develop the PhishNot system, demonstrating its significance in advancing phishing detection methodologies.

3.2 Feature Selection

The feature selection was conducted using the Bi-directional Elimination (BDE) method after dividing the dataset into training (80%) and testing (20%) sets for reproducibility. Starting with all 111 features, an Ordinary Least Squares (OLS) model was applied, and features with p-values over 0.05 were iteratively removed using the BDE method, ultimately selecting 59 features (Fig. 3). The refined model, evaluated using test data, achieved accuracy (91.83%), precision (83.41%), recall (95.28%), F1 Score (88.95%), and ROC AUC Score (97.67%). Key findings include the positive correlation of the “url_shortened” feature with phishing risk, indicating that short URLs, often used by phishers, increase the likelihood of phishing. Conversely, a negative coefficient for “qty_at_params” suggests ‘@’ symbols in URLs may decrease phishing risk, as such patterns are less common in phishing attempts. Additional significant features impacting phishing detection include URL format, use of special characters, and symbols in URLs.

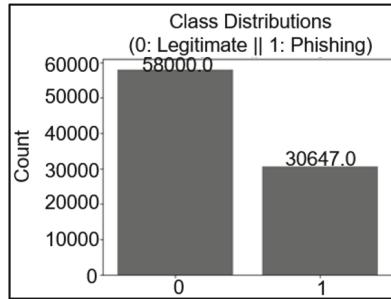


Fig. 2. Class distribution of the dataset utilised in this study.

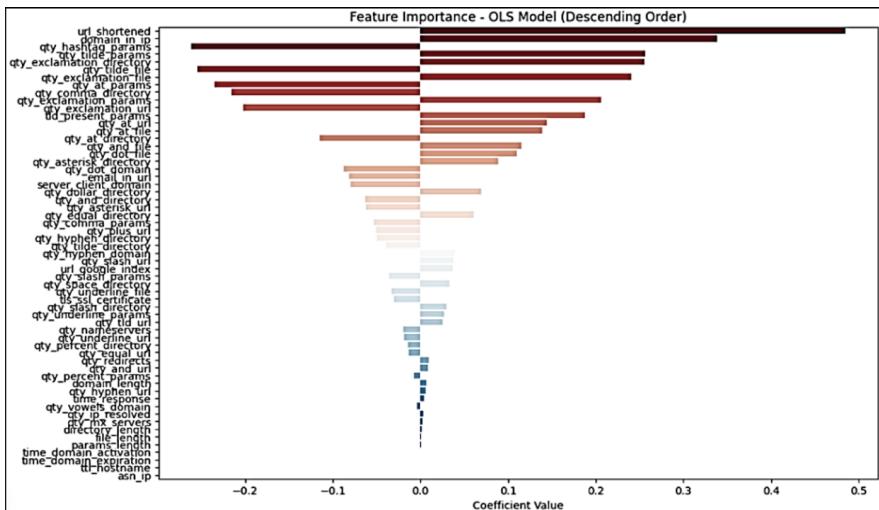


Fig. 3. OLS regression coefficients of the selected features.

3.3 Detection Techniques Implementation

In this study, the dataset was split into 80% training and 20% testing sets. Prior to training the Random Forest, XGBoost, and Autoencoder models, the Synthetic Minority Over-sampling Technique (SMOTE) was used on the training set to balance the class distribution from 46,388 legitimate and 24,529 phishing instances to an equal number for each class. These models were chosen based on their proven efficacy in handling complex data and their widespread use in cybersecurity. Random Forest and XGBoost, as ensemble methods, are noted for their capability to manage imbalanced data and non-linearities, while Autoencoders are effective in detecting phishing through data compression learning. Hyperparameter tuning was conducted via Randomized Search CV in the Scikit-learn library, optimizing parameters like the number of trees, maximum node-splitting features, tree depth, and others. The optimized Random Forest model achieved an AUC-ROC of 0.9953 (Fig. 4), and XGBoost demonstrated a similar performance with an AUC of 0.9949, its precision-recall trade-off shown in Fig. 5. Autoencoders, with

hyperparameters like optimizer type, activation function, and learning rate optimized, yielded an AUC of 0.9619 (Fig. 4). The precise hyperparameter settings for each model are detailed in Table 2.

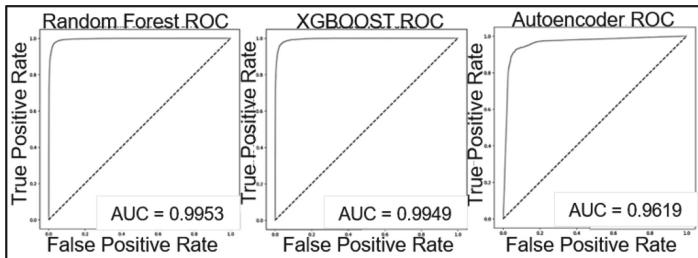


Fig. 4. The ROC curves for the classifiers.

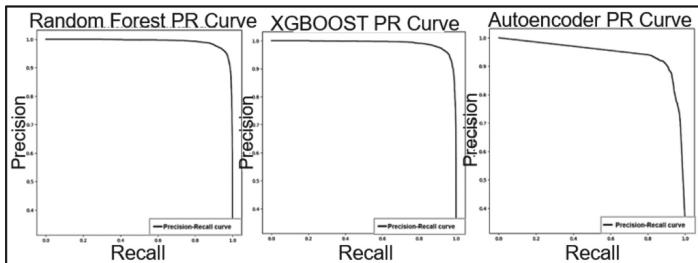


Fig. 5. The Precision-Recall curves for the classifiers.

Table 2. The optimal hyperparameters for the classifiers.

Classifier	Hyperparameter Used
Random forest	number of estimators: 300, minimum samples split: 5, minimum samples leaf: 1, maximum features: sqrt, maximum depth: 40, bootstrap: False
XGBoost	number of estimators: 200, learning rate: 0.1, maximum depth: 7, minimum child weight: 3, column subsampling by tree: 0.6, gamma: 0.2, L1 regularization: 0.4, L2 regularization: 0.3, scale positive weight: 3
Autoencoder	activation function: relu, batch size: 64, number of epochs: 100, size of hidden layer: 128, learning rate: 0.001, optimizer: adam

3.4 Performance Evaluation and Comparison

The ability of a classifier to identify phishing sites needs to be assessed using evaluation metrics such as accuracy, precision, recall, and F1-score. These metrics are critical for evaluating the classifiers' effectiveness. Performance metrics such as accuracy, precision,

recall, and F1-score can be calculated utilizing Python and the Scikit-learn library. The accuracy metric measures the overall credibility of the predictions. Precision measures the proportion of legitimate phishing websites compared to those that were predicted to be such. Recall is the proportion of legitimate phishing websites that the technique accurately detected. The F1 score provides a valuable metric for evaluating a technique's overall performance by calculating the harmonic mean of recall and precision. This approach balances recall and precision, particularly when asymmetrical datasets are involved.

3.5 Web Application Development

The developed phishing detection web application integrates a pre-trained Random Forest classifier and the Google Safe Browsing list to assess and validate user-entered URLs (Fig. 6). Upon URL submission, the application employs the classifier to determine phishing risk and cross-references potentially malicious URLs with Google Safe Browsing, providing a comprehensive safety assessment to the user (Figs. 7 and 8). The system is built using Flask for HTTP request handling and template generation, Flask-Ngrok for internet connectivity, Joblib for loading the Random Forest model, urllib.parse for URL processing, and Google Safe Browsing API for verification. Our evaluation of existing phishing detection methods revealed limitations like data imbalance and over-reliance on specific features. Addressing these issues, we integrated machine learning techniques, particularly Random Forest and XGBoost, with Google Safe Browsing, optimizing feature selection and dataset balancing to refine the detection process. This combination not only leverages the strengths of advanced algorithms and external verification but also aligns with insights from prior research, enhancing the efficacy of phishing detection.

4 Results and Discussion

In our study, the Random Forest classifier demonstrated exceptional performance in identifying phishing websites, achieving an accuracy rate of 97.03%. This accuracy reflects its capability to correctly classify both legitimate and fraudulent websites, with this rate determined by the proportion of true positives and true negatives out of the total dataset. The Random Forest model also maintained a precision rate of 94.74%, effectively minimizing false positives, and a recall rate of 96.76%, indicating its ability to identify phishing websites accurately. Notably, it achieved the highest F1-Score of 95.74%, proving its outstanding overall performance. Additionally, the Random Forest classifier achieved an AUC of 99.51%, showing its excellent capability in distinguishing between phishing and legitimate websites. XGBoost also showed good performance, particularly excelling in recall with a rate of 97.24%, though it experienced slightly lower in precision, F1-Score and AUC. The Autoencoder, with an accuracy rate of 95.97% and slightly lower precision and recall rates of 92.66% and 95.93% respectively, resulted in an F1-Score of 94.27%. Its AUC of 99.18% signifies its effective performance in phishing detection, although slightly behind the other two methods.

Table 3 summarizes the performance of the three classifiers—Random Forest, XGBoost, and Autoencoder—in phishing website identification, revealing different

aspects of their performance. The Random Forest classifier emerges as the best classifier in distinguishing between authentic and phishing websites with a balance of recall, precision, and the highest F1-Score, alongside its high AUC percentage. While XGBoost stands out in recall, it shows a slight trade-off between accuracy and F1-Score. The Autoencoder, despite being the least accurate of the three, still performs considerably well overall. Our work compares Random Forest, XGBoost, and Autoencoder classifiers for phishing website detection, building on earlier research. Rao & Pais' ensemble model had 98.72% accuracy [14], but our technique reveals each classifier's strengths and flaws, enabling a more comprehensive understanding of their capabilities. Our study uses machine learning instead of user input, improving detection accuracy. Our evaluation uses three classifiers with a thorough training-testing split, ensuring fair evaluation, unlike Kumar et al.'s two classifiers on the same dataset [25]. In contrast to Abusaimeh's multiple classifier strategy [5], our study reduces computational complexity and reveals classifier effectiveness. Our detailed examination of numerous classifiers and approaches helps design more robust phishing website detection systems.



Fig. 6. The web application's user interface.

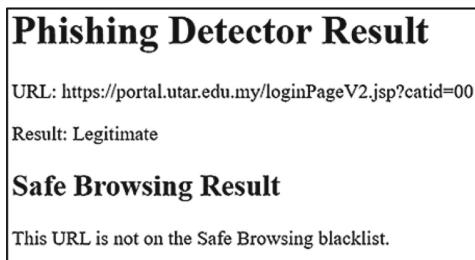


Fig. 7. Result of a phishing link.

Phishing Detector Result

URL: <https://abricy.com/nm/z/?o=ZGlhbmFAc2RqYmNzdGVlbC5jb20=&WhuZ1cekbW9sxyI2dDwRPNUxqHRt3oGKv35yp8CycRdfuaiO4PC9HmOqnamwvreouXUiRC6ZOnJ7tudb4vjhGISIe5BOZ.7G34J>

Result: Phishing

Safe Browsing Result

This URL is not on the Safe Browsing blacklist.

[Go back to home](#)

Fig. 8. Result of a legitimate link.**Table 3.** Performance evaluation of the classifiers.

	Random Forest		XGBoost		Autoencoder	
Predicted True	Legitimate	Phishing	Legitimate	Phishing	Legitimate	Phishing
Legitimate	11283	329	11113	499	11147	465
Phishing	198	5920	169	5949	249	5869
Accuracy	97.03%		96.23%		95.97%	
Precision	94.74%		92.26%		92.66%	
Recall	96.76%		97.24%		95.93%	
F1-score	95.74%		94.68%		94.27%	
AUC	99.51%		99.45%		99.18%	

5 Conclusion

In conclusion, this study has systematically explored the application of machine learning techniques, namely Random Forest, XGBoost, and Autoencoders, in the detection of phishing websites. Through a process of data preparation, feature selection, and model optimization, high levels of accuracy, precision, recall, and F1-Score in phishing detection are achieved. The integration of these machine learning classifiers with the Google Safe Browsing list in a user-friendly web application further enhances the practicality and effectiveness of phishing threat identification and prevention. The findings highlight the need for ongoing updates in feature extraction to address the evolving nature of phishing URLs. Enhanced robustness could be achieved through ensemble methods like stacking or bagging with diverse base learners. Expanding tests to larger, varied datasets will help assess model effectiveness in different situations. Future work should advance

real-time behavioural analysis in phishing detection and explore deep learning to boost the accuracy of anti-phishing tools.

References

1. APWG: Phishing Activity Trends Report 3rd Quarter 2022, 12 December 2022. [Online]. https://docs.apwg.org/reports/apwg_trends_report_q3_2022.pdf. Accessed 25 February 2023
2. Zainab, A., Chaminda, H., Liqaa, N., Imtiaz, K.: Phishing attacks: a recent comprehensive study and a new anatomy. *Front. Comput. Sci.* **3**(2021)
3. IBM: Cost of a data breach, July 2022. [Online]. <https://www.ibm.com/reports/data-breach>. Accessed 25 February 2023
4. Seok, J.B., Sung, B.C.: Deep Character-Level Anomaly Detection Based on a Convolutional Autoencoder for Zero-Day Phishing URL Detection. *Multidisciplinary Digital Publishing Institute*, Seoul (2021)
5. Gupta, B., Yadav, K., Razzak, I., Psannis, K., Castiglione, A., Chang, X.: A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. *Comput. Commun.* **1175**, 47–57 (2021)
6. Yang, P., Zhao, G., Zeng, P.: Phishing website detection based on multidimensional features driven by deep learning. *IEEE Access* **7**, 15196–15209 (2018)
7. Maci, A., Santorsola, A., Coscia, A., Iannaccone, A.: Unbalanced web phishing classification through deep reinforcement learning. *Computers* **12**(6) (2023)
8. Abusaimeh, H.: Detecting the phishing website with the highest accuracy. *TEM J.* 947–953 (2021)
9. Butnaru, A., Mylonas, A., Pitropakis, N.: Towards lightweight url-based phishing detection. *Fut. Internet* **13**(6), 1–15 (2021)
10. Rao, R.S., Umarekar, A., Pais, A.R.: Application of word embedding and machine learning in detecting phishing websites. *Telecommun. Syst.* **79**, 33–45 (2022)
11. Dooremaal, B.V., Burda, P., Allodi, L., Zannone, N.: Combining text and visual features to improve the identification of cloned web pages for early phishing detection. *Vienna* (2021)
12. Abdelnabi, S., Krombholz, K., Fritz, M.: VisualPhishNet: zero-day phishing website detection by visual similarity. In: *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 1681–1698 (2020)
13. Maroofi, S., Korczynski, M., Hesselman, C., Ampeau, B., Duda, A.: COMAR: Classification of compromised versus Maliciously Registered Domains (2020)
14. Rao, R., Pais, A.: Two level filtering mechanism to detect phishing sites using lightweight visual similarity approach. *J. Ambient Intell. Human Comput.* **11**, 3853–3872 (2020)
15. Nathezhtha, T., Sangeetha, D., Vaidehi, V.: WC-PAD: web crawling based phishing attack detection (2019)
16. Azeez, N., Misra, S., Margaret, I., Fernandez-Sanz, L., Abdulhamid, S.: Adopting automated whitelist approach for detecting phishing attacks. *Comput. Secur.* **108** (2021)
17. Cao, Y., Han, W., Le, Y.: Anti-phishing based on automated individual white-list. *Virginia* (2008)
18. Stobbs, I.B., Jacob, S.M.: Phishing web page detection using optimised machine learning. In: *IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 483–490 (2020)
19. Jain, A.K., Gupta, B.B.: A novel approach to protect against phishing attacks at client side using auto-updated white-list. *Inf. Secur.* **9**, 1–11 (2016)
20. Grega, V., Iztok, F.J., Podgorelec, V.: Datasets for phishing websites detection. In: *Data in Brief*, vol. 33 (2020)

21. Bahaghight, M., Ghasemi, M., Ozen, F.: A high-accuracy phishing website detection method based on machine learning. *J. Inf. Secur. Appl.* **77** (2023)
22. Kumar, S., Faizan, A., Viinikainen, A., Hamalainen, T.: Machine Learning Based Spam and Phishing Detection. Springer International Publishing (2018)
23. Feng, T., Yue, C.: Visualising and interpreting RNN Models in URL-based phishing detection. Barcelona (2020)
24. Saha, I., Sarma, D., Chakma, R., Alam, M.N., Sultana, A., Hossain, S.: Phishing attacks detection using deep learning approach. In: 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, IEEE, pp. 1180–1185 (2020)
25. Alani, M.M., Tawfik, H.: PhishNot: a cloud-based machine-learning approach to phishing URL detection. *Comput. Netw.* **218** (2022)



Comparative Performance of Multi-level Pre-trained Embeddings on CNN, LSTM and CNN-LSTM for Hate Speech and Offensive Language Detection

Noor Azeera Abdul Aziz¹(✉), Anazida Zainal², Bander Ali Saleh Al-Rimy³, and Fuad Abdulgaleel Abdoh Ghaleb⁴

¹ Department of Internet Engineering and Computer Science, Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, Bandar Sungai Long, Selangor, Malaysia

azeera@utar.edu.my

² Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Johor, Malaysia
anazida@utm.my

³ School of Computing, University of Portsmouth, Buckingham Building, Lion Terrace, Portsmouth PO1 3HE, UK
bander.al-rimy@port.ac.uk

⁴ College of Computing and Digital Technology, Birmingham City University, Birmingham B4 7XG, UK
Fuad.Ghaleb@bcu.ac.uk

Abstract. With growing concerns over hate speech, social media platforms provide policies for monitoring hate content. Nowadays, platforms like Twitter and Facebook rely on humans and machines as content moderators. As for machine moderators, many studies proposed hate speech detection using machine learning approaches. This study investigated which pre-trained text embedding (Word2Vec, GloVe, FastText, Elmo, and BERT) is the best for each tokenization level (word, subword, and character) and which neural network architecture (CNN, LSTM, and CNN-LSTM) is the best as an encoding method for hate speech and offensive language detection. The character-level GloVe with CNN-LSTM performed best among all tested methods. GloVe (character level) scored 93% for F1-score and 92% for accuracy. At the word level, BERT word embedding with CNN-LSTM had the best classification scores of 90% F1-score and 91% accuracy. At the subword level, CNN-LSTM and CNN fared best with BERT word embeddings, which had 86% for both accuracy and F1-score. The performance findings show that pre-trained embeddings at different tokenization levels capture diverse information. Moreover, with an average of 85% for F1-score and 86% for accuracy, CNN-LSTM yielded the best score for almost all text embedding regardless of the tokenization level compared to CNN and LSTM. These results show that CNN-LSTM complements each other to capture sequential and local patterns in the input text.

Keywords: Hate speech detection · text embedding · neural network · text classification

1 Introduction

Social media has transformed the world by creating a new dimension of communication. People can express and exchange ideas freely through these platforms. Unfortunately, some people have violated freedom of speech by aggressively expressing opinions that incite hatred. Therefore, social media platforms provide policies to monitor and remove hate speech content [1-3].

Many existing studies have employed traditional machine learning methods in hate speech detection. Besides the traditional machine learning method, deep learning has become more prevalent in hate speech detection. In the past several years, text embedding has gradually taken over one-hot representation and become a standard preprocessing method to encode the text in deep learning for text classification. Text embedding will obtain text representation by transforming words into low-dimensional dense vectors to solve the curse of dimensionality issue of one-hot representation when dealing with a large corpus [4, 5]. Unfortunately, text embedding needs high computing power and algorithms to train large vocabulary from scratch [6]. Therefore, pre-trained embeddings such as Word2Vec [7], Global Vectors (GloVe) [8], fastText [9], Embeddings from Language Models (ELMo) [10] and Bidirectional Encoder Representation From Transformers (BERT) [10] have been introduced.

In the hate speech domain, the previous works also implemented various pre-trained embedding techniques to encode the text. As for deep learning methods, many existing studies extract advanced features to detect hate speech, offensive language, and neither text. For example, [11] adapted the pre-trained BERT by fine-tuning this model with a Twitter-specific hate speech dataset. As for the other studies, the authors employed an LSTM network that received input from the previous layer, which used GloVe as word embedding [12, 13]. The following study proposed by [14] implemented BERT to analyze the tokens and then convert the tokens into the vector before the CNN network can use the word vector for feature extraction. Similarly, as proposed by [11], this author utilized the CNN network to extract the features and implemented Word2Vec as the pre-train embedding.

Unfortunately, according to [15], each pre-trained embedding has several limitations and has increased the accuracy result after some improvement in sentiment analysis. Moreover, each pre-trained embedding technique can represent the text at the different tokenization text levels, including character, subword, word, and sentence [16]. Unfortunately, a limited study explores pre-trained embedding techniques for text encoding at different tokenization levels in hate speech, offensive language, and neither class text detection [17]. Moreover, most studies implement the single neural architecture of deep learning methods in the trinary classification of hate speech, offensive language detection, and neither text. For example, some works implement CNN [14, 18], and some works implement LSTM [11, 13, 19] to distinguish hate speech from offensive language. CNN architecture considers local and global features by learning the local features through convolution kernels and combining the local text features to generate overall global features. Unfortunately, CNN cannot capture long-distance dependencies because CNN uses comparatively simple convolutional kernels and cannot maintain the correlation between local and global features [20]. Unlike CNN, LSTM contains three gate components: input, memory, and output, which make this architecture able to handle

long-distance dependencies by storing the history information. Additionally, compared to CNN, which is better at supporting text sequence prediction [21], LSTM is a biased model which always makes decisions based on tail information [22]. To conclude, each neural architecture has a different purpose, advantages, and limitations. Although deep learning can extract hidden information from sentences, a single neural architecture is not enough to capture both dependency types, which are vital indicators for compelling classifier performance.

Therefore, as pre-trained embedding techniques can convert texts into meaningful feature vectors [23] and the enrichment of input data via text encoding can improve text classification [16], this study investigated several pre-trained text embedding techniques such as Word2Vec, GloVe, FastText, Elmo, and BERT at different levels of tokenized input text, such as the word, subword, and character levels to pursue this objective and investigate the best neural network architecture (CNN, LSTM, CNN-LSTM) as an encoding method for hate speech and offensive language detection.

2 The Architecture of HSOLC Detection Model

The architecture of CNN, LSTM, and CNN-LSTM are illustrated in Fig. 1. Three primary layers comprise these architectures. The following subsections discuss each layer in greater detail.

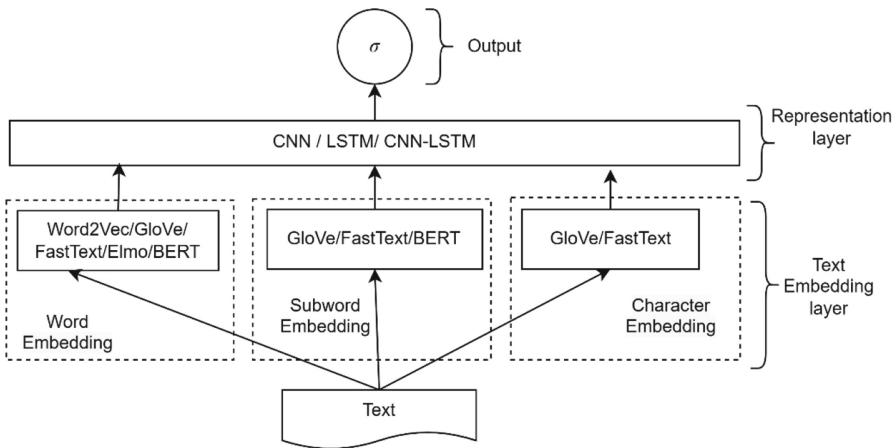


Fig. 1. Architecture of HSOLC Detection Model.

2.1 Text Embedding Layer

Text embedding, which encodes the textual data into vectors, is crucial before employing any machine learning model, including neural networks. This study used pre-trained text embedding to avoid learning the embedding from raw, which causes high computing power. As for pre-trained text embedding, this technique contains two types:

non-contextual (Word2Vec, GloVe, and FastText) and contextual (Elmo and BERT) embedding, and each of them has advantages and disadvantages [24]. A non-contextual approach expedites the coverage of neural networks; however, it disregards polysemy, resulting in unique representations for each token. Word2Vec is a window-based model that does not use the information in the whole document or the subword information. Moreover, Word2Vec cannot discover the Out Of Vocabulary (OOV) terms. In contrast, GloVe and FastText can solve OOV problems as this technique learns the vectors of character n-grams or parts of words. In FastText, the word representation is learned by considering a large window of left and right context words, making FastText able to produce embeddings for misspelt words, uncommon words, and words do not present in the dataset.

On the other hand, a contextual approach is slower but retains context information by utilizing the surrounding words to express the embeddings. ElMo builds a lookup table between the training set words and their pre-trained representations. ElMo uses the pre-trained bidirectional language model (BiLM) to learn the syntax and semantics between words and linguistic context. BiLM captures context-dependent features of the meaning of words. Since the language model aim is built from left to right, adding successive words to a sentence, the primary drawback of ElMo is its inability to consider both the left and right contexts of the target word.

BERT is one of the most effective context and word representations. BERT is based on a multilayer bidirectional transformer trained on plain text for masked word prediction and following sentence prediction tasks. Besides, BERT uses an attention mechanism. Attention is a way to look at how the words in a sentence fit together. Because of this, BERT considers every context of the token surroundings target words, both to the left and right.

Word Embedding. All five well-known pre-trained word embeddings for hate speech detection tasks, including Word2Vec, GloVe, FastText, Elmo, and BERT, are used for word embeddings. In word embedding, the text (tweet) is preprocessed and treated as a sequence of words. For example, Let $\{X_1, \dots, X_T\}$ represents the word sequence of an input text. First, pre-trained embedding is used to obtain the fixed word embedding of each word. This step is repeated for every type of pre-trained word embedding. In this layer, each text is represented by a matrix: $X \in R^{m \times T}$, where m is the dimensions for the word vector, and T is the length of the tweet sentence.

Subword Embedding. A subword is a smaller word unit. In subword embedding, words of the text (tweet) are preprocessed and treated as a sequence of subwords. This work integrated with subword embedding to capture morphological and structural information. For example, assumed characters N-grams $n_c=3$, the subword representation for the word “Hatest” are {ha, hat, ate, est}. The representations could be consists of all character N-grams. Let $\{Sw_1, \dots, Sw_T\}$ represents the subword sequence of a word for input text. This step is repeated for every type of pre-trained embedding. In this layer, each text is represented by a matrix: $Sw \in R^{m \times T}$, where m is the dimensions for the subword vector, and T is the word length in the tweet sentence. This study used GloVe, FastText, and BERT for subword embedding.

Character Embedding. Character and subword both can handle morphological phenomena. The difference is in the character level; the embedding is learned for character n-gram individually. Due to the same reasons in subword embedding, this study used GloVe and FastText for character embedding. In character level, for example, assumed characters N-grams n_c is 2, the subword representation for the word “Hate” is {[h,a,t,e][ha, at, te]}. Let $\{C_1 \dots C_T\}$ represents the character sequence of a word for input text. First, pre-trained embedding is used to obtain the fixed character embedding of each word. This step is repeated for every type of pre-trained embedding. In this layer, each text is represented by a matrix: $C \in R^{m \times T}$, where m is the dimensions for the character vector, and T is the word length in the tweet sentence.

2.2 Representation Layer

In this layer, the word embedding obtained from the previous layer is trained by neural networks. This study considered three common neural network architecture types and has shown promising results in addressing hate speech detection tasks: CNN, LSTM, and CNN-LSTM.

CNN Model. For the first model, CNN [25] learns each input from text embedding later, which performs a discrete convolution on an input matrix. convolution entails applying a filter $W \in R^{k \times m}$ on k windows of words to create a new feature. Therefore, assuming starting from the $t-th$ word, the feature is:

$$f_t = f(w \cdot x_{t:t+k-1} + b) \quad (1)$$

where $b \in R$ is a biassed term to learn, \bullet is matrix multiplication, and f is a nonlinear activation function that lets the network add sparsity to improve the training speed. Each possible window of words in the input text is subjected to this operation to generate a feature map $c = [c_1, c_2, \dots, c_{t-k+1}]$, where $c \in R^{T-k+1}$ using the following:

$$c_j = f(mw_j + b) \quad (2)$$

where b is a bias term and f is the ReLU nonlinear transformation function. The max-pooling operation was implemented over the feature map to capture the significance feature as this operation is frequently employed, and the pooling window includes the maximum value element. Furthermore, n distinct filters are used to execute convolution operations, and all feature maps are concatenated into a single vector $s \in R^n$ to represent the text input.

LSTM Model. LSTM is a kind of Recurrent Neural Networks (RNN) with memory cells that can recall sequence elements over a long time to capture longer-term dependencies. LSTM is used in this study as it can handle the vanishing gradient problem, making it suitable for text classification. LSTM [26] saves long-term dependencies using three gates effectively, input gates, forget gates, and output gates, to control the flow of information through the network. LSTM takes as input x_t, h_{t-1}, c_{t-1} of the $t - th$ word in the input text and produces h_t, c_t based on the following formulas:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) : Input\ gate \quad (3)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) : \text{Forget gate} \quad (4)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) : \text{Output gate} \quad (5)$$

$$\tilde{c}_t = \tanh(W_c x_t + U_i h_{t-1} + b_c) : \text{New memory cell} \quad (6)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t : \text{Final memory cell} \quad (7)$$

$$h_t = o_t \cdot \tanh(c_t) \quad (8)$$

In the above equation, b represents the bias vector, W is used for weight, and x_t is the input vector at the current time step t , whereas f , c_t and o represent input, forget, cell memory and output gates. \tilde{c} is the current cell state; the symbols $\sigma(\cdot)$ and $\tanh(\cdot)$ represent sigmoid activation function and hyperbolic tangent function, and \cdot denotes matrix multiplication.

CNN-LSTM Model. In this CNN-LSTM architecture, the outputs from the CNN layers were channeled into LSTM networks, renowned for their capacity to capture temporal dependencies and long-range contextual insights.

2.3 Output Layer

The probability distribution over all possible categories is provided in the output layer.

3 Experimental Setup

The experiments are divided into several parts based on the tokenization level and the model used. The same experimental setup is used for all text embeddings and tokenization levels.

CNN Architecture Details. (1) Input Layer: one input layer accepting text embeddings of the tokenized tweets. (2) Convolutional Layers: three convolutional layers with filters of varying sizes (e.g., 3×3 , 4×4 , 5×5) to extract different local features from the text. (3) Pooling Layers: three pooling layers, each following a convolutional layer, for dimensionality reduction and feature extraction. (4) Fully Connected Layers: two fully connected layers following the pooling layers to interpret the features for classification. (5) Output Layer: a softmax output layer with units corresponding to the number of classification categories.

LSTM Architecture Details. (1) Input Layer: An input layer that takes in sequences of text embeddings corresponding to the tokenized tweets. (2) LSTM Layers: One or more LSTM layers with 128 LSTM units (neurons). These layers can capture the dependencies and context within the sequence of words in a tweet. (3) Dropout Layers: Dropout is applied to the LSTM layers to prevent overfitting at a rate of around 0.2 to 0.5. (4) Fully Connected Layers: After the LSTM layers, one or more dense layers with an

activation function ReLU are used for further non-linear data processing. (5) Output Layer: A softmax output layer with a number of neurons equal to the number of classes, providing the probability distribution over all possible categories.

CNN-LSTM Architecture Details. (1) Input Layer: Accepts the sequence of text embedding embeddings for each tokenized tweet. (2) Convolutional Layers: One or more 1D convolutional layers to extract local features from the embeddings. The filter size 5 is used to capture n-gram features. (3) Pooling Layers: Max pooling layers follow the convolutional layers to reduce the data's dimensionality and keep only the most salient features. (4) LSTM Layer: An LSTM layer is used to process the sequential data after the convolutional and pooling layers. (5) Fully Connected Layers: Dense layers follow the LSTM layer, typically with a dropout layer in between to reduce overfitting. (6) Output Layer: A softmax layer to classify the tweets into various categories, outputting a probability distribution over the categories.

4 Dataset and Results

This study used a well-known dataset published by [27]. This dataset is the only dataset differentiating between hate speech, offensive language, and neither. Most studies in the same field use this dataset to develop the HSOFC detection model, which is publicly available. The dataset contains 24 783 English tweets: 1430 hate speech tweets, 19 190 offensive language tweets, and 4163 neither. These figures show that offensive language is the majority class in the sample, accounting for 77.4% of the dataset, whereas hate speech accounts for only 5.8%. Using the standard value, the HSOFC dataset is divided into 70% for training and 30% to be accessed for testing. The results for each experiment reported using the standard Precision (P), Recall (R), and F1-score (F1) and accuracy (Acc) are discussed.

4.1 Results and Discussion

The neural network architecture used for these experiments was CNN, LSTM, and CNN-LSTM, performed on the HSOFC dataset containing three classes (hate speech, offensive language, and neither). Table 1 shows the experiment's result for each text embedding technique at different tokenization levels for CNN, LSTM, and CNN-LSTM.

Based on the result in Table 1, GloVe at character level with CNN-LSTM achieved the best result overall among all tested methods. The scores obtained by GloVe (character level) were 93% for F1-score and 92% for accuracy. At the word level, the BERT word embedding layer with CNN-LSTM obtained the best classification scores of 90% for F1-score and 91% for accuracy. Also, at the subword level, BERT word embeddings performed the best for CNN-LSTM and CNN, with a 86% score for accuracy and 86% for the F1-score of CNN.

The results show that each pre-trained embedding at different tokenization captured different information, resulting in different results for the same neural network architecture at different tokenization levels. Therefore, all pre-trained embedding was used to investigate the best combination for HSOFC detection. Moreover, CNN-LSTM yielded

the best score for almost all text embedding regardless of the tokenization level compared to CNN and LSTM. These results show that CNN-LSTM complement each other to capture sequential and local patterns in the input text.

Table 1. Performance of pre-trained text embedding at different tokenization levels for CNN, LSTM, and CNN-LSTM.

Level	Embedding	1_CNN (%)				2_LSTM(%)				3_CNN-LSTM(%)			
		P	R	F1	Acc	P	R	F1	Acc	P	R	F1	Acc
Word	Word2-Vec	83	84	83	84	83	81	81	81	85	83	83	83
	GloVe	86	87	86	87	85	80	80	80	89	90	89	90
	FastText	86	86	86	86	89	86	86	86	89	86	86	86
	Elmo	77	78	77	78	82	83	89	80	87	86	87	84
	BERT	84	85	84	85	50	55	52	55	90	91	90	91
Subword	GloVe	74	81	79	85	88	80	83	80	76	81	78	81
	FastText	74	76	75	76	60	61	59	61	84	86	84	86
	BERT	86	86	86	86	43	48	45	48	80	86	83	86
Character	Glove	82	83	82	83	74	70	90	88	88	80	93	92
	FastText	83	83	83	83	78	81	86	79	72	78	72	78

5 Conclusion

This study investigated the best pre-trained text embedding for each tokenization level and identified the best neural network architecture as an encoding method for hate speech and offensive language detection. To achieve this objective, the architecture of several models was designed. Based on accuracy results, CNN-LSTM is found to be the most effective, incorporating significant features such as GloVe (word level), FastText (word level), Elmo (word level), BERT (word level), FastText (subword level), BERT (subword level), and GloVe (character level). In this work, different text encoding techniques are compared together with several types of network architecture. For future work, more neural architecture, such as Generative Adversarial Networks (GANs) and Multilayer Perceptrons (MLPs), can be explored. Moreover, there is a bias issue in the dataset caused by the imbalance. Solving the imbalanced dataset by merging the dataset with other related hate speech concept datasets is inaccurate as each dataset has a different set of rules to annotate the data. Therefore, more research is needed to prepare the dataset for this domain. Besides that, not all words in a text contribute equally to detecting hate speech. Therefore, attention mechanisms such as self-attention can be explored. This model can selectively focus on relevant information that assists in detecting hate speech and offensive language.

References

1. Twitter: “Twitter Rules and Policies (2020). [Online]. <https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy>. Accessed 30-Dec-2020
2. Facebook: Community Standards (2020). [Online]. https://www.facebook.com/community_standards/hate_speech. Accessed: 30-Dec-2020
3. YouTube: Community Guidelines (2020). [Online]. <https://support.google.com/youtube/answer/2801939?hl=en>. Accessed: 30-Dec-2020
4. Bengio, Y., Ducharme, R., Vincent, P.: A neural probabilistic language model. *J. Mach. Learn. Res.* **3**(6), 1137–1155 (2003)
5. Zulqarnain, M., Ghazali, R., Mazwin, Y., Hassim, M., Rehan, M.: A comparative review on deep learning models for text classification. *Indones. J. Electr. Eng. Comput. Sci.* **19**(1), 325–335 (2020)
6. Fesseha, A., Xiong, S., Emiru, E.D., Diallo, M.: Text classification based on convolutional neural networks and word embedding for low-resource languages : Tigrinya. *Information* **12**(2), 52 (2021)
7. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: 1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings, pp. 1–12 (2013)
8. Pennington, J., Socher, R., Manning, C.D.: GloVe : Global Vectors for Word Representation, pp. 1532–1543 (2014)
9. Zhong, Q.: Bag of Tricks for Effective Language Model Pretraining and Downstream, vol. 5 (2022)
10. Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations (2016)
11. Geet, A., Illina, I., Fohr, D., Geet, A., Illina, I., Bert, D.F.: BERT and fastText embeddings for automatic detection of toxic speech. In: IIE2020—Information Systems and Economic Intelligence (2020)
12. Dorris, W., Hu, R.R., Vishwamitra, N., Luo, F., Costello, M.: Towards automatic detection and explanation of hate speech and offensive language. In: IWSPA 2020—Proceedings of the 6th International Workshop on Security and Privacy Analytics, pp. 23–29 (2020)
13. Bisht, A., Singh, A., Bhaduria, H., Virmani, J., Kriti.: Detection of hate speech and offensive language in twitter data using LSTM model. In: Recent Trends in Image and Signal Processing in Computer Vision, pp. 243–264. Springer (2020)
14. Mozafari, M., Farahbakhsh, R., Crespi, N.: A BERT-based transfer learning approach for hate speech detection in online social media. In: Proceedings of the Eighth International Conference on Complex Networks and Their Applications Complex Network 2019, vol. 881, pp. 928–940 (2019)
15. Mahdi, S., Rahmani, R., Ghodsi, A., Veisi, H.: Sentiment analysis based on improved pre-trained word embeddings. *Expert Syst. Appl.* **117**, 139–147 (2019)
16. Parcheta, Z., Sanchis-Trilles, G., Casacuberta, F., Rendahl, R.: Combining embeddings of input data for text classification. *Neural Process. Lett.* (2020)
17. Alharbi, A.I., Lee, M.: Combining character and word embeddings for the detection of offensive language in Arabic. In: Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, , May, pp. 11–16 (2020)
18. Biere, S.: Hate Speech Detection Using Natural Language Processing Techniques. Vrije Univ. Amsterdam, p. 30 (2018)
19. Bisht, A., Singh, A., Bhaduria, H., Virmani, J., Kriti.: Detection of hate speech and offensive language in twitter data using LSTM model. In: Recent Trends in Image and Signal Processing in Computer Vision, pp. 243–264. Springer, Singapore (2020)

20. Zhang, Y., Zheng, J., Jiang, Y., Huang, G., Chen, R.: A text sentiment classification modeling method based on coordinated CNN-LSTM-attention model. *Chin. J. Electron.* **28**(1), 120–126 (2019)
21. Almuzaini, H.A., Azmi, A.M.: Impact of stemming and word embedding on deep learning-based arabic text categorization. *IEEE Access* **8**, 127913–127928 (2020)
22. Wang, Y., Ma, K., Garcia-hernandez, L., Chen, J., Hou, Z., Ji, K.: A CLSTM-TMN for marketing intention detection. *Eng. Appl. Artif. Intell.* **91**(2020)
23. Rezaeinia, S.M., Rahmani, R., Ghodsi, A., Veisi, H.: Sentiment analysis based on improved pre-trained word embeddings. *Expert Syst. Appl.* **117**, 139–147 (2019)
24. García-Díaz, J.A., Jiménez-Zafra, S.M., García-Cumbreras, M.A., Valencia-García, R.: Evaluating feature combination strategies for hate-speech detection in Spanish using linguistic features and transformers. *Complex Intell. Syst.* **9**(3), 2893–2914 (2023)
25. Kim, Y.: Convolutional neural networks for sentence classification, arXiv Prepr. arXiv1408.5882 (2014)
26. Graves, A., Graves, A.: Long short-term memory. Supervised Seq. Label. with Recurr. neural networks, pp. 37–45 (2012)
27. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language. In: Proc. 11th Int. Conf. Web Soc. Media, ICWSM 2017, pp. 512–515 (2017)



Improved Classifier Chain Method Based on Particle Swarm Optimization and Genetic Algorithm for Multilabel Classification Problem

Abdullahi O. Adeleke¹(✉), Noor A. Samsudin², Shamsul Kamal A. Khalid², and Riswan Efendi³

¹ Department of Computer Science, Crescent University Abeokuta, Sapon, Abeokuta 2104, Ogun State, Nigeria

abdullahi.adeleke@cuab.edu.ng

² Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia

{azah.shamsulk}@uthm.edu.my

³ Faculty of Science and Technology, UIN Sultan Syarif Kasim Riau, Pekanbaru 28293, Indonesia

riswan.efendi@uin-suska.ac.id

Abstract. Classifier chain (CC) is one of the most important MLC methods often applied in multi-labeling tasks. However, the standard CC model fails to recognize distinct labels sequence order in the chain. In this work, a new method is proposed to optimize the chaining order in CC for improved classification performance. The proposed method is based on particle swarm optimization and genetic algorithm. Genetic operators are integrated with the standard PSO algorithm for finding the global best solution representing an optimized label sequence order in the chain. In the experiment, 6 datasets and 8 evaluation metrics are employed. The new method achieved the overall best results including 98.66% accuracy and 97.49% exact match among other metrics applied.

Keywords: Multilabel Text Classification · Heuristic Techniques · Classifier Chain · Particle Swarm Optimization · Genetic Algorithm · Binary Relevance

1 Introduction

Text classification is the task of automatically sorting a set of documents into categories from a predefined set [1, 2]. Essential to classification problems is the prediction output. A classifier is evaluated based on its predictive performance. Significantly, a high performance reflects good efficiency of a classification model. This research work attempts to improve the predictive performance of the standard multi-label classifier chain (CC) method by proposing hybrid heuristic evolutionary based technique. In the experiment, the improved multi-label CC method will be used to find optimized label sequence order in the chain that represents the best solution to the classification problem.

The classifier chain (CC) algorithm is a popular and widely applied MLC method that has the advantage of taking into consideration label correlations which helps to achieve good classification results. This research work attempts to address one of the issues associated with the standard multi-label classifier chain method which is the random label sequence ordering (RLSO) problem [3–5].

1.1 Motivation

In CC, the classifiers are linked together in a chain-like structure allowing for label dependencies i.e., the true label information (prediction) of previous classifiers in the chain are communicated to the next classifier to make prediction of its equivalent label from the set of predefined labels. This label correlation (or dependency) helps to find relevant categories to which instances of data samples belong. This capability gives CC a strong competition over other multi-label classification (MLC) methods such as binary relevance. The method is distinguished due to its simplicity and effective approach to exploit label dependencies.

1.2 Random Label Sequence Ordering (RLSO)

One major factor influencing the performance of the classifier chain method is the chaining order [4, 6]. The chaining order concerns the sequence (pattern) in which the underlying labels are predicted. The classical CC method fails to recognize a distinct chain order i.e., the algorithm does not attempt to optimize the label sequence order in the chain. The original CC makes use of the chaining method in predicting, taking into consideration label dependencies which could help to improve the classification performance. However, the method does not manage to model the chain optimally, rather it adopts random approach to generate the chain order. The first label in the chain is modeled and predicted without considering other labels predictive information, whereas the last label in the sequence order is predicted considering all information from the previous classifiers. Studies [5, 6] have shown that the random label sequence ordering (RLSO) of the classifiers has strong effect on the classification performance. There is potential influence of error propagation along the chain at prediction time when one (or more) of the first set classifiers predict wrongly. This study proposed a new approach to tackle the RLSO issue with the use of hybrid heuristic evolutionary-based algorithms. A new method was developed based on swarm intelligence and genetic algorithm (GA). The new method combined particle swarm optimization (PSO) and GA. The proposed method attempts to overcome the RLSO limitation of the standard CC method.

2 Related Work

The work of [7] proposed a deep neural architecture using bidirectional association-based pooling layers for extracting high-level features and labels for multi-label data. Pearson correlation method was used in accessing the association values. The work used an iterative approach to estimate the degree of association among the neurons. The results proved that the proposed bidirectional neural network significantly reduced the features and labels sets.

The work of [8] proposed a multi-label classification-based ensemble learning for human activity recognition. The proposed approach is based on the application of multi-label classifier chain (CC) associated with four single-label baseline classifiers: Bernoulli naïve bayes, decision trees, logistic regression, and k-nearest neighbor. In addition, a majority voting ensemble classifier method was employed for the recognition task. The results demonstrated that multi-label CC method can effectively handle both complex problems and activity recognition tasks.

There are a number of available existing extensions of the standard CC method [5]. The extended CC variations mostly centered on addressing the random label sequence ordering issue of CC but with limitations particularly complexity. The ensembles of classifier chains (ECC) approach is used to combine random orders to improve the standard CC. The ensembles strategy requires repeating the execution over a number of times until desirable results achieved. The approach could provide accurate CC models but ensembles are computationally exhaustive [5].

3 Method

3.1 Dataset

The datasets as tabulated in Table 1 are from the most commonly applied data used in multi-label classification problems. The proposed classification model is evaluated on 6 of the most widely applied multi-label datasets. These are: enron, medical, genbase, LLOG, yeast, and slashdot datasets.

Table 1. Benchmark ML datasets (with D = no of features; Q = no of class labels; lc = label cardinality).

Dataset	Domain	#Instances	D	Q	lc
Enron	Text	1702	1001	53	3.378
Medical	Text	978	1449	45	1.245
LLOG	Text	1460	1004	75	1.180
slashdot	Text	3782	1079	22	1.181
Genbase	Text	662	1186	27	1.252
Yeast	Text	2417	103	14	4.237

3.2 Data Preprocessing

Feature generation is an important step in data preprocessing which involves the process of extracting features (or keywords) from a normalized data. For the experimental work, the data is first converted to Attribute-Relation File Format (ARFF). Two standard pre-processing techniques: String to Word Vector and TF-IDF are employed in generating and preprocessing features from the textual data.

3.3 Classification (Proposed Model)

The proposed multi-label classification model is based on the use of heuristic evolutionary-based optimization search techniques for finding an optimized label sequence in the chain classifier. The proposed PSOGCC is hybrid of particle swarm optimization (PSO) and genetic algorithm (GA). The model combines the simplicity of PSO with the global explorative capability of GA.

3.3.1 Proposed PSOGCC Description

The process could be broadly categorized into two parts: the first part begins by initializing population of individuals (particles) randomly with an appropriate population size s , position x , and velocity v (peculiar with PSO). A particle is a potential candidate solution representing a label sequence order in the classifier chain. Path representation encoding strategy is used to represent individuals in the population as q -dimensional vectors (where q = number of labels). Each individual is encoded as integer numbers (denoting label sequence indexes) in the range $[1, q]$. In PSOGCC, the particle's previous best position $pbest$ is initialized with a copy of its current position (x). A fitness function (x) defined in Eq. (1) is used to measure the particle's fitness. Global best $gbest$ is initialized with the index of the best (fit) particle. Subsequently, the best solution (in the current population) is passed to GA for the global search.

In the second part, genetic operators of GA are applied, evolving over a repeated loop until a maximum iteration (number of generations) has been reached. The GA phase is further segmented into two sub phases: fitness evaluation (x) (using Eq. (2)) of the current population, and generation of a new population using genetic operators: selection, crossover, and mutation. To compute the fitness of the current population, the study adopted the use of wrapper-based approach. Classifier chain (CC) model is built and evaluated, one for each individual (label sequence). The CC model is built (from the training set T) using the *buildSet*, then subsequently validated using the *validationSet*. Next is parent selection, a step in the evolutionary cycle responsible for defining the individuals in the current generation that will be combined in order to produce new individuals (offspring). Tournament selection approach is employed for selecting the parents (representing the best fit individuals) in the population. The selection strategy is the most adopted of the parent selection techniques.

In this approach, as aforementioned, the proposed PSOGCC randomly choose k individuals from the set population, where k is a user-defined parameter called the tournament size. The tournament size parameter helps to control the rate at which individuals are selected. The idea of the tournament selection strategy is to ensure that the best fitted individuals (candidate solutions) from the population have higher probability of being selected as parents which then necessitates transfer of their genetic material to later generations, hence adding quality to the potential solution to the search problem. The selected k individuals undergo a process synonymous to playing a sport tournament in nature. The process is based on comparing the individuals' fitness values, and the winner (of the tournament) is the individual with the best fitness value which subsequently will reflect (at the end of the cycle) a single optimized label sequence order in the chain. Thereafter, the crossover operation is performed using order crossover (OC)

approach, which resulted in the generation of new population (offspring). The offspring set is subjected to mutation. In order to guarantee global optimum solution, the mutation operator helps to avoid being trapped in the local minima, hence maintains diversity in the population. Consequently, the population is updated replacing the old population (parents) using age-based elitism replacement method, which helps to preserve a small group of elite individuals in the population. This also promotes improved diversity in the population leading to a global best solution. Finally, the program ends with the termination criteria and then returns the global optimum solution $optmL$, which is the (most fitted) candidate solution with the best fitness quality (in the population) representing an optimized label sequence in the classifier chain (Fig. 1).

$$Fitnessf(x) = (\alpha * \text{Acc}) + (\beta * \frac{(N - T)}{N}) \quad (1)$$

Algorithm: Pseudocode of the Proposed PSOGCC Method	
Input:	T (training set)
Output:	$optmL$ (an optimized label sequence in CC)
Step 1:	Initialize population of particles (potential candidate solutions representing the label sequences) with random positions x and velocities v , swarm size s . Set the particle's previous best position to the current position ($pbest = x$)
Step 2:	Given a training set T ; partition the training set T into $buildSet$ and $validationSet$
Step 3:	<i>For</i> all particles (candidate label sequence i) in the current population <i>do</i>
Step 4:	Construct the CC model using $buildSet$ and label sequence i
Step 5:	Compute the fitness (quality) of the CC model using the $validationSet$ and the fitness function $f(x)$ defined in E (1)
Step 6:	Update the population $pbest$ and set the best particle $gbest$ to the current population
Step 7:	<i>end For</i>
Step 8:	Improve the final solution (current population) of the PSO using GA
Step 9:	<i>For</i> all individuals in the current population <i>do</i>
Step 10:	Evaluate the fitness (quality) of the solution using the fitness function $g(x)$ defined in E (2)
Step 11:	Apply Genetic operators:
Step 12:	Using Tournament selection approach <i>SELECT</i> parents (particle with best fitness value)
Step 13:	Generate new particles (child) from the old ones (parents) with <i>Crossover</i> operator
Step 14:	Apply <i>Mutation</i> procedure (to the offspring)
Step 15:	Update particles and population using <i>elitism age-based replacement</i> approach
Step 16:	<i>end For</i>
Step 17:	Until (<i>Max iterations (Number of generations) reached</i>)
Step 18:	return $optmL$ (global optimum solution rep labeled-ordered sequence)

Fig. 1. The Proposed PSOGCC Method.

(α and β) are control parameters for balancing the trade-offs of the best and worst solution in the population; Acc represents the accuracy of the baseline classifier; while (N and T) represent population size and neighborhood size respectively.

$$Fitness\ g(x) = \frac{\left(1 - \left(\frac{HL}{gMean}\right)\right) + Acc + EM + F1}{N} \quad (2)$$

where HL , $gMean$, Acc , EM , and $F1$ are performance metrics (for validating the quality of a method or algorithm).

3.4 Performance Measures

To validate the performance of the proposed multi-label classification model, the study employed 8 standard evaluation metrics commonly used in multi-label classification problems. In addition, the study compared the performance of the improved CC method with the standard classifier chain (CC) and binary relevance methods. The performance evaluators employed in this work include: Accuracy, ExactMatch, HammingLoss, GeometricMean, Precision, Recall, F -measure, and Degree of Balance. The target goal of this research work is to generally improve the predictive performances of the multi-label classifier chain in terms of the above metrics.

4 Results and Discussion

The datasets experimented are of the text application domain. The benchmark datasets include: enron, medical, genbase, LLOG, yeast, and slashdot multi-label datasets. Tables 2, 3, 4, 5, 6, 7, 8, 9 report the results of the multi-label classification methods in terms of Accuracy, ExactMatch, Precision, Recall, F-Measure, HammingLoss, GeometricMean, and Degree of Balance, respectively. The best results are highlighted in bold across the experimental datasets.

Table 2. Predictive Performance of PSOGCC, BR, and CC in terms of Accuracy.

Datasets	PSOGCC	BR	CC
enron	0.4046	0.3671	0.3671
yeast	0.4537	0.4226	0.4219
genbase	0.9866	0.9806	0.9815
medical	0.7463	0.7426	0.7425
LLOG	0.2227	0.2342	0.2219
slashdot	0.3786	0.2860	0.2961

Table 3. Predictive Performance of PSOGCC, BR, and CC in terms of ExactMatch.

Datasets	PSOGCC	BR	CC
enron	0.1261	0.0864	0.0864
yeast	0.1439	0.0643	0.0641
genbase	0.9749	0.9614	0.9734
medical	0.6667	0.6512	0.6509
LLOG	0.1868	0.1982	0.1981
slashdot	0.3204	0.2306	0.2741

Table 4. Predictive Performance of PSOGCC, BR, and CC in terms of HammingLoss.

Datasets	PSOGCC	BR	CC
enron	0.0535	0.0540	0.0542
yeast	0.2642	0.2588	0.2579
genbase	0.0011	0.0121	0.0102
medical	0.0105	0.0106	0.0103
LLOG	0.0211	0.0209	0.0210
slashdot	0.0619	0.0488	0.0394

Table 5. Predictive Performance of PSOGCC, BR, and CC in terms of Precision.

Datasets	PSOGCC	BR	CC
enron	0.5959	0.6574	0.6576
yeast	0.5796	0.5929	0.5950
genbase	0.9950	0.9947	0.9950
medical	0.8778	0.8729	0.8730
LLOG	0.6885	0.6803	0.6810
slashdot	0.8159	0.8119	0.8005

Table 6. Predictive Performance of PSOGCC, BR, and CC in terms of Recall.

Datasets	PSOGCC	BR	CC
enron	0.4858	0.4481	0.4483
yeast	0.6008	0.5613	0.5616
genbase	0.9916	0.9903	0.9908
medical	0.7845	0.7941	0.7940
LLOG	0.2436	0.2544	0.2539
slashdot	0.4010	0.3135	0.3138

Table 7. Predictive Performance of PSOGCC, BR, and CC in terms of F-Measure.

Datasets	PSOGCC	BR	CC
enron	0.5352	0.5329	0.5331
yeast	0.5900	0.5767	0.5778
genbase	0.9933	0.9925	0.9928
medical	0.8285	0.8316	0.8316
LLOG	0.3599	0.3703	0.3699
slashdot	0.5377	0.4523	0.4509

Table 8. Predictive Performance of PSOGCC, BR, and CC in terms of GeometricMean.

Datasets	PSOGCC	BR	CC
enron	0.6091	0.5777	0.5780
yeast	0.6495	0.6405	0.6412
genbase	0.9947	0.9939	0.9941
medical	0.8015	0.8106	0.8101
LLOG	0.2557	0.2676	0.2548
slashdot	0.4159	0.3256	0.3652

Table 9. Predictive Performance of PSOGCC, BR, and CC in terms of Degree of Balance.

Datasets	PSOGCC	BR	CC
enron	0.5092	0.4732	0.4730
yeast	0.5792	0.5515	0.5510
genbase	0.9917	0.9806	0.9910
medical	0.7936	0.8020	0.8020
LLOG	0.2805	0.2909	0.2900
slashdot	0.4276	0.3450	0.3489

From the obtained results, the proposed PSOGCC method was evaluated and compared against standard binary relevance (BR) and classifier chain (CC). The experimenting datasets comprised of 6 benchmark datasets. The ultimate goal of the study is to improve the predictive performance of the multi-label classifier chain method. The results were measured using 8 conventional performance metrics.

Working with the benchmark multi-label datasets produced competitive results. Measuring the Accuracy, ExactMatch, HammingLoss, Precision, Recall, *F*-Measure, GeometricMean, and Degree of Balance are shown in Tables 2, 3, 4, 5, 6, 7, 8, 9 respectively. From the results in Table 2, the proposed PSOGCC had the best performance with PSOGCC achieving the overall highest accuracy value of 98.66% using the genbase multi-label dataset. Excellently, the participating multi-label classifiers: *PSOGCC*, *BR*, and *CC*, obtained above 98% accuracy. These outstanding performances of the classification algorithms showed the classifiers significantly achieved good accuracy results with the genbase dataset. In addition, the classification results showed the algorithms navigate better with the multi-label datasets. However, the multi-label classification methods performed poorly with LLOG dataset. This is largely due to the nature of the dataset. LLOG was primarily used to categorize texts into topics.

The predictive performance of the multi-label classification methods in terms of ExactMatch and HammingLoss as reported in Tables 3 and 4 again reflected satisfactory results. The proposed PSOGCC obtained the best values of 97.49% in terms of ExactMatch and 0.0011 in terms of HammingLoss with the genebase dataset. The original classifier chain (CC) method came in the second position obtaining ExactMatch score of 97.34% and 0.102 HammingLoss value. This explains the limitation associated with the standard binary relevance (BR) method. The BR method ignores labels correlations which often could influence *negatively* the performance of the multi-label classifier. Furthermore, the results obtained by the original CC method, a direct extension of the BR method proposed to address the above said limitation (of BR) have proven to be efficient. The extended method outperformed the standard BR method. Finally, the proposed multi-label classification method, also an extension of the original CC method likewise proved their efficiency over the standard BR and CC methods. This is due to the advantage of the proposed approach strategy in combining the simplicity and good knowledge sharing capability of the swarm-based algorithms with the natural evolution-like process of genetic algorithm which helps to avoid premature convergence in the

search space. Hence, by so doing, promotes population diversity until the global optima solution to the given problem is achieved.

In Tables 5, 6, and 7, the measures of the multi-label classifiers were reported in terms of Precision, Recall, and *F*-Measure respectively. The combination of these performance metrics reflects the efficiency (*how good*) of the classifiers. Consistently, the proposed method achieved the overall best precision, recall, and *f*-measure scores with PSOGCC obtaining the overall best scores of 99.5%, 99.16%, and 99.32% respectively. Closely followed by the standard CC method jointly (with the proposed PSOGCC) obtained the highest precision score of 99.5%, but dropped in terms of recall and *f*-measure recording 99.08% and 99.28% values respectively. These further reiterate the aforementioned why the standard CC and proposed method outperformed the original BR method.

In addition, the decline (in performance) of the standard CC compared to the proposed method re-emphasized on the main drawback (of CC). Its (CC) approach of random label sequence order of classifiers in the chain could have had effect on the algorithm's performance. PSOGCC swarm-based method was proposed to address this random label-sequence issue (associated with the standard CC). As reported in the results, the hybrid evolutionary-based approach outperformed both classifier chain (CC) and binary relevance (BR) multi-label classification methods. This is due to the approach designed by the improved classifier chain method in finding the global solution (*optmL*) representing an optimized label sequence of the classifiers in the chain. Thus, the proposed approach achieved its dedicated goal of improving the predictive performance by achieving the best precision, recall, and *f*-measure scores.

In terms of geometricMean (*gMean*) and Degree of Balance (*dB*), the results are recorded in Tables 8 and 9 respectively. The proposed PSOGCC method achieved the best results of 99.47% (*gMean*) and 99.17% (*dB*) with the genbase dataset. These further showed the effectiveness of the proposed multi-label classification method. Generally, the multi-label classifiers performed satisfactorily achieving above 99% (*gMean*) and 98% (*dB*) values.

5 Conclusion

The study is based on proposing improved multi-label classifier chain (CC) method applicable for multi-label classification problems. The proposed method clearly addressed the aim of the research, which was to generally improve the predictive performance of the standard multi-label classifier chain by addressing the RLSO issue. The new method enabled the searching of a distinct single label sequence order in the chain for multi-label classification problems. The main idea of the proposed classification model relies on the significance of modeling classifiers for multi-label predictions with a single label sequence order in the chain rather than the traditional random label sequence ordering. The experimental results showed that the proposed method performed better than the standard CC method. In the future, the proposed approach will be extended to other multi-label classification problem domains such as image, multimedia, and big data. In addition, the research recommends focusing on proposing new methods and techniques to address several of the multi-label classification (MLC) issues.

References

1. Dogan, T., Uysal, A.K.: A novel term weighting scheme for text classification: TF-MONO. *J. Infometr.* **14**, 101076 (2020)
2. Gargiulo, F., Silvestri, S., Ciampi, M., De Pietro, G.: Deep neural network for hierarchical extreme multi-label text classification. *Appl. Soft Comput. J.* **79**, 125–138 (2019)
3. Al-Salemi, B., Ayob, M., Kendall, G., Noah, S.A.M.: Multi-label Arabic text categorization: a benchmark and baseline comparison of multi-label learning algorithms. *Inf. Process. Manag.* **56**, 212–227 (2019)
4. Jun, X., Lu, Y., Lei, Z., Guolun, D.: Conditional entropy based classifier chains for multi-label classification. *Neurocomputing* **335**, 185–194 (2019)
5. Read, J., Martino, L.: Probabilistic regressor chains with Monte Carlo methods. *Neurocomputing* **413**, 471–486 (2020)
6. Moyano, J.M., Gibaja, E.L., Cios, K.J., Ventura, S.: Review of ensembles of multi-label classifiers: Models, experimental study and prospects. *Inf. Fusion* **44**, 33–45 (2018)
7. Bello, M., Nápoles, G., Sánchez, R., Bello, R., Vanhoof, K.: Deep neural network to extract high-level features and labels in multi-label classification problems. *Neurocomputing* **413**, 259–270 (2020)
8. Jethanandani, M., Sharma, A., Perumal, T., Chang, J.-R.: Multi-label classification based ensemble learning for human activity recognition in smart home. *Internet Things* **12**, 100324 (2020)



Sentiment Analysis on Umrah Packages Review in Malaysia

Deshinta Arrova Dewi¹(✉) ID, Tri Basuki Kurniawan² ID, Mohd Zaki Zakaria³, Shahreen Kasim⁴ ID, and Nur Qasheeh Mustapa³

¹ INTI International University, Jalan Persiaran Perdana Bandar Baru, 71800 Nilai, Malaysia
deshinta.ad@newinti.edu.my

² Universitas Bina Darma, Jl. Jenderal Ahmad Yani No.3, 9/10 Ulu, Palembang, Indonesia

³ Universiti Teknologi Mara (UiTM) Shah Alam, Jalan Ilmu 1/1, 40450 Shah Alam, Selangor, Malaysia

⁴ Universiti Tun Hussain Onn Malaysia, Jalan Persiaran Tun Dr. Ismail, 86400 Parit Raja, Johor, Malaysia

Abstract. Umrah is a well-known pilgrimage site where individuals from around the world come together to receive good fortune from God in the here and now. In Malaysia, there are numerous sorts of Umrah packages promoted by many firms such as Andalusia Travel & Tours, Tiram Travel & Tours, Tabung Haji, and beyond. Because there are so many organizations offering Umrah trips, there are issues when careless people or organizations encourage fraud in Umrah packages to benefit from it. In 2023, the Police in Malaysia have confirmed that they have received 399 allegations regarding suspected Umrah package scams by a local company. The reports include 1,614 pilgrims around the country who are believed to have lost approximately RM14 million. With the world's technological advancements happening at a quick pace, individuals are more comfortable sharing their ideas and opinions on touchy themes on social media, including the experience with the fraud Umrah package. Performing a classification study from these resources develops the sentiment analysis against the available Umrah package. This classification approach can be used in searching applications to assist people in finding better Umrah packages. In this paper, Naïve Bayes, Random Forest, and Support Vector Machine are the classifiers that were employed in the study. The outcomes demonstrate that, out of all the classifiers, the Support Vector Machine has the highest accuracy. The Support Vector Machine (SVM) had a 90.92% accuracy rate. Ninety percent of the results were precise. It displays 90% overall for the f1-score and 91% overall for the total recall. Compared to Naïve Bayes, SVM has greater accuracy.

Keywords: Umrah Sentiment Analysis · Process Innovation · Reviews · Text Analysis

1 Introduction

As Muslims, five key pillars need to be believed, which are a declaration of faith (Shahada), prayers five times a day (Salat), fasting in Ramadan (Saum), purifying tax (Zakat), and religious-related travel, which is a pilgrimage to Makkah (Hajj). There are two types

of religious-related travel, which are Hajj and Non-Hajj (Almuhrzi & Alsawafi, 2017) [1]. Non-Hajj is also known as Umrah. Haq & Jackson (2009) cited by Almuhrzi & Alsawafi (2017) [1] stated Hajj and Umrah are considered as forms of pilgrimage, where Muslims around the world gather to perform to get a good fortune in the future.

Nowadays, religious tourism has become public attention in Malaysia. [2] Majlis Kawal Selia Umrah (MKSU) is a government initiative set up under the Tourism Industry Act 1992 to monitor the implementation of service standards for tour operators and travel agencies in handling Umrah packages. MKSU had issued that the minimum price for an Umrah package should not be less than RM4,900 for the Umrah period of 12 days and 10 nights. Even though the price has been set, many things make a package a different price.

Among the different prices are the chosen dates, in which the peak season such as school holidays, or the end of Ramadan is usually more expensive, transit or continuing flights, frequency of getting food in a day, officer and Mutawwif, transportation, and the distance from accommodation to Haram. This somehow creates a difficult situation for customers to find better and legitimate Umrah packages with reasonable prices. (Vinodhini & Chandrasekaran, 2016) stated in their paper that sentiment analysis is the process of computationally identifying and classifying opinions to determine whether the text sentiment whether positive, natural, or negative [3]. Many people express their opinions on social media such as Facebook, Twitter, Blog, and webpages.

2 Sentiment Analysis on Social Media

These days people tend to post created content and spread it through social media in terms of pictures, videos, testimonials, tweets, blog posts, and everything to promote it rather than brand it. That is the definition of user-generated content, and it has happened since past few years ago (Balahur & Jacquet, 2015) [4]. The most often used social media networks are Twitter, Facebook, YouTube, and Instagram. According to Internetlivestat (2017) cited by Howells & Ertugan (2017), they found approximately 500 million tweets on Twitter per day, which means 6000 per second [5].

2.1 Related Works of Similar

This project reviews various reviews on social media based on different approaches such as Machine-Based Methods like Naïve Bayes, Support Vector Machine, Decision Tree, K-Nearest Neighbour, Deep Learning, or Lexicon-Based Method. Below are several examples of the existing journals for sentiment analysis (Table 1):

Table 1. List of related works

Journal Name	Method Used
Mobile Phone Reviews on Amazon	Support Vector Machine
Sentiment Classification of Online Consumer Reviews	Word Vector Representation
Twitter Sentiment Analysis	Convolutional Neural Network
Online Reviews of Hospitality Services	Naïve Bayes
Text Classification with Wikipedia-base Semantic	Naïve Bayes
Data Mining of Automatically Promotion Tweet for Products and Services	Naïve Bayes
YouTube Comments using Comparative Opinion Mining	Naïve Bayes

2.2 Mobile Phone Reviews from Amazon Using Support Vector Machine

This paper [6] used a lexicon-based approach to improve the classification accuracy for training data generation in the domain “Smartphone”. Since this paper is based on tweets as datasets, it found that there is a limitation in posting tweets because of limited words. Twitter users can type up to 140 words in one tweet. Therefore, people would use the short form in terms of words such as “gd” or “gud” instead of “good” to show a positive opinion. 400,000 reviews were collected from Amazon Mobile Phone reviews to generate the labeled training datasets. Emoji detection, slang words, identification of spelling mistakes, abbreviations, and slang words are the challenges they need to face when it involves opinions on tweets [6].

2.3 Sentiment Classification of Online Consumer Reviews Using Word Vector Representation

Word Vector or word2vec is used to convert reviews into vector representation for classification [7]. In this research paper, researchers used 400,000 consumer reviews in the mobile phone category from Amazon. There are two parts; they demonstrate word2vec to identify the same semantic features in the study and they classify the reviews by using CBOW (continuous bag of words) and skip-gram models with the dissimilar algorithms of machine learning such as Naïve Bayes, SVM, Logistic Regression and Random Forest using 10-folds cross-validation.

2.4 Online Reviews of Hospitality Services Using Naïve Bayes

Prabu (2014) cited by Sánchez-Franco, Navarro-García, & Rondán-Cataluña (2018) stated that travelers would choose and book a hotel that contains many reviews made by previous customers through social media [8]. Based on the statement, most travelers love to read reviews through social media as the information is important for them before start traveling to avoid getting problems along the journey since it is easier and available to access over time. In the research paper by Sánchez-Franco et al. (2018), they analyze

almost 47,172 reviews of hotels in Las Vegas, United States of America, which are registered under Yelp [8].

2.5 Customer Satisfaction Towards Umrah Travel Agencies in Malaysia

In the Umrah travel sector, most agencies provide their clients with some standard services, but obtaining a competitive advantage primarily depends on providing varied service quality, according to Othman et al. (2019) [10]. Customers have the option to select agents that provide high-quality services at comparatively lower costs because numerous organizations are providing Umrah travel services. Umrah travel brokers must increase their responsibilities and provide better services to be successful in the market. Another study by Alghamdi, H. M. (2024) [11] provides a thorough sentiment analysis of tweets over a six-year span that addresses the yearly Hajj journey. Using a large dataset of Arabic tweets, this study effectively performed sentiment analysis using a combination of machine learning and deep learning models. Pre-processing, feature extraction, and sentiment categorization are steps in the process.

3 Methodology

The methodology includes preliminary study, knowledge acquisition, data collection, data pre-processing, data analysis, architecture design, interface design, system development, and evaluation.

3.1 Preliminary Study

The preliminary study contains an initial discussion and steps about the project domain, the technique used, and the problem that occurs relating to the Umrah package with the supervisor. In this phase, there are a lot of sources which are viewed which are related to the project by online reading and searching the Internet such as online newspapers, articles, journals, and many more.

About 30 online readings that are reliable are being reviewed that are related to the project domain. Besides, it is also being searched on social media like Twitter, Instagram, Blogs, YouTube, and others to gain more insight as well. Various articles and journals are taken from reliable sources such as ScienceDirect, Google Scholar, and Scopus. The review about Umrah travel packages was gathered from Facebook and the website.

Since the main objective is to identify the review of the Umrah package in social media in Malaysia, it needs a tool such as Data Miner and Python to scrape the raw data and do the data processing. The process is to scrap and collect data which is the customer review from social media whether in the form of hashtags and comments. The result from data collection was converted and transformed into the proper form in Microsoft Excel with CSV file format to pre-process, which the phase was explained in detail in the next phase.

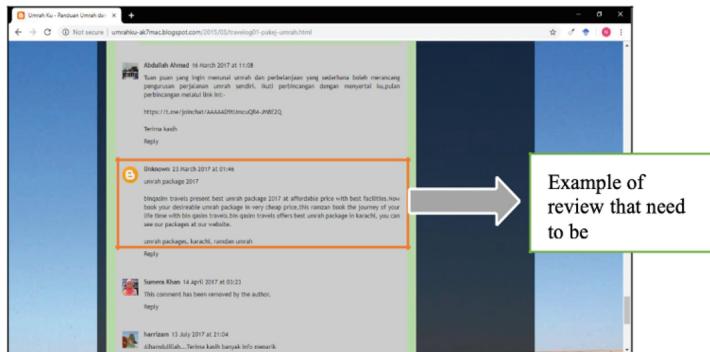


Fig. 1. The interface of Blogspot contains a review of the Umrah package.

Figure 1 shows a few reviews posted by unknown people regarding his/her opinions about Umrah packages. This is an example of a platform used by people to express their views which is not practicable for others to read. Text pre-processing is a data mining technique that involves transforming raw data into an understandable format. (Puteh et al., 2013) stated to prepare for another step of data processing, pre-processing is the main step where it is the process of performing preliminary processing on raw data and the data will transform into an understandable format that will be easy to proceed [9]. In text pre-processing, there are a few steps involved in cleaning the reviews that had been extracted. The steps are lowering the case from the upper case or capital letter, punctuation removal, numbers removal, tokenization, stop word removal, stemming, and lemmatization. This is shown in Fig. 2 below.

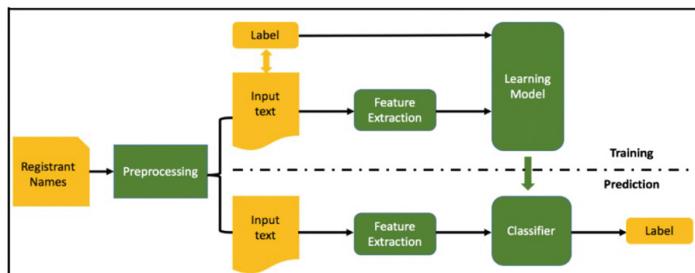


Fig. 2. Steps in text pre-processing

In this project, Python is used as a tool for pre-processing raw data into clean data that includes lowering cases, number removal, punctuation removal, stop words removal, tokenization, stemming, and lemmatization. The raw data as shown in Fig. 3 was collected from many platforms of social media such as Blogs, the website of the Travel agency, and others. The tool used to extract the data was the Data Miner extension provided by Chrome.

	A	B	C	D	E	F	G
3. cuba cek glokal travel...sy dh book n bny deposit utk bln 12 2013 inilamna sahabat tlh selidik juga baik dia travel nlo... tapi tetiba air mata bisa berderak tak berhenti masac bentang solat di Raudhah... seya telah menulum umrah pertama kali pada April 2013.. diukul memang perasan tu sebab tetiba bila kaki je atas karpet hijau tu... xtau kerapna... masab nabi SAW tau dekaje je dgn saya & merikan kedadangan saya, saw dia mcm2... & selawat atas nabi, alhamdullah dapat solat sunat & kail (8 rakaat),selagi tak dihalau oleh wantita' arab yg berpurdah & garang tu.. alhamdullah, semoga syafaat Nabi SAW ambanya membantu yg akhir ketulan, rendu di hal sentiasa bertanding utk Baitullah & ke rumah kelebihanNA lagi, insyaAllah dipermudahkan & Allah menjemput lagi saya sebagai tetamuNA. <p>4. kelebihanNA lagi, insyaAllah dipermudahkan & Allah menjemput lagi saya sebagai tetamuNA.</p> <p>5. mungkinlah agensi ini mungkin lejernya besar. Pada agensi ni, agensi ni yang berpendapat NIAT mereka bukan untuk turut menghadiri masalah seperti visa atau penerbangan sedangkan bayaran penduhung tempahan hotel di Madinah dan Mekah sudah dibayar. Apabila pagi akan menerbangkan jemaah ke tanah suci, maka bayaran berkenaan (mungkin) hangus dan Itulah sebabnya agensi berkenaan gagal mengembangkan sepenuhnya wangi para jemaah. Wallahu alhamad.</p> <p>6. mungkinlah, mungkin mereka tiada NIAT untuk menuju, tapi yang sudah pasti, bilaik jemaah umrah sudah TERTPUU, duit lepas, agenyis atau TIADA NIAT menju tu padail penghingganan diri. Tiada NIAT bawa balki ke semua tu, oleh sebab agenyis tu tiada NIAT untuk menuju, maka semasa kesahalan mereka dianggap tidak berlaku, paciklic mallick semusa yang kena tipu tu..</p> <p>7. enggi umrah dari NIAT sajalah, sungguh adil dunia.</p> <p>8. baru pulang dari trip haji iku lepas, dengan Andalusia Travel. Ini adalah haji kali kedua, yang pertama, 13 tahun lepas dengan Utas. Dua pengalaman jadi berbeza. Trip dengan Andalusia jauh lebih baik dan bermakna dan pada sawa lebih dekat kepada mabur kalan pun bukan mabur, mungkin sebab sebahagian lebih tua, saw Andalusia dah improve atau Utas masa dulu baru naik established. Walaupun hantaran tuh ini, mialah survey yang dilakukan oleh Tabung Hajji, Andalusia menyediakan paket bimbangan igama yang berkali di Maklaihan daran semua paket sveesta, mengalahkan TH travel sendiri. Ini yang penting:</p> <p>9. Annonymous, salam maktaban. Latafat nama cawoi. Maklid andalusia kah?</p>							

Fig. 3. Sample of the raw dataset

According to Mike Driscoll (2019) on the Real Python website, Jupyter Notebook is an open-source web application. Besides, it is a spin-off project from the IPython project, which used to have an IPython Notebook project itself. Jupyter comes from the core supported programming languages that it supports which are Julia, Python, and R. Furthermore, Jupyter ships with the IPython kernel, which allows people to write programs in Python. An example of Python codes is illustrated in Fig. 4.

```
In [1]: import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import time
import re, nltk
import os
import json
from pprint import pprint
from nltk import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
```

Fig. 4. Section of coding in importing packages in Python

After the process of extracting the raw data from social media, the raw data was transformed into cleaned data. This is due to the previous datasets that contain numbers, punctuation, lower or upper case, and so on as explained in detail above. It could affect the accuracy of the result at the end of this project. The cleaned data has been saved in Microsoft Excel as a CSV file.

3.2 Data Analysis

Data analysis is important because conducting the analysis could lead to a better and more appropriate presentation of the results. After having cleansed the data, it will be

analyzed by using a machine learning approach such as Naïve Bayes, Random Forest, and Support Vector Machine (SVM) to classify the reviews whether positive or negative towards the Umrah travel package in Malaysia. Word Cloud is known as a technique for visualizing data where the bigger the size of the words, the more important the words are for better understanding. Figures 5 and 6 are examples of the data analysis performed in this study.

```
In [28]: from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt

In [40]: from wordcloud import WordCloud
        background_color="white",
        max_font_size=150, random_state=42)

# In [47]: import pandas as pd
Tiram = pd.read_csv('C:\\Users\\NUR QASHEEH\\Downloads\\Tiram Travel.csv', encoding = 'iso-8859-1')
Andalusia = pd.read_csv('C:\\Users\\NUR QASHEEH\\Downloads\\Andalusia Travel.csv', encoding = 'iso-8859-1')
Quds = pd.read_csv('C:\\Users\\NUR QASHEEH\\Downloads\\Al-Quds Travel.csv', encoding = 'iso-8859-1')
Haji = pd.read_csv('C:\\Users\\NUR QASHEEH\\Downloads\\Tabung Haji.csv', encoding = 'iso-8859-1')
```

Fig. 5. Section of coding for Word Cloud

```
M In [32]: import matplotlib.pyplot as plt

text = Andalusia.Review
text = " ".join(review for review in Andalusia.Review)

#create and generate a word cloud image:
wordcloud = wc.generate(text)

#display the generated image:
plt.imshow(wordcloud, interpolation = 'bilinear')
plt.axis("off")
plt.show()
```



Fig. 6. Result of Word Cloud

3.3 Interface and Architecture Design

Interface design is a phase where the main system interacts with the end-user. To build the interface design, the interface should be user-friendly and easy to understand how the system works based on the arrangement of the system interface.

A system architecture is designed to represent the project's full system. The process begins with data scraping from social media such as blogs and websites regarding the Umrah travel package based on comments as the features and all data will be stored in the CSV file and continued to data pre-processing where data must be cleaned, integrated, transformed, and reduced so that it does not contain any real-world data such as incomplete, inconsistent, and noisy data. Then data is divided into two parts which are the training set and test set. The training set is the process of making the system recognize the pattern of the dataset before it comes to the test set, so it can classify accurately.

3.4 System Development

In the system development phase, the whole system is combined so that it can work smoothly. It also explains the system engine on how it comprises finding the positive and negative reviews of each Umrah package. The system is proposed to be a dashboard system that will visualize the outcome like a graph for each review of the Umrah package in Malaysia. The dashboard was developed for the user to interact with the system and display the result. It is being developed using a free template from the Internet such as Bootstrap.

4 Analysis and Discussions

This part clarifies and indicates an examination of the outcome of this anticipation. The classifier used consists of supervised learning methods which were Naïve Bayes (Gaussian and Multinomial), Support Vector Machine, and Random Forest. To generate accuracy, a library named Scikit-Learn is used in Python. Scikit-Learn is a machine learning in Python. It provides a wide selection of supervised and unsupervised learning algorithms.

4.1 Naïve Bayes – Gaussian

NAIVE BAYES - GAUSSIAN

```
In [50]: from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
span_detect_model = GaussianNB().fit(X_train, y_train)
all_predictions = span_detect_model.predict(X_train)

from sklearn.metrics import classification_report
print(classification_report(y_train, all_predictions))
print("Final Accuracy: %s"
      % accuracy_score(y_train, span_detect_model.predict(X_train)))

          precision    recall   f1-score   support
          0          0.40      1.00      0.57      144
          1          1.00      0.72      0.84      770
avg / total       0.91      0.76      0.79      914
Final Accuracy: 0.762582056892779
```

Fig. 7. Result of Naïve Bayes - Gaussian

The result of accuracy for Naïve Bayes (Gaussian) in Fig. 7 is 76.26%. The total precision was 100% for positive reviews and 40% for negative reviews. The average total shows that 91% for precision. This accuracy shows that the classifier could be classified positive review. The total of recall was 76% where it can and the total of f1-score was 79%.

4.2 Naïve Bayes – Multinomial

NAIVE BAYES - MULTINOMIAL

```
In [51]: from sklearn.naive_bayes import MultinomialNB
span_detect_model = MultinomialNB().fit(X_train, y_train)
all_predictions = span_detect_model.predict(X_train)

from sklearn.metrics import classification_report
print (classification_report(y_train, all_predictions))

from sklearn.metrics import accuracy_score
print ("Final Accuracy: %s"
      % accuracy_score(y_train, span_detect_model.predict(X_train)))

precision    recall   f1-score   support
          0       0.64      0.56      0.59      144
          1       0.92      0.94      0.93      770
avg / total     0.87      0.88      0.88      914

Final Accuracy: 0.8807439824945296
```

Fig. 8. Result of Naïve Bayes - Multinomial

The accuracy for Naïve Bayes (Multinomial) in Fig. 8 is 88.07%. The total precision for this classifier was 87% whereas the positive review and negative review for precision were 64% and 92%. The total percentage for recall was 88.00% and the total of f1-score was 88%. This classifier shows a slightly higher accuracy compared to the accuracy of Naïve Bayes (Gaussian).

4.3 Support Vector Machine

SUPPORT VECTOR MACHINE CLASSIFIER

```
In [54]: from sklearn import svm
S = svm.SVC(kernel='linear')
span_detect_model = S.fit(X_train, y_train)
all_predictions = span_detect_model.predict(X_train)

from sklearn.metrics import classification_report
print (classification_report(y_train, all_predictions))

from sklearn.metrics import accuracy_score
print ("Final Accuracy: %s"
      % accuracy_score(y_train, span_detect_model.predict(X_train)))

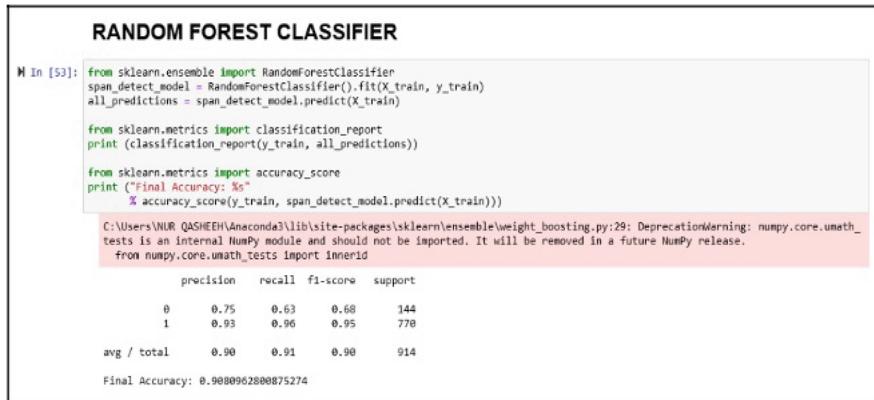
precision    recall   f1-score   support
          0       0.81      0.55      0.66      144
          1       0.92      0.98      0.95      770
avg / total     0.90      0.91      0.90      914

Final Accuracy: 0.9091903719912473
```

Fig. 9. Result of Support Vector Machine

The accuracy for Support Vector Machine (SVM) in Fig. 9 is 90.92%. The total precision was 90%. For the total recall, it shows 91% and for the f1-score, it shows a total of 90%. Accuracy for SVM is higher compared to Naïve Bayes (Gaussian) and Naïve Bayes (Multinomial).

4.4 Random Forest



```

RANDOM FOREST CLASSIFIER

In [53]: from sklearn.ensemble import RandomForestClassifier
span_detect_model = RandomForestClassifier().fit(X_train, y_train)
all_predictions = span_detect_model.predict(X_train)

from sklearn.metrics import classification_report
print (classification_report(y_train, all_predictions))

from sklearn.metrics import accuracy_score
print ("Final Accuracy: %s"
      % accuracy_score(y_train, span_detect_model.predict(X_train)))
C:\Users\NUR ASHRAFI\Anaconda3\lib\site-packages\sklearn\ensemble\weight_boosting.py:29: DeprecationWarning: numpy.core.umath_
tests is an internal NumPy module and should not be imported. It will be removed in a future NumPy release.
  from numpy.core.umath_tests import inner1d
      precision    recall   f1-score   support
          0       0.75     0.63     0.68     144
          1       0.93     0.96     0.95     770
avg / total       0.90     0.91     0.90     914
Final Accuracy: 0.9080962800875274

```

Fig. 10. Result of Random Forest

Finally, Random Forest classifier shows accuracy in Fig. 10 is 90.80%. The total for precision shows 90% while for recall, the total was 91% and for the f1-score, the total was 90%.

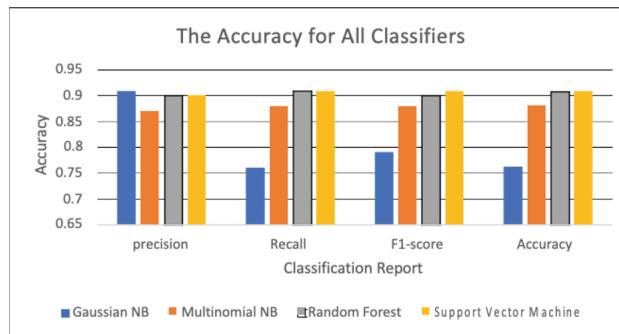
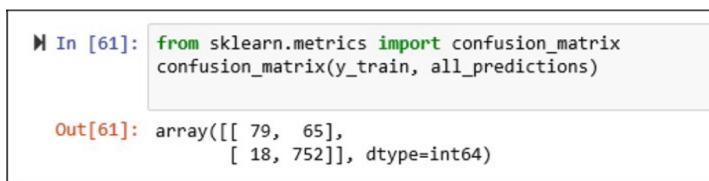
4.5 Analysis

Based on Table 2, it shows the summary of results between the classifiers used in Sentiment Analysis of Umrah Packages Review in Malaysia. Therefore, it can be concluded that Support Vector Machine is the best classifier for this dataset.

Table 2. Summary of accuracy of classifier used.

Classifier	Precision	Recall	f1-score	Accuracy
Gaussian NB	0.91	0.76	0.79	0.7626
Multinomial NB	0.87	0.88	0.88	0.8807
Random Forest	0.90	0.91	0.90	0.9080
Support Vector Machine	0.90	0.91	0.91	0.9091

As a result, based on Figs. 11 and 12, the True Negative is 79, the False Negative is 65, the False Positive is 18 and True Positive is 752.

**Fig. 11.** Bar Charts for all classifiers**Fig. 12.** Confusion Matrix

5 Conclusion and Recommendations

This initiative is intended primarily for those who intend to perform the Umrah in the future. The recommendations gleaned from social media regarding the Umrah packages offered by the Umrah agency can assist individuals in selecting the finest services at reasonable costs. The classifier demonstrates the polarity of every review by training and evaluating the datasets to categorize reviews as neutral, good, or negative depending on their polarity. Making recommendations serves to highlight areas where future performance might be enhanced and improved by collecting a higher number of datasets to produce a better result. Perhaps, this research project could help people in choosing and making decisions regarding what Umrah packages and agencies they should choose before purchasing it.

References

1. Almuhrzi, H.M., Alsawafi, A.M.: Muslim perspectives on spiritual and religious travel beyond Hajj: toward understanding motivations for Umrah travel in Oman. *Tour. Manag. Perspect.* **24**, 235–242 (2017). <https://doi.org/10.1016/j.tmp.2017.07.016>
2. Hassan, S.H., Maghsoudi, A., Nasir, N.I.M.: A conceptual model of perceived value and consumer satisfaction: a survey of Muslim travelers' loyalty on Umrah tour packages. *Int. J. Islamic Mark. Brand.* **1**(3), 215–237 (2016). <https://doi.org/10.1504/IJIMB.2016.075851>
3. Vinodhini, G., Chandrasekaran, R.M.: A comparative performance evaluation of a neural network-based approach for sentiment classification of online reviews. *J. King Saud Univ. Comput. Inf. Sci.* **28**(1), 2–12 (2016). <https://doi.org/10.1016/j.jksuci.2014.03.024>

4. Balahur, A., Jacquet, G.: Sentiment analysis meets social media—challenges and solutions of the field in view of the current information sharing context. *Inf. Process. Manag.* **51**(4), 428–432 (2015). <https://doi.org/10.1016/j.ipm.2015.05.005>
5. Howells, K., Ertugan, A.: Applying fuzzy logic for sentiment analysis of social media network data in marketing. *Proc. Comput. Sci.* **120**, 664–670 (2017). <https://doi.org/10.1016/j.procs.2017.11.293>
6. Rathan, M., Hulipalled, V.R., Venugopal, K.R., Patnaik, L.M.: Consumer insight mining: aspect-based Twitter opinion mining of mobile phone reviews. *Appl. Soft Comput. J.* **68**, 765–773 (2018). <https://doi.org/10.1016/j.asoc.2017.07.056>
7. Bansal, B., Srivastava, S.: Science direct sentiment classification of online consumer reviews using word vector representations. *Proc. Comput. Sci.* **132**, 1147–1153 (2018). <https://doi.org/10.1016/j.procs.2018.05.029>
8. Sánchez-Franco, M.J., Navarro-García, A., Rondán-Cataluña, F.J.: A naive Bayes strategy for classifying customer satisfaction: A study based on online reviews of hospitality services. *J. Bus. Res.* (December), 0–1 (2018). <https://doi.org/10.1016/j.jbusres.2018.12.051>
9. Mazidah, P., Isa, N., Puteh, S., & Redzuan, N.A.: Sentiment mining of Malay newspaper (SAMNews) using the artificial immune system. In: Proceedings of the World Congress on Engineering 2013, III(January) (2013)
10. Othman, B., Harun, A., Rashid, W., Ali, R.: The impact of Umrah service quality on customer satisfaction towards Umrah travel agents in Malaysia. *Manag. Sci. Lett.* **9**(11), 1763–1772 (2019)
11. Alghamdi, H.M.: Unveiling sentiments: a comprehensive analysis of Arabic Hajj-related tweets from 2017–2022 utilizing advanced AI models. *Big Data Cogn. Comput.* **8**(1), 5 (2024)



Opinion Mining System for Influence Detection Using Machine Learning to Secure Business Reputation

Shahrinaz Ismail¹(✉) and Kyi Lin Khant²

¹ Asia Pacific University of Technology and Innovation (APU), Kuala Lumpur, Malaysia
shahrinaz.ismail@apu.edu.my

² Huazhong University of Science and Technology (HUST), Hubei, China

Abstract. The use of social media becomes a must for almost all businesses, including micro and small enterprises. Despite the benefits of social media marketing, the feedbacks and responses retrieved from audience or customers via posts and comments in social media could be a challenge. Comments received could be negative in nature, which could affect the businesses in a negative way. The negative influence could jeopardize the business reputation, especially if the comments are highly negative and go viral. In sustaining their businesses, the micro and small business owners should not be left behind in using the latest analysis method in securing their business reputation. This paper proposes a simple website that makes use of the back-end sentiment analysis, to detect negative influence on their products and businesses. The results are presented in the form of process flow and interface design of the proposed website, made simple for the target users.

Keywords: Opinion Mining · Sentiment Analysis · Machine Learning · Influence Detection · Securing Reputation

1 Introduction

Small businesses can benefit greatly from social media marketing as a means of developing their brands and strengthening client relationships. The three main benefits of social media marketing for small businesses are being able to interact with the audience, keeping up with the latest trends, and raising brand awareness [1]. Nearly 90% of marketers claimed that their social media marketing activities have boosted traffic and exposure for their company, in which over 50% of them observed an increase of sales after two years using social media marketing strategies [2]. These statistics proved that the role of social media for small businesses is growing. However, most people are not knowledgeable enough and are struggling because of the bad reviews of their products and brands, even when their products or services are good. The main negative impact of social media on business comes from the risk of bad reviews, turning away the interest of potential consumers in products. According to recent studies, people treat evaluations on well-known social media platforms, like Facebook or Twitter (former name for X),

just as seriously as they would a friend's suggestion. Consequently, whether the criticism is warranted or not, it hinders the capacity to attract new clients, causing adverse effects on the expansion of the brand and company [3].

Bad reviews are one of the biggest problems for the micro-entrepreneurs and small businesses. However, there is a way to analyze the reviews and comments on social media, in which many are not aware of. Sentiment analysis, another word for opinion mining [4], is a technique used to determine whether data, e.g., social media posts, is positive, negative, or neutral. It is performed on textual data to help businesses understand the sentiment on their brand and products, received in the form of customers' needs and satisfaction, as retrieved in social media posts and comments.

There are a lot of different social media platforms made available for businesses, and a lot of ways to advertise a product. According to the statistics, 75% of Malaysians use the Internet regularly to read news and keep themselves updated with current events, and 72% use it to keep in touch with friends via social media. Besides, 96.4% of Malaysian Internet users access the internet with a smartphone [5], and this allows users to instantly post reviews and opinions on products and brand in the palm of their hands, anywhere and at any time. The most popular social media being WhatsApp and Facebook (over 90%), followed by Instagram, Telegram, Facebook Messenger, Tik Tok, and X. However, for business advertising, Twitter stands in fourth place after Facebook, Instagram, and Tik Tok [5], partly because the interface is very simple and easy to use, especially in retrieving the data for businesses.

Since social media is heavily used global wide, there are problems that micro-entrepreneurs and small business owners may face. Businesses may suffer in losing their reputations due to cyberattacks on their social media posts. People can easily share and comment on misleading statements and fake news, stating things that never happened. These activities will cause the victims, i.e., businesses, to lose their branding image due to negative comments, since negative comments can cause negative influence on the brand or products offered by them. This cyberattack can be considered as cyberbullying, especially when the negative influence is intended to cause harm on the victims. There are times when the posts or comments are a mixture of positive and negative words that lead to confusion to the readers, causing the business owners to feel unsure on how to respond. Despite the efforts shown by previous research on solutions to curb cyberbullying, most of them are focused only on prevention methods in general, not directly for the benefits of small business owners and microentrepreneurs.

Considering the need to facilitate and educate microentrepreneurs and small business owners on this matter, this research puts forth an initiative to develop an easy-to-use analytical tool, backed by machine learning (ML), for sentiment analysis on social media posts and comments. The goal of this research is guided by the objectives of investigating related works on sentiment analysis, analyzing sentiments in social media posts using ML, and developing a website for sentiment analysis on social media posts. The outcome of this research will benefit in creating awareness among the community and microentrepreneurs on the impact of sentiment in social media posts, especially the negative ones. With the help of this platform, microenterprises can decide and improve their products and brands based on the sentiments understood from the posts, to prevent the brand image from getting further damage.

2 Related Works

2.1 Sentiment Analysis

Sentiment analysis implies the involvement of analysis on numerous sentiments expressed by people from online or in their reviews on various businesses. When searching for movie reviews before watching one, for example, there are certain techniques that are available to analyze the movie reviews. In a broader sense, sentiment analysis and opinion mining are methods for finding, retrieving, and extracting knowledge and opinions from the massive amounts of textual material on the World Wide Web (WWW). Often linked to sentiment analysis, opinion mining makes use of computation techniques to extract, classify, understand, and assess the opinions expressed in various sources found on the Internet [4], most commonly social media.

Opinion mining and sentiment analysis are frequently used for analysis on customer reviews in social media [6] because they emphasize the extraction of sentiment polarity and user opinions. Since sentiments are reflected by the emotion, opinion mining that involves sentiment analysis is defined as a series of processes to identify and extract negative (and positive) sentiments from the text, usable for decision-making [4].

In order to perform the sentiment analysis, five steps are needed [7]: (i) **Goal Setting**: knowing what to find out, i.e., establishing the objective and involves deciding on the range of the text content to be analyzed; (ii) **Text Preprocessing**: identifying the source of data, e.g., the web or a microblogging site, which then unnecessary words are removed from the text, and the emotional symbols that people use in texts are organized into words, resulting in the text being placed into the processing system; (iii) **Parsing the Content**: breaking down the text content into individual components to better understand the sentiment expressed in it, which include identifying and separating out phrases, words, and sentences, and then analyzing their meaning, tone, and context; (iv) **Text Refinement**: finding the stop words and synonyms, in which a collection of pre-trained word vectors and a sentiment lexicon are annotated with real-valued sentiment scores; and (v) **Analysis and Scoring**: identifying and rating sentiment-bearing sentences from the data, with a technique of scoring involves analyzing how strongly a feeling is expressed.

There are different types of sentiment analysis, performed in different ways based on different perspectives. Fine-grained sentiment, emotion detection, aspect-based and intent analysis are some of the different types of sentiment analysis from the perspective of the users and researchers [8]. Regardless of the various types available, they are insufficient for a development of a prototype model on sentiment analysis. In addition, sentiment analysis is used based on the words, in which many different levels of words exist. This justifies the reason for applying the type of sentiment that has the specific level of words to do the sentiment analysis. The most appropriate way to develop this prototype model is word-base, which are:

- **Document Level:** The opinion mining is solely based on the document, in which the polarity is determined by the entirety of the document, whether the available opinions, i.e., sentiments or feelings, give a favorable or negative sentiment, often focusing on one subject. It benefits by obtaining most of the polarity of a certain characteristic, with a drawback of its inability to capture people's preferences [9].

- **Sentence Level:** Each sentence polarity is processed and examined to provide a positive, negative, or neutral judgement on it. User perceptions, perspectives, and opinions regarding the statement make up subjective sentences. When there is no implication of a viewpoint, a sentence is neutral, and it is more likely to be classified as an objective sentence that provides information instead of a subjective sentence that expresses the ideas and opinions of the subject. The benefit is the categorization of subjectivity and objectivity. ML detects subjective sentences, but sentiment analysis has a constraint at the phrase level [10].
- **Aspect Level:** Also known as the feature level, or entity level, which provides output that expresses the result as an opinion. It is the most in-depth kind of sentiment analysis. The goal value and the two outcomes are both regarded as either positive or negative. Target opinion aids in understanding the significance of this level by providing emotional input on entities and their properties. At this level, reviews, feedback, comments, and other functions are conducted [10].

2.2 Supervised Machine Learning Approach

Using the technique of machine learning (ML), which is a form of artificial intelligence (AI), software programs may anticipate outcomes more accurately without having to be explicitly instructed to do so. In forecasting new output values, ML algorithms use past data as input [11], making ML the basic need for sentiment analysis.

Supervised learning consists of several classifiers. In general, one of the several types of datasets used in supervised learning systems is the labelled training dataset. Each class has its own characteristics, advantages, and labels that may be applied to the system. Upon arrival, each word is given a label based on its kind, or class, and associated characteristics [12]. Starting with the ***Probabilistic Classifiers***, mixture of models is used, in which every class is a part of the mixture. The probability of picking a certain word for each mixture component is provided by a generative model for that component. Generative classifiers are another name for these classifiers [13]. Naïve Bayes is a commonly used probabilistic classification model in ML and natural language processing (NLP) tasks, such as sentiment analysis, spam filtering, and document categorization. The model is based on Bayes' theorem, which states that the probability of a hypothesis, or a class label for a given input, can be calculated based on the prior probability of the hypothesis and the likelihood of the input given the hypothesis. The Naive Bayes model is trained on a labeled dataset, where each data point consists of a set of features (words or other variables) and a class label. The model calculates the prior probability of each class label and the likelihood of each feature given each class label. It then uses these probabilities to predict the class label of new input data [8, 14].

Under the same Probabilistic Classifiers are the Bayesian Network and Maximum Entropy. Bayesian Network is a probabilistic graphical model that represents a set of variables and their conditional dependencies using a directed acyclic graph (DAG), used to model uncertain relationships between variables and to make probabilistic predictions based on evidence. Each node in a Bayesian network represents a random variable, and the edges represent the probabilistic dependencies between the variables. Each node has a probability distribution that describes the probability of its value given the values of its parent nodes. [8]. On the other hand, the Maximum Entropy is based on a principle of

statistical inference used to estimate probability distributions, based on the idea that when there is incomplete information about a probability distribution, the distribution with the maximum entropy (i.e., the most “uninformative” distribution) should be chosen. The goal is to find a probability distribution that satisfies some constraints while having the maximum entropy, in which the constraints can be in the form of moment matching or expectations of some features of the distribution [15].

The second common group of classifiers under supervised learning is the *Linear Classifiers*, which consist of Support Vector Machines (SVM) and Neural Network (NN). For SVM, the fundamental premise is to identify linear separators in the search space that can best divide the various classes, using hyperplane to best separate between the classes. SVMs are used for a variety of tasks, including categorizing reviews according to their quality. On another note, the NN gets its name from its fundamental unit, the neuron, in which a neural network has numerous numbers of neurons. The vector overline, which represents the word frequencies in the text, is used to indicate the inputs to the neurons. Despite being under linear classifiers, non-linear boundaries could be implemented using multilayer NNs. The multi-layered network is employed to create several piecewise linear bounds, in which the neurons in the later layers get input from the neurons in the earlier layers. Due to the necessity of backpropagating through several layers, the training process is more complicated in NN [14].

In complementing the ML approach for sentiment analysis, Lexicon-based approach is adopted. Using a lexicon, the sentiment orientation is determined by the words or phrases that appear in a text [16]. In many tasks involving sentiment categorization, opinion words are used. Other desired states are expressed using positive opinion words, whereas some undesirable ones are expressed with negative opinion words. Additionally, there are idioms and expressions of opinion, together known as the opinion lexicon. There are three basic methods for gathering the list of words that express opinions: the manual technique that requires a lot of time and is not used exclusively; the Dictionary-based approach; and the Corpus-based approach.

The *Dictionary-based* approaches take the polarity of each sentence in a text and extract it. The sentiment is then classified by analyzing the meaning of the opinion words. These methods basically employ a dictionary of words linked to their semantic significance, as a language’s vocabulary is its lexicon [17]. However, the dictionary-based approach has a limitation in terms of its inability to identify opinion words with context- and domain-specific orientations.

The *Corpus-based* approaches are methods in natural language processing (NLP) that rely on analyzing large collections of text, known as corpora, to build language models and perform various language-related tasks. These approaches gained their popularity in recent years, thanks to the availability of large amounts of digital text and advances in computational power. The challenge of discovering opinion words with context-specific orientations is overcome by the corpus-based technique, as it relies on grammatical patterns or patterns that happen simultaneously with a seed list of opinion words in a big corpus to locate more opinion terms. A list of seed adjectives and a set of linguistic restrictions are employed to find new adjective terms and their orientations, building the concept known as “emotion constancy” [17].

Corpus-based approaches are available in terms of statistics and semantic. Statistically, sentiment analysis is a type of text or opinion mining that uses computational linguistics, NLP, and text assessment to extract quantitative information from a piece of text corpus and categorize it into positive, negative, and other degrees of intention. The independence test and goodness-of-fit test for hypotheses testing both employ this sentiment analysis report. On the other hand, the semantic method uses many ideas to determine how related phrases are and directly assigns sentiment ratings. Words with comparable emotion levels have similar meaning, and by iteratively adding synonyms and antonyms to the initial set, a list of emotional words could be generated. The sentiment polarity of an unknown word may then be calculated using the relative number of positive and negative synonyms for that word [8].

Table 1 summarizes the related works on sentiment analysis, with accuracy success rate as achieved by the previous researchers.

Table 1. Summary of related works on sentiment analysis with accuracy success rate.

Author (Year)	Approaches	Accuracy Success Rate
Godsay (2018) [7]	SVM and Neural Network (NN)	SVM: 87% NN: 68%
Medhat (2014) [13]	ML-based and Lexicon-based	Naïve Bayes: 75% Bayesian Network: 62% Maximum Entropy: 59%
Hu (2019) [12]	Categories of Sentiment Analysis: Extraction using BERT for Span-based purposes	Document Level: 82% Sentence Level: 92%
Cai (2020) [15]	Categories of Sentiment Analysis: Aspect-Category-based with Hierarchical Graph Convolutional Neural Network (CNN)	Aspect Level: 86%
Saad & Yang (2019) [18]	Ordinal Regression	Multinomial Logistic Regression: 67% SVR: 82%
Geetha (2021) [19]	Fine-tuned BERT Base Uncased model	BERT: 88%

3 Methodology

Despite the large number of articles found, only a few articles that did sentiment analysis for the benefit of small businesses. Hence, a website is necessary to send queries to the back-end sentiment analysis engine and retrieve results to be displayed to the target users, i.e., the business owners. The website design needs to be simple and user-friendly, yet able to describe the influence of the sentiments or opinions.

In preparation for the web interface, the process starts with dataset identification and acquisition. It is necessary to get sufficient and suitable dataset to perform the opinion mining process. Using the dataset from Kaggle.com, a Twitter (X) dataset developed by Kazanova [20] is chosen. The name of the dataset is “Sentiment140 with 1.6 million tweets” and it has 1.6 million tweets.

In order to analyze sentiments in the social media posts using ML, two processes are performed: the machine learning process; and the lexicon part. In the machine learning process, there are four steps: Data Preprocessing, Feature Extraction, Modelling, and Performance Evaluation. Inside them, there are Exploratory Data Analysis (EDA), tokenization, tagging and entity.

3.1 Data Preprocessing and Feature Extraction

Data preprocessing in ML is a data mining approach that converts raw data into a format that is legible and intelligible. It requires feature extraction, a technique of turning unprocessed raw data into numerical features that may be processed while keeping the original information of the data set intact. Compared to directly applying ML to the raw data, it produces superior outcomes.

Exploratory Data Analysis (EDA) is an approach to analyze and understand data through summary statistics, visualizations, and other techniques. The goal is to identify patterns, relationships, and anomalies in data that can be used to make informed decisions about how to proceed with further analysis or modeling. It helps to identify potential issues with the data, e.g., missing values, outliers, and skewness, and provides insights into the underlying structure of the data. It is an important step in the data analysis process and can inform the choice of modeling techniques and algorithms to be used. For this research, EDA process is performed to clean the unnecessary data and to see the data head and data information. This is to ensure that it can easily determine to choose the right column of the dataset.

Tokenization is a technique used in natural language processing (NLP) to break down phrases and paragraphs into simpler language-assignable elements. The collecting of data (a sentence) and its breakdown into comprehensible components (words) are the first steps of the NLP process. Tokenization is used to decompose the string into its components so that it can be interpreted by a machine. By breaking a statement down into its constituent pieces, a machine can comprehend both the parts and the whole. This eases the software to comprehend both the individual words and how they fit into the overall text. It is crucial in enabling the computer to count the frequency of certain words and the locations in the text where they usually appear, in which the NLP will depend on at the later stage. These processes are necessary, since there are a lot of the text and comments in the dataset that need to be trained for future uses of the analysis. Besides, the sentence of the text can be visualized easily in a simple form.

Tagging, or tag, is the first stage in part-of-speech tagging. It is done with the help of the Default Tagger class. The only parameter that the Default Tagger class accepts is “tag”. The most effective application of Default Tagger is when it is put to use with the most popular part-of-speech tag. Consequently, a noun tag is advised. By tagging the text, the property of the tag is defined, and all the required texts need to be organized as an entity for the next processes. Entity extraction is a text analysis method that

automatically extracts particular information from unstructured text and organizes it into specified categories using NLP. These groups are known entities, which are noun-representing words or phrases. This is applied to proper names, numerical expressions of time or quantity, e.g., dates, phone numbers, or monetary amounts. Once the text has gone through this process, it is easy to identify which word is belong to which category, and it is easy for the process to work on the sentiment analysis.

3.2 Modelling

A file that has been taught to detect particular patterns is known as a ML model. By giving a model an algorithm, it may use to analyze and learn from a set of data by training it over those data. For this research, two models are used for testing and comparison of the models: Roberta and Vader.

RoBERTa stands for Robustly Optimized BERT Pre-training. BERT, or Bidirectional Encoder Representations from Transformers, is a Natural Language Processing (NLP) model proposed by researchers at Google Research in 2018. When it was proposed, it achieved state-of-the-art accuracy on many NLP and NLU tasks. BERT and RoBERTA are almost the same. However, in the coding of python, there are two packages, i.e., Transformers and Torch, which are necessary to perform the sentiment analysis. **Transformer** in NLP tries to tackle sequence-to-sequence problems while skillfully managing long-range relationships. Instead of using sequence-alignment to compute representations of its input and output, it completely depends on self-attention. In python, this is very useful for the NLP related processes including the sentiment analysis. **Torch**, on the other hand, is a Deep Learning tensor library that has been optimized for usage with GPUs and CPUs. It is built using python and the Torch framework. Torch is preferred over other deep learning frameworks like TensorFlow and Kera's because it is entirely pythonic and employs dynamic computation graphs. It enables real-time testing and execution of certain sections of code by researchers, programmers, and NN debuggers. It allows portion of the code to be examined without having to wait for the implementation of the complete program. For the purpose of the sentiment analysis, the Roberta model is related to the use of the Transformers and Torch alternatively. The use of the Transformer pipeline helps the Torch on the polarity score (i.e., the score based on a person's feeling) of the sentiment analysis.

VADER, or Valence Aware Dictionary and sentiment Reasoner, is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. In VADER, a combination of a list of lexical features (e.g., words) is used to generally label according to their semantic orientation as either positive or negative. Apart from the positivity and negativity score, VADER also indicates how positive or negative a sentiment is. The VADER model is mainly used with Sentiment Intensity Analyzer (SIA) to compute the value of the sentiment analysis.

Sentiment Intensity Analyzer is an object that measures the polarity score, i.e., a method that gives the score of the categories of Positive, Negative, Neutral, and Compound. The compound score is the sum of positive, negative, and neutral scores, which is then normalized between -1 (for extreme negative) and + 1 (for extreme positive).

3.3 Performance Evaluation

Performance evaluation of the trained ML models is calculated in ML using performance assessment measures. This aids in determining how well the ML model performs on a dataset that it has never encountered before. RoBERTa gives out the result of the individual words in term of positive, negative, and neutral, while VADER, although the same as the Roberta, additionally gives all the total score of the whole sentiment results. There is not much difference between the two models, but in order to plot the results, there are no alternative visualization for the RoBERTa. Meanwhile, there is a package named “Textblob” that helps on the visualization of the VADER results.

Textblob is a python library for processing textual data. It provides an easy-to-use method for performing common NLP tasks, e.g., sentiment analysis, spelling correction, and translation. Since RoBERTa is unable to visualize the results, VADER is the only choice for data visualization. In the final stage of this research, only VADER is applied to produce the sentiment analysis.

3.4 Interface Development

Three web frameworks are tested out before determining the right one for this research: Flask, Django and Streamlits. For the implementation, only Streamlits is used, due to the following justifications.

Flask, a Web application framework programmable in python, is not stable in terms of its functionalities, despite being flexible, lightweight, and scalable. **Django**, a high-level python web framework encourages rapid development and clean, pragmatic design. However, it often crashes due to the localhost error during the development stage on local site, since the Django parameters work on very high usage of the system random access memory (RAM), causing unsuccessful results for the sentiment analysis.

Streamlits is the most suitable website framework in python. It allows re-use of any python code priorly written, saving considerable amount of time, compared to non-python-based tools that forces all codes for visualizations to be re-written.

4 Results and Analysis

Exploratory Data Analysis (EDA) helps to identify any potential issues with the data that may need to be addressed before proceeding with further analysis or modeling. In this training, only 500 sample of the dataset is applied for the purpose of reducing the time consumption and r . The results of the EDA are as shown in Fig. 1. With the sample of 500 comments from the users, the highest rating star is 5 stars. Looking at this, it can be able to predict that there will be more positive reviews than the negative reviews according to the sample of the datasets.

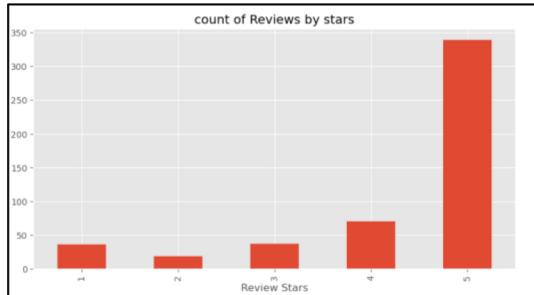
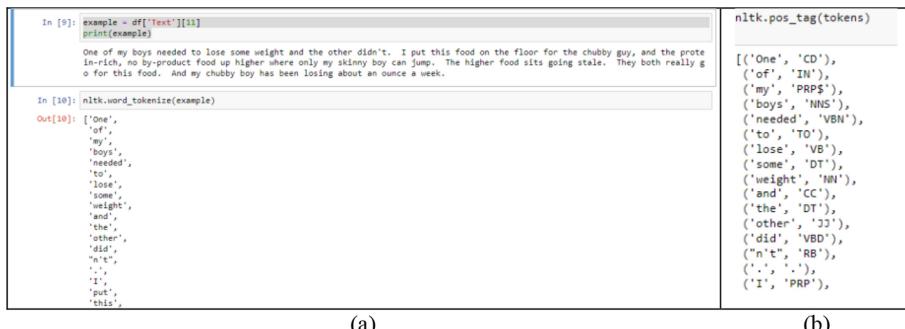


Fig. 1. Results of EDA.

Tokenization is an important step in sentiment analysis, which involves the process of breaking down a text document into smaller units called tokens. It plays a crucial role in this process as they help to identify individual words and phrases that are indicative of positive, negative, or neutral sentiment. In these processes, only one of the sentences is taken from the dataset to see how the tokenization get the token and the results are shown in Fig. 2(a). All the words are separated into single pieces, known as tokens.



(a)

(b)

Fig. 2. (a) Tokenization. (b) Tagging and results.

Tagging is a common technique to categorize the sentiment of a given text into positive, negative, or neutral category. The text is tagged based on the sentiment words or phrases that appear in the text. This process can be performed after the tokenization. Since, tokenization separates the words into token, it is easy to get tagged, as shown in Fig. 2(b).

In sentiment analysis, an entity is a named real-world object (e.g., a person, organization, or product) of which opinions or sentiments are expressed in a text. The task of entity recognition in sentiment analysis is to identify the entities mentioned in the text, and classify them into predefined categories, i.e., person, location, product, and such.

Website Development is the final part of this research. Before designing the interface and features, it is required to design the system architecture. As shown in Fig. 3, once the user starts entering the text and choosing the analysis, the website would send the

request through Streamlits and python. The model then processes the text data and give the sentiment results, which is processed through Textblob for visualization. In this visualization, the user can see the polarity score, either positive, negative, or neutral, along with the graph of the polarity score.

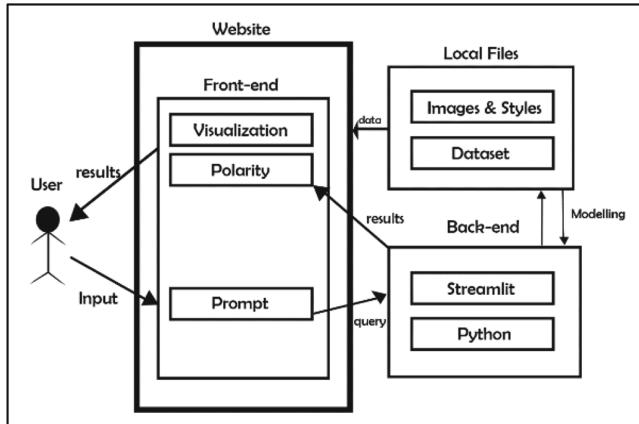


Fig. 3. System architecture.

The website interface includes a slide bar of menu, as shown Fig. 4, which is optional for the users to choose. The menu includes links to Home, Sentence Level Analysis, Dataset Analysis, and Contact. This website is limited to work on the sentence level in the first stage of development, since dataset level needs a lot of parameters that require a proper domain to process the dataset and the hyperparameters. Upon choosing the sentence level, a text box is made available, as shown in Fig. 4.

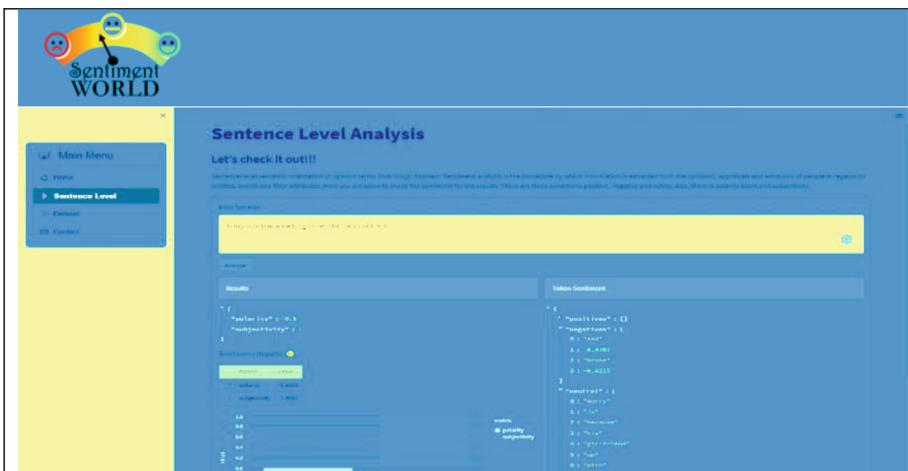


Fig. 4. The web interface with sentence level sentiment analysis and an example.

For the sentence level, there are three main outputs, which are polarity score, subjectivity, and sentiment. Polarity score is the sentiment analysis result that stands between -1 to +1, with subjectivity that expresses how the sentence can be subjective between that range. There are three types of sentiment results, which are Positive, Negative, and Neutral. These results are made available in table and graphical formats.

Testing is performed to know how the website handles the input with the number of words of the texts. In this analysis, there are three tests with: 65 words; 165 words; and 320 words. Figure 5 shows the testing results of r with 65 words, as it analyzed all the sentiments of the text. The sample text is a description about a girl who is suffering from an illness. For this test, the result shows negative side, and the polarity score shows below 0, since it is negative.

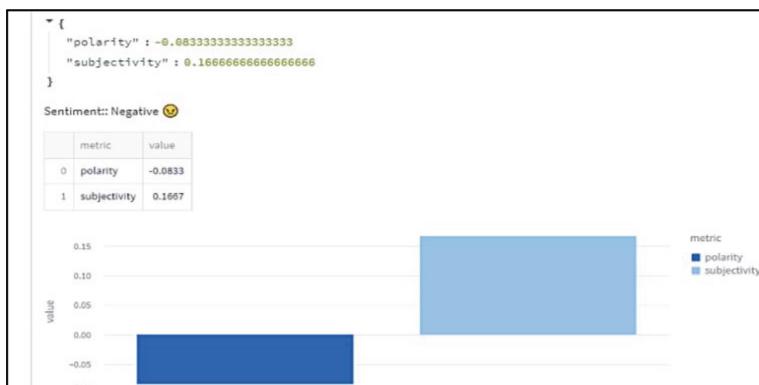


Fig. 5. Testing the sentiment analysis in the website with 65 words.

Figure 6 shows the results of the website performance with 165 words, analyzing all the sentiment of the text. A random text is used for this test, and the website can handle the input text of the 165 words properly.

Figure 7 shows the test results with 320 words, as the website analyzed all the sentiment of the text. Overall, the performance is somewhat good, except for a lot of decreases in polarity score and calculation, but it still can produce the sentiment results of the text. It can be concluded that the website can handle the undefinable texts and a large amount of the texts.

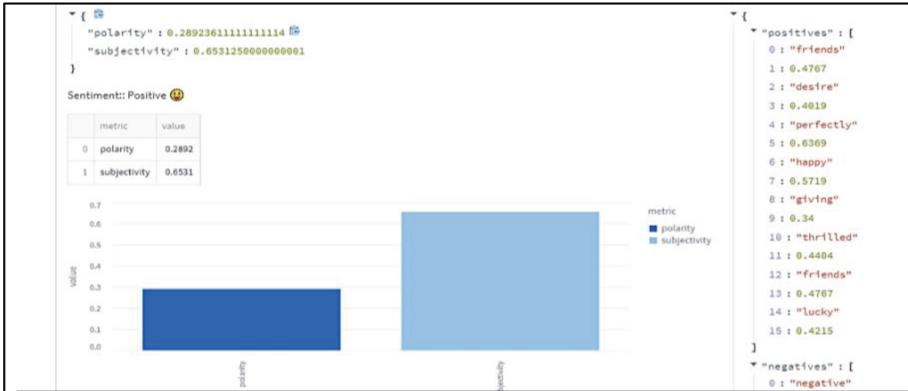


Fig. 6. Testing the sentiment analysis in the website with 165 words.



Fig. 7. Testing the sentiment analysis in the website with 320 words.

5 Conclusion

Sentiment analysis helps businesses to understand their customers' opinions, emotions, and attitudes towards their products, services, and brand. This information can be used to improve customer experience, customer engagement, and customer satisfaction. Nevertheless, further testing on user acceptance and usability will be required to prove the success of the implementation.

This research has brought forth the development of website for the target users, i.e., small business owners, with a stable sentiment analysis model in the back end. Not only small businesses owners, this website can support big companies with lots of comments as dataset, but a very powerful platform is required to process the data and hyperparameter of the dataset. Future improvement of this research can support businesses with better impact and sustainability, in terms of embedding escalation protocols when high negative influence or opinion is detected in the system, and measuring reduction of reputation damage due to negative influence.

References

1. Sociallybuzz. Truths About Social Media Marketing for Small Business: Stats, Benefits, Tips and Importance. Sociallybuzz. (2022). Retrieved from <https://www.sociallybuzz.com/benefits-of-social-media-marketing-for-small-business/>
2. Mansfield, M.: Social Media Marketing Statistics Important to Small Businesses. Small Business Trends. (December 6, 2016). Retrieved from <https://smallbiztrends.com/2016/12/social-media-marketing-statistics.html>
3. NetReputation: Negative Social Media Impacts on Business. Net Reputation (2022). Retrieved from <https://www.netreputation.com/negative-social-media-impacts/>
4. Kasinathan, V., Mohamed, M.N.A., Ji, L.Y., Mustapha, A., Rani, M.F.C.A., Manikam, S.: Opinion mining in the airline industry: learning social sentiments and insights. In: Uden, L., Liberona, D. (eds) Learning Technology for Education Challenges. LTEC 2021. Communications in Computer and Information Science, vol. 1428. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-81350-5_15
5. DigitalBusinessLab.: Social Media Penetration in Malaysia [Research]. Digital Business Lab. (2022). Retrieved from <https://digital-business-lab.com/2022/07/%E2%91%A1-social-media-penetration-in-malaysia-research/>
6. Kim, Y., Jeong, S.R.: Opinion-mining methodology for social media analytics. KSII Tran. Internet Inf. Syst. **9**(1), 391–406 (2015)
7. Godsay, M.: The process of sentiment analysis: a study. Int. J. Comput. Appl. **126**(7), 26–30 (2018)
8. Goyal, K.: Top 4 Types of Sentiment Analysis & Where to Use. Up Grad (2020). Retrieved from <https://www.upgrad.com/blog/types-of-sentiment-analysis/>
9. Yu, L.C., Wang, J., Zhang, X.: Refining Word Embeddings for Sentiment Analysis. (2017). https://www.researchgate.net/publication/322582837_Refining_Word_EMBEDDINGS_for_Sentiment_Analysis
10. Mehta, P.: A review on sentiment analysis methodologies, practices and applications. Int. J. Sci. Technol. Res. **9** (2020)
11. Burns, E.: What is machine learning and why is it important? TechTarget. (2022). Retrieved from <https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML>
12. Hu, M. Open-Domain Targeted Sentiment Analysis via Span-Based Extraction and Classification (2019). <https://arxiv.org/abs/1906.03820>
13. Medhat, Y.: Sentiment analysis algorithms and applications: a survey. Vol. 5, pp. 1093–1113 (2014). <https://www.sciencedirect.com/science/article/pii/S2090447914000550>
14. Mach, M.: Lexicon-based Sentiment Analysis Using the Particle Swarm Optimization. (2020). <https://www.mdpi.com/2079-9292/9/8/1317>
15. Cai, H. Aspect-Category based Sentiment Analysis with Hierarchical Graph Convolutional Network. (2020). <https://aclanthology.org/2020.coling-main.72/>
16. Jurek, A., Mulvenna, M.D., Bi, Y.: Improved lexicon-based sentiment analysis for social media analytics. Secur. Inform. **4**(9) (2015). <https://doi.org/10.1186/s13388-015-0024-x>
17. Sushmitha, R., Haripriya, V.: Sentiment analysis: facebook status message. Int. J. Eng. Res. Technol. (IJERT) **4**(27) (2020)
18. Saad, S.E., Yang, J.: Twitter sentiment analysis based on ordinal regression. IEEE Access **7**, 163677–163685 (2019). <https://doi.org/10.1109/ACCESS.2019.2952127>
19. Geetha, M.P., Karthika Renuka, D.: Improving the performance of aspect-based sentiment analysis using fine-tuned Bert Base Uncased model. Int. J. Intell. Netw. **2**, 64–69 (2021). <https://doi.org/10.1016/j.ijin.2021.06.005>
20. Kazanova, M.M.: Sentiment140 dataset with 1.6 million tweets. Kaggle. (2017). Retrieved from <https://www.kaggle.com/datasets/kazanova/sentiment140>



A Presentation Mining Framework: From Text Mining to Mind Mapping

Vinothini Kasinathan¹ and Aida Mustapha²(✉)

¹ School of Computing, Asia Pacific University of Technology and Innovation Taman Teknologi Malaysia, 57000 Kuala Lumpur, Malaysia
vinothini@apu.edu.my

² Faculty of Applied Sciences and Technology, Universiti Tun Hussein Onn Malaysia, KM1 Jalan Pagoh, 84600 Pagoh, Johor, Malaysia
aidam@uthm.edu.my

Abstract. Slide-based presentations, such as those created using PowerPoint, have become ubiquitous in educational settings, offering educators a convenient tool for conveying information to students. However, a significant challenge arises from the inherent linear structure of these presentations, which may not align with the non-linear process of knowledge reconstruction undertaken by students. This discrepancy often leads to variations in students' understanding and interpretation of the presented material, potentially resulting in misinterpretation of the original content sequence of the slides. Recognizing this challenge, this paper proposes a presentation mining framework to address the gap between the linear knowledge construction by presenters and the non-linear knowledge reconstruction by students. By leveraging text mining techniques, the framework aims to uncover keywords and key phrases embedded within slide content, facilitating the generation of a visual representation in the form of mind map that captures the underlying structure and relationships of the extracted information. The mind maps generated are then evaluated against a domain ontology, a dictionary, and subject matter experts based on averaged precision, recall, and F-measure score. The results showed that the proposed framework is able to extract and generate reasonable visualization from presentation slides.

Keywords: presentation mining · keyphrase extraction · concept map

1 Introduction

In high school and tertiary education settings, slide presentations are popular as they offer educators a versatile and effective tool for delivering course content, engaging students, and facilitating learning. By leveraging the benefits of visual communication and multimedia integration, slide presentations contribute to a more interactive and enriched learning experience for students. Slide presentations are often created using presentation software such as Microsoft PowerPoint, Apple Keynote, Google Slides, or similar tools, prepared by the presenter who

assume the role of subject matter expert. Kinchin and Cabot [3] believed that sequential layout in presentation slides does not correspond to students' knowledge and understanding. This is because the original sequence of presentation delivery is often reconstructed differently by the audiences or the students [1]. This disparity frequently results in discrepancies in students' comprehension and interpretation of the material presented, possibly leading to misinterpretation of the original sequence of slide content.

Research has shown that concept mapping is able to help students to visualize the content at the abstract level, which are closer to the expert knowledge structure as opposed to the linear structure [1]. Concept mapping has been widely adopted in academic teaching especially in computer-based courses [9]. The abstract representation of the relationships among the key topics helps the represent tacit knowledge delivered by the presenter. Concept mapping is similar to a mind map, which is a type of graphical knowledge display used to visualize outline information popularized by Buzan [2]. Both concept maps and mind maps are proven to enhance memory due to its nature in capitalizing the potential of both left and right brain [12]. However, constructing a concept map or a mind map manually requires a person to have in-depth understanding and identifying keywords takes up a lot of time [4].

To automate the process, a mechanism to extract the key topics from slide presentation is imperative, before a concept map or a mind map can be constructed. Automatic keyphrase extraction in Natural Language Processing (NLP) covers two types of algorithms; keyphrase assignment and keyphrase extraction. Keyphrase assignment can be considered as alignment task, whereby keyphrase for a given document is selected from an existing vocabulary with the aid of machine learning algorithm. This method involves keyphrase assigned document to act as training material, key phrases for input document are chosen based on classes of keyphrase in the mentioned vocabulary [5]. On the other hand, keyphrase extraction does not perform selection on the predefined vocabulary, instead, it extracts from the document itself. Candidate keyphrases are examined on several condition such as frequency and length [10].

Nonetheless, text in slide presentations are usually in chunks or fragments because few key phrases are sufficient to give basic idea and represent the concept of the document or text [6]. To address this gap, this research proposes a framework called the Presentation Mining, whereby this framework is able to automatically extract keywords and key phrases from a collection of presentation slides and generate a visual knowledge display in the form of mind map based on the keywords and keyphrases extracted. The objective of this research is two-fold. The first is to deliver a quality keyphrase extraction algorithm and the second is to deliver a quality output to the user, which is the mind map. The final output will be compared against a matching domain ontology, a dictionary, and human expert.

The remainder of this paper proceeds as follows. Section 2 presents the materials and methods to achieve this objective. Section 3 presents the implementation results, and finally Sect. 4 concludes the paper with some indication for future works.

2 Materials and Methods

Presentation mining is a new approach proposed by this research with the aim to support learning pedagogies using technology like the PowerPoint slides. The motivation behind this framework is to assist visual learners to capture the gist from their lecture in the form of visual display such as the mind map. Figure 1 shows the proposed presentation mining framework that consists of three stages, which are pre-processing, presentation mining, and visualization.

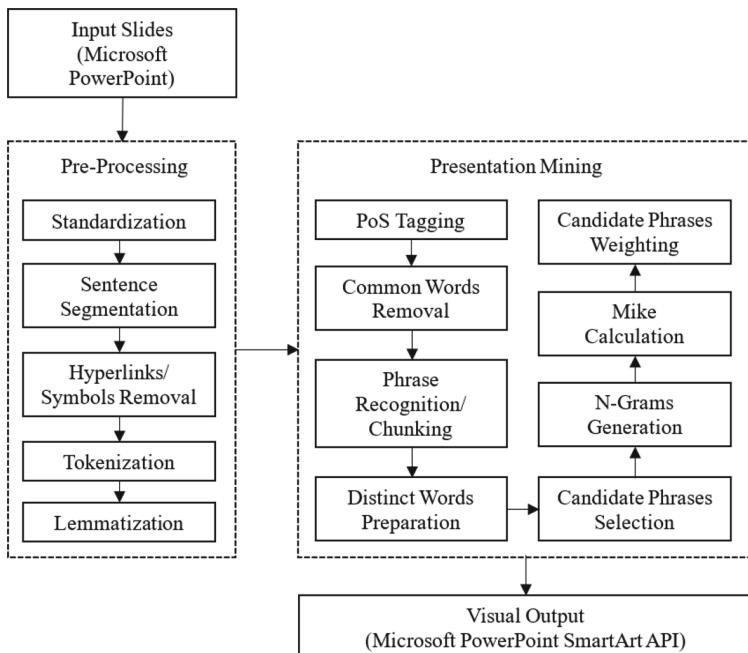


Fig. 1. Presentation Mining Framework

2.1 Pre-processing Stage

There are six pre-processing tasks to be performed within the pre-processing stage, starting with the standardization step, where multiple variations of English characters are turned to single unified forms to simplify the input. Sentence segmentation step breaks the raw string of content into separate sentences. Next, the URLs removal step eliminates any raw URLs included within the slide content. Tokenization step breaks each sentence to smaller tokens based on the nature of the characters (e.g., numbers, symbols) within the sentence.

Meanwhile, and lemmatization extracts the lemma of each word within the sentence to make understanding of similar patterns within the sentence easier for

the program. The final pre-processing task is the symbols removal step, aims at removing non-letter and non-number characters, and passes the finalized input ready for the mining stage.

2.2 Presentation Mining Stage

This stage focus on keyphrase extraction. The process begins further breaking down input phrases from pre-processing stage into part-of-speech sequences, and tagged according to their word category disambiguation such as adjective, determiner or proper noun. After having the words shortlisted, phrases would be formed by combination of the remained word multiples, and then those formed phrases would be shortlisted through a set of various criteria. Next, having the phrases shortlisted, then n -grams (unigrams, bigrams, and trigrams) of those phrases would be generated separately for the next calculations and form various candidates. As a final step, the highest-ranking candidate would be selected as the document keyphrases and would be passed to the visualization component for output generation. The algorithm that does the final ranking with the formula as shown in Eq. 1,

$$C = S(A) - \frac{S(A \cap B)}{S(B)} \quad (1)$$

where $S(A)$ is the total number of occurrences of the n -grams. $S(A \cap B)$ is the total number of occurrences of n -grams, which are the keyphrases. $S(B)$ is the total number of keyphrases. The dependent variable C may have a non-negative value for any measure of A, B . Although this formula does not directly contain a Term Frequency (TF) element, but it does work in a similar structure to a TF formula as its roots also refer to the frequency of the keyphrases. Equation 2 shows the arithmetic calculation based on this formula, while having all the $S(A)$, $S(B)$, and $S(A \cap B)$ counted and prepared in the earlier stages of the system.

$$\text{weight} = \text{frequency} - \frac{\text{totalKeyPhraseWithNGram}}{\text{totalKeyPhrases}} \quad (2)$$

2.3 Visualization Stage

The last component of the system, visualization, is designed to shape the extracted information into a user readable format by presenting the keyphrases in form of a visual knowledge display as well as producing a report of the document content in web format. The output from the presentation mining system is in the form of the visual knowledge display, which is mind map generated in Powerpoint (pptx) without edit restriction. The generation of a mind map in .pptx format was highly reliant on the close integration of Microsoft C#.NET APIs with the Microsoft PowerPoint. This research utilized a template of the Microsoft PowerPoint Smart Art objects, and added multiple branches based on slide numbers and their keyphrase content.

2.4 Evaluation Strategies

The mind maps that are generated based on the keyphrases extraction algorithm are evaluated against a domain ontology, a dictionary, and subject matter experts. The results are measured by precision and recall criteria defined by the number of matches between the system-extracted keyphrases against the human-extracted keyphrases. In the context of this research, precision is the ratio of candidates categorized correctly as keyphrases to the total number of candidates that were categorized as relevant. Recall is the ratio of the relevant keyphrases selected and categorized correctly, to the total number of relevant keyphrases in the full collection. F-measure is a weighted harmonic of the precision and recall and are appropriate when averaging rates and frequencies.

Ontology-Based Evaluation. Figure 2 shows the AI ontology developed based on the widely used AI textbook by Russell and Norvig [8]. This textbook is chosen because its worldwide use able to provide the research with a big collection of course slides as the test data. The datasets involved in the experiments include 500 pages of presentation slides for Artificial Intelligence course at introductory level in over 1,200 universities and 100 countries worldwide. The slides from the textbook are of varying topics, length, and structure. This ontology will be used as the source of comparison against the output from the proposed keyphrase extraction algorithm.

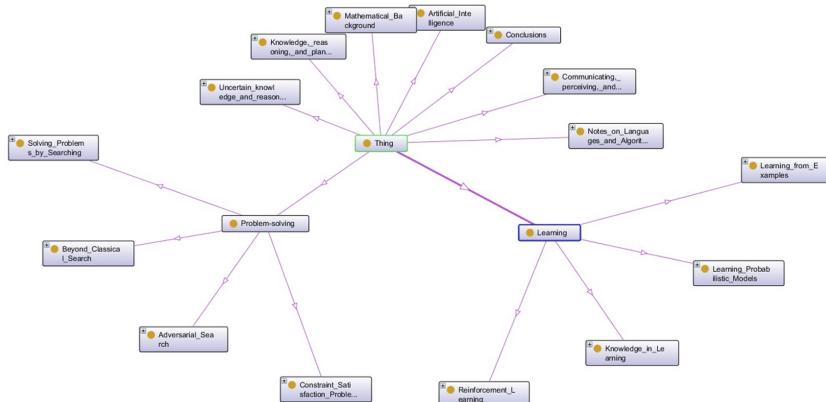


Fig. 2. Ontology for Artificial Intelligence based on [8]

Dictionary-Based Evaluation. The second source to compare the output of the keyphrase extraction algorithm is the International Dictionary of Artificial Intelligence [7]. This dictionary was chosen because it is a commonly accepted source and is mainly used as a reference book in Artificial Intelligence courses

across the world. In preparing the dictionary data, the PDF version of this dictionary was first converted to HTML format using an online free convertor tool (<http://www.zamzar.com/>). Next, the keyphrases from the HTML document were separated using a short JavaScript snippet using various format of the keyphrases in the generated HTML in order to differentiate them from the rest of the text content. The extracted list of keyphrases were then stored in a plain text file with the keyphrases separated using a new line character.

Expert Evaluation. Following [11], the performance of the keyphrase extraction algorithm is also evaluated by human expert evaluation. In this research, expert reviews are solicited from a set of questionnaires. The questionnaire is sent to more than ten (10) domain experts from both public and private universities throughout the world as well in Malaysia. The domain experts are carefully chosen from highly qualified, well-experienced academicians and researchers from the field of Artificial Intelligence domain with some experience using the Artificial Intelligence textbook [8].

3 Results and Discussion

A prototype of presentation mining system that takes in a set of presentation slides as the input is developed to produce a single mind map as the output. Figure 3 shows the interface for the presentation mining system, where users are provided with simplified options to convert slides into mind maps. While the system allows batch processing by providing the users with an option to input multiple .pptx files at a time, it displays the current file being processed and the total number of files that has been processed in a period of time. Upon completion of each conversion, the two buttons at the bottom of the screen are enabled, allowing user to view both the HTML and .pptx output files.



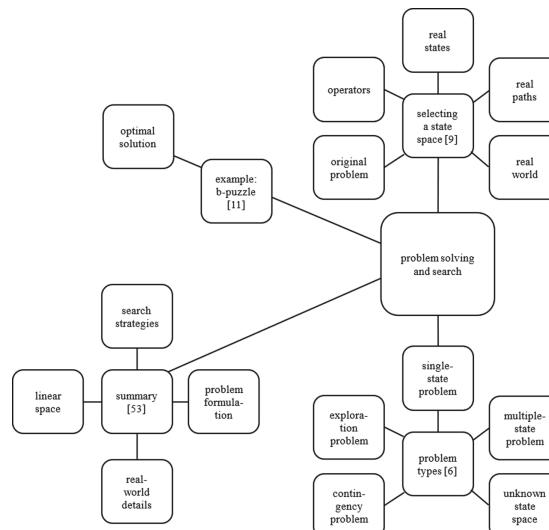
Fig. 3. Interface for the Presentation Mining System

Table 1 shows the summary of count for keywords and keyphrases extracted by the keyphrase extraction algorithm compared to the number of pages that a presentation slide has.

Table 1. Summary of Number of Slides and Extracted Keywords and Keyphrases

Slide	Title	#Slides	# Extracted
Chapter 1	Artificial Intelligence	13	30
Chapter 2	Intelligent Agents	16	23
Chapter 3	Problem Solving and Search	63	24
Chapter 4a	Informed Search Algorithms	34	25
Chapter 4b	Constraint Satisfaction Problems	27	28
Chapter 5	Game Playing	26	23
Chapter 6	Logical Agents	30	24
Chapter 7	First-Order Logic	22	24
Chapter 9a	Inference in First-Order Logic	16	27
Chapter 9b	Industrial-Strength Inference	13	28
Chapter 11	Planning	16	27
Chapter 13	Planning and Acting	15	22
Chapter 13	Uncertainty	17	22
Chapter 15a	Belief Networks	24	22
Chapter 15b	Inference in Belief Networks	27	27
Chapter 16	Rational Decisions	22	29
Chapter 17a	Complex Decisions	10	25

Figure 4 shows the mind map generated from the presentation mining system in Microsoft PowerPoint (*.pptx) using the Microsoft C#/.NET API. Although the Smart Art radial cluster with 3-level node can be break up into more nodes, it has been standardized to 5 nodes in the third level, based on recommendation by

**Fig. 4.** Mind Map Generated for Chapter03.pptx

the Buzan technique. Next, the keywords and keyphrases extracted are evaluated by comparing them against a domain-specific ontology, a dictionary, and expert evaluation.

3.1 Ontology-Based Evaluation

In this evaluation method, the extracted keyphrases by the MiKe algorithm were searched for and compared against the list of keyphrases mapped in the .owl file developed based on the textbook “Artificial Intelligence: A Modern Approach” [8]. The matches were then marked as TRUE in the results table. Table 2 shows the results from the keyphrase extraction as compared to the AI ontology. The results showed that MiKe algorithm had 96 keywords and keyphrases matched with the ontology, proving the keywords and keyphrases accuracy.

Table 2. Results Extracted and Matched against AI Ontology

Slide	# Extracted	# Matched	Precision	Recall	F-Measure
Chapter 1	30	7	0.86	0.38	0.52
Chapter 2	23	4	0.75	0.38	0.50
Chapter 3	24	11	0.82	0.82	0.82
Chapter 4a	25	5	0.80	0.27	0.40
Chapter 4b	28	5	0.60	0.21	0.32
Chapter 5	23	4	1.00	0.27	0.42
Chapter 6	24	9	0.67	0.55	0.60
Chapter 7	24	7	0.57	0.44	0.50
Chapter 9a	27	4	0.75	0.21	0.70
Chapter 9b	28	4	1.00	0.25	0.40
Chapter 11	27	7	0.57	0.40	0.47
Chapter 13	22	2	1.00	0.20	0.33
Chapter 13	22	6	0.67	0.40	0.50
Chapter 15a	22	2	1.00	0.33	0.50
Chapter 15b	27	6	0.83	0.33	0.48
Chapter 16	29	7	0.57	0.31	0.40
Chapter 17a	25	6	1.00	0.35	0.52
Average	25.29	5.64	0.79	0.36	0.49

3.2 Dictionary-Based Evaluation

In this evaluation method, the HTML pages of the Online Dictionary of Artificial Intelligence [7] were first converted to plain text format, and put together as a single file for the test program to use. The presentation mining system then searched for each and every extracted keyword and keyphrase from the

input slides inside the new text file generated from the dictionary, and mark any matches as TRUE in the results table. Table 3 shows the keyphrase extraction results by the MiKe algorithm as compared to the AI dictionary, which has 24 matched words with the online dictionary. AI Dictionary online covers only certain chapters in the AI areas.

Table 3. Results Extracted and Matched against AI Dictionary

Slide	# Extracted	# Matched	Precision	Recall	F-Measure
Chapter 1	30	5	1.00	0.46	0.63
Chapter 2	23	0	0.00	0.00	0.00
Chapter 3	24	3	1.00	0.27	0.43
Chapter 4a	25	1	1.00	0.07	0.13
Chapter 4b	28	2	0.50	0.07	0.12
Chapter 5	23	3	0.67	0.15	0.25
Chapter 6	24	2	1.00	0.18	0.31
Chapter 7	24	0	0.00	0.00	0.00
Chapter 9a	27	0	0.00	0.00	0.00
Chapter 9b	28	2	1.00	0.13	0.22
Chapter 11	27	1	1.00	0.10	0.18
Chapter 13	22	0	0.00	0.00	0.00
Chapter 13	22	1	1.00	0.09	0.08
Chapter 15a	22	0	0.00	0.00	0.00
Chapter 15b	27	2	1.00	0.13	0.23
Chapter 16	29	0	0.00	0.00	0.00
Chapter 17a	25	2	1.00	0.11	0.21
Average	25.29	1.41	0.60	0.10	0.20

3.3 Expert Evaluation

For expert evaluation, the keywords and keyphrases extracted by the presentation mining system were listed into a questionnaire to be evaluated by domain expert, which in this research are academicians teaching the course Artificial Intelligence all across the world. The purpose of this questionnaire is to solicit experts' opinion on given keywords and keyphrases that have been extracted from the slides are correct and valid keywords and keyphrase from the domain.

The questionnaire was prepared chapter by chapter with a total of 17 chapters. The keywords and keyphrases were put into a table format for the Expert Evaluator to mark TRUE for the keywords and keyphrases that are correct. Four (4) valuable experts' opinion were received from the total of 11 invitations sent out. Human Expert 1 (E1) and Human Expert 2 (E2) completed 17 chapters

of the questionnaire. Human Expert 3 (E3) and Human Expert (E4) completed random chapters that covers combined output from all 17 chapters. Table 4 shows the evaluation results based on averaged precision and recall values as shown in Table 4.

Table 4. Comparison of Evaluations by Human Experts

	Averaged Precision	Averaged Recall	Averaged F-Measure
Human Expert 1	0.724588235	0.987764706	0.822823529
Human Expert 2	0.866647059	0.995470588	0.923058824
Human Expert 3	1	0.982764706	0.990882353
Human Expert 4	1	0.982764706	0.990882353

Based on the results from three evaluations, the keyphrase extraction algorithm, MiKe, within the Presentation Mining system has shown promising performance. The difference between the values of precision, recall, and F-measure across AI ontology, AI dictionary, and human experts indicate the importance of the source knowledge base for the domain under study. Note that MiKe is based on keyphrase weights and can be significantly improved by training with Large Language Models (LLM).

4 Conclusions

This research proposed a Presentation Mining framework, in which the implementation system is able to mine keywords and key phrases from a collection of PowerPoint slides and to generate a visual display such as the mind map based on the extracted words and phrases. Since there is no standard corpus of testing keyphrases extraction, the slides for the widely used Artificial Intelligence textbook called the Artificial Intelligence: A Modern Approach (AIMA) written by [8] were used as the benchmark dataset. As there are also no uniform standard for the evaluation of keyphrases extraction, the performance of the keyphrase extraction algorithm was compared using three methods; against an AI ontology, an AI dictionary, and human expert evaluations. The results showed that the proposed presentation mining system was best evaluated by human experts with average of 89% precision, 98% recall, and 93% F-Measure. The next best evaluation method was by using the ontology with average of 79% precision, 36% recall, and 49% F-measure. The results showed that the proposed framework is able to extract and generate reasonable visualization from presentation slides.

References

1. Hay, D., Kinchin, I., Lygo-Baker, S.: Making learning visible: the role of concept mapping in higher education. *Stud. High. Educ.* **33**(3), 295–311 (2008)
2. Keet, C.M.: Mind maps. In: *The What and How of Modelling Information and Knowledge*, pp. 13–23. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-39695-3_2
3. Kinchin, I., Cabot, L.: Using concept mapping principles in powerpoint. *Eur. J. Dent. Educ.* **11**(4), 194–199 (2007)
4. Kudelić, R., Konecki, M., Maleković, M.: Mind map generator software model with text mining algorithm. In: *Proceedings of the ITI 2011, 33rd International Conference on Information Technology Interfaces*, pp. 487–494. IEEE (2011)
5. Liu, F., Huang, X., Huang, W., Duan, S.X.: Performance evaluation of keyword extraction methods and visualization for student online comments. *Symmetry* **12**(11), 1923 (2020)
6. Parida, U., Nayak, M., Nayak, A.K.: Insight into diverse keyphrase extraction techniques from text documents. In: Mishra, D., Buyya, R., Mohapatra, P., Patnaik, S. (eds.) *Intelligent and Cloud Computing. SIST*, vol. 194, pp. 405–413. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-5971-6_44
7. Raynor, W.: *International Dictionary of Artificial Intelligence*. Routledge, London (2020)
8. Russell, S.J., Norvig, P.: *Artificial intelligence a modern approach*. London (2010)
9. Stamper, J., Gaind, B., Thankachan, K., Nguyen, H., Moore, S.: Hierarchical concept map generation from course data. In: *AAAI 2023 Workshop on Artificial Intelligence in Education (AI4Edu)* (2023)
10. Supendi, Y., Yulianto, E., Dewi, D.A., Syarif, K.: Thesaurus-based query expansion on information retrieval to improve the quality of document searching result. *J. Theor. Appl. Information Technol.* **101**(21) (2023)
11. Vveinhardt, J., Gulbovaitė, E.: Expert evaluation of diagnostic instrument for personal and organizational value congruence. *J. Bus. Ethics* **136**, 481–501 (2016)
12. Yue, M., Zhang, M., Zhang, C., Jin, C.: The effectiveness of concept mapping on development of critical thinking in nursing education: a systematic review and meta-analysis. *Nurse Educ. Today* **52**, 87–94 (2017)



Enhancing Network Intrusion Detection Systems Through Dimensionality Reduction

Mosleh M. Abualhaj^{1(✉)}, Sumaya N. Al-Khatib¹, Ali Al-Allawee², Alhamza Munther³, and Mohammed Anbar⁴

¹ Department of Networks and Cybersecurity,
Al-Ahliyya Amman University, Amman 19328, Jordan
{m.abualhaj,sumayakh}@ammanu.edu.jo

² IRIMAS Lab, University of Haute Alsace, 68000 Colmar, France
³ University Technology and Applied Sciences, Sur 411, Oman
⁴ National Advanced IPv6 Centre, Universiti Sains Malaysia,
11700 Gelugor, Penang, Malaysia

Abstract. The proliferation of data on the Internet has intensified the difficulties faced by network intrusion detection systems (NIDS) in handling large dimensions of data that include irrelevant and duplicate elements. This process entails a significant investment of time and effort in accurately identifying the attack while also experiencing a rise in false alarms. Applying dimensionality reduction may solve this problem. This paper employs and compares two feature selection methods, the Whale Optimization Algorithm (WOA) and Hawks Optimization (HHO). These methods reduce the number of features by finding the minimum number without affecting the performance of the NIDS system. The main idea is to select the key features from the NSL-KDD datasets using WOA and HHO. The results have shown a better performance of the NIDS system when using HHO rather than WOA with binary classification. The HHO and WOA achieved 93.83% and 92.83% accuracy, respectively. These results were achieved using the Support Vector Machine Classifier (SVM).

Keywords: Feature selection · WOA · HHO · NIDS · SVM

1 Introduction

Companies and individuals increasingly depend on cyberspace services. The high dependency on the cyberspace service comes at the cost of security and privacy. A sheer volume of cyberattacks target cyberspace services. These cyberattacks are causing a considerable loss of money and information [1, 2]. Several cybersecurity solutions are available to protect against cyberattacks, including the Network Intrusion Detection System (NIDS). The NIDS systems are using advanced techniques to detect many advanced cyberattacks. For instance, the NIDS systems

are using Machine Learning (ML) to protect against advanced cyberattacks that are based on artificial intelligence (AI) [3,4]. ML is a subset of AI that focuses on developing systems capable of learning from existing data in order to generate accurate predictions about new, unknown data. Enhancing the NIDS systems with ML enhances its capacity to identify and respond to threats. Nevertheless, a significant challenge arises when merging the NIDS with ML due to the substantial volume of data that the NIDS systems must process [5,6]. ML employs feature selection algorithms, which are specialized sorts of algorithms, to manage vast quantities of data effectively and enhance the detection capabilities of the NIDS in identifying cyberattacks. Feature selection is a technique that lowers the number of input variables in a model by including only relevant data and excluding irrelevant or noisy data. Metaheuristic algorithms are computational intelligence paradigms specifically employed for solving complex optimization issues such as feature selection [7,8]. This study will evaluate the efficacy of the Whale Optimization Algorithm (WOA) and the Harris Hawks Optimization (HHO) algorithm in the NIDS systems utilizing the Support Vector Machine Classifier (SVM) [9,10].

2 Related Works

This section provides an overview of previous research that conducted feature selection using the NSLKDD dataset. Vinutha et al. [11] investigated several feature selection techniques on the NSLKDD dataset, including Chi-square, Info Gain, SU, Cfs, and Gain Ratio. The experiments of several ensemble and single classifiers using WEKA with these feature selection techniques showed that AdaBoost greatly improved the classification accuracy. Nskh et al. [12] suggested a model that utilizes the Principal Component Analysis (PCA) method for feature selection to reduce the data size. The suggested model uses a 10% subset of the KDDCup'99 dataset for training and the complete dataset for testing. The performance of the proposed model was tested using the SVM classifier. Then, the results obtained from SVM without PCA and from SVM with PCS were compared. The comparison results showed that the accuracy increases and the detection time decreases when PCA is used with SVM. Ingre et al. [13] proposed an NIDS model that employs feature selection and Artificial Neural Networks (ANN) to detect the attack. The employed selected features are based on previous studies and manual selection. The resulting NSLKDD dataset, after feature selection, contains 29 features rather than 41. The model obtained 81.2% accuracy with binary classification and 79.9% with multiclass classification. Al-Jarrah et al. [14] introduced a NIDS model that uses two new feature selection techniques: Random Forest- Forward Selection Ranking (RF-FSR) and Random Forest- Backward Elimination Ranking (RF-BER). The evaluation results showed that the selected features by the suggested techniques effectively improved the accuracy of the introduced NIDS model. However, it is important to note that these results were obtained by 10-fold cross-validation, which may reflect a different accuracy level when applied to the whole testing dataset. Gaikwad et al. [15] introduced the GA feature selection strategy to identify the most

relevant features from 41 features in the NSLKDD dataset. They evaluated the accuracy of the selected features using the Bagging method in ML to create an NIDS, with the Partial Decision Tree (DT) rule as the underlying classifier. The findings demonstrated that the 10-fold cross-validation achieved an accuracy rate of 99.71%, while the test dataset yielded an accuracy rate of 78.37%, surpassing other classifiers. Pervez et al. [16] introduced wrapper feature selection and assessed its performance using the NSLKDD dataset. They employed SVM and attained an accuracy of 91% using three features and 99% using 41 features on the entire training data. However, when tested, the accuracy dropped to 82.37% using 14 features in binary classification.

3 NSLKDD Dataset Preprocessing

In order to analyze the effectiveness of the WOA and HHO optimizers, the NSLKDD dataset will be utilized, and the SVM classifier will be essential in this evaluation. According to [17,18], the NSLKDD dataset includes forty features that can be used to identify attacks. These features are land, wrong_fragment, dst_host_same_src_port_rate, num_failed_logins, num_compromised, num_shells, srv_count, same_srv_rate, dst_host_count, dst_host_srv_count, num_outbound_cmds, dst_host_error_rate, Count, dst_bytes, dst_host_srv_error_rate, srv_diff_host_rate, root_shell, su_attempted, service, src_bytes, dst_host_srv_diff_host_rate, srv_error_rate, hot, logged_in, srv_error_rate, num_access_files, Flag, error_rate, dst_host_diff_srv_rate, diff_srv_rate, protocol_type, dst_host_error_rate, num_root, num_file_creations, is_guest_login, is_host_login, urgent, dst_host_srv_error_rate, dst_host_same_srv_rate, and error_rate. In addition to that, the NSLKDD dataset includes 148,518 records representing attacks as well as normal data. The data pertaining to the attack can be classified into four distinct categories: Denial of Service (DoS), Probe, Remote to Local (R2L), and User to Root (U2R). Every single one of the four categories includes a number of subcategories [18–20]. Only the binary classification of data, which categorizes data as either normal or attack, is taken into consideration in this paper. Because of this, the word “Attack” will be substituted for each and every label in the output column, regardless of the different types or subtypes of attacks. The output column will, therefore, only include two values, which are Normal and Attack, for the purpose of binary classification.

A portion of the NSLKDD dataset consists of textual data, which poses a hindrance to the functioning of the ML algorithms. Consequently, the Label-Encoder technique substituted the textual data with numerical data. The Label-Encoder approach assigns numerical values to each category inside a textual feature, ranging from 0 to n. Here, n represents the feature’s total number of distinct values minus 1. As an illustration, the “flag” attribute consists of 11 distinct values: OTH, REJ, RSTO, RSTOS0, RSTR, S0, S1, S2, S3, SF, and SH. The Label-Encoder algorithm assigns the values 0, 1, 2, 3, ..., and 11 to replace these 11 variables. Another instance is the output column, which comprises two distinct values: Normal and Attack. The Label-Encoder algorithm substitutes

these two values with 0 and 1, correspondingly. Furthermore, certain attributes of the NSLKDD dataset exhibit data that is spread across wide ranges, affecting the performance of ML algorithms. Hence, the Min-Max scaler method substitutes the values of these features with narrow ranges ranging from 0 to 1, thereby preventing ML systems from being influenced by huge data [17,18]. Tables 1 and 2 display a subset of the NSLKDD dataset before and after applying the Label-Encoder and Min-Max scaler algorithms.

Table 1. Sample of the NSLKDD dataset before Label-Encoder and Min-Max scaler algorithms

No	Instances	Output
1	tcp, private, S0, 0, 0, 0	Attack
2	icmp, urp.i, SF, 181, 0, 0	Normal
3	udp, domain_u, SF, 42, 42, 0	Normal
4	tcp, private, S0, 0, 0, 0	Attack
5	tcp, ftp_data, SF, 12, 0, 0	Normal
6	tcp, auth, S0, 0, 0, 0	Attack

Table 2. Sample of the NSLKDD dataset after Label-Encoder and Min-Max scaler algorithms

No	Instances	Output
1	0.5, 0.6875, 0.5, 0, 0, 0	1
2	0, 0.9375, 0.9, 2.02050601507989E-06, 0, 0	0
3	1, 0.171875, 0.9, 4.68846699631799E-07, 5.97554125599048E-06, 0	0
4	0.5, 0.6875, 0.5, 0, 0, 0	1
5	0.5, 0.296875, 0.9, 1.33956199894799E-07, 0, 0	0
6	0.5, 0.046875, 0.5, 0, 0, 0	1

In the preprocessing phase of the NSLKDD dataset, the feature selection procedure follows translating the data into a numerical format and scaling it to a consistent range. The WOA and HHO metaheuristic algorithms will be utilized to make feature selections on the NSLKDD dataset. The WOA imitates the hunting patterns of whales, while the HHO imitates the hunting patterns of hawks. There are multiple benefits associated with the two algorithms: i) Applicability in a diverse set of optimization problems, ii) Reliable for both exploration and exploitation and iii) Straightforward algorithms that may be readily comprehended and applied. Nevertheless, the efficacy of these algorithms may fluctuate depending on the particular problem to which they are applied. The selection of either WOA or HHO algorithms is contingent upon the specific attributes of the problem being addressed and the available computational resources [9, 10, 21–23].

The WOA and HHO algorithms are extensively utilized in the realm of cybersecurity, specifically in conjunction with NIDS systems [9, 22]. Both of these algorithms have been utilized on the NSLKDD dataset in order to assess their effectiveness with NIDS systems. The WOA algorithm has chosen 16 features from the NSLKDD dataset out of a total of 40. These features are: is_host_login, dst_host_error_rate, srv_count, dst_host_same_src_port_rate, same_srv_rate, service, num_failed_logins, num_outbound_cmds, srv_diff_host_rate, src_bytes, num_access_files, error_rate, num_root, is_guest_login, Flag, and srv_error_rate. However, the HHO algorithm has chosen 13 features from the NSLKDD dataset out of a total of 40. These features are dst_host_diff_srv_rate, Count, dst_host_count, hot, dst_host_srv_count, Flag, dst_bytes, diff_srv_rate, num_access_files, protocol_type, src_bytes, urgent, and dst_host_same_src_port_rate.

4 Attack Prediction Using SVM

An SVM is a type of ML algorithm that is used for supervised classification problems. The objective is to identify an ideal hyperplane that effectively separates distinct classes in the feature space. The support vectors are the data points closest to the decision boundary. Figure 1 shows the SVM algorithm. The SVM aims to optimize the margin, the distance between the decision boundary, and the closest data point from each class. This optimization enhances the model's robustness and ability to generalize. The technique is capable of dealing with non-linear boundaries by utilizing kernel functions. Additionally, a regularization parameter is employed to strike a balance between a broader margin and accurately identifying training points.

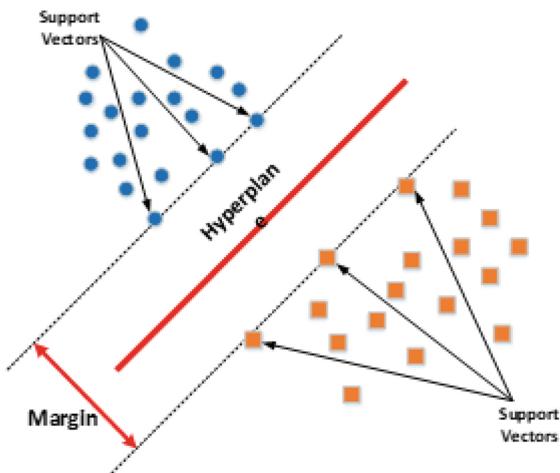


Fig. 1. The SVM algorithm [22]

5 Performance Evaluation

The PC that was used in the implementation has the following specifications: Intel Corei9-14900K Processor 24 (8 P-cores + 16 E-cores, Speed 6 GHz, 36M Cache, and 32 threads), 256 GB SSD, 32 GB RAM, and Ubuntu 21.9 O.S. Python has been utilized to implement WOA, HHO, and SVM to attain the results. The K-Fold Cross-Validation technique was employed to validate the results using a value of k equal to 5. The confusion matrix provides a summary of the predictions made by a model on a classification problem, showing the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. The Accuracy (Eq. 1), Precision (Eq. 2), Recall (Eq. 3), Matthews Correlation Coefficients (MCC) (Eq. 4), and F1-Score (Eq. 5) metrics were derived from the confusion matrix [16,24].

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (2)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (3)$$

$$MCC = \frac{((TP * TN) - (FP * FN))}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}} \quad (4)$$

$$F1-score = 2 \frac{(Precision * Recall)}{(Precision + Recall)} \quad (5)$$

Figure 2 illustrates the Accuracy of the HHO in comparison to the WOA. The HHO algorithm attained an accuracy rate of 93.83%, whilst the WOA algorithm acquired an accuracy rate of 92.83%. The HHO algorithm demonstrates superior performance compared to the WOA algorithm, with a 1.00% higher accuracy in finding attacks by NIDS systems utilizing the NSLKDD dataset and SVM algorithm. Figure 3 illustrates the Precision of the HHO in comparison to the WOA. The HHO algorithm attained a Precision rate of 93.83%, whilst the WOA algorithm acquired a Precision rate of 92.83%. The HHO algorithm demonstrates superior performance compared to the WOA algorithm, with a 1.00% higher Precision in finding attacks by NIDS systems utilizing the NSLKDD dataset and SVM classifier.

Figure 4 illustrates the Recall of the HHO in comparison to the WOA. The HHO algorithm attained a Recall rate of 93.83%, whilst the WOA algorithm acquired a Recall rate of 92.83%. The HHO algorithm demonstrates superior performance compared to the WOA algorithm, with a 1.00% higher Recall in finding attacks by NIDS systems utilizing the NSLKDD dataset and SVM classifier. Figure 5 illustrates the MCC of the HHO in comparison to the WOA. The HHO algorithm attained an MCC rate of 90.03%, whilst the WOA algorithm acquired an MCC rate of 88.37%. The HHO algorithm demonstrates superior

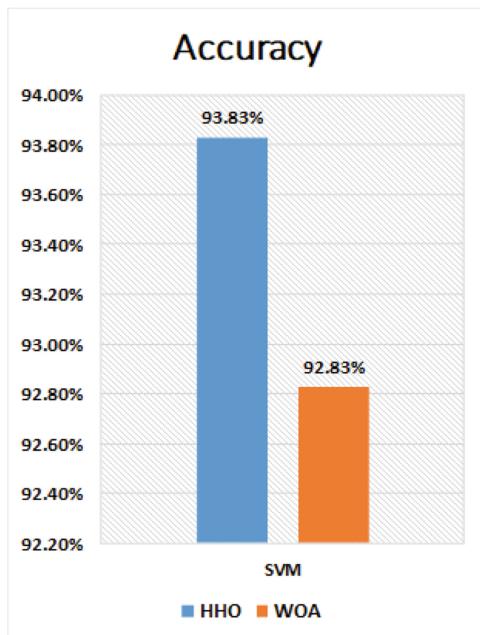


Fig. 2. Accuracy of the WOA and HHO algorithms

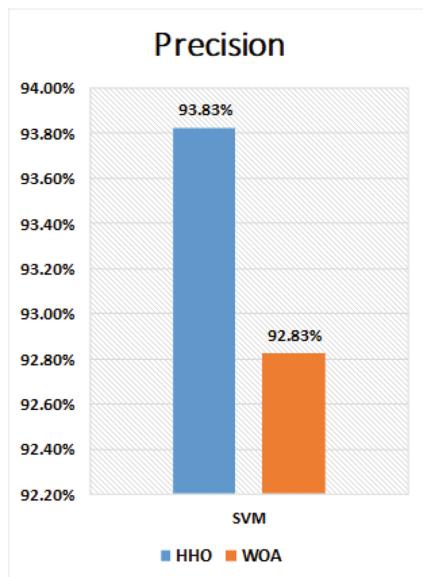


Fig. 3. Precision

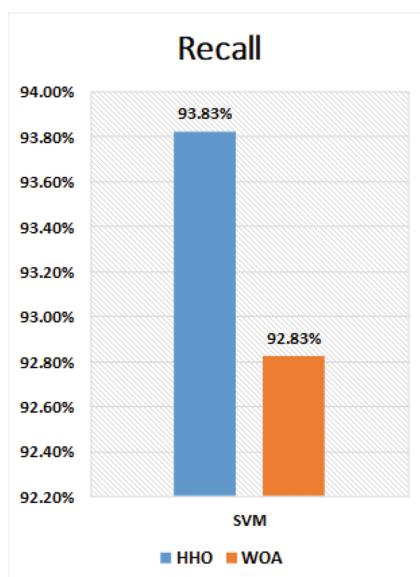
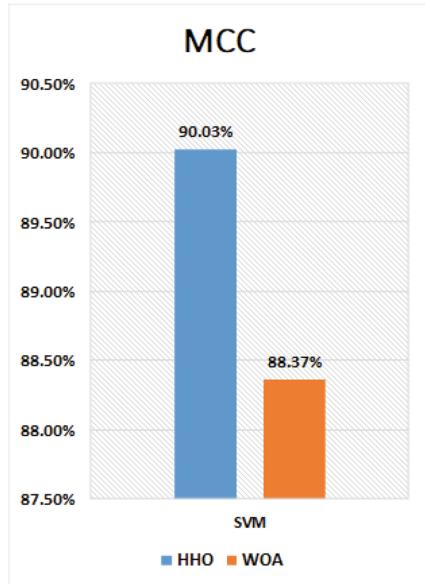
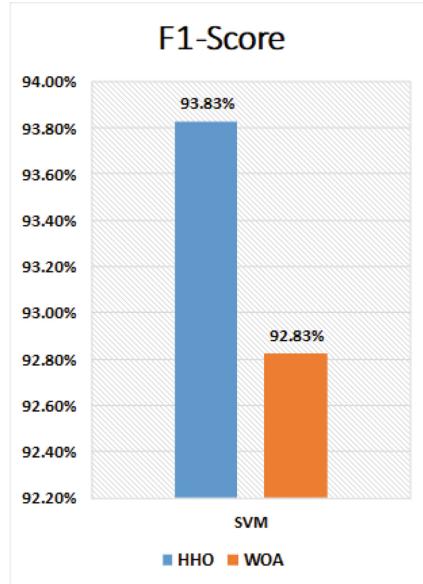


Fig. 4. Recall

**Fig. 5.** MCC**Fig. 6.** F1-Score

performance compared to the WOA algorithm, with a 1.66% higher MCC in finding attacks by NIDS systems utilizing the NSLKDD dataset and SVM classifier. Figure 6 illustrates the F1-Score of the HHO in comparison to the WOA. The HHO algorithm attained an F1-Score rate of 93.83%, whilst the WOA algorithm acquired an F1-Score rate of 92.83%. The HHO algorithm demonstrates superior performance compared to the WOA algorithm, with a 1.00% higher F1-Score in finding attacks by NIDS systems utilizing the NSLKDD dataset and SVM classifier.

6 Conclusions and Future Work

The primary objective of feature selection is to decrease the dataset size and enhance the NIDS system's performance. This work introduced feature selection-based optimization algorithms, namely the WO and the HHO algorithms. Additionally, we utilized SVM for data classification. The comparative analysis of the two optimizers demonstrated that the HHO optimizer can attain a remarkable accuracy of 93.83% when selecting 13 features. Simultaneously, the WOA optimizer can attain a slightly lower accuracy of 92.83% by selecting 16 features. Furthermore, the decrease in the quantity of features resulted in a decrease in the duration of both the training and testing processes. In the future, our plan is to establish connections from real-time networks in order to test the two optimizers. We will utilize a sophisticated classifier to achieve enhanced detection

capabilities, specifically for identifying different types of attacks. This is particularly challenging in the NIDS system, as the behavior of these attacks closely resembles normal network activity. Hence, detecting it is a significant challenge.

References

- Bang, M., Saraswat, H.: Building an effective and efficient continuous web application security program. In: 2016 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), London, United Kingdom, pp. 1–4 (2016). <https://doi.org/10.1109/CyberSA.2016.7503287>
- Abualhaj, M., et al.: A fine-tuning of decision tree classifier for ransomware detection based on memory data. *Int. J. Data Netw. Sci.* **8**(2), 733–742 (2024)
- Chandre, P.R., Mahalle, P.N., Shinde, G.R.: Machine learning based novel approach for intrusion detection and prevention system: a tool based verification. In: IEEE Global Conference on Wireless Computing and Networking (GCWCN), Lonavala, India, pp. 135–140 (2018). <https://doi.org/10.1109/GCWCN.2018.8668618>
- Abualhaj, M., Abu-Shareha, A., Shambour, Q., Alsaaidah, A., Al-Khatib, S., Anbar, M.: Customized K-nearest neighbors' algorithm for malware detection. *Int. J. Data Netw. Sci.* **8**(1), 431–438 (2024)
- Zhou, P.-Z., Zhang, H., Liang, W.: Research on hybrid intrusion detection based on improved Harris Hawk optimization algorithm. *Connection Sci.* **35**(1), 1–24 (2023). <https://doi.org/10.1080/09540091.2023.2195595>
- Yedukondalu, G., Bindu, G.H., Pavan, J., Venkatesh, G., SaiTeja, A.: Intrusion detection system framework using machine learning. In: 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, pp. 1224–1230 (2021). <https://doi.org/10.1109/ICIRCA51532.2021.9544717>
- Mendes, H., Quincozes, S.E., Quincozes, V.E.: A web user interface tool for metaheuristics-based feature selection assessment for IDSs. In: 6th Cyber Security in Networking Conference, Rio de Janeiro, Brazil (2022). <https://doi.org/10.1109/csnet56116.2022.9955616>
- Manjur, K., Fadi, A., Abdalla, A., Mosleh, M.: A three layered decentralized IoT biometric architecture for city lockdown during COVID-19 outbreak. *IEEE Access* **8**, 163608–163617 (2020)
- Alazab, M., Abu Khurma, R., Castillo, P., Abu-Salih, B., Martín, A., Camacho, D.: An effective networks intrusion detection approach based on hybrid Harris Hawks and multi-layer perceptron. *Egypt. Inform. J.* **25**(1), 1–9 (2024). <https://doi.org/10.1016/j.eij.2023.100423>
- Mirjalili, S., Lewis, A.: The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67 (2016). <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- Vinutha, H.P., Poornima, B.: An ensemble classifier approach on different feature selection methods for intrusion detection. In: Bhateja, V., Nguyen, B.L., Nguyen, N.G., Satapathy, S.C., Le, D.-N. (eds.) *Information Systems Design and Intelligent Applications*. AISC, vol. 672, pp. 442–451. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-7512-4_44
- Praneeth, N., Varma, M.N., Roshan, N.: Principle component analysis based intrusion detection system using support vector machine. In: International Conference on Recent Trends in Electronics Information Communication Technology, Bangalore, India, pp. 1344–1350 (2016)

13. Ingre, B., Anamika, Y.: Performance analysis of NSL-KDD dataset using ANN. In: International Conference on Signal Processing and Communication Engineering Systems, Guntur, India, pp. 92–96 (2015)
14. Al-Jarrah, O.Y., Siddiqui, A., Elsalamouny, M., Yoo, P.D., Muhaidat, S., Kim, K.: Machine-learning-based feature selection techniques for large-scale network intrusion detection. In: International Conference on Distributed Computing Systems Workshops, Madrid, Spain, pp. 177–181 (2014)
15. Gaikwad, D., Thool, R.C.: Intrusion detection system using bagging with partial decision tree base classifier. *Procedia Comput. Sci.* **49**(1), 92–98 (2015)
16. Muhammad Shkil, P., Dewan, M.F.: Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs. In: International Conference on Software, Knowledge, Information Management and Applications, Dhaka, Bangladesh, pp. 1–6 (2014)
17. Al-Mimi, H., Hamad, N.A., Abualhaj, M.M., Daoud, M.S., Al-Dahoud, A., Rasmi, M.: An enhanced intrusion detection system for protecting HTTP services from attacks. *Int. J. Adv. Soft Comput. Appl.* **15**(3) (2023)
18. Çavuşoğlu, Ü.: A new hybrid approach for intrusion detection using machine learning methods. *Appl. Intell.* **49**, 2735–2761 (2019). <https://doi.org/10.1007/s10489-018-01408-x>
19. Mosleh, A., Ahmad, A., Mohammad, O., Yousef, A., Mahran, A., Mohammad, A.: A paradigm for DoS attack disclosure using machine learning techniques. *Int. J. Adv. Comput. Sci. Appl.* **13**(3) (2022)
20. Al-Mimi, H., Hamad, N.A., Abualhaj, M.M.: A model for the disclosure of probe attacks based on the utilization of machine learning algorithms. In: 10th International Conference on Electrical and Electronics Engineering (ICEEE), Istanbul, Turkiye, pp. 241–247 (2023). <https://doi.org/10.1109/ICEEE59925.2023.00051>
21. Ruba, A., Awadallah, M.A., Ibrahim, A.: Binary Harris hawks optimisation filter based approach for feature selection. In: Palestinian International Conference on Information and Communication Technology (PICICT), Gaza, State of Palestine, pp. 59–64 (2021). <https://doi.org/10.1109/picict53635.2021.00022>
22. Basu, D., Singh, M., Gupta, A., Ranjan, S., Pareta, D.N., Biswa, M.: Survey paper: whale optimization algorithm and its variant applications. In: International Conference on Innovative Practices in Technology and Management (ICIPTM), Noida, India (2021). <https://doi.org/10.1109/iciptm52218.2021.9388344>
23. Nadimi-Shahraki, M., Zamani, H., Asghari Varzaneh, Z.: A systematic review of the whale optimization algorithm: theoretical foundation, improvements, and hybridizations. *Arch. Comput. Methods Eng.* **30**, 4113–4159 (2023). <https://doi.org/10.1007/s11831-023-09928-7>
24. Al-Mimi, H.M., Hamad, N.A., Abualhaj, M.M., Al-Khatib, S.N., Hiari, M.O.: Improved intrusion detection system to alleviate attacks on DNS service. *J. Comput. Sci.* **19**(12), 1549–1560 (2023). <https://doi.org/10.3844/jcssp.2023.1549.1560>



Performance Evaluation of Whale and Harris Hawks Optimization Algorithms with Intrusion Prevention Systems

Mosleh M. Abualhaj^{1(✉)}, Ahmad Adel Abu-Shareha², Ali Al-Allawee³, Alhamza Munther⁴, and Mohammed Anbar⁵

¹ Department of Networks and Cybersecurity, Al-Ahliyya Amman University, Amman 19328, Jordan
m.abualhaj@ammanu.edu.jo

² Department of Data Science and Artificial Intelligence, Al-Ahliyya Amman University, Amman 19328, Jordan

³ IRIMAS Lab, University of Haute Alsace, 68000 Colmar, France

⁴ University Technology and Applied Sciences, Sur 411, Oman

⁵ National Advanced IPv6 Centre, Universiti Sains Malaysia, 11700 Gelugor, Penang, Malaysia

Abstract. Intrusion Prevention Systems (IPS) protect computer networks from hostile activities. This study investigates the effectiveness of two optimization algorithms, Hawks Optimization (HHO) and Whale Optimization Algorithm (WOA), in selecting features to improve the performance of Intrusion Prevention Systems (IPS). The NSL-KDD dataset is employed as a standard to assess the influence of specific features on classification accuracy. The study uses the K-nearest neighbors (KNN) classification algorithm to assess the efficacy of feature selection techniques. The implementation is carried out using Python, using its flexible libraries and frameworks. Comparative investigation shows that when combining the chosen features, KNN achieves better outcomes than using the whole dataset of features. The results demonstrate that the combination of KNN with WOA provides a remarkable accuracy rate of 97.42%, highlighting the effectiveness of WOA in improving IPS performance. On the other hand, the combination of KNN and HHO demonstrates an impressive accuracy rate of 96.87%.

Keywords: WOA · HHO · KNN · IPS · NSL-KDD

1 Introduction

The cyber-world is the interconnected computers and digital networks where huge amounts of information flow worldwide. Individuals rely heavily on this information to study, work, and communicate. However, cybersecurity is one

of the key issues that should be considered carefully while using the cyber-world [1,2]. Numerous cyber-attacks surround the cyber-world, including Probe attacks, ransomware, DHCP spoofing, and viruses. Cybersecurity developers use numerous cybersecurity tools to defend the cyber-world, such as Spam filtering, firewalls, and intrusion prevention systems (IPS). However, the attackers continually invent new advanced methods to evade the cybersecurity tools. Currently, attackers are widely employing artificial intelligence (AI) tools to advance the attack tools. Hence, cybersecurity tools must be developed to use specific AI algorithms to keep up with the new sophisticated attacks [3,4].

Machine Learning (ML) is a field of AI that cybersecurity developers wildly use to advance the techniques used by cybersecurity tools, including IPS systems, therefore improving the attack detection rate. ML algorithms are fed by raw data obtained from analyzing previous attacks. The ML algorithms then process this data and generate new rules that make IPS forecast new unseen attacks. As the data size increases, so does the complexity of analyzing the data and generating new rules, negatively impacting the IPS detection rate [5,6]. To handle the large data size that the IPS deals with, the ML uses feature selection algorithms to select the key attack features that can be used to detect the attack. Therefore, reducing the size of the processed data reduces the complexity of the IPS rules and enhances the attack detection rate. Metaheuristic algorithms are widely used with IPS systems for feature selection. Metaheuristic algorithms are proven to be efficient and improve the performance of IPS systems [7,8]. In this article, we will evaluate the performance of the whale optimization algorithm (WOA) and Harris Hawks optimization (HHO) with IPS systems, utilizing the K-nearest neighbors (KNN) ML algorithm [9,10].

This paper is organized as follows: Sect. 2 briefs some ML works that use the NSL-KDD dataset. The NSL-KDD dataset used in the evaluation is discussed in Sect. 3. Section 4 spotlights the HHO and WOA metaheuristic algorithms. Section 5 details the HHO and WOA comparative models, including data pre-processing, feature selection, and classification. Section 6 analyzes the HHO and GWO performance using the confusion matrix metrics. Finally, Sect. 7 concludes the paper.

2 Related Work

Bajaj et al. [11] utilized the information gain model to select features and subsequently employed J48, Naïve Bayes, NB tree, support vector machine (SVM), and simple cart algorithms for binary classification. Bhoria et al. [12] employ cart 4.5 to detect DoS attacks. A 6-fold cross-validation technique is utilized on a 20% subset of the NSL KDD dataset for training and testing purposes. The collection has 22,495 records encompassing both typical instances and DoS attacks. Ibrahim et al. [13] utilized Self-Organizing Maps (SOM) to analyze the NSLKDD and KDD 99 datasets. The findings demonstrate that the binary classification performance on the KDD 99 dataset was superior to that of the NSL dataset. In their survey work, Bhuyan et al. [14] examined many approaches to anomaly

detection, including statistical methods, classification-based methods, clustering and outlier-based methods, soft computing techniques, knowledge-based methods, and combination learners. These methods are utilized for the training and testing dataset or cross-validation to test the intrusion system. However, cross-validation only measures the accuracy of detecting known attacks. Imran et al. [15] utilized Linear Discriminant Analysis (LDA) and Genetic Algorithm to select features. They subsequently employed the Radial Basis Function as the feature classifier. The cross-validation technique trains and tests 20% of the NSL KDD training dataset.

3 NSL KDD Dataset

Multiple datasets are employed to assess the efficacy of IPS systems. Several datasets, including the DARPA datasets, the KDD 99 intrusion data (generated from the DARPA 98 dataset), and the NSL-KDD dataset [16, 17], are publicly accessible to evaluate the IDS system. This paper will use the NSL-KDD dataset to compare the WOA and the HHO performance with the IPS system. The NSL-KDD dataset comprises 40 features, not including the output column. The features are `srv_diff_host_rate`, `src_bytes`, `protocol_type`, `dst_host_error_rate`, `srv_error_rate`, `land`, `dst_host_same_srv_rate`, `wrong_fragment`, `dst_host_same_src_port_rate`, `num_failed_logins`, `logged_in`, `num_compromised`, `hot`, `srv_error_rate`, `root_shell`, `dst_host_srv_diff_host_rate`, `num_root`, `num_shells`, `su_attempted`, `srv_count`, `dst_bytes`, `num_access_files`, `dst_host_diff_srv_rate`, `urgent`, `dst_host_srv_error_rate`, `dst_host_count`, `is_guest_login`, `diff_srv_rate`, `Count`, `num_outbound_cmds`, `service`, `error_rate`, `error_rate`, `is_host_login`, `Flag`, `dst_host_srv_count`, `num_file_creations`, `same_srv_rate`, `dst_host_error_rate`, and `dst_host_srv_error_rate`. Furthermore, the NSL-KDD dataset has 148,518 instances, consisting of attack and benign instances, which are used for training and testing. The NSL-KDD dataset contains instances categorized into four primary forms of attack [17–19]:

- Denial of Service (DoS) attack (53387 instances): preventing legitimate access to the resources, including computer systems, online services, or networks.
- Probe attack (14077 instances): gathering information about the target, including computer systems, online services, or networks.
- Remote to Local (R2L) attack (3880 instances): illegal access to a remote computer system, online services, or network.
- User to Root (U2R) attack: non-root user intended to gain root access to a computer system, online services, or network.

4 HHO and WOA Optimizers

The main goal of this article is to evaluate the of HHO and WOA optimizers in improving the performance of the IDS systems. This section briefs these two optimizers.

4.1 HHO Optimizer

Numerous active and time-varying exploration and exploitation stages are included in the HHO method, a well-known swarm-based optimization technique that does not require gradients. As a result of its adaptable structure, high performance, and high-quality outcomes, HHO has garnered a growing amount of attention from researchers. The primary reasoning for the HHO approach is derived from the cooperative behavior and chasing tactics of Harris' hawks in the wild, which are associated with a technique known as "surprise pounce." The behavior of HHO hunting is depicted in Fig. 1. Regarding algorithmic behavior, one of the most important and useful characteristics of HHO is its ability to escape energy. The escaping energy parameter has dynamic, random, and time-varying characteristics, which have the potential to improve further and harmonize the exploratory and exploitative patterns of HHO. A smooth transition between exploration and exploitation can also be carried out with the assistance of this component, which facilitates HHO [9,20].



Fig. 1. HHO hunting behavior [21]

4.2 WOA Optimizer

The natural hunting activity of humpback whales served as the source of inspiration for constructing the WOA optimizer algorithm—the humpback whale hunts by focusing on groups of small fish located at the water's surface. Figure 2 shows that the whales produce characteristic bubbles along a spiral motion to encircle and capture their prey. By employing encircling the prey, hunting for prey, and spiral bubble-net attacking strategies, the WOA optimizer algorithm can imitate the whales' behavior. In the WOA optimizer algorithm, the exploration phase is represented by searching for prey, while the exploitation phase is represented by the spiral bubble-net attacking strategy [10,22,23].

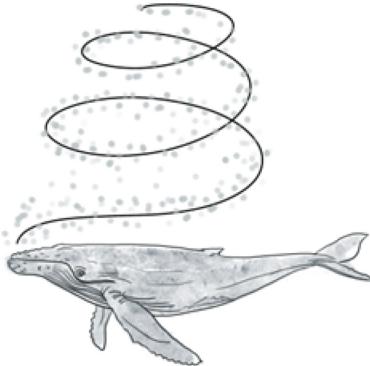


Fig. 2. WOA Hunting behavior [24]

When the WOA algorithm is used in prey encirclement, it considers the optimal solution to be either the target prey or something close to it within the search area. During prey searching, the WOA algorithm investigates the problem space to locate areas that have yet to be explored and to improve the diversity of the population. Finally, the WOA optimizer has successfully modeled the spiral bubble-net attack by employing shrinking encircling and spiral updating position techniques. One way to formulate the shrinking encircling technique is to decrease the value of the convergence variable. On the other hand, the spiral updating position technique is responsible for calculating the distance between the best solution achieved up to this point. It then advances in a spiral pattern from the current location toward an optimal solution, beginning with the current position [10, 22, 23].

5 HHO and WOA Comparative Framework

This section illustrates the designed framework for assessing the WOA and HHO optimizers in enhancing the IPS systems performance in the attacks detection. Figure 3 displays the framework that has been utilized to compare WOA and HHO optimizers.

5.1 Data Preprocessing

The NSLKDD dataset includes texts and numeric data types in its data collection. In addition, the numerical data is distributed throughout a wide range of values. To ensure that the KNN ML classifier functions more effectively, processing the data inside the NSLKDD dataset is necessary. In processing the data inside the NSLKDD dataset, Label Encoder is applied to categorical features, converting them into numerical representations. This step is essential as many machine learning models require numerical input. Simultaneously, the Min-Max Scaler is employed to normalize and scale numerical features, ensuring that all

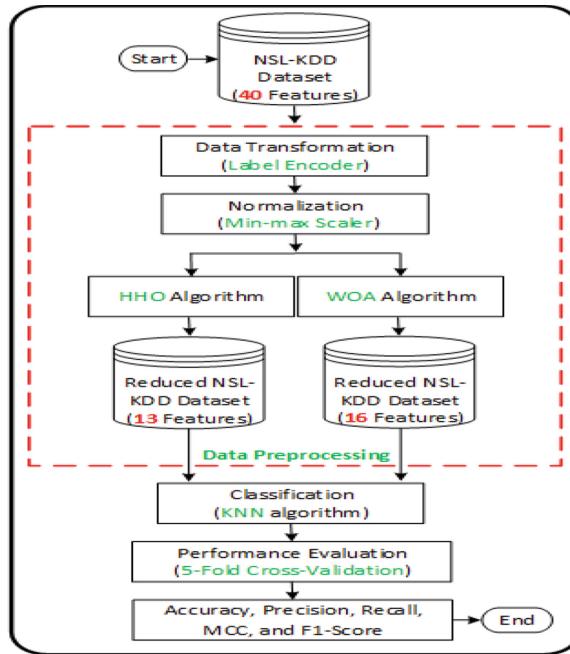


Fig. 3. WOA and HHO optimizers comparative framework

features contribute equally to the model training process. Scaling is particularly important when features have different scales, as it helps prevent certain features from dominating the learning process. Initially, the Label-Encoder mechanism transforms the textual data into numerical data. For example, TCP, UDP, and ICMP are the three values included in the protocol feature of the NSLKDD dataset. These values were transformed into 0, 1, and 2 using the Label Encoding mechanism. Because the article concentrates solely on binary classification, the output column has a zero value for benign instances and one for attack instances. The second step uses the Min-Max scaler mechanism to scale the numeric data between 0 and 1 to eliminate bias toward the large values [16, 17]. The Min-Max scaler mechanism is used after the data has been converted into numbers. A sample of the NSLKDD dataset is presented in Tables 1 and 2, respectively, before and after the preprocessing steps.

5.2 Feature Selection Using WOA and HHO Optimizers

The feature selection method is a crucial stage in data pre-processing in data mining. Its purpose is to identify the subset of relevant features and create a new subset with these features. An ML framework that utilizes the relevant subset of features will yield a superior attack detection rate compared to a framework that incorporates the whole dataset of features. The

Table 1. Sample of the NSLKDD dataset before preprocessing

No	Instances	Output
1	Icmp, eco.i, SF, 8, 0, 0	Benign
2	Tcp, http, SF, 303, 555, 0	Attack
3	Tcp, private, REJ, 0, 0, 0	Benign
4	Udp, domain_u, SF, 45, 45, 0	Attack
5	Udp, private, SF, 105, 147,0	Attack
6	Udp, domain_u, SF, 43, 43, 0	Attack

Table 2. Sample of the NSLKDD dataset after preprocessing

No	Instances	Output
1	0, 0.043478261, 0.9, 5.80E-09, 0, 0	0
2	0.5, 0.028985507, 0.9, 2.20E-07, 4.24E-07, 0	1
3	0.5, 0.072463768, 0.1, 0, 0, 0	0
4	1, 0.086956522, 0.9, 3.26E-08, 3.44E-08, 0	1
5	1, 0.072463768, 0.9, 7.61E-08, 1.12E-07, 0	1
6	1, 0.086956522, 0.9, 3.12E-08, 3.28E-08,0	1

researchers employ algorithms that replicate the innate behavior of the animal in its natural habitat when searching for food. These algorithms choose features based on a workspace search that produces appropriate and optimal subsets of features. This article will utilize the WOA and HHO to improve the efficiency of the IPS system. The WOA and HHO optimizers have been applied to the NSL-KDD dataset to select the most relevant features that give the best performance to the IPS systems. The WOA has selected 16 features of NSL-KDD dataset: src_bytes, service, num_failed_logins, Flag, num_root, num_outbound_cmds, is_host_login, is_guest_login, srv_count, srv_diff_host_rate, serror_rate, num_access_files, dst_host_same_src_port_rate, same_srv_rate, srv_serror_rate, and dst_host_error_rate. On the other hand, the HHO has selected 13 features of NSL-KDD dataset: src_bytes, protocol_type, dst_bytes, dst_host_count, hot, Count, diff_srv_rate, num_access_files, dst_host_srv_count, Flag, urgent, dst_host_diff_srv_rate, and dst_host_same_src_port_rate.

5.3 Using KNN for Attack Classification

The KNN algorithm is a fundamental and indispensable classification approach in machine learning. This algorithm falls under supervised learning and is widely used in pattern recognition, data mining, and intrusion detection. KNN is extensively utilized in cybersecurity due to its non-parametric nature, which means it does not rely on any underlying assumptions about data distribution. KNN utilizes the notion of similarity to forecast the label or value of a new data point

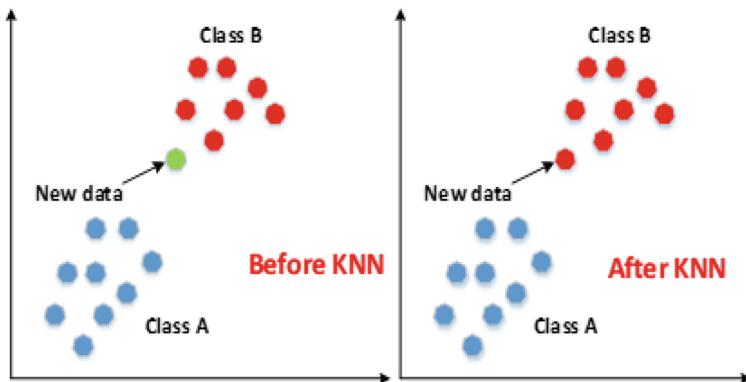


Fig. 4. KNN classifier

by considering its K nearest neighbors in the training dataset. Figure 4 provides a clear explanation of the KNN method. [16, 25].

6 Performance Evaluation

The comparison of WOA and HHO was conducted on a PC with Intel Core i9 processor i9-9900K (8 Core, 16 Threads, 16 MB Cache, 3.60 GHz up to 5.00 GHz), 16 GB RAM, 128 GB SSD, and Ubuntu 21.10 O.S. Python was used to implement WOA, HHO, and RF to obtain to results. Several libraries from Python were used including `mealpy.swarm_based.HHO`, '`mealpy.swarm_based.WOA`', '`sklearn.preprocessing`', '`pandas`', '`train_test_split`', '`KNeighborsClassifier`', and '`numpy`'. The K-Fold Cross-Validation method will be used to divide the NSL-KDD dataset into five groups to validate the results. The utilization of the confusion matrix obtained the results. The confusion matrix comprises four primary elements: TP stands for true positive, TN stands for true negative, FP stands for false positive, and FN stands for false negative. To evaluate the effectiveness of the two algorithms, WOA and HHO, the normal metrics produced from the confusion matrix were utilized. Accuracy (Acc), Precision (Pre), Recall (Rec), Matthews Correlation Coefficients (MCC), and F1-Score are the metrics derived from the confusion matrix [26].

Figures 5, 6, 7, 8, and 9 show the Acc, Pre, Re, MCC, and F1-Score, respectively, of the WOA and HHO optimizers with the KNN classifier. Notably, the WOA and HHO optimizers have attained a remarkable result in enhancing attack discovery by the IPS system. However, the WOA optimizer has achieved a better result than the HHO optimizer with all evaluation metrics. The WOA algorithm has attained Acc, Pre, Re, and F1-Score of 97.42%, while the HHO optimizer has 96.87%. Therefore, the WOA optimizer has attained an enhancement of 0.55% with the four metrics over the HHO optimizer. As for the MCC metric, the WOA optimizer has attained a value of 95.89%, while the HHO optimizer has attained

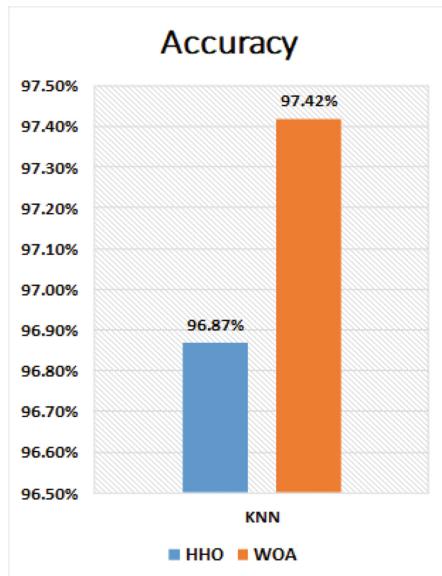


Fig. 5. Acc of the WOA and HHO optimizers

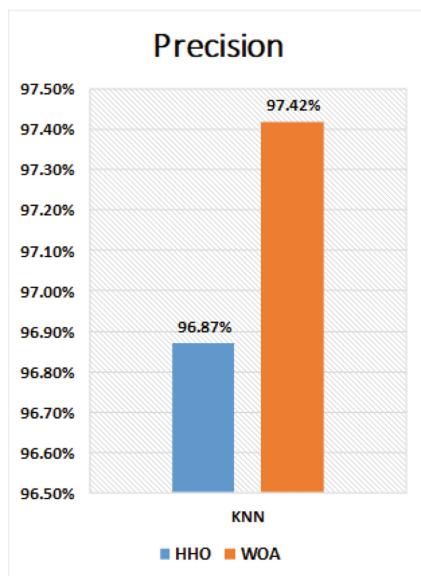


Fig. 6. Precision

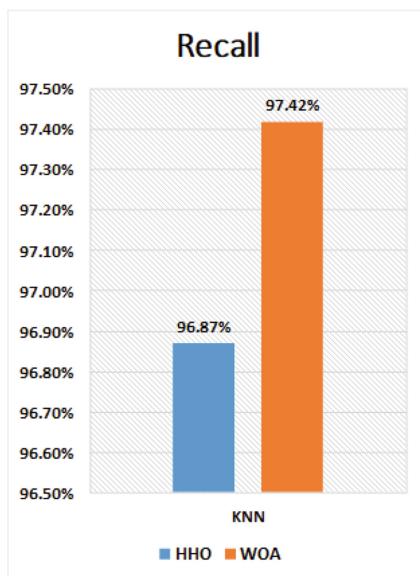
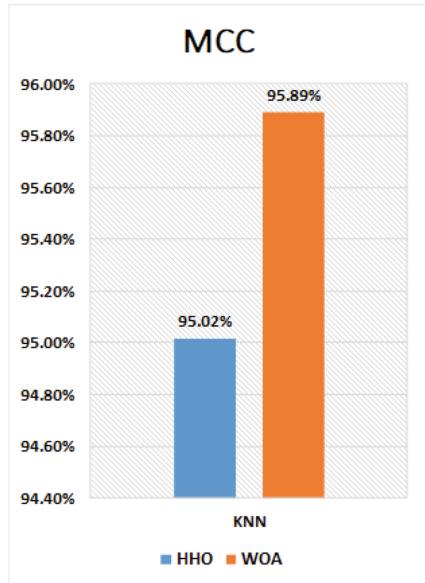
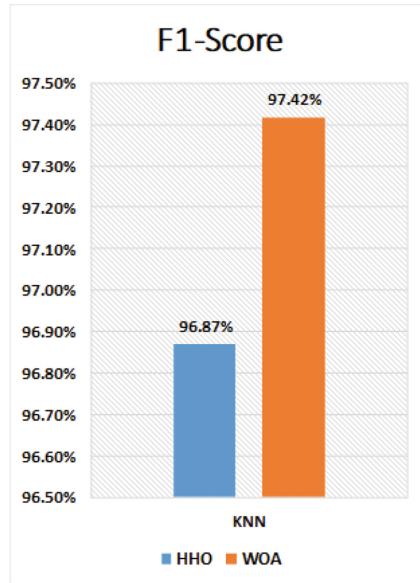


Fig. 7. Recall

**Fig. 8.** MCC**Fig. 9.** F1-Score

a value of 95.02%. Therefore, the WOA optimizer has attained an enhancement of 0.87% with the MCC metric over the HHO optimizer. Consequently, the WOA optimizer is a promising algorithm that can enhance the IPS system's detection of attacks more than the HHO optimizer when using the KNN classifier.

7 Conclusion

This study has explored the field of IPS systems, particularly examining how feature selection optimization might improve their performance. The study compared two advanced optimization algorithms, HHO and WOA, using the NSL-KDD dataset as a benchmark and employing KNN for classification. The implementation was executed using Python, capitalizing on its versatility for effective experimentation. The obtained results emphasize the critical significance of feature selection in enhancing the accuracy of IPS. Both HHO and WOA have proven to help identify significant features, leading to enhanced outcomes in intrusion detection. During the comparison assessment, WOA demonstrated a little advantage over HHO, with an impressive accuracy rate of 97.42% using the KNN algorithm. This result showcases the exceptional ability of WOA to optimize feature selection for intrusion detection. Meanwhile, the KNN algorithm with the HHO attained an impressive accuracy of 96.87%, demonstrating the method's efficacy in this particular situation.

References

- Bang, M., Saraswat, H.: Building an effective and efficient continuous web application security program. In: 2016 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), pp. 1–4 (2016). <https://doi.org/10.1109/CyberSA.2016.7503287>
- Abualhaj, M., et al.: A fine-tuning of decision tree classifier for ransomware detection based on memory data. *Int. J. Data Netw. Sci.* **8**(2), 733–742 (2024)
- Chandre, P.R., Mahalle, P.N., Shinde, G.R.: Machine learning based novel approach for intrusion detection and prevention system: a tool based verification. In: IEEE Global Conference on Wireless Computing and Networking (GCWCN), pp. 135–140 (2018). <https://doi.org/10.1109/GCWCN.2018.8668618>
- Abualhaj, M., Abu-Shareha, A., Shambour, Q., Alsaaidah, A., Al-Khatib, S., Anbar, M.: Customized K-nearest neighbors' algorithm for malware detection. *Int. J. Data Netw. Sci.* **8**(1), 431–438 (2024)
- Zhou, P.-Z., Zhang, H., Liang, W.: Research on hybrid intrusion detection based on improved Harris Hawk optimization algorithm. *Connection Sci.* **35**(1), 1–24 (2023). <https://doi.org/10.1080/09540091.2023.2195595>
- Yedukondalu, G., Bindu, G.H., Pavan, J., Venkatesh, G., SaiTeja, A.: Intrusion detection system framework using machine learning. In: 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, pp. 1224–1230 (2021). <https://doi.org/10.1109/ICIRCA51532.2021.9544717>
- Mendes, H., Quincozes, S.E., Quincozes, V.E.: A web user interface tool for metaheuristics-based feature selection assessment for IDSs (2022). <https://doi.org/10.1109/csnet56116.2022.9955616>
- Kolhar, M., Al-Turjman, F., Alameen, A., Abualhaj, M.M.: A three layered decentralized IoT biometric architecture for city lockdown during COVID-19 outbreak. *IEEE Access* **8**, 163608–163617 (2020)
- Alazab, M., Abu Khurma, R., Castillo, P., Abu-Salih, B., Martín, A., Camacho, D.: An effective networks intrusion detection approach based on hybrid Harris Hawks and multi-layer perceptron. *Egypt. Inform. J.* **25**(1), 1–9 (2024). <https://doi.org/10.1016/j.eij.2023.100423>
- Mirjalili, S., Lewis, A.: The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67 (2016). <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- Bajaj, K., Arora, A.: Improving the intrusion detection using discriminative machine learning approach and improve the time complexity by data mining feature selection methods. *Int. J. Comput. Appl.* **76**(1), 5–11 (2013). ISSN: 0975-8887
- Bhuyan, M., Bhattacharyya, D.K., Kalita, J.K.: Network anomaly detection: methods, systems and tools. *IEEE Commun. Surv. Tutor.* **16**(1), 303–336 (2014)
- Ibrahim, L.M., Basheer, D.T., Mahamod, M.S.: A comparison study for intrusion database (KDD99, NSL-KDD) based on self organization map (SOM) artificial neural network. *J. Eng. Sci. Technol.* **8**(1), 107–119 (2013)
- Bhoria, P., Kanwal Garg, K.: Determining feature set of DOS attacks. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **3**(5), 875–878 (2013)
- Imran, H.M., Abdullah, A.B., Hussain, M., Palaniappan, S., Ahmad, I.: Intrusions detection based on optimum features subset and efficient dataset selection. *Int. J. Eng. Innov. Technol. (IJEIT)* **2**(6), 265–270 (2012)
- Pervez, M.S., Farid, D.M.: Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs. In: International Conference on Software,

- Knowledge, Information Management and Applications, Dhaka, Bangladesh, p. 1 (2014)
- 17. Al-Mimi, H., Hamad, N.A., Abualhaj, M.M., Daoud, M.S., Al-Dahoud, A., Rasmi, M.: An enhanced intrusion detection system for protecting HTTP services from attacks. *Int. J. Adv. Soft Comput. Appl.* **15**(3) (2023)
 - 18. Çavuşoğlu, Ü.: A new hybrid approach for intrusion detection using machine learning methods. *Appl. Intell.* **49**, 2735–2761 (2019). <https://doi.org/10.1007/s10489-018-01408-x>
 - 19. Abualhaj, M.M., Abu-Shareha, A.A., Hiari, M.O., Alrabanah, Y., Al-Zyoud, M., Alsharaiah, M.A.: A paradigm for DoS attack disclosure using machine learning techniques. *Int. J. Adv. Comput. Sci. Appl.* **13**(3) (2022)
 - 20. Al-Mimi, H., Hamad, N.A., Abualhaj, M.M.: A model for the disclosure of probe attacks based on the utilization of machine learning algorithms. In: 2023 10th International Conference on Electrical and Electronics Engineering (ICEEEE), Istanbul, Turkiye, pp. 241–247 (2023). <https://doi.org/10.1109/ICEEEE59925.2023.00051>
 - 21. Gölcük, İ., Ozsoydan, F.B.: Quantum particles-enhanced multiple Harris Hawks swarms for dynamic optimization problems. *Expert Syst. Appl.* **167**, 114202 (2021)
 - 22. Shrivhare, B.D., Singh, M., Gupta, A., Ranjan, S., Pareta, D.N., Sahu, B.M.: Survey paper: whale optimization algorithm and its variant applications (2021). <https://doi.org/10.1109/iciptm52218.2021.9388344>
 - 23. Nadimi-Shahraki, M., Zamani, H., Asghari Varzaneh, Z.: A systematic review of the whale optimization algorithm: theoretical foundation, improvements, and hybridizations. *Arch. Comput. Methods Eng.* **30**, 4113–4159 (2023). <https://doi.org/10.1007/s11831-023-09928-7>
 - 24. Dokeroglu, T., Deniz, A., Kiziloz, H.E.: A comprehensive survey on recent metaheuristics for feature selection. *Neurocomputing* **494**, 269–296 (2022)
 - 25. Pandey, A., Jain, A.: Comparative analysis of KNN algorithm using various normalization techniques. *Int. J. Comput. Netw. Inf. Secur.* **10**(11), 36 (2017)
 - 26. Al-Mimi, H.M., Hamad, N.A., Abualhaj, M.M., Al-Khatib, S.N., Hiari, M.O.: Improved intrusion detection system to alleviate attacks on DNS service. *J. Comput. Sci.* **19**(12), 1549–1560 (2023). <https://doi.org/10.3844/jcssp.2023.1549.1560>



Domestic Solid Waste Prediction with an Enhanced LSTM with SigmoReLU and RAdam Optimizer

Abdulrahman Sharaf Mohammed Fadhel¹(✉), Rozaida Ghazali¹,
Mohd Razali Md Tomari², Yana Mazwin Mohmad Hassim¹,
Abdullahi Abdi Abubakar Hassan¹, and Lokman Hakim Ismail³

¹ Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Johor, Malaysia
abisharf@gmail.com

² Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Johor, Malaysia

³ Faculty of Civil Engineering and Built Environment, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Johor, Malaysia

Abstract. A novel approach is presented to address the prediction challenge in domestic solid waste generation through the application of machine learning techniques. To overcome the limitations inherent in capturing intricate temporal patterns faced by conventional Long Short-Term Memory (LSTM) models designed for time series forecasting, an enhanced variant, termed e-LSTM, is introduced. This model incorporates crucial enhancements to rectify standard LSTM shortcomings. Introducing a hybrid activation function, SigmoRelu, bolsters the model's capacity to grasp complex time series patterns. Furthermore, the RAdam optimizer is employed to optimize the learning process and improve convergence. Dropout layers are seamlessly integrated within the LSTM architecture to counter overfitting, ensuring robust generalization to novel data. A series of comprehensive experiments is conducted to compare the performance of the e-LSTM model against standard LSTM and GRU models, showcasing its noteworthy advancements. Notably, the e-LSTM model demonstrates superior predictive accuracy in forecasting waste generation compared to standard LSTM and GRU models. In essence, the proposed e-LSTM model represents a significant stride in domestic solid waste prediction, effectively mitigating the limitations of traditional LSTM models. The synergistic integration of SigmoRelu activation, RAdam optimization, and dropout mechanisms results in a resilient and accurate predictive framework. Empirical results affirm the model's superiority, establishing it as a valuable tool for waste management applications and decision-making processes.

Keywords: Domestic Solid Waste · Time series forecasting · Hybrid activation function · LSTM · RADAM · Machine learning

1 Introduction

Effective waste management and regulatory decision-making hinge on accurate waste forecasting. However, predicting domestic solid waste encounters challenges due to limited methods and historical data, impeding volume estimates and policy creation. Solid waste prediction play crucial roles in environmental planning, considering factors like population growth, historical trends, and socioeconomic aspects [4]. The current methods for predicting domestic solid waste encounter obstacles stemming from complex patterns, limited data availability, and evolving technologies. These challenges underscore the need to enhance prediction accuracy through advanced models and improved data quality [9]. Although intelligent techniques such as machine learning algorithms, data mining, GIS, and predictive analytics with IoT devices contribute to accurate waste prediction and resource allocation, there is still room for improvement [2]. Recent advancements in deep learning research, particularly methods like Convolution Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), offer promising avenues for revolutionizing waste prediction [4]. In this context, Long Short-Term Memory (LSTM), a type of RNN, holds particular potential. To address common challenges in waste forecasting, this paper introduces an enhanced version of LSTM known as eLSTM. The e-LSTM model incorporates innovative solutions such as dropout layers to combat overfitting, the RAdam optimizer for optimization, and the SigmoReLU activation function to mitigate vanishing gradients. These techniques empower e-LSTM for analyzing time-series data, and experimentation is essential for achieving optimal results [6].

2 Literature Review

2.1 Overview of Malaysian Domestic Solid Waste

An Overview of Scheduled Wastes Management in Malaysia discusses the challenges of solid waste management in the country [4]. Malaysia faces solid waste issues due to urbanization, industrialization, and population growth. The nation has taken steps like 3R principles (Reduce, Reuse, Recycle), waste segregation, and waste management facilities. Still, concerns persist, including infrastructure gaps and non-recyclable waste [4]. Collaboration between government, industries, and the public is crucial for sustainable waste management and a greener future [3]. Malaysia aimed to predict solid waste generation accurately. Researchers used machine learning techniques and historical data to create models considering factors like population growth, urbanization, and policies [13]. Results showed machine learning's effectiveness in predicting waste generation. Models forecast waste quantities with high accuracy, aiding municipal authorities and waste management agencies to allocate resources efficiently [9]. In the Malaysian study, machine learning is used to examine the influence of seasonal variation on municipal solid waste composition. Large datasets from different regions and times of the year are analyzed using advanced algorithms [8]. The goal is to create a predictive model that forecasts waste composition accurately across changing climates. This research aids waste management strategies, guiding informed decisions for optimized waste handling and resource recovery [6].

2.2 Long Short-Term Memory (LSTM)

LSTM, a type of recurrent neural network (RNN) architecture, is well-suited for modeling sequential data and effectively addresses the vanishing gradient problem [2]. Notable components of LSTM include memory cells, forget gate, input gate, output gate, and a well-maintained gradient flow. Widely applicable across various domains such as Natural Language Processing (NLP), speech recognition, and time series forecasting, LSTM networks involve forward propagation through three essential thresholds forget, input and output gate [7]. LSTM stands out in predicting both the generation and composition of solid waste, successfully capturing intricate temporal patterns in waste data. Leveraging memory cells and gating mechanisms, LSTM excels in handling short-term fluctuations and long-term dependencies inherent in time-series waste data [12]. Its predictive capabilities extend to estimating waste generation rates, assessing composition variations, and facilitating tasks like collection scheduling, recycling strategy formulation, and planning treatment facilities. This makes LSTM a valuable tool for fostering sustainable waste management practices [11].

3 The Proposed e-LSTM (Method)

3.1 The Architecture of e-LSTM

In traditional LSTM networks, information processing relies on three critical components within the hidden-layer cell structure the forget, input, and output gates. While effective in many applications, standard LSTM models face challenges, particularly in handling vanishing gradients [1]. To address these shortcomings, the e-LSTM model introduces innovative solutions. One key enhancement is the integration of a hybrid activation function called SigmoReLU, which combines features from both the Sigmoid and ReLU functions Eq. (1). Unlike the standard Sigmoid activation function, which can lead to vanishing gradients, SigmoReLU offers improved performance [10]. Additionally, the e-LSTM architecture includes multiple layers, each utilizing the SigmoReLU activation function. Dropout layers are also incorporated to prevent overfitting, enhancing the model's robustness and ability to generalize to new data refer to Fig. 1. Overall, these enhancements aim to address the limitations of standard LSTM models and improve the performance of e-LSTM in handling complex sequential patterns.

$$\text{SigmoReLU}(\sigma) = \max(0, x) + \text{sigmoid}(x) \quad (1)$$

3.2 RAdam Optimizer and Dropout Mechanism

The e-LSTM model's efficiency and performance are significantly boosted by the implementation of the rectified Adam optimizer (RAdam) and dropout mechanism. RAdam contributes to adaptive learning rate adjustments, enhancing the model's convergence by dynamically adapting the learning rate for each parameter during training based on gradient history [11].

This adaptability proves valuable in handling noisy waste data patterns with irregular trends and variations commonly found in real-world waste generation data. The adaptive

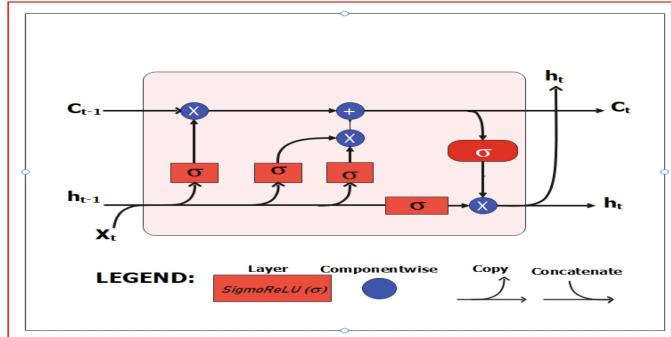


Fig. 1. The e-LSTM Architecture

approach of RAdam is particularly beneficial for time series forecasting, such as domestic solid waste generation, resulting in improved e-LSTM model performance and more precise predictions [5]. The enhanced e-LSTM model architecture comprises an input layer integrating LSTM, activation, and dropout layers, three hidden layers each featuring activation and dropout layers, and an output layer consisting of dense and activation layers. The LSTM and activation layers facilitate the flow of information, ensuring the model's robust generalization. Dropout layers are strategically employed during training to exclude neural units and mitigate overfitting.

3.3 Hybrid Activation Function (SigmoReLU)

The hybrid activation function (SigmoReLU), a fusion of Rectified Linear Unit (ReLU) and Sigmoid functions, was incorporated to enhance the e-LSTM model's predictive performance for domestic solid waste. SigmoReLU combines ReLU for non-negative values and sigmoid for values between (0 and 1) [8]. Introducing non-linearity to neural networks, enriching their capacity to learn complex input-output relationships. SigmoReLU mitigates the vanishing gradient problem through ReLU, stabilizing learning during back-propagation therefore combining ReLU and Sigmoid ensures numerical stability during training, guarding against computational issues arising from extreme values [5]. This hybrid activation bolsters model representational power, activating neurons for specific input features while smoothly transforming inputs into probabilistic outputs [8]. The SigmoReLU hybrid activation boosts the e-LSTM in predicting domestic solid waste and improves the non-linearity, gradient handling, numerical stability and enhancing time series pattern capture. SigmoReLU incorporation further aids domestic solid waste rate prediction.

4 Research Method and Materials

4.1 Summary of the Dataset

The research dataset was gathered from the Labis and Segamat landfill sites in the Johor region of Southern Peninsular Malaysia, covering a period of three years (2020–2023). It consists of daily measurements of domestic solid waste generation, represented as (Net WT). The dataset details are summarized in Table 1.

Table 1. Summary of Datasets

Summary	Segamat Landfill	Labis Landfill
Total Rows	1069	1069
Missing Values	198 rows	169 rows
Mean (Net WT)	328,813.47	309,433.43
Standard Devision	160,203.58	155,566.85
Minimum (Net WT)	9,070	613
Maximum (Net WT)	680,520	773,850
Median (50th %)	347,550	362,410
25th Percentile	193,870	191,400
75th Percentile	467,205	415,952.5

4.2 Model Training and Optimization Strategies

This research employed a train-test holdout validation scheme for conducting experiments. The dataset was divided using a 75–25 train-test split, meaning that it was split into two portions, with 75% of the data allocated for training and 25% for testing. As a result, the model was trained using 825 samples, and then its performance was evaluated on the remaining 270 samples. This approach ensured unbiased assessment on unseen data, emphasizing the model's generalization capabilities. To normalize data and facilitate effective pattern learning, Min-Max Scaling with MinMaxScaler was applied. For time series forecasting of daily domestic solid waste income rates, the TimeSeriesGenerator method was employed, this technique allowed the e-LSTM model to capture temporal dependencies and historical context, improving predictive accuracy by considering historical trends. To prevent overfitting, a dropout layer was introduced between e-LSTM layers. Various dropout values, selected between (0.1 to 0.5) through a trial-and-error approach, were assessed. The dropout layer effectively reduced overfitting, minimizing the gap between training and test errors from (3.94% to 0.0019%). However, optimal dropout values should be carefully selected, as excessively large values may harm model performance and prediction accuracy.

4.3 Performance Evaluation Measures

The study conducted a comprehensive assessment of the e-LSTM model by comparing it with standard LSTM and gated recurrent unit (GRU) models for predicting domestic solid waste generation rates. Various metrics, including mean score error (MSE), mean absolute error (MAE), RMSE, R2, and accuracy, were employed to ensure a fair comparison. Lower MSE, MAE, and RMSE, along with higher R2 and improved accuracy, collectively indicated the superior forecasting capabilities of the e-LSTM model, which holds practical applications in waste management and related fields. Additionally, the evaluation included specific performance measures such as training and testing loss, maximum residual error (Eq. (2)), variance score (Eq. (3)), the R2 score (Eq. (4)), and the model accuracy (Eq. (5)), which provided insights into the model's predictive precision. These evaluation metrics collectively validated the precision and reliability of the proposed e-LSTM predictive model.

$$\text{Max Error} = \max(\text{abs}(test - predictions)) \quad (2)$$

$$\text{Explained Variance Score} = 1 - \frac{\text{Var}(test)}{\text{Var}(test - prediction)} \quad (3)$$

$$R2 = \frac{\sum_i^n (\text{test}[i] - \text{prediction}[i])^2}{\sum_i^n (\text{test}[i] - \text{mean}[\text{test}])^2} \quad (4)$$

$$\text{Accuracy} = [1 - \text{abs}(\text{test} - \text{Prediction}) / (\text{Prediction})] * 100 \quad (5)$$

4.4 Future Prediction with e-LSTM

The e-LSTM model is designed to forecast future domestic solid waste generation by analyzing recent data. It follows a loop mechanism to process input data and generate predictions for the upcoming days. During this process, the input data is reshaped, and the predictions are stored in arrays and lists. Each time step is iterated through, predicting the waste generation for the next day, and these forecasts are stored in a list. Additionally, arrays are used to represent both the original and predicted days. In essence, this model facilitates the forecasting and storage of future waste generation values based on historical data.

5 Results and Discussions

5.1 Comparing e-LSTM with Alternative Machine Learning Models

The e-LSTM model outperformed the standard LSTM and gated recurrent unit (GRU) models in a comprehensive performance evaluation using metrics like (MSE, MAE, RMSE, R2 score and accuracy). The results consistently showed significantly lower MSE and MAE values, indicating superior pattern and trend capturing Fig. 2. The eLSTM model exhibited higher accuracy, making a higher percentage of correct predictions, and lower RMSE values, depicting closer predictions to actual values and precise data variability representation. The higher R2 score for e-LSTM suggests adeptness in capturing data variability and reliable representation of variable relationships Table 2.

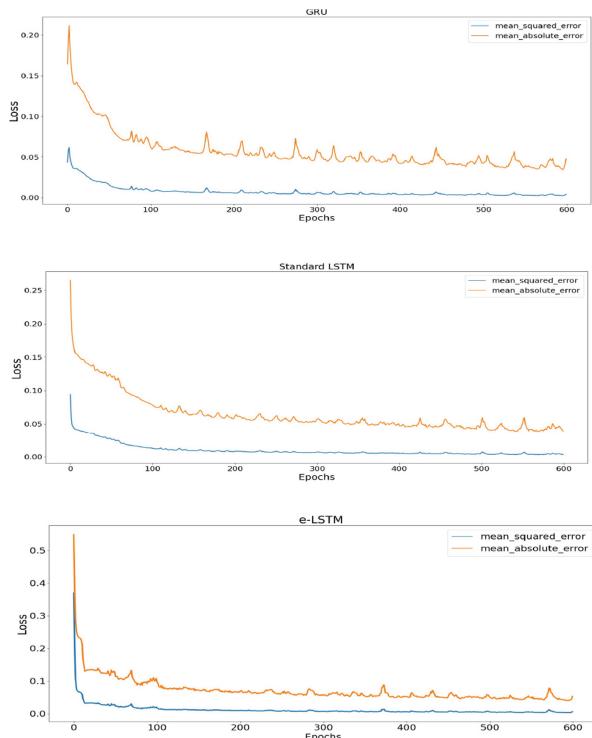


Fig. 2. Comparative Analysis of GRU, LSTM and e-LSTM Models: MSE and MAE Performance Evaluation

Table 2. Summary of comparison models performance

Models	MSE	MAE	RMSE	MAX Error	Variance Score	R2	Accuracy
GRU	0.0187	0.0802	0.089	0.424	0.85	0.83	0.87
LSTM	0.0054	0.0587	0.074	0.263	0.87	0.85	0.88
E-LSTM	0.0028	0.0409	0.065	0.231	0.90	0.92	0.93

5.2 The e-LSTM Model Prediction

The e-LSTM predictions for both the Segamat and Labis landfill sites showcased impressive and precise forecast capabilities. Utilizing the e-LSTM model, the predictions effectively showcased the model's competence in capturing temporal patterns and variations in the daily generation of domestic solid waste. For the Segamat and Labis landfills site, the e-LSTM model exhibited remarkable predictive capabilities by generating forecasts that closely matched the real waste generation data, as depicted in (Figs. 3 and 4). The model's precision in its predictions was evident through its lower Mean Squared Error (MSE) and Mean Absolute Error (MAE) when compared to alternative models. Additionally, the Root Mean Squared Error (RMSE) values were minimized, indicating



Fig. 3. Domestic Solid Waste Generation: Testing vs. Prediction for Segamat Landfill Sites

consistent proximity between the e-LSTM predictions and the observed values. This high degree of accuracy was further supported by the model's impressive R2 score, highlighting a strong correlation between the predicted and actual waste generation.

The e-LSTM model demonstrated precise waste generation forecasts, reflected in lower MSE and MAE values. Reduced RMSE and increased R2 scores indicated accurate predictions. Compared to other models (refer to Table 2), the e-LSTM displayed superior accuracy highlighting its potential in waste management for precise trend. The e-LSTM model consistently aligns predicted values with actual ones, demonstrating strong correspondence. These predictions extend number of months ahead showcasing its accuracy in capturing waste generation trends (refer to Fig. 5).

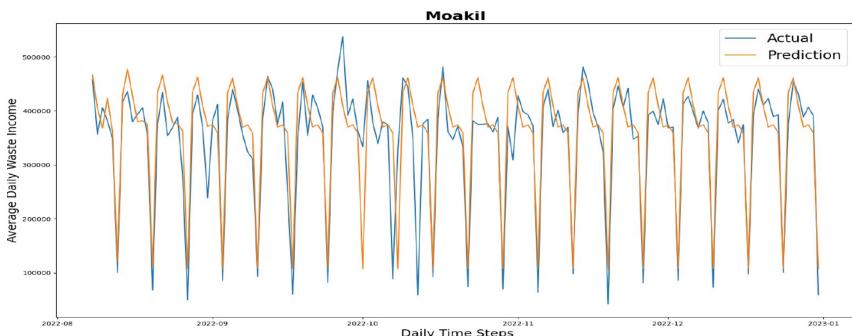


Fig. 4. Domestic Solid Waste Generation: Testing vs. Prediction for Labis forecasting at Segamat and Labis landfill sites

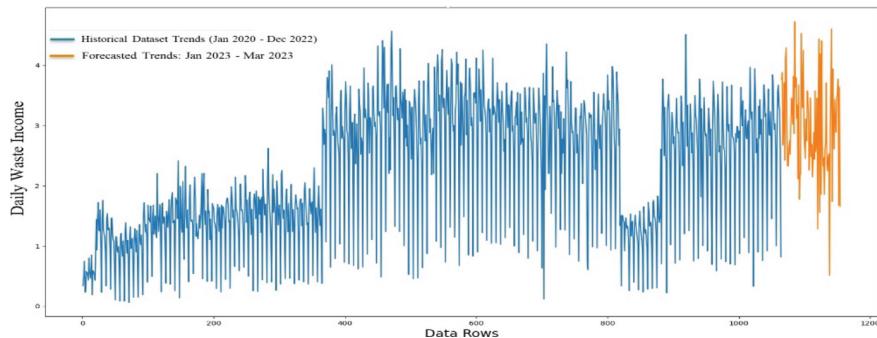


Fig. 5. The e-LSTM Predictions for the next 3 Months ahead (Jan 2023 – Mar 2023) for Segamat Landfill

6 Conclusion

In conclusion, this model is a major step forward in overcoming the shortcomings of conventional LSTMs and improving the accuracy of waste prediction. The integration of innovative features, such as the dropout layers and the SigmoReLU activation function as well as the RAdam optimization, shows remarkable proficiency in predicting the generation of domestic solid waste. The ability to capture complex temporal patterns and solve problems such as overfitting or vanishing gradients is a major development in waste prediction technologies. With its high accuracy and precision, e-LSTM has great potential to optimize waste management strategies as well as to support sustainability initiatives.

Acknowledgments. This research was supported by the Ministry of Higher Education (MOHE) through the Fundamental Research Grant Scheme (FRGS/1/2020/ICT02/UTHM/02/1).

References

1. Radam, D.N.: A Nested LSTM-Based Time Series Prediction Method for Human–Computer Intelligent Systems, Electronics **12**(14) (2023)
2. Siami-Namini, S., Tavakoli, N., Namin, A.S.: The Performance of LSTM and BiLSTM in Forecasting Time Series. In: 2019 IEEE International Conference on Big Data (Big Data), pp. 3285–3292 (2019)
3. Lin, K., Zhao, J.: Kuo: toward smarter management and recovery of municipal solid waste: a critical review on deep learning approaches. J. Clean. Prod. **346**, 130943 (2022)
4. Hamid, S., Isa, C.M., Felix, S.N., Mustaffa, N.K.: Sustainable management using recycle and reuse of construction waste materials in malaysia. ESTEEM Acad. J **16**, 47–58 (2020)
5. Tavakoli, M., Agostinelli, F., Baldi, P.: SPLASH: Learn-able activation functions for improving accuracy and adversarial robustness. Neural Netw. **140**, 1–12 (2021)
6. Adeleke, O., Akinlabi, T., Jen, I., Dunmade: A machine learning approach for investigating the impact of seasonal variation on physical composition of municipal solid waste. J. Relia. Intell. Environ. **9**, 2022–2022

7. Younes, M.K., Nopiah, Z.M., Basri, N.E.A.: Landfill area estimation based on integrated waste disposal options and solid waste forecasting using modified ANFIS model. *Waste Manage.* **55**, 3–11 (2016)
8. Li, Q.: Knowledge structure of technology licensing based on co-keywords network: a review and future directions. *Int. Rev. Econ. Fin.* **66**, 154–165 (2020)
9. Oguz-Ekim, P.: Machine learning approaches for municipal solid waste generation forecasting. *Environm. Eng. Sci.* **38**(6), 489–499 (2021)
10. Hoque, M.M., Rahman, M.T.U.: Landfill area estimation based on solid waste collection prediction using ANN model and final waste disposal options. *J. Clean. Prod.* **256**, 120387 (2020)
11. Cui, K., et al.: Detection of long-term effect in forecasting municipal solid waste using a long short-term memory neural network. *J. Clean. Product.* **290**, 125187–125187 (2021)
12. Al-Jumeily, D., Ghazali, R., Hussain, A.: Predicting physical time series using dynamic ridge polynomial neural networks. *PLOS ONE* **9**(8) (2014)
13. Ghazali, R., Al-Jumeily, D.: Application of Pi-Sigma Neural Networks and Ridge Polynomial Neural Networks to Financial Time Series Prediction. *Artificial Higher Order Neural Networks for Economics and Business* pp. 271–293 (2008). <https://doi.org/10.4018/978-1-59904-897-0.ch012>



Sounds Prediction Instruments Based Using K-Means and Bat Algorithm

Rozlini Mohamed^(✉), Noor Azah Samsuddin, and Munirah Mohd Yusof

Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia
{rozlini, azah, munirah}@uthm.edu.my

Abstract. This study focuses on predicting sound categories in music datasets, employing classification to match predicted sounds with correct instrument categories. Sound classification in music datasets is challenging due to factors such as sampling and digital representation. Discretization at a sufficiently high rate helps preserve the original signal than ensures that classify all relevant information to appropriate group. This study proposes a novel approach that integrates K-Means clustering and the Bat Algorithm for efficient data discretization, aiming to improve sound category prediction accuracy. Unlike existing methods, our approach optimizes the data split points, leading to more informative and well-separated groups for classification. We tested our method with two classifiers, Naive Bayes and K-Nearest Neighbors, and observed significant improvements in classification performance compared to using raw data or other discretization methods. This approach offers a promising way to enhance the accuracy of sound prediction in music applications.

Keywords: Bat Algorithm · Discretization · Prediction · K-Means

1 Introduction

Prediction in data science involves using machine learning models to forecast future outcomes based on historical data patterns [1]. These models analyze and learn from existing data to identify trends and relationships, with regression predicting continuous outcomes and classification assigning categorical labels. The applications of predictive analytics are diverse, spanning finance, sales, weather forecasting, healthcare, fraud detection, and personalized recommendations. The accuracy of predictions is essential for making informed decisions and optimizing strategies across various domains.

The choice of features can significantly impact model performance, posing a challenge for prediction [2]. Including irrelevant or redundant features may result in sub-optimal predictions. Addressing these challenges often requires a combination of careful data pre-processing, feature engineering, and model selection. Regular updates and adaptations to changing conditions help maintain the model accuracy and relevance over time.

In this study, the primary objective is to ensure accurate matching of predicted sounds with the correct category of musical instruments. Classification serves as the

key approach to address this challenge, representing a problem-solving methodology aimed at categorizing or labeling items into predefined groups or classes based on their distinctive features [3]. The classification process involves employing machine learning models to assign input data into predefined categories, typically initiated by collecting a labeled dataset where each example includes features and a corresponding class label. Nevertheless, challenges in classification persist, particularly pertaining to the accuracy of grouping data into the correct label, which significantly influences the overall performance of the classification task.

According to [4], discretization processes conducted during the preprocessing phase can enhance classification performance. Discretization, the conversion of continuous data into discrete form [5], is a crucial step in various data analysis and machine learning tasks, significantly influencing downstream tasks like classification and prediction. There are several methods can be used to perform including K-Means classifier can be employed as a discretization technique in certain contexts [6].

Optimization algorithms are crucial in discretization, helping identify optimal thresholds for transforming continuous data into discrete intervals [7]. The Bat Algorithm, a notable member of swarm-based optimization approaches, refines the discrete representation of data by mimicking bats echolocation behavior in a collaborative process. This iterative optimization contributes to improving the efficiency of the discretization process.

This paper predicts the sound category based on the sounds produced by instruments in a music dataset through discretization. KB method is an integration of K-Means and Bat Algorithm [4] employed as discretization method. K-means is one the simple but powerful discretization method but lack in determine k value. While Bat Algorithm is Bat Algorithm, including its global optimization capabilities, adaptability, balance between exploration and exploitation, robustness makes it a promising approach for discretization tasks. The aim of this study is to evaluate the classification accuracy and recall results. There are two objectives to be achieved: firstly, the classification results before and after the discretization process, and secondly, the classification results with discretization K-Means only and integrating K-Means with Bat algorithm.

This paper is organized as follows: Related works are presented in Sect. 2. Section 3 discusses the methodology based on K-Means and Bat algorithm, along with a description of the dataset. Section 3.4 covers the results and the discussion of these results, while Sect. 4 presents the conclusion.

2 Related Work

2.1 K-Means Classifier as Discretization Method

K-Means is a popular centroid-based clustering algorithm used in machine learning for unsupervised learning. It aims to partition and observations into k clusters, where each observation belongs to the cluster with the nearest centroid. The algorithm (show in Fig. 1) works by iteratively assigning data points to the nearest cluster and updating the centroids until the cluster assignments no longer change. One of the main challenges of K-Means is choosing the appropriate value of k , the number of clusters, and it is known

to be sensitive to the initial placement of the centroids. Despite its limitations, K-Means is fast and simple, making it widely used for clustering data points [5].

The algorithm iteratively divides data points into k clusters by minimizing the variance within each cluster. To determine the best value for k , the elbow method is often used, which involves graphing the inertia (a distance-based metric) and identifying the point where the inertia begins to decrease at a slower rate. This point is a good estimate for the best value of k .

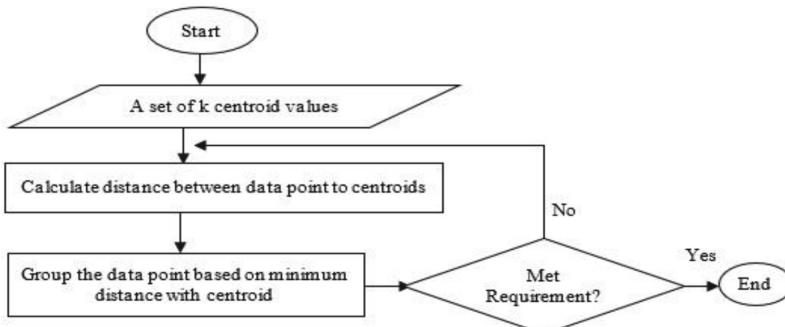


Fig. 1. K-Means Flowchart

K-Means classifier is a popular unsupervised machine learning technique that can also be used as a discretization method. K-Means discretization is a straightforward approach. In the context of discretization, the goal is to partition continuous data into a set of distinct, non-overlapping intervals or bins. K-Means clustering achieves this by grouping similar data points into clusters based on their feature similarity. Each cluster centroid represents a potential bin or interval for discretization.

Discretization with K-Means were employed in [8] in voice conversion. From [9] used K-Means clustering optimized by a genetic algorithm (GA) for discretization. The proposed method performed significantly better than the other discretization techniques. In [10] K-Means able to improve credit risk prediction using logistic regression.

2.2 Bat Algorithm

The Bat algorithm is a metaheuristic algorithm (show in Fig. 2) for global optimization inspired by the echolocation behavior of microbats. It was developed by Xin-She Yang in [11] and has been found to be efficient in dealing with various optimization problems in diverse fields such as power and energy systems, economic load dispatch problems, engineering design, image processing, and medical applications. The algorithm is a population-based metaheuristic algorithm for solving continuous optimization problems and has been used in areas such as cloud computing, feature selection, image processing, and control engineering. The Bat algorithm is a swarm-based algorithm that uses the mechanism bats use to situate their prey, echolocation. It has demonstrated excellent efficiency in solving continuous optimization problems and has been the subject of

various applications and literature reviews [12]. The Bat Algorithm is a population-based metaheuristics algorithm that has been used in various applications, including classification [13, 14].

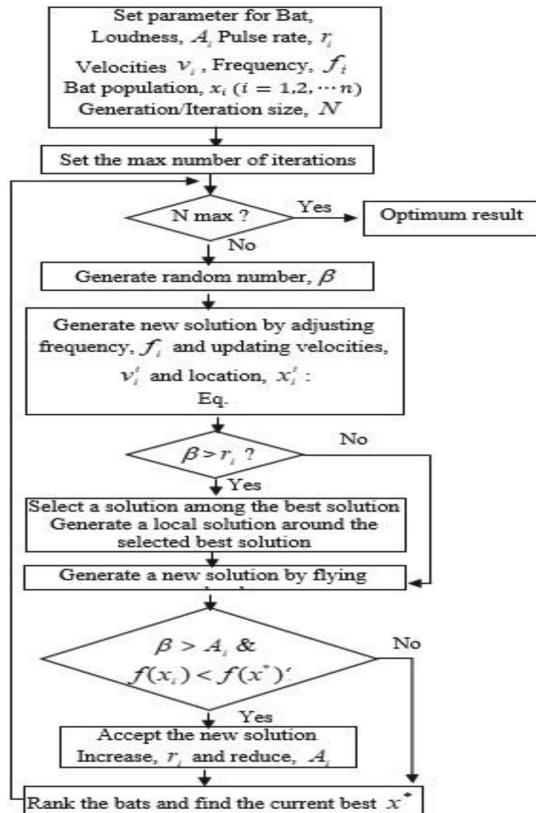
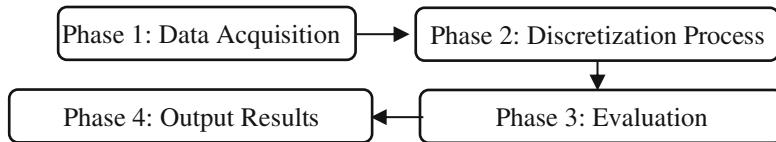


Fig. 2. Bat Algorithm Flowchart

3 Methodology

There are four main phases in research framework: data acquisition, discretization process, evaluation and output result as shown in Fig. 3.

**Fig. 3.** Research Framework

Phase 1: Data Acquisition

Real dataset has been used in this research and obtained from Rick Shriver website for Musical Instruments of Peninsular Malaysia (<https://www.rickshriver.net/hires.htm>) [15]. The music dataset contains continuous data features and corresponding to instrument as class labels for classification purpose. This dataset falls into four main categories. Wind instruments, known as aerophones, include those producing sound through air vibration. Stringed instruments, or chordophones, are either plucked or bowed. Percussion instruments, called idiophones, are either struck or shaken to create sound. The largest group, membranophones, comprises drums, named for the stretched membrane that produces sound when struck (Table 1).

Table 1. Information on Music Dataset.

Category	Number of sounds	Music instruments characteristic	Musical instrument	Number of instances
Idiophones	75	Instruments that are either struck or shaken	Gong, Angklung, Canang, wash boards	279
Membranophones	41	The skin or membrane stretched out on the instrument to produce its sound or drum	Kompang, Beduk, Rebana, tambourine	279
Chordophones	11	Stringed instruments that can either be plucked or bowed	Gambus, Rebab, piano, guitar	279
Aerophones	23	Wind instruments	Bamboo Flute, Seruling, recorder	279

Phase 2: Optimize Discretization

During this phase, the music dataset is converted into discrete features by using integration techniques between K-Means classifier and Bat algorithm, KB method. The KB algorithm shown in Fig. 4. Table 2 illustrates each position, comprising a length of k , where k represents the total number of attributes within the dataset. Instances information is given by $S = \{s_1, s_2, \dots, s_n\}$ where n is the number of solutions. For each solution, $S_i = \{a_{i1}, a_{i2}, \dots, a_{ik}\}$, where k indicate the number of attributes for the S_i solution

in the dataset. The number of groups is equal to number of k . The centroid values are determined by the minimum fitness function values from the bat algorithm. Assuming minimum fitness function come from 1st solution, S_1 thus the attribute, $a_{i1}, a_{i2}, \dots, a_{ik}$, has selected as the centroid. The next step to convert from continuous into discrete. By using K-Means calculate the distance between features or data point with these centroids. The feature will be group based on minimum distance.

- STEP 1** : Set parameter
Objective function $f(x)$, $x=(x_1, \dots, x_d)^T$
Loudness, A , Pulse rate, r_i , Velocities v_i , Frequency, f_i
Bat population, x_i ($i = 1, 2, \dots, n$) from continuous Dataset, ds ,
Generation/Iteration size, N , current iteration/counter, t
- STEP 2** : Generate new solutions by adjusting frequency, f_i and updating velocities, v_i^t and locations/solutions, x_i^t
- STEP 3** : Repeat **STEP 4** until **STEP 12** for $t \leq N$
- STEP 4** : Generate random number, $rand$
- STEP 5** : **if** ($rand > r_i$)
Select a solution among the best solutions
Generate a local solution around the selected best solution
end if
- STEP 6** : Generate a new solution by flying randomly
- STEP 7** : **if** ($rand < A_i$ & $f(x_i) < f(x^*)$)
Accept the new solutions
Increase r_i and reduce A_i
end if
- STEP 8** : Rank the bats and find the current best x^*
- STEP 9** : **DETERMINE a set of k points as centroids**
- STEP 10** : Calculate distance between datapoint to each centroid
- STEP 11** : Group the data point based on minimum distance with centroid
- STEP 12** : Repeat **STEP 11** and **STEP 12** until data point remain in the same cluster
- STEP 13** : Generate the discrete dataset

Fig. 4. Bat algorithm

Table 2. Dataset format

Instances	Attributes					Fitness function
	a_1	a_2	a_3	\dots	a_k	
S_1	a_{11}	a_{12}	a_{13}	a_{14}	a_{1k}	Fitness function S_1
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
S_n	a_{n1}	a_{n2}	a_{n3}	a_{n4}	a_{nk}	Fitness function S_n

Phase 3: Evaluation

After the discretization stage, the next step involves evaluating the effectiveness of the KB methods. Two classifiers, Naïve Bayes and K-nearest neighbor, are employed in the experiment. The classifier, often employed as a benchmark in classification [16], is used to classify music datasets both before and after the discretization process. This is done

to verify the efficiency of the proposed method, and the performance metrics detailing this will be explained in the following section.

Phase 4: Output Result

The effectiveness of the suggested method is assessed through two evolution matrices: accuracy and recall from Naïve Bayes and K-nearest neighbor classifiers. The highest result is 1, while the lowest is 0. Results and Discussion.

The experiment investigates the effectiveness of the KB method for predicting sound categories in a music dataset, involving accuracy and recall analyses using Naïve Bayes and K-nearest neighbor classifiers for individual instruments. These results are then compared against both the original dataset and discretized data from the original K-Means. Figures 5 and 6 illustrate the classification performance for each instrument using Naïve Bayes and K-nearest neighbors in the music dataset. Figure 5(a) and (b) depict accuracy and recall, respectively, using Naïve Bayes. Similarly, Fig. 6(a) and (b) illustrate the accuracy and recall, respectively, employing K-nearest neighbors.

Figure 5 presents the accuracy performance metrics classification using k-Nearest Neighbors (kNN) and Naive Bayes (NB) for dataset without discretization and after discretization, across different categories of musical instruments. Notably, the Naive Bayes method with K-Means clustering (NB-kM) stands out for its comparatively lower accuracy scores in several categories compared to other methods. For instance, in the classification of aerophones, chordophones, and idiophones, NB-kM consistently trails behind other methods, including both kNN and other NB variations (NB-Ori and NB-kB). This suggests a potential limitation of the NB-kM approach in accurately classifying these instrument categories. However, it's essential to note that NB-kM performs relatively better in the membranophones category compared to some other categories, although still not surpassing the accuracy achieved by kNN methods. This analysis underscores the importance of considering the specific characteristics of the dataset and the suitability of different classification algorithms for achieving optimal performance. Further investigation into the underlying reasons for the lower performance of NB-kM compared to other methods could provide valuable insights for refining classification strategies in similar tasks.

Figure 6 illustrates the performance scores of recall classification methods for dataset without discretization and after discretization applied to categorize different types of musical instruments the recall performance metrics using k-Nearest Neighbors (kNN) and Naive Bayes (NB). Notably, the Naive Bayes method with K-Means clustering (NB-kM) exhibits comparatively lower accuracy scores in several categories compared to other methods. For instance, in the classification of aerophones, chordophones, and membranophones, NB-kM consistently trails behind other methods, including both kNN and other NB variations (NB-Ori and NB-kB). This suggests a potential limitation of the NB-kM approach in accurately classifying these instrument categories. However, it's essential to note that NB-kM performs relatively better in the idiophones category compared to some other categories, although still not surpassing the accuracy achieved by kNN methods. This analysis underscores the importance of considering the specific characteristics of the dataset and the suitability of different classification algorithms for achieving optimal performance. Further investigation into the underlying reasons for

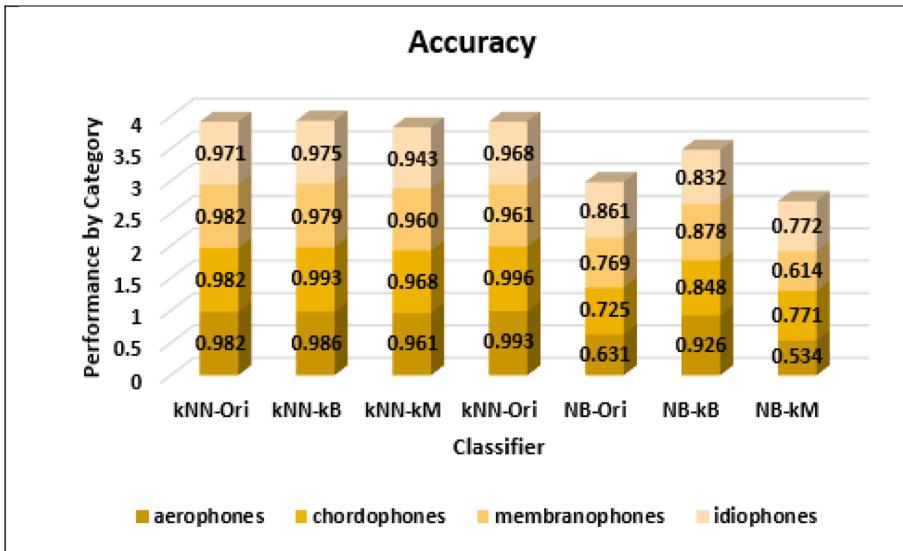


Fig. 5. Accuracy comparison for all method with K-Nearest Neighbor and Naïve Bayes Classifier

the lower performance of NB-kM compared to other methods could provide valuable insights for refining classification strategies in similar tasks.

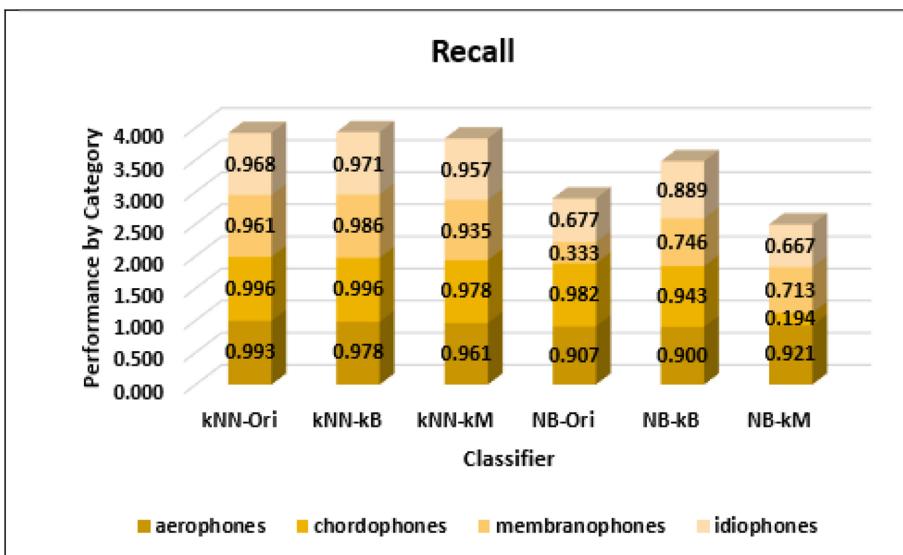


Fig. 6. K-nearest neighbor Classifier Performance

From Fig. 7, show Naive Bayes, both accuracy and recall show improvements when using the KB method, while the K-Means method leads to a decrease in both metrics. This suggests that the KB method enhances the overall performance of NB, making it more accurate in predicting classes, while K-Means has a negative impact. In the case of K-nearest neighbor, both accuracy and recall remain consistently high across all conditions. The KB method does not significantly alter the performance, and the K-Means method maintains high accuracy and recall but with a slight decrease compared to the original data. Comparing the two algorithms, kNN generally exhibits higher accuracy and recall than NB across all conditions. Additionally, the KB method seems to have a more positive impact on NB's performance compared to its impact on kNN.

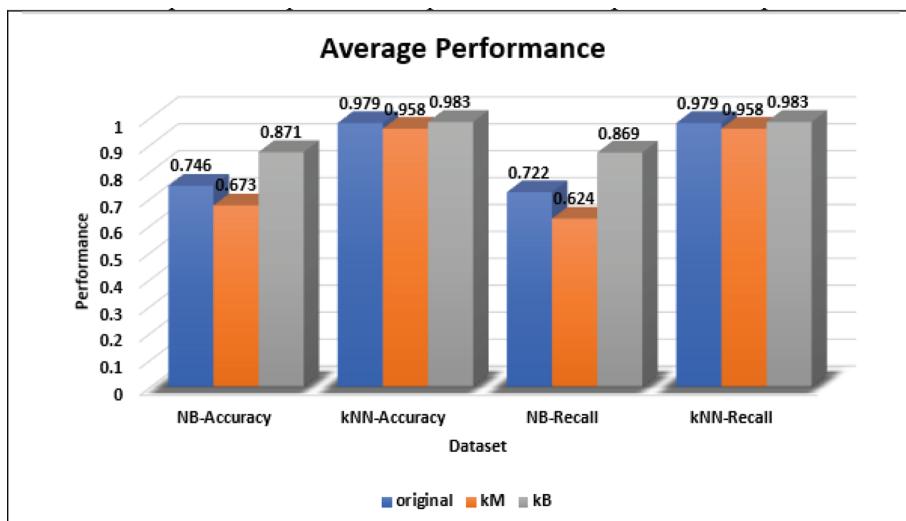


Fig. 7. Average Performance

4 Conclusion

Optimizing the discretization process is crucial in data preprocessing, and the selection of an appropriate algorithm is key for maximizing machine learning model potential. Based on the experiment, the objectives are achieved, the classification results show discretization process able to improve classification performance through the the classification results with discretization K-Means only and integrating K-Means with Bat algorithm. The future research will be focus on the classification problem such as imbalance dataset and feature selection.

References

1. Liu, J., et al.: A clustering-based feature enhancement method for short-term natural gas consumption forecasting. Energy 278(2023). <https://doi.org/10.1016/j.energy.2023.128022>

2. Xie, K., Ge, X., Alvi, H.A.K., Liu, K., Song, J., Yu, Q.: An adaptive segmentation and OTSU-based anomaly classification method for CNV detection using NGS data. *BMC Genomics* **25**, 1–11 (2024). <https://doi.org/10.1186/s12864-024-10018-6>
3. Zhang, X., Yu, L.: Consumer credit risk assessment: a review from the state-of-the-art classification algorithms. *Expert System with Applications* **237**, 121848 (2024). <https://doi.org/10.1016/j.eswa.2023.121484>
4. Mohamed, R., Samsudin, N.A.: A New Discretization Approach of Bat and K-Means. *Int. J. of Adv. Comp. Sc. and App.* **12**, 510–516(2021). <https://doi.org/10.14569/IJACSA.2021.0120159>
5. Peker, N., Kubat, C.: Application of Chi-square discretization algorithms to ensemble classification methods. *Expert Systems with Applications* **185**, 115540 (2021). <https://doi.org/10.1016/j.eswa.2021.115540>
6. Komarasamy, G., Wahi, A.: An optimized K-Means clustering technique using bat algorithm. *Eur. J. Sci. Res.* **84**, 263–273 (2012)
7. Ying, X., Xiaobo, L., Qian, L.A.: discrete teaching–learning based optimization algorithm with local search for rescue task allocation and scheduling. *Applied Soft Computing* **134**, 109980 (2023). <https://doi.org/10.1016/j.asoc.2022.109980>
8. Huang, W.C., Yang, S.W., Hayashi, T., Toda, T.: A Comparative Study of Self-Supervised Speech Representation Based Voice Conversion. *IEEE Journal of Selected Topics in Signal Processing* **16**, 1308–1318 (2022). <https://doi.org/10.1109/JSTSP.2022.3193761>
9. Dwiputrantri, T.H., Setiawan, N.A. and Adji, T.B.: Rough-set-theory-based classification with optimized K-Means discretization. *Technologies* **10**, 51 (2022). <https://doi.org/10.3390/technologies10020051>
10. Fuentes Cabrera, J.G., Pérez Vicente, H.A., Maldonado, S., Velasco, J.: Combination of unsupervised discretization methods for credit risk. *PLoS ONE* **18**, e0289130 (2023). <https://doi.org/10.1371/journal.pone.0289130>
11. Yang, X.S.: A new metaheuristic Bat-inspired Algorithm. *Studies in Computational Intelligence* **284**, 65–74 (2010). https://doi.org/10.1007/978-3-642-12538-6_6
12. Delalić, S., Alihodžić, A., Tuba, M., Selmanović, E., Hasić, D.: Discrete Bat Algorithm for Event Planning optimization. In: 43rd International Convention on Information, Communication and Electronic Technology (MIPRO), pp. 1085–1090. Opatija, Croatia (2020). <https://doi.org/10.23919/MIPRO48935.2020.9245276>
13. Mohammed, H. Ibrahim.: WBA-DNN: A hybrid weight bat algorithm with deep neural network for classification of poisonous and harmful wild plants. *Computers and Electronics in Agriculture* **190**, 106478 (2021). <https://doi.org/10.1016/j.compag.2021.106478>
14. Al-Betar, M.A., Alomari, O.A., Abu-Romman, S.M.: A TRIZ-inspired bat algorithm for gene selection in cancer classification. *Genomics* **112**, 114–126 (2020). <https://doi.org/10.1016/j.ygeno.2019.09.015>
15. Senan, N., Ibrahim, R., Nawi, N.M., Yanto, I.T.R., Herawan, T.: Rough and Soft Set Approaches for Attributes Selection of Traditional Malay Musical Instruments Sounds Classification. *Int. J. of Softw. Sci. Computat. Intell.* **4**, 14–20 (2012). <https://doi.org/10.4018/jssci.2012040102>
16. Basit, M.A., Liu, C., Zhao, E.: SDI: A tool for speech differentiation in user identification. *Expert Systems with Applications* **243**, 122866 (2024). <https://doi.org/10.1016/j.eswa.2023.122866>



A Comparative Study on Ant-Colony Algorithm and Genetic Algorithm for Mobile Robot Planning

Piraviendran a/l Rajendran and Muhaini Othman

Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat Johor, Malaysia
muhaini@uthm.edu.my

Abstract. This paper investigates the optimization of routing for robotics vehicles in automated warehouses in Malaysia. Focusing on routing optimization, the study evaluates Ant-Colony Optimization (ACO) and Genetic Algorithm (GA) in Mobile Robot Planning. Key challenges include efficient routing among task scheduling, and path planning complexities. Objectives include analyzing features of mobile robot planning and representing them in ACO and GA, implementation of ACO and GA algorithms for solving routing problems using dataset, and evaluating their performance. The research anticipates significant contributions to algorithmic solutions, utilizing Python-based experiments aligned with Software Engineering practice, providing practical insights for routing optimization in automated warehouses. Results indicate that ACO outperforms GA in minimizing travel distance, establishing it as the superior routing algorithm for both case studies. Case study 1, the ACO algorithm achieved a best distance of 1036 (u) with execution time 1.67 (s), while the GA algorithm resulted in a best distance 1062 (u) with execution time 0.08 (s). For case study 2, the ACO algorithm achieved a best distance of 1071 (u) with execution time 1.91 (s), while the GA algorithm resulted in a best distance of 1082 (u) with execution time 0.08 (s). Multiple code execution cycles are conducted to provide average findings, ensuring the strength and consistency of the assessment. In conclusion, the study successfully identifies key features in warehouses routing, implements ACO and GA algorithms, and evaluates the performance based on achieved routes and distance.

Keywords: Automated Warehouse · Ant-Colony Optimization (ACO) · Genetic Algorithm (GA) · Routing Optimization · Mobile Robot Planning · Travel Distance · Execution Time

1 Introduction

Over the past five decades, robotics has undergone significant evolution, particularly with the widespread adoption of industrial robots since the 1960s [1]. Robotics has diversified into various classifications, playing a pivotal role in Industry 4.0 across multiple sectors [2–4]. Warehousing has transitioned from manual handling to automation, facilitated by smart automated warehouses integrating IoT, AI, and data analytics [5–8].

Autonomous guided vehicles, crucial in automating material handling processes within warehouses, have seen significant adoption, particularly in Malaysia with the rise of Automated Storage and Retrieval Systems (ASRS) [9–13]. Routing algorithms, notably dynamic ones like ACO and GA, play a vital role in optimizing paths within warehouse networks. [14]. ACO, inspired by ant foraging behavior, calculates optimized paths [16]. While GA simulates natural selection to discover optimal solutions for robotic vehicles, determining parameters like speed and movement angle [1]. In this paper, cranes, conveyors, and an Electrified Monorail System (EMS) are chosen as robotic vehicles for algorithm application which play vital roles in lifting, transporting, and conveying goods [7]. The focus of the research is on optimizing algorithms, specifically ACO and GA, for cranes and conveyors in automated warehouses in Malaysia, addressing the challenge of routing optimization in complex environments [8]. The study identifies key challenges in routing optimization within automated warehouses, focusing on efficient routing amidst task scheduling complexities and intricate path planning. Addressing these challenges necessitates sophisticated algorithms capable of dynamically adapting to changing environments and optimizing routes in real-time. The research aims to explore and implement optimization algorithms, specifically ACO and GA, for mobile robot planning in warehouses. Objectives include investigating the features of ACO and GA, implementing them using relevant datasets, and evaluating their performance to contribute to advancements in automated warehouse logistics.

2 Related Works

The Genetic Algorithm, introduced in 1975, mimics natural genetic processes, utilizing a population-based approach to evolve potential solutions through generations. This method has proven successful in warehouse operation optimizations, demonstrating adaptability to non-continuous goal functions and the ability to handle multiple variables [20–22]. Ant Colony Optimization (ACO), inspired by the foraging behavior of ants, is a metaheuristic algorithm widely used in combinatorial optimization problems. Introduced by Marco Dorigo in the early 1990s, ACO employs virtual ants to incrementally construct solutions based on local information and pheromone trails. The algorithm converges towards optimal or near-optimal solutions through an iterative process, effectively balancing exploration, and exploitation in the solution space [23–26]. The literature further delves into mobile crane operation planning, addressing factors such as crane selection based on site conditions, utilizing fuzzy logic, stochastic artificial neural networks, and building information modeling. While specific discussions on ACO and GA in mobile crane operation planning are limited, these optimization techniques have proven valuable in minimizing operational costs, enhancing performance, and ensuring safety in mobile crane applications [27, 28]. The summarized literature review provides a foundation for the research's focus on optimizing routing paths for robotic vehicles in the context of warehouse automation.

2.1 Comparison Study of Previous Researcher Work on Genetic Algorithms

In this section, a comparison study of previous researchers' work is presented, focusing on Genetic Algorithm (GA) techniques. The Table 1 below provides an overview of

the selected research papers, their evaluation, methods employed, programs used, and conclusions drawn (Table 2).

Table 1. Comparison on Genetic Algorithm

Author Researcher	Technique Evaluation	Method	Program
GA-based Optimization Method for Mobile Crane Repositioning Route Planning	Propose RPOS for mobile crane repositioning using genetic algorithm	Uses DA and GA for optimizing mobile crane relocation	MATLAB
Path Planning for Multiple AGV Systems Using Genetic Algorithm in Warehouse	Introduces modified genetic algorithm with a single chromosome for enhance performance	Enhanced genetic algorithm with a single chromosome and modified crossover for improved performance	It focuses on the algorithms aspects and simulation results

2.2 Comparison Study of Previous Researcher Work on Ant-Colony Optimization

Table 2. Comparison on Ant-Colony Optimization

Author Researcher	Technique Evaluation	Method	Program
Ant Colony Optimization for Real-World Vehicle Routing Problem	Explores VRP variations, delving into ACO basics & its real-world applications in scenarios like time windows for a grocery chain, pickup and delivery for a distribution company, and an online VRP	Metaheuristic ant colony optimization	-
Automated Lift Planning Method for Mobile Cranes	ACO in lift planning: Crane selection, localization, and path planning	Integration of ACO into automated lift planning processes	Use simulation tools for ACO implementations

3 Methodology

3.1 Research Framework

Figure 1 shows the flowchart of research framework represent the structured approach that guides the optimization process for robotic vehicle routing in automated warehouses.

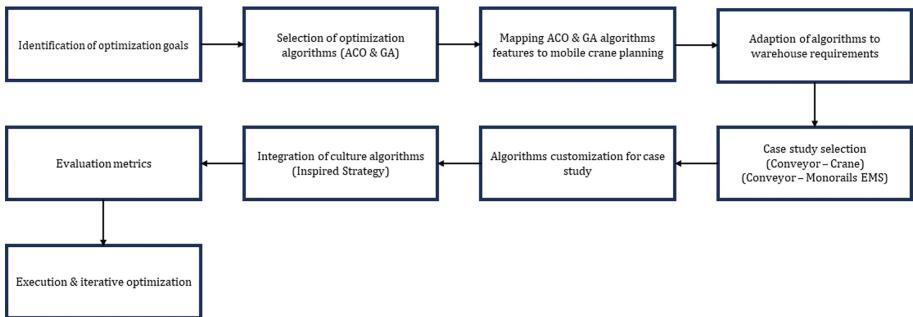


Fig. 1. Research Framework

3.2 Research Activities

A comprehensive overview of the research activities involved in applying the genetic algorithm and ant-colony algorithm to the case studies of robotic vehicle routing optimization in automated warehouses. The steps and strategies employed in each algorithm are outlined below in Fig. 2. The genetic algorithm begins by initializing parameters and creating an initial population of routes. Each route's fitness is evaluated based on optimization objectives like minimizing distance. Selection then picks individuals for the next generation using roulette selection, favoring those with higher fitness. Crossover combines genetic information from two routes to create offspring, while mutation introduces random changes to maintain diversity. The new population is delivered to the belief space, integrating past knowledge. The algorithm terminates when a predefined condition is met, outputting the best route found. Overall, the genetic algorithm efficiently optimizes robotic vehicle routing by evolving routes over successive generations. The Ant-Colony algorithm begins with initialization, where ants are placed at starting locations and initial pheromone levels on route segments are set based on distance factors. Key parameters such as num_ants and num_iterations are defined to influence exploration and convergence. Ants then select the next node using a roulette method based on transfer probabilities derived from pheromone levels and heuristic information. Regular updates to pheromone levels along chosen routes adaptively improve routing choices. An iteration control mechanism ensures convergence, with elite ants guiding others towards optimal solutions. Finally, the algorithm outputs the optimal route based on accumulated pheromone levels, incorporating a fallback approach to handle challenging scenarios.

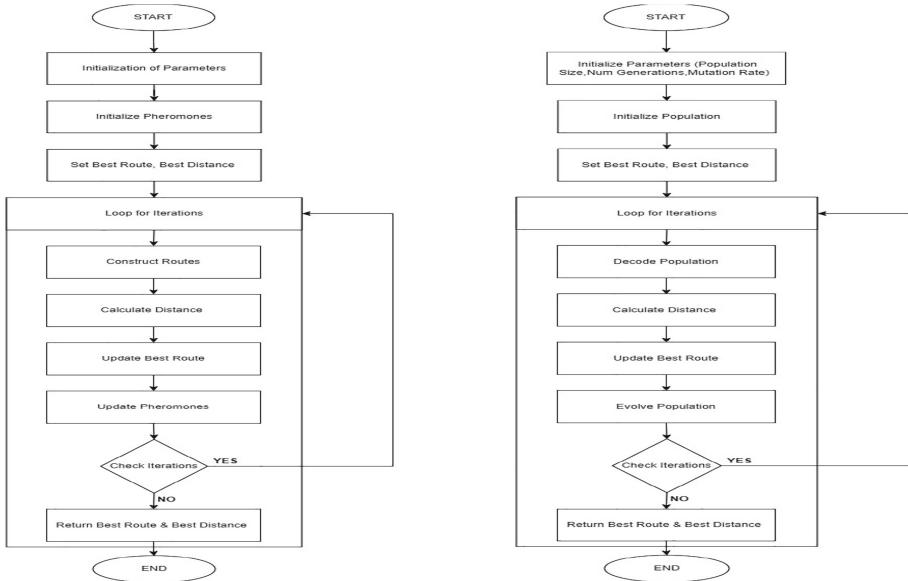


Fig. 2. Steps of ACO and GA

3.3 Algorithm Implementation

Ant Colony Optimization (ACO) and Genetic Algorithm (GA) implemented using the Python programming language in Spyder software to tackle the challenges associated with optimizing robotic vehicle routing in automated warehouses. The ACO algorithm will undergo modifications to integrate constraints and requirements, focusing on enhancing pheromone update rules and developing effective heuristics for route selection. Leveraging Python libraries such as NumPy and SciPy, the implementation will ensure efficient execution and visualization capabilities through Matplotlib. Similarly, the GA algorithm will be customized to handle the complexities of robotic vehicle routing, involving the design of appropriate chromosome representations, selection of genetic operators (crossover, mutation), and exploration of encoding schemes. Utilizing Python libraries like DEAP or PyGMO, the implementation aims to deliver robust solutions for optimizing robotic vehicle routing and contribute to advancements in warehouse automation.

3.4 Dataset

In this study, data collected from automated warehouses to ensure a comprehensive analysis. The datasets collected from the real-world automated warehouse facility played an important role in the study's subsequent phases. The Warehouse Layout Data provided insights into the physical arrangement of conveyors and crane systems, offering details on positions, CRANE numbers, rack numbers, and position. Task Execution Data, logged the movements of items within the warehouse, including parameters like Source and Destination Nodes, Start and End Positions, and timestamps for each movement. Two

case studies further enriched the dataset, focusing on the movement of products from conveyors to cranes and conveyors to monorails (EMS), capturing details such as starting position, ending position, and total time taken. These datasets, encompassing spatial layout and operational dynamics, served as foundational resources for the comparative analysis of routing algorithms.

3.5 Evaluation Metrics

Distance Efficiency

Distance efficiency measures the total distance traveled by the robotics vehicles in the optimized routes, with the objective of minimizing the overall route length. The primary focus is on optimizing the spatial aspects of routing, aiming to reduce the total distance covered. The evaluation metric is defined as the sum of distance traveled by the vehicles to traverse the optimized routes. Let D_{best} represent the total distance for the best route identified by the algorithm. The formula for distance efficiency is shown in (1):

$$D_{best} = \sum_{i=1}^{N-1} \text{distance}(\text{node}_i, \text{node}_{i+1}) \quad (1)$$

where $D_{best} = \sum_{i=1}^{N-1} \text{distance}(\text{node}_i, \text{node}_{i+1})$. Where N signifies the total number of nodes in the route. Distance $(\text{node}_i, \text{node}_{i+1})$ represents the spatial distance between node_i and node_{i+1} . Minimizing indicates more efficient routing and improved distance efficiency in navigating automated warehouse tasks.

Time Efficiency

Time efficiency, as described previously, assesses the time taken by the vehicles to complete their assigned tasks, aiming to minimize task completion time. The combination of these two metrics, distance efficiency and time efficiency, provides a comprehensive analysis of the ACO and GA algorithms' effectiveness in optimizing both spatial and temporal aspects of robotic vehicle routing. The comparison between the algorithms will consider their abilities to minimize D_{best} and the total time taken, contributing valuable information for algorithm selection in warehouse automation scenarios.

4 Results and Discussion

Using ACO and GA on the dataset, independent experiments were conducted for each case study to analyze and compare the outcomes of optimizing robotic vehicle routing within automated warehouses. The data were averaged over several experiment cycles. Conveyor-to-crane and conveyor-to-monorail scenarios were the focus of the tests, which offered detailed insights into algorithmic performance in certain operating environments. Significant variations in route pathways were examined using descriptive statistics, taking nodes and edges into account. The differences between ACO and GA's node prioritization and edge connections were displayed graphically on a graph. The X-Y graph, which showed the node index and processing time, clarified each algorithm's efficiency. Notable differences in Node Index and Edge between ACO and GA were investigated, providing information on how algorithms select routes and priorities positions. Multiple

experiments were conducted for each case study, focusing on conveyor-to-crane and conveyor-to-monorail scenarios. The outcomes were analyzed and compared, considering route pathways, node prioritization, and edge connections. Descriptive statistics and graphical representations, including X-Y graphs, were employed to highlight notable differences in algorithmic efficiency, providing insights into the advantages and flexibility of ACO and GA in robotic vehicle routing optimization.

Table 3. Results for ACO and GA

Algorithm	Case Study One	Case Study Two
ACO	Best Distance: 1032 (u) Execution Time: 1.67 (s)	Best Distance: 1071 (u) Execution Time: 1.91 (s)
GA	Best Distance: 1062 (u) Execution Time: 0.08 (s)	Best Distance: 1082 (u) Execution Time: 0.08 (s)

In Table 3, the ACO algorithm demonstrated superior performance in both Case Study 1 and Case Study 2, achieving best distances of 1032 and 1071 units, respectively, compared to GA's distances of 1062 and 1082 units. This signifies ACO's effectiveness in optimizing travel distances within automated warehouses. However, GA highlighted remarkable computational efficiency, completing optimization in 0.08 s for both case studies, while ACO required 1.67 s for Case Study 1 and 1.91 s for Case Study 2. Despite ACO's longer execution times, its ability to yield shorter distances suggests a trade-off between optimization and computational efficiency.

Distance Comparison

In evaluating the performance of the ACO and GA algorithms, the following distance comparisons were observed. The analysis is based on the average results obtained through multiple runs using dataset variations in Python code, ensuring a comprehensive evaluation. Based on the results collected from the implementation, a comparison between ACO and GA is illustrated in Table 4 for both the case studies in terms of distance.

Table 4. Average Distance Comparison for ACO and GA

Algorithm	Case Study One	Case Study Two
ACO	Mean: 1057 Mode: 1032 Standard Deviation: 16.43	Mean: 1074 Mode: 1071 Standard Deviation: 47.20
GA	Mean: 1073 Mode: 1062 Standard Deviation: 11.84	Mean: 1065 Mode: (1062,1072,1090) Standard Deviation: 36.05

Execution Time

In analyzing the performance of both algorithms across multiple instances in each case

study, the average execution time is calculated based on multiple runs of the experiments. The mean represents the central tendency, indicating the typical time taken, while the standard deviation provides a measure of the variability or dispersion of the execution times. The reported average execution times and standard deviations capture the trends and variations observed across these multiple runs. The average execution times for both algorithms in each case study are as follows in Table 5:

Table 5. Average Execution Time Comparison for ACO and GA

Algorithm	Case Study One (s)	Case Study Two (s)
ACO	Mean: 2.08 Mode: (2.08, 2.19) Standard Deviation: 0.39	Mean: 1.75 Mode: (1.63, 1.70, 1.73) Standard Deviation: 0.06
GA	Mean: 0.11 Mode: 0.11 Standard Deviation: 0.05	Mean: 0.52 Mode: (0.16, 0.26, 0.62, 0.80) Standard Deviation: 0.05

5 Conclusion

In conclusion, the comparison between Ant Colony Optimization (ACO) and Genetic Algorithm (GA) algorithms in terms of distance and execution time favored ACO, demonstrating its superiority in minimizing travel distances for robotic vehicles in automated warehouses. ACO consistently outperformed GA across both case studies. The study concludes that ACO is more suitable for optimizing automated warehouses due to its effectiveness in enhancing efficiency and productivity. ACO's adaptability to dynamic environments and superior performance compared to GA further solidify its suitability for routing optimization. Factors beyond performance metrics, such as adaptability to dynamic environments, scalability, robustness against uncertainties, ease of implementation and maintenance, and compatibility with existing infrastructure, are crucial when selecting the optimal algorithm for routing optimization in automated warehouses. Recommendations for further experimentation include exploring hybrid approaches, evaluating algorithms under varying conditions, and conducting real-world pilot tests or simulations. Ultimately, the findings of the paper are expected to contribute practically by informing decision-making processes for warehouse operators and logistics managers, enabling them to choose the most appropriate algorithm and improve overall warehouse efficiency.

References

1. Vrcan, Ž, Lovrin, N.: Genetic algorithm based optimisation of conveyor belt material cross section area. *Teh. Vjesn.* **17**(2), 137–143 (2010)
2. Michael Christofia, E.T., Pereira, V., Tarba, S., Makrides, A.: Artificial intelligence, robotics, advanced technologies and human resource management: a systematic review. *Int. J. Hum. Resour. Manag.* **33**(6), 1237–1266 (2022)

3. Goel, R., Gupta, P.: A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development (2020)
4. Jorge Ribeiro, S.P., Lima, R., Eckhardt, T.: Robotic Process Automation and Artificial Intelligence in Industry 4.0 – A Literature review. *Procedia Comput. Sci.* **181**, 51–58 (2021)
5. van Geest, M., Tekinerdogan, B., Catal, C.: Smart warehouses: rationale, challenges and solution directions. *Appl. Sci.* **12**(1), 1–16 (2022)
6. Zeinab, S.N.G., Ahmed, E., Saeed, R.A., Mukherjee, A.: Energy optimization in low-power wide area networks by using heuristic techniques. *LPWAN Technol. IoT M2M Appl. Acad. Press*, no. ISBN 9780128188804, pp. 199–223 (2020)
7. Gwak, H.S., Lee, H.C., Choi, B.Y., Mi, Y.: Ga-based optimization method for mobile crane repositioning route planning. *Appl. Sci.* **11**(13) (2021)
8. Bell, J.E., McMullen, P.R.: Ant colony optimization techniques for the vehicle routing problem. *Adv. Eng. Informatics* **18**(1), 41–48 (2004)
9. Sadeghi, M., Nikfar, M., Momeni, F.: Optimizing warehouse operations for environmental sustainability: A simulation study for reducing carbon emissions and maximizing space utilization **4**, 35–44 (2024)
10. Javaid, M., Haleem, A., Singh, R.P., Suman, R.: Substantial capabilities of robotics in enhancing industry 4.0 implementation. *Cogn. Robot.* **1**(June), 58–75 (2021)
11. Matters, L.: HSS Magazine - Swisslog wins major order from IKEA Supply Malaysia (2019)
12. Pengerang Integrated Complex (PIC) Achieves New Milestones: PETRONAS Media Centre. Retrieved from (2019)
13. Reserve, T.M.: Coca-Cola positive on Malaysia's outlook (2022)
14. Talaviya, T., Shah, D., Patel, N., Yagnik, H., Shah, M.: Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides. *Artif. Intell. Agric.* **4**, 58–73 (2020)
15. Wahdan, M.: Motion planning for autonomous mobile robots. *AEJ - Alexandria Eng. J.* **44**(1), 51–57 (2005)
16. Zhang, S., Pu, J., Si, Y., Sun, L.: Path planning for mobile robot using an enhanced ant colony optimization and path geometric optimization. *Int. J. Adv. Robot. Syst.* **18**(3), 1–15 (2021)
17. Dere, M.E.: Optimum path planning for mobile robots, MS Dissertation. Konya Tech. Univ. (2019)
18. Shi, K., et al.: Path planning optimization of intelligent vehicle based on improved genetic and ant colony hybrid algorithm. *Front. Bioeng. Biotechnol.* **10**(July), 1–17 (2022)
19. Zhen, L., Li, H.: A literature review of smart warehouse operations management. *Front. Eng. Manag.* **9**(1), 31–55 (2022)
20. Zhang, H.Y., Lin, W.M., Chen, A.X.: Path planning for the mobile robot: A review. *Symmetry (Basel)* **10**(10) (2018)
21. Krishnan, E.R.K., Wahab, S.N.: A qualitative case study on the adoption of smart warehouse approaches in Malaysia. *E3S Web Conf.*, vol. 136 (2019)
22. Hassanat, A., et al.: Choosing mutation and crossover ratios for genetic algorithms-a review with a new dynamic approach. *Inf.* **10**(12) (2019)
23. Yan, Y., Li, Q., Zhang, C., Wang, L., Liao, J.: Recent advances in ant colony optimization and its application in engineering optimization problems. *Swarm and Evolutionary Computation* **100637**, 54 (2020)
24. Wang, B.L.R., Gao, Y.: A novel ant colony optimization algorithm based on reciprocal strategy and pheromone perturbation. *J. Intell. Manuf.* **29**, 1521–1534 (2018)
25. S, S., Roy, S.: A hybrid ant colony optimization algorithm for multi-objective economic dispatch problem. *Swarm Evol. Comput.* **48**, 43–55 (2019)
26. Rodrigues, V.M.C.F.A., Lopes, L.R.: Ant colony optimization for the minimum spanning tree problem with additional constraints. *Swarm Evol. Comput.* **60**, 100736 (2020)

27. Di Wu, Y.L., Wang, X.: Algorithm of crane selection for heavy lifts. *J. Comput. Civ. Eng.* **25**(1), 57–65 (2011)
28. Sawhney, A., Mund, A.: IntelliCranes: an integrated crane type and model selection system **19**(2), 227–237 (2001). 2010
29. Guo, H., Zhou, Y., Pan, Z., Lin, X.: Automated lift planning methods for mobile cranes. *Autom. Constr.* **132**(July), 103982 (2021)



Enhanced Air Quality Index Prediction Using a Hybrid Convolutional Network

Pei-Chun Lin^{1(✉)}, Nureize Arbaiy², Chen-Yu Yu¹, and Mohd Zaki Mohd Salikon²

¹ Department of Information Engineering and Computer Science, Feng Chia University,
No. 100, Wenhwa Rd, Taichung, Taiwan
peichunpcclin@gmail.com

² Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn
Malaysia, 86400 Batu Pahat, Johor, Malaysia

Abstract. Accurate air quality forecasting is critical for decreasing pollution and protecting public health. A hybrid model combining the Temporal Convolution Network (TCN) and the Graph Convolution Network (GCN) has been developed to predict air pollution with high accuracy and minimise the associated health risks. Because air quality data has two crucial components: temporal trends and spatial linkages, the combination of TCN and GCN is required. The GCN model learns the complicated architecture of each observatory, whereas the TCN model uses past data to detect deviations. The Graph Temporal Convolution Network (GTCN) model was evaluated using six important variables: station names, Air Quality Index (AQI), data timestamps, longitude, and latitude. Our GTCN outperformed other researchers' models on real-world data between February and July 2021. The results demonstrated the lowest Mean Absolute Error (MAE) of approximately 4.78 and the lowest Root Mean Square Error (RMSE) of approximately 6.67. Through precise air quality forecasting, people can pre-know how to protect themselves and prepare outdoor dresses well to reduce exposure to air pollution and related health hazards.

Keywords: Air Quality Index · Graph Convolution Network · Temporal Convolution Network · Uncertainty · Prediction Model

1 Introduction

In recent years, public health, environmental sustainability, and economic growth have all been significantly affected by air quality [1]. Due to rapid industrialization and urbanization in many parts of the world, air pollution has become a serious problem that requires immediate action [2]. Precise air quality forecast extends beyond short-term projections, considering spatial relationships, historical trends, and a range of data sources to guide mitigation activities. Prediction is a valuable tool for taking preventive action, allocating resources properly, and raising awareness. Accordingly, precisely anticipating pollution levels is a critical indicator for air quality regulation [3]. It has the potential to influence decision-making processes ranging from personal behaviour to public policy and sustainable urban design by giving precise, timely information on pollutant concentrations

(4). Governments and public health organisations can utilise precise projections to act immediately to safeguard individuals from high levels of pollution.

Traditional air quality forecasts often utilise statistical models based on mathematical equations to simulate the physical factors that control air pollution [5]. These models often rely on historical data and a restricted set of variables, such as traffic, weather, and industrial pollution. These models may not always fully reflect the complex and dynamic interactions between these variables, but they can still be useful for learning about air quality. When compared to conventional models, machine learning techniques provide a more sophisticated and flexible approach to forecasting air quality [6]. However, they do necessitate the ability to understand and analyse data, as well as a large volume of high-quality data. Artificial Neural Networks (ANN), Support Vector Machines (SVM), Random Forest (RF), and Deep Learning models such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) are among the most utilised machine learning techniques for predicting air quality [7–9]. Other machine learning approaches, such as Decision Trees (DT), Bayesian Networks (BN), and K-Nearest Neighbour (KNN), have also been used to forecast air quality, but less frequently [7, 10].

Graph convolutional networks (GCNs) and temporal convolutional networks (TCNs) have showed great potential in air quality prediction due to their ability to replicate complex spatial and temporal correlations in data. Zhang et al. [10] discovered that by using graph-based representations that capture the relationships between stations, GCNs may successfully depict the spatial dependencies between air quality monitoring stations. TCNs use convolutional layers to capture patterns in time series data, allowing them to accurately represent temporal dependencies in air quality data. It has been established that combining GCNs and TCNs in air quality prediction models improves forecast accuracy and enables the identification of potential pollution sources. A GCN-TCN model was used to forecast Beijing, China's air quality [10, 11]. The method beat earlier machine learning models in terms of prediction accuracy and was able to determine how different pollution sources affected the overall air quality.

This article proposes using the GCN and TCN techniques to forecast future AQI using the latitude and longitude of the observation station, as well as prior AQI readings. This study uses spatial and temporal correlations to forecast AQIs for future time points. Because air pollutants are mobile in the atmosphere and travel to different places depending on meteorological circumstances, there are both temporal and spatial correlations involved. The suggested model employs temporal convolutional networks to learn minor dynamic changes in historical data and graph convolutional networks to learn complicated topology patterns between observation stations. This model can forecast the AQI at future time points, acting as an early warning system for people to take appropriate action and giving the government information on the trend in air quality to help it create pertinent legislation.

This paper is organized as follows: the first section provides the introduction. The second section introduces the related works. The experiment method is described in the third section. The experimental results and discussions are presented in the fourth section. The last section gives the conclusion and future work.

2 Related Works

The AQI is a statistic used in several countries to assess the quality of air [12]. There are also numerous researchers working on AQI furcation using various models [13–17]. In Taiwan, the Environmental Protection Agency incorporated long-standing dual air quality indicators into the AQI, such as the Air Pollution Index (PSI) and Fine Suspended Particulate Index (DAQI). The primary components include ozone (O₃), suspended particles (PM10), small, suspended particulates (PM2.5), nitrogen dioxide (NO₂), sulphur dioxide (SO₂), and carbon monoxide (CO). The sub-indicator values of various pollutants are transformed depending on their impact on human health, and the maximum value of each sub-indicator value for the day is used to generate the AQI value for the observation station.

Many studies have focused on predicting the AQI to preserve human health, such as Lah, et al. [18] used the ARIMA model to estimate the historical AQI value. Furthermore, the relationship between the AQI and nine factors (PM2.5, PM10, SO₂, NO₂, CO, O₃, maximum temperature, minimum temperature, and wind direction) is investigated in Jiao, et al.’s research works [19] before utilizing the LSTM model to predict the AQI using these variables. Meanwhile, a new framework, MTMC-NLSTM, is presented in Jin, et al.’s research works [20]. It employs nested LSTM, which boosts the size of each LSTM unit by one. To forecast multivariate AQI data, MTMC (multi-task multi-channel) is used. The experimental results show that the DL model with DSWT can predict events more accurately and with fewer errors. Lah, et al. [18] and Zhao, et al. [21] provide good prediction results, however, they only analyze the temporal relationship and ignore the geographical correlation.

Cui [22], on the other hand, presented a novel traffic forecasting technique. They used Graph Convolutional Networks (GCN) to learn about the intricate structure of roadways and Gated Recurrent Units (GRU) to grasp past traffic patterns. Based on experimental results and dynamic changes in the data, the suggested approach may provide spatiotemporal correlation in traffic data and can be applied to a variety of correlation-based predictions. He and colleagues proposed a new deep learning framework known as TGC-LSTM. Road interactions in the traffic network were identified using graph convolutional networks and LSTM. The findings of the experiment show that models that incorporate both temporal and spatial correlations perform better in predictions.

Deep learning has evolved substantially in recent years, resulting in much more accurate models. Gopali et al. [24] suggested a temporal convolutional network model (TCN) for detecting time series abnormalities. TCN was used to learn the typical pattern of sequence data, and anomalies were recognized using a specified threshold and the aberrant error’s degree of divergence. Kipf et al. [25] investigated the performance of LSTM and TCN for detecting anomalies in time series data. The results revealed that TCN outperformed LSTM, demonstrating that it can predict data more accurately and quickly utilizing time series.

From the above works, we can see that practically all studies employed LSTM, GCN, GRU, or TCN to combine each model as TGC-LSTM, TCN + LSTM. Nevertheless, those approaches all have limitations, such as the inability to analyze the space-time situation simultaneously. Due to this reason, our study employs both GCN and TCN

model to forecast Taiwan's air quality to consider more impact factors in space-time locations.

3 Experimental Method

The proposed system architectural design is shown in Fig. 1. The process began with data splitting, which divided the historical source data into latitude and longitude and AQI values. The data is then preprocessed. After training, the GCN and TCN models are used to obtain temporal and spatial correlations. Finally, obtain a model capable of forecasting AQI with temporal and geographical correlations.

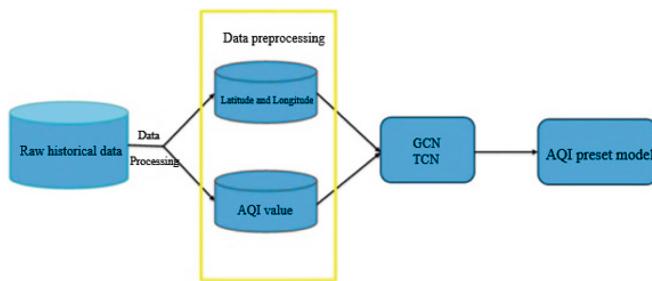


Fig. 1. System Architecture Diagram

The environmental data open platform of the Republic of China's Environmental Protection Agency serves as the primary source of experimental data in this paper. The original data contains a total of 25 observation items, such as meteorological and pollution indicators that have an impact on the AQI. The specific items have “Site Name”, “Country”, “AQI”, “Data Creation Date”, “Longitude”, “Latitude”, “Status”, “SO2”, ..., “PM2.5”, “Wind speed”, and “Site ID”. We chose six items for our experimental data that are “Site Name”, “Country” (the station name), “AQI”, “Data Creation Date”, “Longitude”, and “Latitude”.

Table 1. Data included Empty Data

Area	County	AQI	Time
Daicheng	Changhua	108	2021/3/17 10:00
		111	2021/3/17 11:00
		N	2021/3/17 12:00
		N	2021/3/17 13:00
		103	2021/3/17 14:00

We utilized data from February 2021 to January 2022, as some observation stations were destroyed in 2021. We incorporated the updated information from the revised observation stations, which included the addition of several new stations. The initial phase involved organizing the original data and removing irrelevant information. For further data analysis, we retained the observation station name, AQI value, longitude, and latitude. In Table 1, we present examples of five scenarios. During the AQI processing phase, the program scanned the data and populated any empty AQI value columns with N .

Then, through Eq. (1), Eq. (2) and Eq. (3), the historical AQI value is updated. $X(t)$ represents the data at time point t . If the time point t is N , but the time point $t - 1$ and the time point $t + 1$ are not N , fill in the data at the time point t through Eq. (1).

$$X_{(t)} = \frac{X_{(t-1)} + X_{(t+1)}}{2}$$

$$\text{if } X_{(t-1)}, X_{(t+1)} \neq N \text{ and } X_{(t)} = N \quad (1)$$

If the data at time point t and time point $t + 1$ are N , and the data at time point $t - 1$ and time point $t - 2$ are not N , then fill in the data at time point t through Eq. (2). Since the range of the AQI value is between 0 and 500, doing so can ensure that the data at the time point t will not be negative or out of range due to drastic fluctuations.

$$X_{(t)} = \frac{X_{(t-2)} + X_{(t-1)}}{2}$$

$$\text{if } X_{(t-2)}, X_{(t-1)} \neq N \text{ and } X_{(t)}, X_{(t+1)} = N \quad (2)$$

Similarly, if the data at time point t and time point $t - 1$ are N , and the data at time point $t + 1$ and time point $t + 2$ are not N , fill in the data at time point t through Eq. (3). The final completed data is shown in Table 2. In Tables 1 and 2, The initial value of data N is obtained by adding 108 and 111, dividing the result by 2, and rounding to 110. Equation (2) is used to calculate this. The second data N is then calculated through Eq. (1). The result is rounded to 107 after adding 110 to 103 and dividing the total by 2.

$$X_{(t)} = \frac{X_{(t+1)} + X_{(t+2)}}{2}$$

$$\text{if } X_{(t+1)}, X_{(t+2)} \neq N \text{ and } X_{(t)}, X_{(t-1)} = N \quad (3)$$

The filling formulas in Eqs. 1–3 represent better filling strategies. It can prevent the AQI from quickly rising or falling due to emergencies while also ensuring that the AQI does not drop below zero.

Table 2. Filled empty data to complete information.

Area	County	AQI	Time
Daicheng	Changhua	108	2021/3/17 10:00
		111	2021/3/17 11:00
		110	2021/3/17 12:00
		107	2021/3/17 13:00
		103	2021/3/17 14:00

3.1 The GCN Model

Deep learning has recently gained prominence in a variety of industries. However, their research subjects are frequently limited to Euclidean data, even though many essential data sets exist in the actual world as graphs. Because the distances between observation stations in different locations are not equal, a graph-structured topology network is established, indicating that it is difficult to utilise CNN to process such non-Euclidean data to derive spatial correlation. As a result, we use graph convolutional networks, which can handle data with non-Euclidean structures.

We use an undirected graph \mathbf{G} to represent the relationship between observation stations and observation stations. Use the unweighted graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ to describe the topological structure between the geographical locations of the observation stations in each region. Each observatory represents a node, and is the set of all nodes, which means $= \mathbf{V}\{v_1, v_2, v_3 \dots, v_n\}$. is the set of all observatories and the connected edges between them.

In the past related papers [19] the process of graph convolution uses the Chebyshev polynomial $T_k(x)$ to do the approximation. The Chebyshev polynomial is defined recursively. The definition formula is Eq. (4), and the graph convolution formula approximated by Chebyshev polynomials can be redefined as Eq. (5).

$$T_0(x) = 1, T_1(x) = x, T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x), x \in [-1, 1] \quad (4)$$

$$y = g_{\theta^*}x \approx \sum_{k=0}^k \theta'_k T_k(\tilde{\mathbf{L}})x \quad (5)$$

Next, to reduce the time complexity in the process of model training and learning, we set k to 1, that is, only consider the adjacent one-step nodes each time, so the whole formula can be rewritten as Eq. (6).

$$y = \theta'_0 x + \theta'_1 \tilde{\mathbf{L}}x \quad (6)$$

Among them, $\tilde{\mathbf{L}} = \frac{2L}{\lambda_{\max}} - I$. So, the formula can be rewritten as Eq. (7).

$$y = \theta'_0 x + \theta'_1 \left(\frac{2L}{\lambda_{\max}} - I \right) x \quad (7)$$

L represents the normalized Laplacian matrix, then further set λ max to 2 and define $L = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$ through the normalized Laplacian matrix, rewrite the formula as Eq. (8).

$$y = \theta'_0 x - \theta'_1 \left(D^{-\frac{1}{2}} WD^{-\frac{1}{2}} \right) x \quad (8)$$

Finally, to reduce the parameters of the model, let $\theta_0' = -\theta_1'$, and re-normalize, rewrite the formula as Eq. (9), where $\tilde{W} = W + I$, $\tilde{D}_{ii} = \Sigma_j \tilde{W}_{ij}$ is the adjacency matrix, and D is the degree matrix.

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

In the experiment, the adjacency matrix W is calculated through latitude and longitude. First, we take out the latitude and longitude of 81 observation stations in the original data. The distribution of observation stations' latitude and longitude data (partial data) are shown in Table 3.

Table 3. Observation station latitude and longitude

Area	County	Longitude	Latitude
Erlin	Changhua	120.4097	23.92518
Sanchong	New Taipei	121.4938	25.07261
Sanyi	Miaoli	120.7588	24.38294
Tucheng	New Taipei	121.4519	24.98253

Since the atmosphere on the Earth belongs to a system that changes at any time, there are different degrees of influence between the observation stations. First, we use Eq. (10) to calculate the observation stations (w, x) and the distance between Observation stations (y, z) on the earth. The calculated distance (partial data) is shown in Table 4.

$$dist = 2r * \arcsin \left(\sqrt{\left(\sin \frac{w-y}{2} \right)^2 + \cos w * \cos y * \left(\sin \frac{x-z}{2} \right)^2} \right) \quad (10)$$

Among them, w and y represent latitude, x and z represent longitude, and r represents the radius of the earth. Then we use the distance calculated by Eq. (10) and substitute it with Eq. (11) to calculate the distance between the stations. Weight k , the range of k is between 0 and 1. 0 means the distance between two points is farther, and 1 means the distance between two points is closer. To prevent excessive dependence between points, we set the parameter z at 0.5, , and the final adjacent matrix (partial data) is shown in Table 5.

$$W(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } k < z, k = \exp \left(-\frac{(dist(a, b))^2}{\sigma^2} \right) \\ k & \text{if } k \geq z \end{cases} \quad (11)$$

Table 4. The distance between observation stations

		Erlin	Sanchong	Sanyi	Tucheng
		Changhua	New Taipei	Miaoli	New Taipei
Erlin	Changhua	00.00	168.01	61.89	157.74
Sanchong	New Taipei	168.01	0.00	106.60	10.84
Sanyi	Miaoli	61.89	106.61	00.00	96.59
Tucheng	New Taipei	157.74	10.84	96.59	00.00

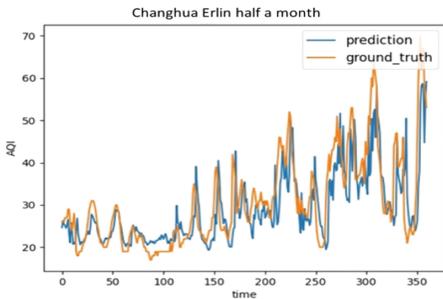
Table 5. The Weight of Distance

		Erlin	Sanchong	Sanyi	Tucheng
		Changhua	New Taipei	Miaoli	New Taipei
Erlin	Changhua	1	0	0.6732	0
Sanchong	New Taipei	0	1	0	0.9879
Sanyi	Miaoli	0.6732	0	1	0
Tucheng	New Taipei	0	0.9879	0	1

3.2 The TCN Model

The Time Convolutional Network (TCN) is derived from the Convolutional Neural Network (CNN) model and is utilized in this paper to process time series data. It comprises key components such as one-dimensional convolutional network, atrous convolution, causal convolution, and residual connection. To ensure consistency between input and output layers' lengths, the TCN employs hole convolution. This technique adapts the sequence length of each hidden layer to match the input sequence length, with the hole convolution employing exponentially increasing values (e.g., 1, 2, 4, 8...) to determine the number of spaces between messages that need to be traversed to reach the next layer.

To prevent the information from being leaked during the training process of the model, the temporal convolutional network adopts causal convolution, so that the output y_t at the time point t will only receive the information x_t at the time point t and the information before the time point t influenced by x_1, x_2, \dots, x_{t-1} . With the increase of the neural network layer, many problems may be encountered when the depth is deeper. This is such as gradient disappearance and gradient explosion, resulting in no way to update the weight, and eventually the accuracy decreases. TCN uses residual poor connections to solve this problem. It makes the entire network architecture achieve good results when the number of layers is very deep.

**Fig. 2.** AQI prediction**Table 6.** The data from February to July.

Method	MAE	RMSE
ARIMA	23.97	29.92
LSTM	5.22	7.33
GCN	4.91	6.92
GRU	4.83	7.16
TGCN	4.86	6.74
TCN	5.24	7.82
GCTN	4.78	6.67

4 Experimental Result and Discussion

The data set is divided into three parts: training, validation, and testing, in the ratio 7:1:2. Then we use the geographical location to find Changhua Erlin in central Taiwan. In the following trials, we made predictions with the Chang-hua Erlin as our goal. In Fig. 2, the x-axis indicates the future time point, indicating that the prediction is based on the first data point in the test set data; the unit is an hour; and the y-axis shows the AQI value. Figure 2 illustrates the real values and forecasted value map, which were generated by dividing the data set by half a year, from February to July of the next year, 2021. The following experiment describes how to use the training and verification sets to predict the test sets AQI.

We compare the model incorporating temporal and space correlation to the ARIMA, LSTM, GCN, GRU, TGCN, and TCN models. In Table 6, statistics from half a year are used to forecast the next half month. We can see that the model's prediction value when paired with TCN and GCN is lower. This suggests that our GCTN model performs better in predicting Changhua Erlin's AQI.

5 Conclusion

In this study, we introduce a novel approach, the GTCN model, for predicting Taiwan's AQI by incorporating both temporal and spatial correlations. Unlike traditional methods that focus solely on temporal correlation, the GTCN model analyzes complex temporal and spatial correlations among multiple observation stations across various counties and cities. From our experiments, the GTCN model demonstrates superior predictive accuracy in both short- and long-term settings compared to existing models. By leveraging real datasets and strict comparisons with other researchers' models, we established the effectiveness of the GTCN model in accurately forecasting AQIs. In particular, the GTCN model achieves the smallest mean absolute error (MAE) of about 4.78 and the smallest root mean square error (RMSE) of about 6.67 when tested on data from February to July 2021, outperforming other models during this period. Our research works emphasize the potential of the GTCN model as a valuable tool for environmental protection

and public health improvement. In future works, we would like to develop predictive applications that continuously analyze AQIs in real-time. These applications will not only provide users with up-to-date AQI statistics but also offer guidance on preventive measures against air pollution at various levels.

References

1. Mujtaba, G., Shahzad, S.J.H.: Air pollution, economic growth and public health: implications for sustainable development in OECD countries. *Environ. Sci. Pollut. Res.* **28**, 12686–12698 (2021)
2. Liang, L., Wang, Z., Li, J.: The effect of urbanization on environmental pollution in rapidly developing urban agglomerations. *J. Clean. Prod.* **237**, 117649 (2019)
3. Janarthanan, R., Partheeban, P., Somasundaram, K., Elamparithi, P.N.: A deep learning approach for prediction of air quality index in a metropolitan city. *Sustain. Cities Soc.* **67**, 102720 (2021)
4. World Health Organization: The world health report 2002: reducing risks, promoting healthy life. World Health Organization (2002)
5. Kao, J., Huang, S.: Forecasts using neural network versus box-jenkins methodology for ambient air quality monitoring data. *Journal of the Air & Waste Management Association* (2000)
6. Ivatt, P., Evans, M.: Improving the prediction of an atmospheric chemistry transport model using gradient boosted regression trees (2019)
7. Khattak, A.M., Khan, Z.: Machine learning-based air quality prediction: a review. *Environ. Sci. Pollut. Res.* **28**(10), 11916–11936 (2021)
8. Cheng, J., Zhu, Y., Wang, X., Zhang, J., Chen, J., Xu, B.: A comprehensive review of machine learning applications in air quality prediction. *Sci. Total Environ.* **717**, 137222 (2020)
9. Zhang, L., Wang, Y.: Prediction of air pollution using machine learning algorithms: a review. *J. Environ. Sci.* **107**, 128–141 (2021)
10. Zhang, X., Zhang, Y., Zhao, Y., Xu, X., Wei, W.: Prediction of air quality with graph convolutional networks. *Int. J. Environ. Res. Public Health* **18**(2), 576 (2021). <https://doi.org/10.3390/ijerph18020576>
11. Li, K., Zhao, J., Zhu, K., Huang, S.: A graph convolutional network-temporal convolutional network model for air quality prediction in Beijing. *China. Environmental Pollution* **277**, 116838 (2021). <https://doi.org/10.1016/j.envpol.2021.116838>
12. Kumari, S., Jain, M.K.: A critical review on air quality index. *Environmental Pollution: Select Proceedings of ICWEES-2016*, pp. 87–102 (2018)
13. Andrašić, P., Radišić, T., Novak, D., Juričić, B.: Subjective air traffic complexity estimation using artificial neural networks. *Promet-Traffic & Transport.* **31**(4), 377–386 (2019)
14. Xie, H., Zhang, M., Ge, J., Dong, X., Chen, H.: Learning air traffic as images: a deep convolutional neural network for airspace operation complexity evaluation. *Complexity* **2021**, 1–16 (2021)
15. Triantafyllou, S.A.: A detailed study on the 8 queens problem based on algorithmic approaches implemented in PASCAL programming language. In: Silhavy, R., Silhavy, P. (eds.) *Software Engineering Research in System Science. CSOC 2023. Lecture Notes in Networks and Systems*, vol 722. Springer, Cham (2023)
16. Triantafyllou, S.A.: Work in progress: educational technology and knowledge tracing models. In: 2022 IEEE World Engineering Education Conference (EDUNINE), pp. 1–4. Santos, Brazil (2022)

17. Triantafyllou, S.A.: Magic Squares in Order $4k+2$. 2022 30th National Conference with International Participation (TELECOM), pp. 1–4. Sofia, Bulgaria (2022)
18. Lah, M.S.C., Arbaiy, N., Lin, P.C.: Forecasting of ARIMA air pollution with improved fuzzy data preparation. In: AIP Conference Proceedings, Vol. 2644, No. 1, p. 040006. AIP Publishing LLC (2022)
19. Jiao, Y., Wang, Z., Zhang, Y.: Prediction of air quality index based on LSTM. In: 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), pp. 17–20. IEEE (2019)
20. Jin, N., Zeng, Y., Yan, K., Ji, Z.: Multivariate air quality forecasting with nested long short-term memory neural network. *IEEE Trans. Industr. Inf.* **17**(12), 8514–8522 (2021)
21. Zhao, L., et al.: T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **21**(9), 3848–3858 (2019)
22. Cui, Z., Henrickson, K., Ke, R., Wang, Y.: Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Trans. Intell. Transp. Syst.* **21**(11), 4883–4894 (2019)
23. He, Y., Zhao, J.: Temporal convolutional networks for anomaly detection in time series. In: *Journal of Physics: Conference Series*, Vol. 1213, No. 4, p. 042050. IOP Publishing (2019)
24. Gopali, S., Abri, F., Siami-Namin, S., Namin, A.S.: A Comparison of TCN and LSTM Models in Detecting Anomalies in Time Series Data. In: 2021 IEEE International Conference on Big Data (Big Data), pp. 2415–2420. IEEE (2021)
25. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2016). arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907)



Filter Method Feature Selection Techniques for Solid Waste Prediction Based on GRU Deep Learning Model

Tuba Batool¹✉, Siti Hajar Arbain¹, Rozaida Ghazali¹, Lokman Hakim Ismail¹, and Irfan Javid²

¹ Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn, Johor, Malaysia

hi210012@student.uthm.edu.my, {sitihajara, rozaida, lokman}@uthm.edu.my

² Department of Computer Science and IT, University of Poonch Rawalakot, Poonch, Pakistan
irfanjavid@upr.edu.pk

Abstract. Municipal solid waste management (MSW) is an imperative aspect towards the development of a sustainable and healthy communities. Hence, it's important to develop a strong system to handle and dispose the MSW. In this regard, the early prediction of waste generation can play an important role in facilitating the municipal authorities for the development of a proper MSW management system and resource allocation for the collection and disposal of the MSW. Many researchers have contributed in early predictions of waste generation. However, most of the studies have chosen random variables for making predictions without considering any feature selection technique which eventually leads to over-fitting or under-fitting issue that reduces the accuracy of the model's prediction. Moreover, some researchers have conducted studies with feature selection techniques to enhance the model predictions, however, they have not focused on the comparison evaluation of different feature selection techniques to present the best technique for the MSW generation forecast. Therefore, this study has been conducted to reveal the importance of feature selection techniques, making the right choices about its selection and its influence on a model's prediction accuracy. This study has compared three most commonly used filter based feature selection techniques i.e. Pearson's correlation coefficient, mutual information, and analysis of variance for MSW generation forecast based on Gated Recurrent Unit model. Three distinct error matrices, namely Root Mean Square Error, Mean Absolute Error, and Regression values have been used to assess the GRU model's performance. Notably, the PCC-GRU approach demonstrated enhanced performance compared to the MI-GRU and ANOVA-GRU approaches, as evidenced by achieving the lowest values for RMSE and MAE, explicitly 0.0848 and 0.0647, respectively.

Keywords: Municipal solid waste · Feature selection · Pearson's correlation coefficient · Mutual information · Analysis of variance · GRU

1 Introduction

Municipal solid waste (MSW) is garbage that is generated as a result of households, organizations, and other nonindustrial activities. According to a Global Market Insights report, the size of the MSW management market was ramped up to a whopping \$117 billion in 2022 [1]. The massive increment in waste production has become a paramount factor in the degradation of the quality of life, therefore, a sound framework for MSW management is required to be developed. In this regard, an accurate prediction of MSW generation can play a vital role in developing an effective system. MSW prediction can't be done directly and depends upon a particular region's social, demographic, and economic factors. However, developing countries have uncertain and insufficient data, therefore, appropriate models are required to be developed for accurate predictions.

Different researchers have used different models, like regression techniques and time series methodology, however according to a study, time series methods performed better to forecast the MSW generation [2]. To enhance the accuracy of time series models a hierarchical neural network known as an artificial neural network (ANN) has been utilized for the prediction of MSW generation [2]. Furthermore, to upgrade the ANN model, researchers come up with many deep learning techniques which include long short-term memory (LSTM) and gated recurrent unit (GRU) models.

The GRU model was introduced back in 2014 by Kyunghyun Cho et al. It consists of two gates, an update gate and a reset gate. GRU model has outstandingly performed in the prediction of MSW production. Moreover, a deep learning model can come up with the optimum results by training the model on the best features. Therefore, there are different feature selection techniques that are responsible for providing the best features for model training and hence play an important part in enhancing the accuracy of the model.

Furthermore, feature selection is one of the methods used for reducing the dimensionality where the most relevant features are chosen, and the irrelevant and redundant features are eliminated. For best feature selection, many researchers have conducted various studies on filter-based feature selection techniques to enhance the accuracy of model's prediction. However, researchers have not focused on the comparison of different filtered method techniques to present the best technique for feature selection in the MSW generation forecast. This study aims to assess and compare the efficacy of prevalent filter methods, and to ascertain the optimal subset of attributes through a comprehensive evaluation of these filter methods based on the GRU model for the prediction of MSW generation.

2 Literature Review

2.1 Deep Learning Models for Waste Generation Prediction

Deep learning models are neural networks with at least three layers and reduced data pre-processing in comparison to machine learning models. Three of the most popular deep learning models in use today in research are Gated Recurrent Unit Networks (GRUs), Long Short-Term Memory Networks (LSTMs), and Recurrent Neural Networks (RNNs). A study has used RNN – LSTM to forecast the MSW disposal rate and the results revealed

that the performance of the model was acceptable with the value of R2 ranging from 0.7 to 0.8 [3]. Another study has been conducted to evaluate the performance of LSTM and GRU models, incorporating them into a gray relational analysis (GRA). The results have revealed that, among the examined configurations, GRA-GRU presented superior performance according to the applied evaluation criteria. [4].

Additionally, a study [5] has been contributed to the prediction of MSW by using multisite LSTM and the results revealed that the multisite LSTM performance was superior. Another study [6] has been conducted in Shanghai for the prediction of MSW generation by using LSTM and a 1-dimensional CNN-LSTM model, the results have exposed the superior performance of 1-dimensional CNN- LSTM model with the R2 value of 0.95. Moreover, a conducted study [7] has predicted the solid waste quantity in Sousse city, Tunisia, and compared bidirectional LSTM (BLSTM) with other models including LSTM, according to the results BLSTM has outperformed with the MSE value of 0.15 and MAE value of 0.21. Another study has been conducted to predict waste generation in urban regions using Regularized Noise-based Gated Recurrent Unit (RNGRU). According to the results, RNGRU outperformed with the MAE value of 0.0147 and MSE value of 0.001 [8]. A summary of the deep learning models used to forecast solid waste is given in Table 1.

Table 1. A summary of deep learning models in solid waste predictions

Model	MAE	MSE	RMSE	R ²	Study
RNN-LSTM	-	-	72–95	0.70–0.86	[3]
GRA-GRU	18.801	-	21.830	-	[4]
Multisite LSTM	0.41	-	0.50	-	[5]
1-D, CNN-LSTM	-	0.006	-	0.95	[6]
BiLSTM	0.21	0.15	-	-	[7]
RNGRU	0.014	0.001	-	-	[8]

2.2 Filter Method Based Feature Selection Techniques in Waste Prediction

Feature selection techniques involve selecting the most pertinent features within a dataset by eliminating redundant and irrelevant features. The application of appropriate feature selection technique is imperative to develop the best feature subset. There are three common feature selection techniques i.e.: filtered, wrapper, and embedded. In this study, the filtered methods have been taken into account. Filtered method techniques select or eliminate a variable based on the relation of that variable with the target variable. Various researchers have employed diverse filtering methods to construct the optimal subset of variables. A study [9] has been conducted for solid waste generation prediction by developing the dataset through the PCC technique and using it with the ANN model and the outcomes have shown the increased accuracy of the model from R2 = 0.916 to R2 = 0.968.

Additionally, the authors conducted research [10] in which Mutual Information (MI) combined with LSTM was compared with standalone LSTM. According to the results, MI-LSTM outperformed LSTM, achieving an R² value of 0.9023, while LSTM alone yielded an R² value of 0.7524. In a comparative study [11] the performance of LSTM cyclic (LSTMC), LSTM, GRU, and GRU cyclic (GRUC) models with Mutual Information (MI) was evaluated. The results indicated that MI with LSTMC outperformed other approaches, achieving a minor RMSE value of 0.7057. Moreover, a study [12] has used ANOVA and response surface methodology (RSM) with ANN model and results revealed that ANOVA with ANN performed better with the value of R² = 0.9982 and RMSE = 0.0084. Table 2 presents a summary of filter method techniques.

Table 2. A summary of filtered techniques in solid waste predictions

Model	MAE	MSE	RMSE	R ²	Study
PCC-ANN	-	-	17.6	0.968	[9]
MI-LSTM	-	-	-	0.9024	[10]
MI-LSTMC	-	-	0.7057	-	[11]
ANOVA-ANN	-	-	0.0084	0.9982	[12]

3 Methodology

In this section, the detailed methodology of the conducted study has been discussed which includes: data collection sources, the application of PCC, MI, and ANOVA to choose the optimum features, the development of the GRU model, and the evaluation matrices used for the analysis of model's performance. The general workflow of this study has been shown in Fig. 1.

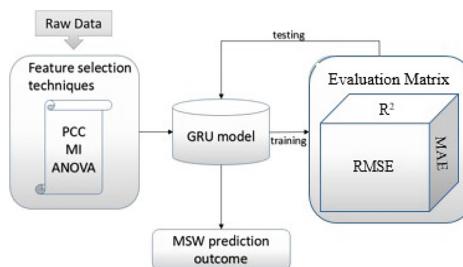


Fig. 1. General workflow of the study

3.1 Data Collection

The data has been collected for 16 different developed countries through the Organization for Economic Co-operation and Development (OECD) via their website link <https://data.oecd.org>. The dataset contains 15 different variables for the past 25 years (1996–2020) which can be categorized as demographic, social, and economic factors. The dataset was partitioned into three subdivisions: training data, comprising 80% of the dataset; testing data, representing 10% of the dataset; and validation data, accounting for the remaining 10% of the dataset.

3.2 Application of Feature Selection Techniques

The performance of any prediction model is intricately linked to the precision of the dataset utilized. The accuracy of a dataset infers the best features that contain most relevant information while minimizing noise and redundancy. Thus, the utilization of a feature selection technique is important to generate an optimal dataset by selecting the most relevant features while eliminating the redundant and irrelevant features. This study has chosen three different, most commonly used filtered-based feature selection techniques to compare their results to analyze the best approach for forecasting MSW generation by using the GRU model. The threshold value used for selecting the optimum features has been set to the value of 0.82.

3.2.1 Parson's Correlation Coefficient

The Pearson correlation coefficient technique is a filtered-based statistical approach to finding out the linear relationship between the input and the output features. The PCC of two variables I and T could be calculated by dividing the covariance of both variables with the product of the standard deviation of both attributes as shown in Eq. (1).

$$r_{IT} = \frac{\text{Covariance}(I, T)}{\sqrt{\text{variance}I} \cdot \sqrt{\text{variance}T}} \quad (1)$$

whereas I and T represent input and target variables respectively. The values of r_{IT} range from -1 to 1.

3.2.2 Mutual Information

Mutual information technique has been widely used with the purpose of computing the arbitrary dependency relation between two features by using the statistical approach. The mutual information value for two distinct variables A and B can be calculated by using the Eq. (2).

$$M(A; B) = E(A) - E(A|B) \quad (2)$$

where $M(A; B)$ signifies the MI for two variables A and B, $E(A)$ denotes the entropy of the variable A, and $E(A|B)$ symbolizes the conditional entropy for A given B. MI has values between 0 to ∞ .

3.2.3 Analysis of Variance

Analysis of variance is one statistical technique for determining the impact of an independent feature on a dependent feature. ANOVA can be calculated using a numeral of statistical representations, including degrees of freedom, mean square between groups (MSB), and mean square of errors (MSE). The formulas for the computation of ANOVA between the variables have been shown in Eqs. (3), (4), and (5).

$$V = MSB/MSE \quad (3)$$

where $MSB = \text{Sum of squares between group (SSB)}/\text{degree of freedom (a - 1)}$ =

$$SSB/a - 1 \quad (4)$$

And $MSE = \text{Sum of squares of error (SSE) / degree of freedom.}$

$$(X - a) = SSE/X - a \quad (5)$$

The values of variance range from 0 to $+\infty$.

3.3 Model Development

The GRU model has been used in this study for the prediction of MSW generation in developed countries. The GRU model has also a gated architecture like LSTM to deal with the learning of long-term dependencies, but, it has no output gate, and therefore it has fewer parameters than an LSTM.

For the current study, 100 epochs have been used to train the GRU model, and every single iteration during the training phase utilizes the random weights. A dropout value of 0.02 has been set in the layers to handle the over-fitting issue. An optimizer known as “Adam” has also been used to enhance the model’s proficiency. The overall model’s performance has been assessed through error matrices.

3.4 Performance Evaluation Criteria

In this study, three different error matrices Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Regression values (R2) have been utilized to determine the performance of the GRU model in the prediction of MSW generation.

Regression analysis (R2) has been considered one of the most prevailing approaches, which is based on a statistical method that calculates the association’s strength that exists between the observed values and the forecasted value. A model’s performance is considered to be high when the value of regression is high. The regression values can be computed by using the equation (6).

$$R2 = 1 - \frac{SSR}{SST} = 1 - \frac{\sum_{i=1}^T (Pi - Qi)^2}{\sum_{i=1}^T (\bar{Pi} - \bar{Qi})^2} \quad (6)$$

where R^2 = regression analysis, SSR = squared sum of residuals, SST = total sum of squares, T = total observations, P_i = Actual value at the i^{th} point and Q_i = forecasted value at the i^{th} point.

Another matrix used in this study, RMSE can be computed by using Eq. (7).

$$RMSE = \sqrt{\frac{\sum_{i=1}^T (Pi - Qi)^2}{T}} \quad (7)$$

Additionally, for detailed analysis, the error matrix MAE has also been utilized in this study which can be calculated by using Eq. (8).

$$MAE = \frac{\sum_{i=1}^T |Pi - Qi|}{T} \quad (8)$$

4 Results

The findings indicate that the Pearson's correlation coefficient feature selection technique outperformed Mutual Information and Analysis of Variance, providing the optimal subset of the data consisting of 5 features, as illustrated in Fig. 2. The values for performance evaluation matrices have revealed that PCC- GRU has outperformed MI-GRU and ANOVA-GRU approaches. The values for error matrices, MAE, and RMSE have been shown in Fig. 3. According to the results, the PCC-GRU technique has the lowest values for MAE being 0.0647 and RMSE being 0.0848. Additionally, the MAE and RMSE values for MI-GRU are 0.0692 and 0.092 respectively, however the number of features selected by MI-GRU is 6. And the MAE and RMSE values are highest for ANOVA-GRU that has provided a subset with 8 features, with MAE being 0.084 and RMSE being 0.1367. Moreover, for a detailed analysis of the model's performance, the regression values have also been evaluated for all three approaches i.e. PCC-GRU, MI-GRU, and ANOVA-GRU. The regression quantitative values help in assessing the model's accuracy of the prediction results. The higher the value of regression higher the accuracy of the model's forecasts. According to the results, the PCC-GRU approach has the most significant values for R. The value of R for the PCC-GRU is 0.939 as illustrated in Fig. 4.

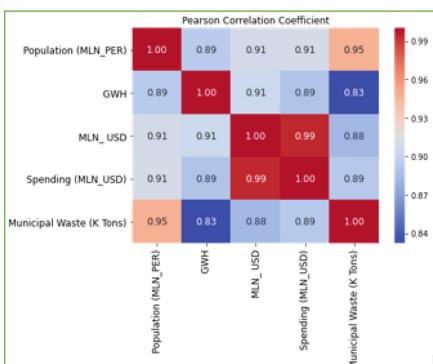


Fig. 2. The data subset through PCC.

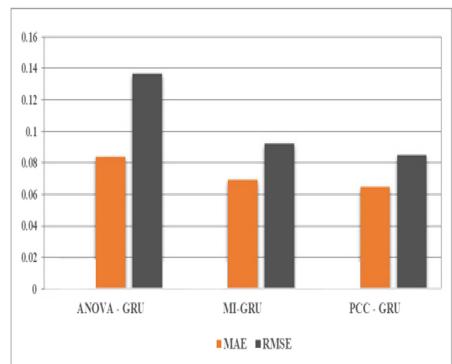


Fig. 3. MAE and RMSE values of PCC, MI, and ANOVA based on GRU model.

Furthermore, the regression values for the MI-GRU model is 0.90369 as shown in Fig. 5. The results clearly depict the acceptable performance of the MI-GRU model. Additionally, the value of regression has also been computed for the ANOVA-GRU approach and the outcome has revealed that the ANOVA- GRU approach has the least prediction accuracy, as the regression values are lowest for all stages and the MAE, and RMSE values are highest. According to the results, the regression value in of the ANOVA-GRU is 0.88 as shown in Fig. 6.

The disparities observed in the results of the evaluation matrices for the three approaches, namely PCC-GRU, MI-GRU, and ANOVA-GRU, unequivocally underscore the critical significance of selecting an appropriate feature selection technique. The optimum feature selection technique will benefit in improving the accuracy results, reducing the training time, and avoiding the over-fitting problems of the model by choosing the most effective features. In this study, the PCC approach has given the optimum subset of the data which eventually leads to enhance the accuracy of the predicted results and reduced training time of the GRU model. The overall difference between the actual and PCC-GRU predicted values has been shown in Fig. 7 where high accuracy can be observed.

Additionally, MI-GRU has followed the PCC- GRU in the accuracy results where the acceptable performance of the MI-GRU approach can be observed. And the least prediction accuracy results could be observed for ANOVA-GRU which reveals that the subset of the data provided by the ANOVA approach has not provided the optimum features for forecasting the waste generation. The conducted stud will help the researchers to understand the importance of using the feature selection techniques and making the right choices for its selection. Moreover, this study has also revealed that PCC feature selection is the optimum choice when dealing with the time series data with linear relationships among features.

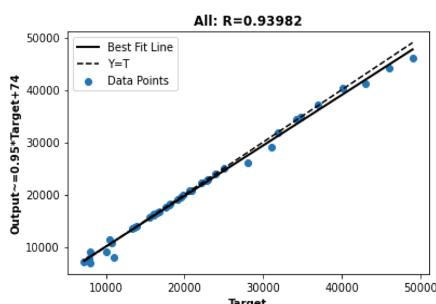


Fig. 4. Regression value of ANOVA-GRU

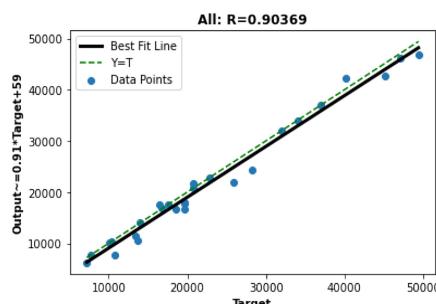


Fig. 5. Regression value of MI-GRU

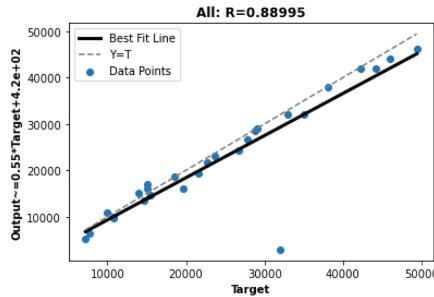


Fig. 6. Regression value of PCC-GRU

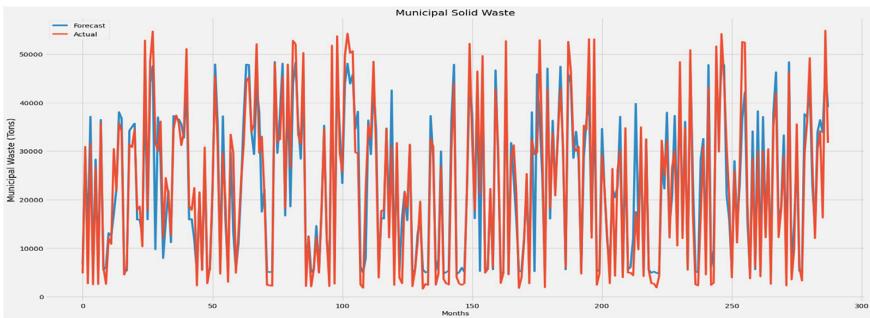


Fig. 7. MSW prediction accuracy of the PCC-GRU approach

5 Conclusion

The literature review of the conducted study has revealed the contribution of different researchers in the prediction of solid waste generation. However, it could be noticed that most of the conducted studies for MSW prediction have not considered feature selection techniques and have randomly chosen the variables which has compromised the accuracy results of a model's prediction. Therefore, this study has contributed in revealing the importance of feature selection techniques, making the right choices about its selection and its influence on the accuracy of a model's prediction.

Moreover, the conducted study has compared commonly used filter methods i.e. PCC, MI, and ANOVA to come up with the most effective filter method technique for the prediction of MSW generation. The input features have been chosen based on the statistical values of PCC, MI, and ANOVA, whereas the target variable is the generated MSW.

Additionally, the GRU model has been developed for the prediction of MSW generation based on the features selected by PCC, MI, and ANOVA. The results have revealed that, for continues dataset like solid waste data, PCC performed better than the other two approaches. Future research in this field could delve into the application of diverse deep learning and machine learning models, considering not just filter techniques but also incorporating wrapper and embedded techniques for feature selection. Exploring additional research avenues, such as evaluating the impact of feature selection on model

interpretability and investigating ensemble methods that amalgamate the strengths of multiple feature selection techniques, could provide valuable insights, such as catering more accurate subset of features for enhancing the model's accuracy.

Acknowledgments. This research was supported by Universiti Tun Hussein Onn Malaysia.

References

1. Municipal Solid Waste Management: <https://www.gminsights.com/industry/analysis/municipal-solid-waste-management-market>. Accessed 01 june 2023
2. BoranWu, D.: Detection of long-term effect in forecasting municipal solid waste using a long short-term memory neural network. *J. Clean. Prod.* (2020). <https://doi.org/10.1016/j.jclepro.2020.125187>
3. Kenneth, K., Adusei, K.T.W.N., Mahmud, T.S., Karimi, N., Lakhan, C.: Exploring the use of astronomical seasons in municipal solid waste disposal rates modeling. *Sustainable Cities and Society* **86**, 104115 (2022). ISSN 2210-6707
4. Liu, B., Zhang, L., Wang, Q.: Demand gap analysis of municipal solid waste landfill in Beijing: Based on the municipal solid waste generation. *Waste Management* **134**, 42–51 (2021)
5. Cubillos, M.: Multi-site household waste generation forecasting using a deep learning approach. *Waste Manage.* **115**, 8–14 (2020)
6. Lin, K., Zhao, Y., Kuo, J.-H.: Deep learning hybrid predictions for the amount of municipal solid waste: A case study in Shanghai. *Chemosphere* **307**, 136119 (2022)
7. Jammeli, H., Ksantini, R., Abdelaziz, F., Masri, H.: Sequential Artificial Intelligence Models to Forecast Urban Solid Waste in the City of Sousse, Tunisia. *IEEE Transactions on Engineering Management*. 1–11 (2021). <https://doi.org/10.1109/TEM.2021.3081609>
8. Rashmi, G.: Regularized noise based GRU model to forecast solid waste generation in the urban region. *Turkish J. Comp. Math. Edu. (TURCOMAT)* **12**(10), 5449–5458 (2021)
9. He, T., Niu, D., Chen, G., Wu, F., Chen, Y.: Exploring key components of municipal solid waste in prediction of moisture content in different functional areas using artificial neural network. *Sustainability* **14**(23), 15544 (2022)
10. Li, D., Li, Z., Sun, K.: Development of a novel soft sensor with long short-term memory network and normalized mutual information feature selection. *Mathematical Problems in Engineering*, 1–11 (2020)
11. Lv, N., et al.: A long short-term memory cyclic model with mutual information for hydrology forecasting: a case study in the xixian basin. *Advances in Water Resources* **141**, 103622 (2020). <https://doi.org/10.1016/j.advwatres.2020.103622>
12. Igwegbe, C.A.: Bio-coagulation-flocculation (BCF) of municipal solid waste leachate using picralima nitida extract: RSM and ANN modelling. *Current Research in Green and Sustainable Chemistry* **4**, 100078 (2021)



Spiking Neural Network for Microseismic Events Detection Using Distributed Acoustic Sensing Data

Mohd Safuan Bin Shahabudin^(✉), Nor Farisha Binti Muhamad Krishnan,
and Farahida Hanim Binti Mausor

Universiti Teknologi PETRONAS, 32610 Seri Iskandar, Perak, Malaysia
mohd_22010924@utp.edu.my

Abstract. Microseismic detection events are critical in monitoring subsurface activities, including hydraulic fracturing, enhanced oil recovery, carbon dioxide, or natural gas geological storage and reservoir characterization, to guarantee safe and efficient energy extraction. This presents significant challenges as it generates large amounts of data. Despite the recent machine learning techniques being increasingly integrated into fiber-optic distributed acoustic sensor (DAS) systems to enhance their intelligent recognition capabilities, there is still a need to solve this. In some research, it has been observed that computational speed is time-consuming and overfitting, thus necessitating a more extensive investigation of this analysis. This study proposes a novel approach using DAS data to enhance microseismic event detection's precision, overfitting issue, and interpretability. This approach utilizes a specifically designed neural network architecture. The deep learning approach is highly effective for the real-time management of the substantial amounts of data recorded by DAS equipment. Three phases of research methodology are proposed. The contribution of this research is that spiking neural network architecture for microseismic detection will bring advancements in microseismic monitoring.

Keywords: Deep Learning · Signal Processing · Microseismic · Event Detection · Spiking Neural Network · Distributed Acoustic Sensing

1 Introduction

Microseismic events, which are natural seismic events, are significant in passive seismology studies. Microseismic events can be characterized by seismic signals of low magnitude that are often invisible to human senses but can be detected using susceptible monitoring equipment, including geophones, hydrophones, and Distributed Acoustic Sensing (DAS) systems. Unlike geophones and hydrophones, DAS utilizes standard optical fibers to create a vast array of sensors capable of capturing acoustic signals along the entire fiber length [1, 2]. With many sensors and a high sampling frequency exceeding 1000 Hz, substantial amounts of DAS data are recorded, resulting in an approximate data size of 650 GB per day to terabytes per day for a two km-long cable sampled

at 2000 Hz with a spatial sampling interval of 1 m [3, 4]. Therefore, advanced signal processing techniques and algorithms are employed to analyze recorded seismic data to identify microseismic events.

Several comparisons have been published for DAS data to evaluate the effectiveness of traditional microseismic event detection methods, such as short-term average/long-term average (STA/LTA) statistical algorithm-based. However, these methods are susceptible to noise and may not detect weak events, which makes them unsuitable for DAS data [5]. These tools also require a certain level of human intervention, such as setting thresholds and performing quality control, which makes them inefficient in handling large volumes of data produced by DAS. However, these techniques that rely on well-known signal-processing methods are often computationally inefficient when dealing with the large data volumes generated by DAS systems.

Machine learning techniques, widely used to improve processing speed and accuracy, have made significant advancements in recent years, demonstrating the potential for rapid analysis of DAS data [5]. Furthermore, developing and researching algorithms for event detection and signal processing using classical machine learning [6] or deep learning [1, 7–11] techniques remains complex despite the extensive literature on this study. Little attention is given to the role of human knowledge and its iterative development throughout the process. Several machine learning techniques have been proposed in the literature to enhance the processing speed of DAS microseismic data, including the Convolutional Neural Network (CNN) and a cutting-edge neural network designed for image object detection [5, 12]. These techniques often demonstrate superior performance compared to traditional methods but require additional effort to reduce computation time using a dataset exhibiting high diversity to encompass most features from field data.

The main aim of this study is to identify areas of deficiency in the application of machine learning techniques to the processing of DAS data in the context of Microseismic event detection. A Spiking Neural Network (SNN) is introduced to achieve better signal identification results for DAS data. The structure of the paper is organized as follows. The literature review is presented in Sect. 2. The methodology utilized in the study is explained in Sect. 3, whereas Sect. 4 provides an in-depth description of the findings and analysis. The paper is concluded in Sect. 5.

2 Literature Review

Exploring machine learning techniques for improving the accuracy and efficiency of microseismic event detection by utilizing DAS data is a crucial focal area in contemporary research. Numerous algorithms have been proposed in machine learning for microseismic event detection. These algorithms include the Fingerprint and Similarity Thresholding (FAST) algorithm, Haar Cascades (HC), Long-Short-Term-Memory (LSTM), CNN-RNN, Bi-Directional LSTM, and U-Net architecture. However, it is worth noting that most of these algorithms have been developed explicitly for traditional geophone-based data acquisition systems [13].

However, using machine learning techniques on DAS data for microseismic event detection has yet to be extensively studied, revealing a significant research gap. Only several machine-learning methods have been developed and published to enhance the

efficiency of DAS microseismic data. Prominent instances include the implementation of the CNN [3, 14–16] and the YOLOv3, regarded as a leading technique for image object detection [4]. Researchers have devoted significant efforts to enhancing its detection and characterization capability. However, the capability still needs improvement, and the development cycle is lengthy. Furthermore, the limited emphasis on neural networks for efficiency compromises the interpretability crucial for instilling confidence in applications.

3 Methods

Machine learning workflow commences with data collection, followed by data cleansing and exploration utilizing the Pandas library in Python. Exploratory data analysis investigates the distribution, relationships, and relative importance of the input features concerning the output features. The Anaconda package integrates Python libraries and an integrated development environment (IDE) for the complete workflow, as depicted in Fig. 1.

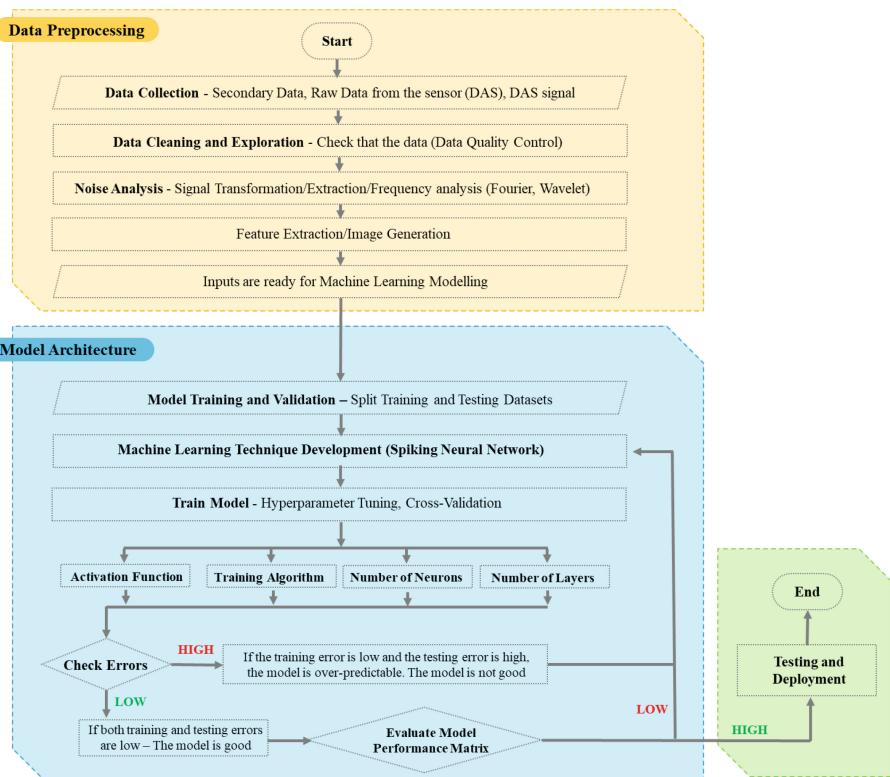


Fig. 1. Workflow of machine learning modeling.

3.1 Data Collection, Cleaning and Exploration

The entire dataset consisted of 40 microseismic events detected and recorded over an 11-day acquisition period, with moment magnitudes ranging from -1.5 to 0.5 and a total size of 13TB. The data represents a passive Vertical Seismic Profiling (VSP) and a continuous Silixa iDAS Carina DAS data for 1280 channels recorded from April- May 2019, collected from Beaver County, Utah, near Milford.

During the data preprocessing phase of the DAS project, the raw data is refined for comprehensive analysis. SEGY data, a standard format for storing seismic data, is managed to ensure consistency and compatibility in the seismic data domain. The seismic datasets are stored in SEG-Y format, which is widely used in industry. SEG-Y encapsulates trace headers and data samples. The analysis workflow uses NumPy for efficient numerical operations and ObsPy for specialized tools in seismological data processing. SEGY and Python-based analysis provide a versatile approach to exploring seismic datasets. Computing signal-to-noise ratio (SNR) is a significant part of this section. SNR calculates the clarity of signals compared to background noise, contributing to data quality assessment.

Numerous signal preprocessing techniques have been examined in the scientific literature, including widely used methods such as Fast Fourier Transform (FFT), Wavelet Packet Transform (WPT), Discrete Fourier Transform (DFT), Continuous Wavelet Transform (CWT), etc. These techniques operate in either the time or frequency domain, resulting in enhanced interpretability of signals [17]. To calculate the SNR, the one-dimensional n-point DFT is computed using the standard and efficient FFT algorithm, and the result is stored in matrix A. The Power Spectral Density (PSD) is subsequently determined using the following mathematical expression:

$$PSD = \frac{2 \cdot |A^t|^2}{N_t^2} \cdot \Delta t \cdot N_t \quad (1)$$

$|A^t|$ is the absolute value of the transpose of matrix A. Δt is the time step between samples. N_t is the number of time samples. The Root Mean Square (RMS) value is subsequently determined over a designated frequency range using the Power Spectral Density function of the traces.

$$RMS_{signal} = \sqrt{\sum_{f=h_s^{start}}^{h_s^{end}} psdf \cdot \Delta f} \quad (2)$$

$psdf$ is the PSD value at frequency f . s represents either signal or noise, e.g., RMS_{signal} is the RMS value for signal. h_s^{start} is the starting frequency range for s , e.g., if s represents a signal, then h_s^{start} is the starting frequency for the signal. h_s^{end} is the ending frequency range for s . SNR is computed using the following formula:

$$SNR = 20 \times \log_{10} \frac{RMS_{signal}}{RMS_{noise}} dB \quad (3)$$

RMS noise levels are evaluated in different frequency ranges. The remaining frequencies are used to calculate RMS_{Signal} . SNR analysis is done at randomly selected time intervals.

3.2 Model Training and Validation

The dataset was split into a 7:3 ratio for training and validation. The test dataset was only used for predictions after training. Seismic waveforms are transformed into 256×256 -pixel images before being integrated into the network. Input to the network is organized into batches of size 32. This choice was based on tests that showed no noticeable improvements in performance with varying batch sizes. A batch size of 32 balances efficiency and effectiveness. Two measures have been implemented to counter overfitting. Firstly, the performance of the model's validation dataset is assessed following every epoch, and the model weights are saved only in the event of discernible enhancement. Secondly, A validation dataset, comprising a randomly selected 30% subset of the complete dataset, is utilized to assess the network's ability to generalize after each training epoch [18].

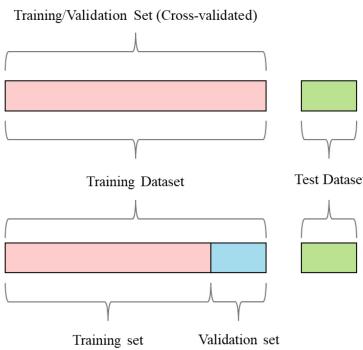


Fig. 2. Training and testing machine learning models.

Based on Fig. 2, cross-validation techniques for optimizing hyperparameters in machine learning are used. Cross-validation involves dividing the training dataset into training and validation sets. Different partitioning techniques can be used. Cross-validation treats the training and validation sets as a single dataset for better evaluation. In machine learning, the term ‘training dataset’ is used instead of ‘training set’ and ‘validation set’ due to cross-validation. The raw training dataset is the original microseismic data without segmentation and labeling [18]. Model performance on the validation dataset is monitored after each epoch. Model weights are stored only if there is an improvement. A validation dataset comprises 30% of the entire dataset and is chosen randomly. This allows evaluation of the network’s ability to generalize.

3.3 Model Evaluation

The performance metrics, including accuracy, precision, recall, and F1 score, are crucial for evaluating the model’s efficacy, calculated using established formulas to assess accuracy, speed, efficiency, and overall performance against predefined objectives. Evaluating the computational efficiency of the proposed methodology is crucial for comprehending its practical applicability, particularly in real-time scenarios. This entails considerations

of processing speed, resource utilization, and the overall efficiency of the implemented Deep Learning models.

$$\text{Processing Speed} = \frac{\text{Number of Data Points Processed}}{\text{Time Taken}} \quad (4)$$

$$\text{Resource Utilization} = \frac{\text{Used Resources}}{\text{Total Resources}} \quad (5)$$

$$\text{Viability Index} = w_1 \times \text{Processing Speed} + w_2 \times (1 - \text{Resource Utilization}) \quad (6)$$

Here, w_1 and w_2 are weights assigned based on the relative importance of processing speed and resource utilization.

To evaluate the progress made by proposing deep learning architecture, a thorough comparison will be conducted against established baseline models. In the selection of baseline models, the CNN was chosen [3, 4, 14–16]. This comparative analysis intends to offer insights into the progress made by the proposed methodology.

3.4 Proposed Approach: SNN Architecture

A systematic study on utilizing Machine Learning in DAS and detecting microseismic events from the literature will be conducted, and suitable models will be proposed. CNN is a potential candidate due to its ability to work in hierarchies, provide translation invariance, and exhibit robustness to noise, enabling effective feature extraction and pattern recognition. However, specific challenges may arise when utilizing CNN for DAS Data in signal processing, particularly in detecting microseismic events. Certain studies have reported limitations of CNN [3, 4, 14–16].

Other techniques could outperform the CNN, such as the SNN, due to the ability to capture spatiotemporal patterns, be energy-efficient, and provide robust, translation-invariant features [19]. An initial study has identified the SNN proposed for microseismic event detection, which is displayed in Fig. 3. This architectural design provides significant benefits for the detection of microseismic events as it enables real-time processing of seismic signals, thus allowing for the identification of subtle seismic events amidst background noise.

After completing the preprocessing stage, spike encoding transforms the preprocessed data into spike trains. These spike trains represent neuronal firing patterns over time, as shown in Fig. 3. This process involves the implementation of encoding methodologies, with rate encoding and temporal encoding being the most suitable [20]. Following this, within the SNN, the spiking neuron is responsible for processing the incoming spike input by considering the specific neuron dynamics. It further integrates the spike over some time to compute the membrane potential and initiates a spike event upon reaching a certain threshold. A neuronal model known as the Adaptive Exponential Integrate-and-Fire Model (AdEx) was utilized in this study. The AdEx neuron model, with its adaptive mechanisms capturing spike-frequency adaptation, is well-suited for processing DAS data, efficiently addressing the dynamic nature of acoustic signals along fiber optic cables and maintaining energy-efficient computational alongside biological realism compared to other models [21]. Spike-Timing-Dependent Plasticity (STDP) is

a significant selection due to its capacity to regulate synaptic weights based on the precise timing of pre-and postsynaptic spikes, enabling the network to learn temporal relationships between input patterns efficiently [20].

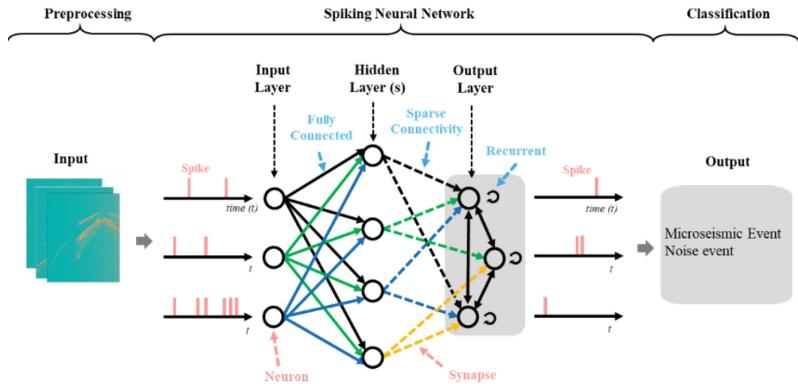


Fig. 3. Proposed SNN architecture for microseismic event detection [20].

Moreover, using supervised learning algorithms, such as SpikeProp, can further boost the SNN's capability to adjust to specific tasks or datasets, offering flexibility and robustness in learning complex patterns from data. Training the SNN necessitates the optimization of various parameters, such as neuronal dynamics and synaptic weights. It frequently entails the utilization of supervised learning algorithms to reduce a specified loss function and guarantee the efficient detection of microseismic events. Finally, the final layer combines spiking patterns to forecast microseismic occurrences, utilizing techniques such as thresholding or sophisticated decoding algorithms to extract significant data from spatiotemporal neural patterns. Developing a robust and accurate microseismic event detection system with SNN architecture requires paramount consideration and integration of each component.

4 Results and Discussion

The experimental results highlight the performance of SNN and CNN architectures for the two categories of occurrences: microseismic and noise. Throughout the training phase, the SNN achieved an accuracy of 80.23%, a recall of 79%, a precision of 79.4%, and an F1-score of 79%. On the other hand, CNN exhibited slightly lower performance metrics with an accuracy of 78.9%, a recall of 78.4%, a precision of 77.5%, and an F1-score of 78.1%. The SNN exhibited better results during the training phase than the CNN across all metrics, as shown in Fig. 4. During the transition to the testing phase, the SNN maintained its superiority over the CNN by achieving an accuracy of 81.1%, a recall of 79.5%, a precision of 79.8%, and an F1-score of 80%. Nonetheless, CNN's performance faced a small drop during the testing phase, yielding an accuracy of 78.7%, a recall of 78%, a precision of 76.8%, and an F1-score of 77.8.

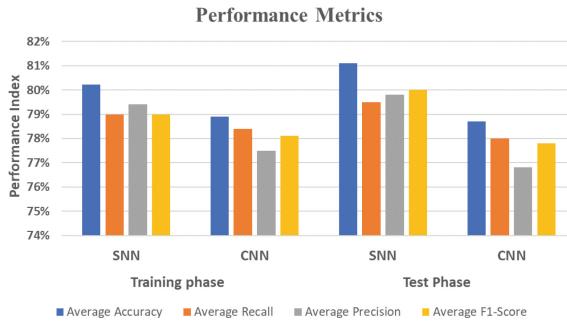


Fig. 4. Performance metrics for training and test phase

The results emphasize the strength of the SNN architecture, which displayed exceptional performance in training and illustrated improved overfitting and stability during the testing phase in contrast to the CNN. Furthermore, SNN outperforms CNN in various metrics, highlighting its interpretability as it adeptly captures and interprets complex temporal patterns in the data, highlighting its prowess in extracting meaningful information. This illustrates the capacity of the SNN as a model that is interpretable and efficient for tasks that call for a nuanced examination of temporal dynamics. The outstanding performance of the SNN can be credited to its spatiotemporal processing capabilities, which effectively capture dynamic patterns and temporal dependencies in the data. Additionally, the event-driven nature of SNNs allows for practical computation and energy usage, making them viable options to traditional CNNs, especially in tasks that require temporal information. Further investigation and fine-tuning of SNN architectures show potential for progressing neural network-based learning and inference.

Based on Fig. 5, the time cost during the test phase indicates that the SNN took around 0.587 s, whereas the CNN needed 0.224 s for inference. Despite the superior performance of SNN, it has resulted in a longer time cost than CNN. This discrepancy in time cost can be ascribed to the inherent differences in computational mechanisms between the two architectures. Despite the advantages of the event-driven nature of SNNs regarding energy efficiency and real-time processing, it might lead to prolonged inference times because of the asynchronous nature of spike-based computations and the need for extra processing steps to interpret spike trains. Conversely, CNNs, more customary and forward-propagating, often manifest swifter inference times, exceptionally when executed on fine-tuned hardware architectures like Graphics Processing Units (GPUs).

However, it is crucial to weigh the trade-offs between inference speed and performance metrics. Although the CNN achieved faster inference times, the SNN's superior performance in accuracy, recall, precision, and F1-score during the test phase suggests its effectiveness in capturing complex temporal patterns and extracting meaningful information from the data. Hence, the slightly longer time cost of the SNN may be justifiable, given its enhanced interpretability and performance.

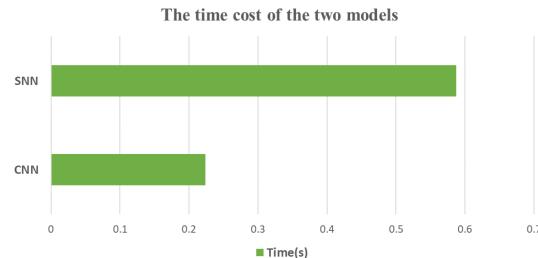


Fig. 5. Time of the two models during the test phase

Moreover, the continuous research and optimization efforts in SNN architectures, algorithmic improvements, and hardware acceleration techniques may alleviate the time cost associated with SNN inference, further enhancing their practical viability for real-world applications. Overall, the SNN exhibits a slightly longer time cost than the CNN during the test phase; nevertheless, its superior performance and interpretability emphasize its potential as a valuable alternative for tasks necessitating nuanced analysis of temporal dynamics.

5 Conclusions

This study concludes by laying the groundwork for an innovative investigation into improving the detection of microseismic events in reservoirs with DAS data. This study addresses the current obstacles in effectively detecting microseismic events by developing an SNN architecture for microseismic event detection while addressing efficiency, model interpretability, and overfitting challenges. Based on the proposed method opens possibilities for creating an optimized SNN architecture and integration with transfer learning. The speed of inference for SNNs can be enhanced by optimizing spike-based computations and interpreting spike trains. Fine-tuning the algorithms and hardware implementations for event-driven processing and improving the efficiency of spike encoding and decoding mechanisms can result in a notable decrease in inference times while upholding the network's performance and interpretability.

Acknowledgments. The author would like to thank Total Energies Ep MY (Grant No: 015MD0-167) and Universiti Teknologi Petronas for supporting this study.

References

- Shiloh, L., Eyal, A., Giryes, R.: Efficient processing of distributed acoustic sensing data using a deep learning approach. *J. Lightwave Technol.* **37**(18), 4755–4762 (2019)
- Spikes, K.T., et al.: Comparison of geophone and surface-deployed distributed acoustic sensing seismic data. *Geophysics* **84**(2), A25–A29 (2019)
- Ma, Y., et al.: Machine learning-assisted processing workflow for multi-fiber DAS microseismic data. *Front. Earth Sci.* **11**, 1096212 (2023)

4. Stork, A.L., et al.: Application of machine learning to microseismic event detection in distributed acoustic sensing data. *Geophysics* **85**(5), KS149–KS160 (2020)
5. Binder, G., Chakraborty, D.: Detecting microseismic events in downhole distributed acoustic sensing data using convolutional neural networks. In: SEG International Exposition and Annual Meeting (2019)
6. Tejedor, J., et al.: A multi-position approach in a smart fiber-optic surveillance system for pipeline integrity threat detection. *Electronics* **10**(6), 712 (2021)
7. Shi, Y., et al.: An easy access method for event recognition of Φ -OTDR sensing system based on transfer learning. *J. Lightwave Technol.* **39**(13), 4548–4555 (2021)
8. Shi, Y., et al.: An event recognition method for Phi-OTDR sensing system based on deep learning. *Sensors* **19**(15) (2019)
9. Peng, Z., et al.: Distributed fiber sensor and machine learning data analytics for pipeline protection against extrinsic intrusions and intrinsic corruptions. *Opt. Express* **28**(19), 27277–27292 (2020)
10. Li, J.C., et al.: Pattern recognition for distributed optical fiber vibration sensing: a review. *IEEE Sens. J.* **21**(10), 11983–11998 (2021)
11. Wu, H., et al.: Pattern recognition in distributed fiber-optic acoustic sensor using an intensity and phase stacked convolutional neural network with data augmentation. *Opt. Express* **29**(3), 3269–3283 (2021)
12. Huot, F., et al.: Detecting microseismic events on DAS fiber with super-human accuracy. In: SEG/AAPG/SEPM First International Meeting for Applied Geoscience & Energy (2021)
13. Anikiev, D., et al.: Machine learning in microseismic monitoring. *Earth-Science Reviews*, 239 (2023)
14. Wamriew, D., et al.: Deep neural networks for detection and location of microseismic events and velocity model inversion from microseismic data acquired by distributed acoustic sensing array. *Sensors* **21**(19), 17 (2021)
15. Wamriew, D., et al.: Microseismic monitoring and analysis using cutting-edge technology: a key enabler for reservoir characterization. *Remote Sensing* **14**(14), 3417 (2022)
16. Huot, F., et al.: Detection and characterization of microseismic events from fiber-optic das data using deep learning. *Seismol. Res. Lett.* **93**(5), 2543–2553 (2022)
17. Kandamali, D.F., et al.: Machine learning methods for identification and classification of events in phi-OTDR systems: a review. *Appl. Opt.* **61**(11), 2975–2997 (2022)
18. Stone, M.: Cross-validatory choice and assessment of statistical predictions. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* **36**(2), 111–133 (1974)
19. Auge, D., et al.: A survey of encoding techniques for signal processing in spiking neural networks. *Neural Process. Lett.* **53**(6), 4693–4710 (2021)
20. Liu, F., et al.: SSTDP: supervised spike timing dependent plasticity for efficient spiking neural network training. *Frontiers in Neuroscience* **15** (2021)
21. Sanaullah, et al., Exploring spiking neural networks: a comprehensive analysis of mathematical models and applications. *Frontiers in Computational Neuroscience*, **17** (2023)



Battery Electric Vehicle Charging Load Forecasting Using LSTM on STL Trend, Seasonality, and Residual Decomposition

Syahrizal Salleh , Roslinazairimah Zakaria , and Siti Roslindar Yaziz

Centre for Mathematical Sciences, Universiti Malaysia Pahang Al-Sultan Abdullah, Kuantan,
Pahang, Malaysia

roslinazairimah@umpsa.edu.my

Abstract. To overcome the challenge of limited high-resolution Battery Electric Vehicle (BEV) charging data, a unique feature engineering technique was implemented. The start-stop electricity charging data from the My Electric Avenue project underwent transformation into a count of concurrent active charging events at 1-min intervals. In an effort to enhance prediction accuracy, the transformed BEV charging data was subjected to decomposition into its trend, seasonality, and residual components using the Seasonal-Trend decomposition using Loess (STL) procedure. Acknowledging the nonlinear, dynamic, and noisy characteristics inherent in BEV charging behavior, the Long Short-Term Memory (LSTM) network was chosen to model electricity demand arising from multiple concurrent BEV charging events. In the model optimization process, hyperparameter tuning focused on adjusting the number of epochs at intervals of 1, 10, 20, 30, 40, and 50. The prediction values for the STL-LSTM decomposed trend, seasonality, and residual components achieved their lowest Mean Absolute Percentage Error (MAPE) values against decomposed testing data at 0.03%, 4.37%, and 27.11%, respectively. The corresponding Root Mean Squared Error (RMSE) values were 0.01, 0.08, and 0.49. Upon reconstruction and comparison against observed testing data, the resulting lowest MAPE occurred at epochs 10, 30, and 40 for trend, seasonality, and residuals, respectively, with values of 1.21% and RMSE 0.51. This is marginally lower than the forecast using the LSTM model on observed data, which recorded a MAPE of 1.38% at RMSE 0.51. The findings underscore the suitability of the STL-LSTM model, tailored for 1-min resolution electricity demand, for electric utility companies aiming to forecast very short-term loads.

Keywords: Battery Electric Vehicle · STL Decomposition · Long Short-Term Memory · Feature Engineering · Charging Behavior

1 Introduction

The Battery Electric Vehicles (BEVs) has garnered widespread acceptance as a means to mitigate global warming by reducing greenhouse gas (GHG) emissions into the atmosphere. Since 2016, carbon dioxide (CO_2) emissions from Internal Combustion Engine Vehicles (ICEVs) have constituted approximately one-quarter of global emissions [1].

The United States stands as the world's second-largest contributor to Electric Vehicle (EV) growth, with EV sales representing about 30% of the global market [1]. The International Energy Agency forecasts that the global EV fleet could reach 250 million by 2030 [2].

As the number of BEV adoptions rises and becomes significant relative to ICEVs, simultaneous charging of multiple BEVs strains the electric grid, presenting challenges related to grid capacity. Electric utility companies actively forecast demand capacity from BEVs to mitigate losses resulting from overgeneration. Moreover, these companies are compelled to enhance electricity generation and grid infrastructure to accommodate rising demands that exceed current capacity levels.

To support the rapidly growing adoption of BEVs, California has invested approximately \$400 million since 2010 to drive BEV adoption and support future infrastructure development [1]. Electricity possesses a distinctive characteristic wherein, in principle, demand and supply equilibrium is imperative and must be maintained continuously [3].

2 Objectives

With the rapid adoption of BEVs, addressing various challenges is crucial to establish a robust ecosystem. One of the most significant issues revolves around the charging infrastructure for BEVs. As of 2020, the number of charging stations has surged to nearly 10 million charging points, with 25% of them located at workplaces [4]. Despite BEV fleet charging loads currently representing a relatively small share of global electricity demand, this load is expected to escalate to 670 GW by the year 2030 [4].

While numerous studies have focused on the demand load of public charging networks ([4–6]), these constitute less than 8% of total BEV chargers [5]. The primary reason for this is the lack of data from private chargers.

To overcome the limitation that has resulted in a lack of studies on BEV charging behavior at private chargers, this research leverages secondary data obtained from the My Electric Avenue program. Initiated in 2013, the My Electric Avenue program loaned approximately 300 units of Nissan Leaf to participants with the explicit purpose of collecting pertinent BEV usage behavior data. The start and stop charging data derived from the My Electric Avenue program is processed and transformed using dedicated feature engineering method into continuous data, making it suitable for in-depth time series analysis.

Prior to building models using the transformed data, a crucial step involves decomposing the data into its trend, seasonality, and residual components. This decomposition is achieved using the Seasonal-Trend decomposition with Loess (STL) method [7]. The motivation behind this decomposition is to assess whether it enhances the overall accuracy of the forecast.

Recognizing the nonlinear, dynamic, and noisy characteristics inherent in BEV charging behavior, the chosen modeling approach for predicting electricity demand arising from multiple concurrent BEV charging events was the Long Short-Term Memory (LSTM) network. Each decomposed component is then individually modeled using an LSTM network. The forecasted data from these individual models are reconstructed, and their effectiveness is assessed by comparing them with the testing data using error metrics. Subsequently, the error metrics obtained from these models are compared against

the error metrics derived from the forecast using the LSTM model on the transformed data. This comprehensive comparison serves to gauge the performance and reliability of the proposed approach in predicting concurrent active charging events based on the My Electric Avenue dataset.

3 Methodology

3.1 Feature Engineering

The primary objective of the My Electric Avenue program was to conduct a comprehensive study of user behavior, with a specific emphasis on gaining insights into BEV charging patterns. As part of the initiative, participants were equipped with Level 1 home chargers. The classification and rated values for BEV charging are categorized into three levels. Level 1 represents slow charging, accommodating up to 1.8 kW at 120 V AC on a single-phase electric power supply system suitable for home usage [5].

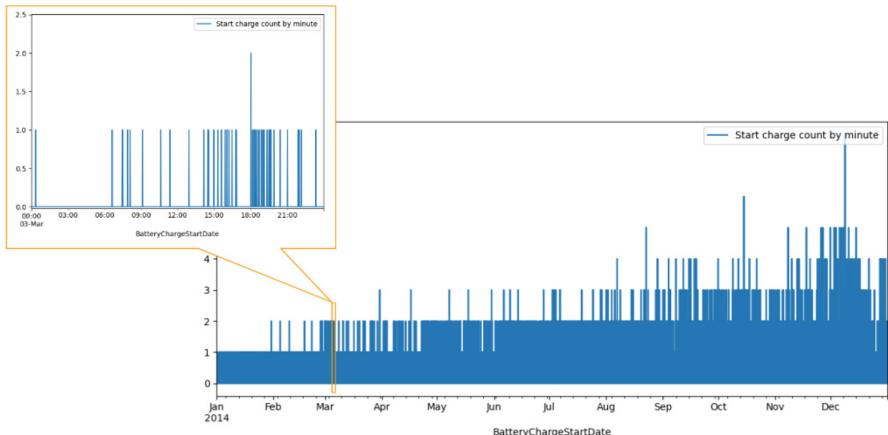


Fig. 1. Start charging count by minutes

The gathered data included crucial details such as charger locations, timestamps marking the initiation and conclusion of charging events, battery charge levels at the start and end of charging events, and the distances covered by the participating vehicles. Figure 1 visually depicts the start charging timestamps' behavior, where each start charging event is highlighted by a one-minute pulse. Similarly, Fig. 2 showcases stop charging events, with each marked by a one-minute pulse. Figure 3 is the feature engineered data that provides insights into the count of concurrent active charging instances, illustrating moments when multiple BEVs are concurrently actively charging and drawing electric power from the power source at a 1-min resolution. Moving forward, this data will be referred to as observed data.

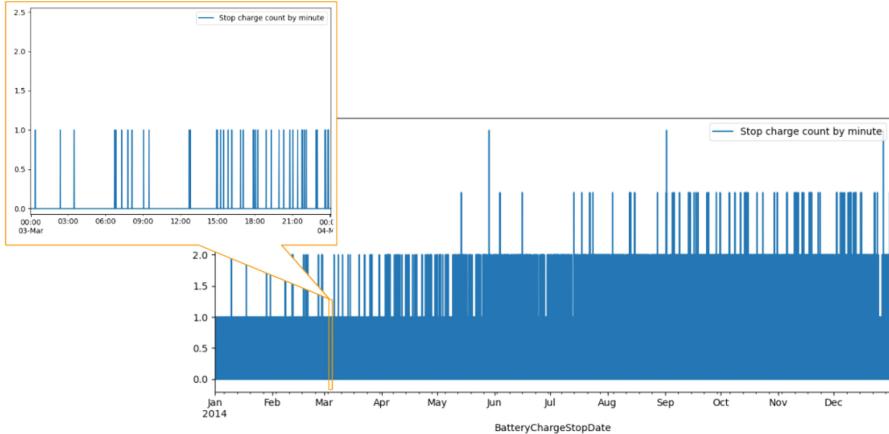


Fig. 2. Stop charging count by minutes

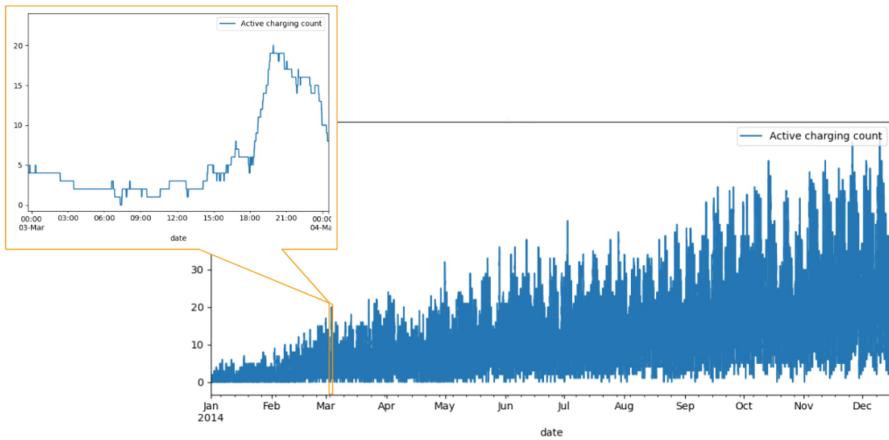


Fig. 3. Feature engineered concurrent active charging count by minutes

3.2 STL Decomposition

The Seasonal-Trend decomposition using Loess (STL) procedure, initially introduced by [7], serves as a filtering method designed for the decomposition of time series observed data into distinct components, including trend, seasonal, and residual components. Moving forward in this study, the feature-engineered data depicted in Fig. 3 will be referred to as the observed data. The decomposed observed data can be reconstructed by adding its three components, as illustrated in Eq. 1

$$Y_v = T_v + S_v + R_v \quad (1)$$

where, the observed data, trend component, seasonal component, and residual components are denoted by Y_v , T_v , S_v , and R_v , respectively, for $v = 1$ to N .

STL comprises a series of smoothing operations involving locally weighted regression. STL is characterized by six parameters that require user input: the number of observations (n_p), the number of passes through the inner loop (n_i), the number of robustness iterations of the outer loop (n_o), the smoothing parameter for the low-pass filter (n_l), the smoothing parameter for the trend component (n_t), and the smoothing parameter for the seasonal component (n_s).

Upon observation, it becomes evident that the observed data exhibits a daily seasonality pattern, with the number of concurrent charging instances peaking at approximately 8 pm and reaching a minimum around 8 am as shown in the snippet window in Fig. 3. In light of this insight, the parameter n_s can be set to the total number of minutes in a day, considering that the observed data is captured at a 1-min resolution. Figure 4 shows plot of decomposed components of the observed data in Fig. 3.

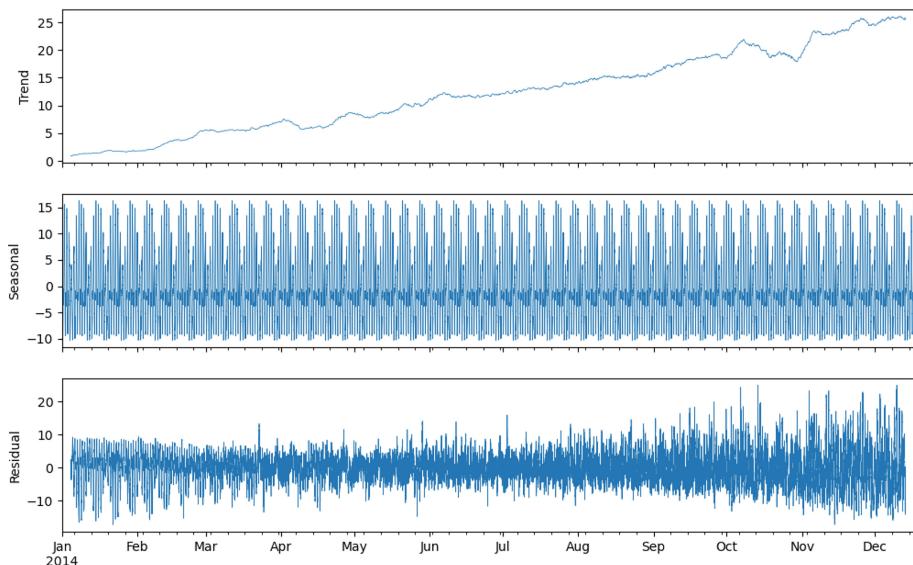


Fig. 4. Decomposed observed data into its trend, seasonal, and residual components

3.3 Recurrent Neural Network Model

The Recurrent Neural Network (RNN) is a specialized neural network architecture tailored for processing sequential data. LSTM network successfully address the shortcomings of traditional RNNs by incorporating gate control mechanisms that integrate short-term memory with long-term memory [8].

Figure 5 visualizes a single LSTM cell, its input layer, output layer, and internal hidden layers. The LSTM cell take input from previous cell state memory (c_{t-1}), previous cell hidden state (h_{t-1}), and current input data (x_t). Internally, the intermediate state of forget gate (f_t), input gate (i_t), and prior cell state (c_t) is calculated. Output from the

LSTM cell are current cell state (c_t), current hidden cell state (h_t), and output gate state (o_t).

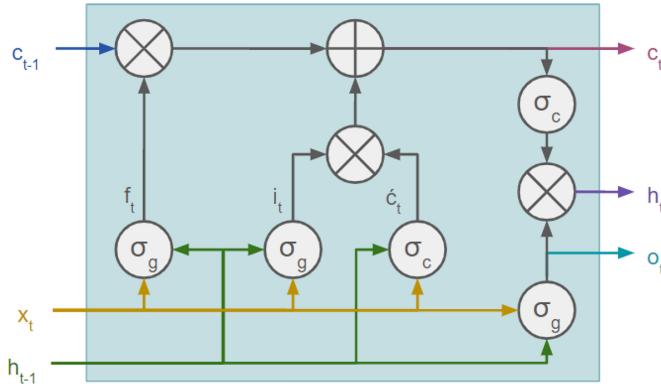


Fig. 5. Visualization of single LSTM cell

The LSTM cell is defined by Eq. 2–7.

$$f_t = \sigma_g(W_f * x_t + U_f * h_{t-1} + b_f) \quad (2)$$

$$i_t = \sigma_g(W_i * x_t + U_i * h_{t-1} + b_i) \quad (3)$$

$$o_t = \sigma_g(W_o * x_t + U_o * h_{t-1} + b_o) \quad (4)$$

$$\tilde{c}_t = \sigma_c(W_c * x_t + U_c * h_{t-1} + b_c) \quad (5)$$

$$\tilde{c}_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \quad (6)$$

$$h_t = o_t \cdot \sigma_c(c_t) \quad (7)$$

where f_t is for the forget gate, i_t for the input gate, o_t is for the output gate, \tilde{c}_t is the prior cell state, c_t is for the current cell state, and h_t is for the hidden state. The constants W_f , W_i , W_o , W_c , U_f , U_i , U_o , and U_c are weight matrices. The constants b_f , b_i , b_o , and b_c are biases. Both weight matrices and biases are not time-dependant. The σ_g is a sigmoid function defined by Eq. 8, and the σ_c is tanh hyperbolic tangent function defined by Eq. 9.

$$\sigma_g(x) = (1 + e^{-x})^{-1} \quad (8)$$

$$\sigma_c(x) = \left(\frac{e^{2x} - 1}{e^{2x} + 1} \right) \quad (9)$$

The hidden LSTM layer is structured as a sequential single layer, encompassing 125 units of LSTM cells. In the training process, the data was divided into sequences of 60 data points, each associated with a single expected output. The expected output corresponds to the subsequent data point in the training dataset.

3.4 One-Step Ahead Forecasting

The LSTM model employed in this study is configured for supervised learning. To impart the behavioral patterns of the data to the LSTM network, each of the STL decomposed components of the observed data, as depicted in Fig. 4, is partitioned into training and testing sets. Specifically, the training data comprises 80% of the dataset, starting from the initial entry, while the testing data consists of the remaining subsequent 20% of the dataset [9]. Based on the observed data, the training dataset contains 452,304 data points, while the testing dataset consists of the remaining 50,256 data points.

The model was trained on the training data over 1, 10, 20, 30, 40, and 50 epoch iterations, and the loss function for the final epoch was recorded. Subsequently, a one-step ahead forecast was executed using the fitted model. The forecasted results were then compared to the testing data, and the comparison was quantified using Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE).

4 Results and Discussion

The number of epochs in this study is constrained to a range between 1 and 50 epochs. The forecasted values are validated against the testing data, and their corresponding RMSE and MAPE values are recorded. Each epoch run takes approximately 410 s to complete.

During each epoch run, the value of the loss function is recorded. A loss function serves as a mathematical construct, mapping an event to a real number. In the context of optimization problems, the primary objective is to minimize this loss function. Within the field of neural networks, the lowest point of the loss function is referred to as the global minimum. Table 1 summarizes model loss, and forecast error metrics per epoch run for each decomposed components.

Table 1. Forecast loss and error metrics of each decomposed components. The first column indicate number of epochs selected for the model

Ep	Trend			Seasonal			Residual		
	Loss	RMSE	MAPE	Loss	RMSE	MAPE	Loss	RMSE	MAPE
1	1.38×10^{-4}	0.48	1.94%	1.34×10^{-4}	0.09	5.99%	1.99×10^{-4}	0.82	103.55%
10	4.06×10^{-6}	0.09	0.37%	1.08×10^{-5}	0.08	4.37%	6.75×10^{-5}	0.58	64.09%
20	1.94×10^{-6}	0.06	0.24%	8.71×10^{-6}	0.08	4.67%	6.56×10^{-5}	0.55	53.83%
30	1.34×10^{-6}	0.04	0.16%	8.01×10^{-6}	0.08	4.46%	6.50×10^{-5}	0.49	27.11%
40	1.11×10^{-6}	0.01	0.03%	7.81×10^{-6}	0.09	5.06%	6.46×10^{-5}	0.50	29.91%
50	9.66×10^{-7}	0.03	0.12%	7.87×10^{-6}	0.09	5.55%	6.43×10^{-5}	0.49	27.23%

Based on the results in Table 1, the loss function values for the decomposed components that are trend, seasonal, and residual, demonstrate a consistent decrease with an increase in the number of epochs. It is conceivable that the global minima for all the components possibly lie beyond the epoch range covered by this study.

The error metrics for the trend decomposed components highlight the benefits of the decomposition process, demonstrating exceptionally small RMSE and MAPE values. At epoch 40, its RMSE and MAPE stand at 0.01 and 0.03%, respectively. The lowest RMSE and MAPE values for the seasonal decomposed components are 0.09 and 5.06%, respectively, at epoch 30. As for the residuals decomposed components, the lowest RMSE and MAPE values are 0.49 and 27.11%, respectively, at epoch 30. Figure 6 visualizes relationship between MAPE of each decomposed components as number of epoch run increases.

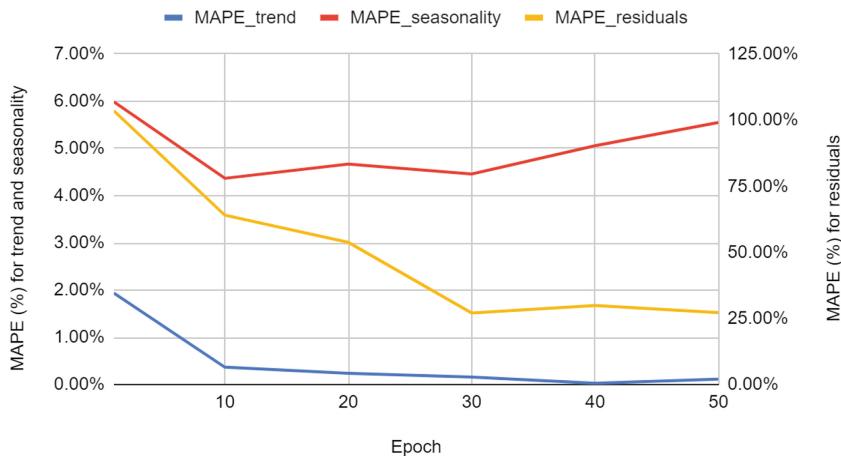


Fig. 6. Plot of the decomposed components MAPE vs epoch intervals at 1, 10, 20, 30, 40, and 50

The reconstructed forecasted value for each decomposed component is obtained by summing up all the individual components at each timestamp, as indicated in Eq. 1. In the process of selecting the combination with the lowest RMSE and MAPE values, an exhaustive search is conducted across all possible combinations of epochs for the trend, seasonal, and residual components. This meticulous exploration aims to identify the optimal configuration that yields the most accurate forecasting results.

There are a total of 216 possible combinations when considering different numbers of epochs for each decomposed component. The 10 lowest MAPE combinations are tabulated in Table 2.

In order to evaluate the effectiveness of the proposed method, an LSTM model is constructed using the observed data. Subsequently, the forecasted values derived from this method are compared against the observed forecast data, employing metrics such as MAPE and RMSE as benchmarks. The objective is to ascertain whether the lowest MAPE achieved through this method outperforms the MAPE of the observed forecast data. This comparative analysis provides valuable insights into the method's efficacy and its ability to yield more accurate predictions compared to the baseline forecast using LSTM on observed data. Table 3 compares result from this study with the baseline forecast.

Table 2. Top ten lowest forecasted values of decomposed observed data that yield lower MAPE results using STL-LSTM

Epoch			Forecast evaluation	
Trend	Seasonal	Residual	RMSE	MAPE
10	30	40	0.51	1.21%
10	50	40	0.51	1.22%
20	10	40	0.51	1.22%
30	20	50	0.50	1.23%
50	20	50	0.50	1.24%
10	10	40	0.51	1.24%
20	20	30	0.50	1.24%
30	20	30	0.50	1.24%
20	30	40	0.51	1.24%
10	50	30	0.51	1.25%

Table 3. Comparison between one-step ahead LSTM forecast on observed data and reconstructed one-step ahead STL-LSTM forecast

	Forecast evaluation		Epoch			
Model	RMSE	MAPE	Observed	Trend	Seasonal	Residual
LSTM	0.51	1.38%	20			
STL-LSTM	0.51	1.21%		10	30	40

5 Conclusions

The STL-LSTM method proposed in this study has achieved slightly lower MAPE and RMSE values compared to the baseline forecast using observed data, which recorded a MAPE of 1.38% and RMSE 0.51, at epoch 20. Notably, the MAPE results from the decomposed trend and seasonality components exhibit favorable performance. However, there is room for improvement in the MAPE result from the residual components, suggesting potential enhancements for better overall accuracy in predicting BEV charging behavior.

Load forecasting spans various temporal scales, including long-term, medium-term, short-term, and ultra-short-term forecasting, each tailored to specific prediction periods. Ultra-short-term load forecasting (USTLF) is a specialized focus that involves estimating power consumption within a timeframe ranging from a few minutes to hours ahead. This type of forecasting is integral for facilitating rational tariff adjustments, ensuring smooth system operation, and enhancing economic efficiency [10].

The approach demonstrated by the LSTM network and 1-min resolution data in predicting BEV charging behavior can be extrapolated to forecast USTLF. This application

becomes particularly relevant in the context of a smart electric supply grid. Leveraging the same principles of utilizing temporal patterns and historical data for accurate short-term predictions, the LSTM network can optimize the operations and efficiency of the electric grid in handling dynamic and rapidly changing load scenarios. The network's ability to capture sequential dependencies in the data positions it effectively for forecasting and optimizing electricity demand at an ultra-short-term level.

Acknowledgement. This research was funded by Universiti Malaysia Pahang Al-Sultan Abdullah (grant number RDU233008) under a special publication grant from the Research and Innovation Department, and Doctorate Research Scheme (Salleh, S.)

References

1. Yi, Z., Liu, X.C., Wei, R., Chen, X., Dai, J.: Electric vehicle charging demand forecasting using deep learning model. *J. Intell. Transp. Syst.* **26**(6), 690–703 (2022)
2. Mohsenimanesh, A., Entchev, E., Bosnjak, F.: Hybrid Model Based on an SD Selection, CEEMDAN, and Deep Learning for Short-Term Load Forecasting of an Electric Vehicle Fleet. *Appl. Sci.* **12**(18), 9288 (2022)
3. Nguyen, H., Hansen, C.K.: Short-term electricity load forecasting with Time Series Analysis. In: 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), pp. 214–221. IEEE (2017)
4. Dokur, E., Erdogan, N., Kucuksari, S.: EV fleet charging load forecasting based on multiple decomposition with CEEMDAN and swarm decomposition. *IEEE Access* **10**, 62330–62340 (2022)
5. Chang, M., Bae, S., Cha, G., Yoo, J.: Aggregated electric vehicle fast-charging power demand analysis and forecast based on LSTM neural network. *Sustainability* **13**(24), 13783 (2021)
6. Buzna, L., De Falco, P., Khormali, S., Proto, D., Straka, M.: Electric vehicle load forecasting: A comparison between time series and machine learning approaches. In: 2019 1st International Conference on Energy Transition in the Mediterranean Area (SyNERGY MED), pp. 1–5. IEEE (2019)
7. Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I.S.T.L.: A seasonal-trend decomposition. *J. Off. Stat.* **6**(1), 3–73 (1990)
8. Memory, L.S.T.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (2010)
9. Koohfar, S., Woldemariam, W., Kumar, A.: Prediction of Electric Vehicles Charging Demand: A Transformer-Based Deep Learning Approach. *Sustain.* **15**(3), 2105 (2023)
10. Tong, C., Zhang, L., Li, H., Ding, Y.: Attention-based temporal–spatial convolutional network for ultra-short-term load forecasting. *Electr. Power Syst. Res.* **220**, 109329 (2023)



Convolutional Neural Network Using Regularized Conditional Entropy Loss (CNNRCoE) for MNIST Handwritten Digits Classification

Ashikin Ali^(✉), Norhalina Senan, and Norhanifah Murli

Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn

Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia

ashikinbintiali@gmail.com, halina@uthm.edu.my

Abstract. Convolutional Neural Networks (CNNs) have demonstrated remarkable performance in various image classification tasks, including the classification of handwritten digits in the MNIST dataset. It is evident that it has a vast go, but it too faces certain limitations such as overconfidence, complexity and overfitting. Hence, an approach leveraging Regularized Conditional Entropy (RCE) as loss function within a CNN architecture for enhanced accuracy in digit classification as a whole called Convolutional Neural Network using Regularized Conditional Entropy Loss (CNNRCoE). The regularization technique employed aims to mitigate overfitting and improve generalization by penalizing the complexity of the model. In this study, extensive experiments have been conducted on the MNIST Handwritten digits inclusive of 5 datasets to evaluate the effectiveness of the proposed method, comparing it with traditional CNN architectures. The results demonstrate that the integration of RCoE within the CNN framework yields superior performance, achieving state-of-the-art accuracy at about 98% with the increase about 20%, while maintaining sturdiness against overfitting.

Keywords: Convolutional Neural Netowrk · Conditional Entropy · Regularization · Classification

1 Introduction

CNN are evolved with various image classification such as fashion images [1], medical images [2], face images [3], also handwriting recognition [4, 5] which chosen as domain focus of this study. Handwritten digit classification is a fundamental task in the field of pattern recognition and machine learning, with applications ranging from postal automation to bank check processing [6]. The MNIST dataset, comprising a collection of grayscales images of handwritten digits [7], has long served as a benchmark dataset for evaluating the efficacy of various classification algorithm. CNNs have emerged as a powerful tool for image classification and recognition tasks [8], demonstrating exceptional performance in accurately identifying handwritten digits. In recent years, advancements

in deep learning have led to the development of sophisticated CNN architectures capable of achieving remarkable accuracy on the MNIST dataset. However, as the complexity of neural networks increases, so does the risk of overfitting and overconfidence, wherein the model memorizes the training data's noise that resulting in poorer model performance rather than learning generalizable features [9]. Therefore, this study wholly motivated to focus on the overfitting limitation of CNN by observing and focusing on the training process of CNN to reduce the complexity and avoid overfitting of the training nature by working on the loss function of CNN. To address this challenge, regularization techniques have been introduced to prevent overfitting and enhance the model's ability to generalize to unseen data [10].

In this paper, an approach, termed as CNNRCoE for handwritten digit classification on the MNIST dataset have been implemented, this study is an extended work from the work by Xu [11]. Xu implemented mutual information approach without regularization and achieved 72.46%. This method leverages regularized conditional entropy loss (RCoE), a regularization technique designed to penalize the complexity of the model and encourage the learning of discriminative features. By integrating RCoE within the CNN architecture, the aim to improve the model's generalization performance while maintaining high accuracy on digit classification tasks.

The primary objective of this study is to investigate the effectiveness of CNNRCoE in comparison to traditional CNN architectures for handwritten digit classification. Extensive experiments conducted to evaluate the proposed method's performance in terms of classification Accuracy, Precision, Recall and F1 Score. Furthermore, provided insights into the interpretability of the model's decisions through visualization techniques, offering a deeper understanding of the features learned by the network. Overall, this work contributes to the ongoing efforts in advancing the state-of-the-art in handwritten digit classification, demonstrating the potential of regularization techniques in enhancing the performance and reliability of CNN models on the MNIST dataset. The rest of the paper is organized as followings; Sect. 2 explains the theory of CNN for MNIST digit classification, regularization and conditional entropy. Section 3 visualizes the implementation of the proposed method. Section 4 describes the experimental results. Section 5 concludes the whole paper also discuss the future work.

2 Literature Review

Deep learning techniques, particularly CNNs, have demonstrated remarkable performance in various image classification tasks, including the recognition of handwritten digits [12, 13]. The MNIST dataset, comprising a collection of 28×28 grayscale images of handwritten digits, has long served as a benchmark for evaluating the efficacy of new algorithms in this domain. In this literature review, exploration of recent advancements in CNN architectures and loss functions tailored for MNIST digit classification is aligned.

2.1 CNN Architectures for MNIST Handwritten Digit Classification

Over the past few years, researchers have proposed numerous CNN architectures to enhance the accuracy and efficiency of MNIST digit classification. LeNet-5, introduced by LeCun in 1998 [14], was one of the pioneering CNN architectures for this task, consisting of convolutional and pooling layers followed by fully connected layers. Since then, deeper and more complex architectures have been developed. For instance, variants of the ResNet and DenseNet architectures have been adapted and optimized for MNIST, leveraging skip connections and dense connectivity to facilitate feature reuse and gradient flow, thereby improving performance [15]. Figure 1 demonstrates the visual of handwritten digit classification process using CNN. In the interim, Fig. 2 visualizes MNIST handwritten Digits.

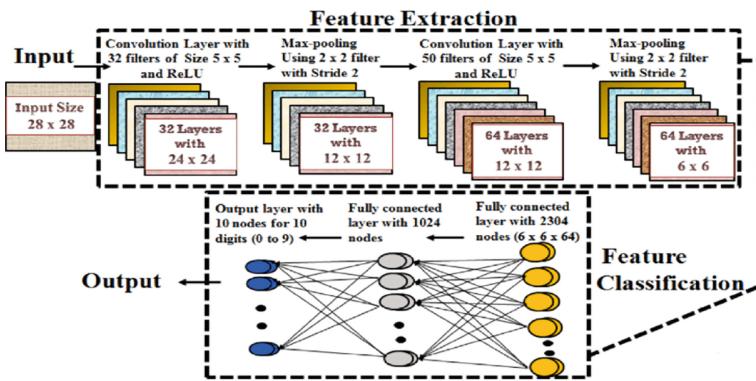


Fig. 1. A visualization of Handwritten Digit Classification using CNN



Fig. 2. A visualization of MNIST Handwritten Digits

Followings are the factors of variation that leads to the benchmarks as shown in Table 1:

- i. *mnist-rot*: the digits were rotated by an angle generated uniformly between 0 and 2π radians. Thus, the factors of variation are the rotation angle and the factors of variation already contained in MNIST, such as hand-writing style.

- ii. *mnist-back-rand*: a random background was inserted in the digit image. Each pixel value of the background was generated uniformly between 0 and 255.
- iii. *mnist-back-image*: a patch from a black and white image was used as the background for the digit image. The patches were extracted randomly from a set of 20 images downloaded from the internet. Patches which had low pixel variance contained little texture were ignored.
- iv. *mnist-rot-back-image*: the perturbations used in *mnist-rot* and *mnist-back-image* was combined.

Table 1. MNIST Handwritten Variations.

MNIST Handwritten Digits Types						
<i>mnist-back-rand</i>						
<i>mnist-rot</i>						
<i>mnist-back-image</i>						
<i>mnist-rot-back-image</i>						

2.2 Regularization Techniques

Regularization serves as a method to incorporate prior knowledge, often referred to as common sense, into the parameters of a learning algorithm. This knowledge is particularly valuable when the available data is insufficient. Broadly speaking, any addition to the learning process or prediction mechanism aimed at addressing data scarcity is considered regularization. By introducing a controlled level of bias, regularization effectively reduces model variance. In the realm of machine learning, data points can be decomposed into a pattern component and stochastic noise. It is essential for a machine learning algorithm to accurately capture the underlying pattern while disregarding the noise. CNNs, like other neural networks, are susceptible to overfitting, especially when dealing with complex datasets with limited samples. Algorithms that excessively focus on fitting noise and idiosyncrasies alongside the pattern are prone to overfitting. Regularization, in such cases, assists in selecting an appropriate model complexity, ensuring improved predictive performance on unseen data in the future [16]. Figure 3 shows the overview of regularization techniques for neural networks.

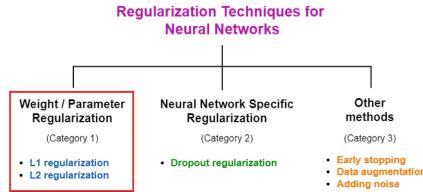


Fig. 3. Overview of regularization techniques for neural networks

L1 regularization, L2 regularization, dropout, and batch normalization are widely used techniques to achieve this goal in deep learning models. However, recent research has shown that L2 regularization techniques tailored to specific loss functions and tasks can further enhance performance [17]. By adding an L2 penalty term to the loss function, the model learns to minimize not only the training error but also the magnitude of the weights. This encourages the network to learn more robust and generalizable features from the data, as it reduces the influence of noise and outliers. L2 regularization provides a mechanism to control the complexity of the CNN model by adjusting the regularization strength. The regularization parameter, often denoted as λ , one can balance between fitting the training data well and keeping the model simple. This flexibility is crucial for achieving optimal performance on validation or test data. Overall, L2 regularization is a versatile technique that addresses key challenges in training CNNs, including overfitting, generalization, and model complexity control, making it a valuable component of the training process [15]. Figure 4 illustrates L2 regularization.

L2 Regularization

$$\text{Modified loss function} = \text{Loss function} + \lambda \sum_{i=1}^n W_i^2$$

Fig. 4. L2 Regularization representation

Modified Regularized Conditional Entropy Loss (RCoE) using L2 regularization in this study delightedly is one such technique that effectively balances the trade-off between accuracy and model complexity by penalizing high-confidence misclassifications.

2.3 Conditional Entropy Loss

Conditional Entropy Loss (CoE) is a loss function that measures the uncertainty associated with predictions. Unlike traditional loss functions such as cross-entropy, CoE takes into account the conditional probability distribution of predictions given the input data [17]. By optimizing the model to minimize conditional entropy, CoE encourages the model to produce not only accurate but also confident predictions. This is particularly beneficial for tasks with high class imbalance or noisy labels as shown in Fig. 3.

Following is the mathematical representation for CoE:

$$H(Y|X) = \frac{1}{n} \sum_{n=1}^N L(\sum P(Y|X) * \log(P(Y|X))) \quad (1)$$

$H(Y|X)$ represents the conditional entropy, for instance the uncertainty of the predicted labels given the input features. $P(y|x)$ is the predicted probability of class y given input x . N is the number of classes. The equation calculates the average uncertainty or entropy of the predicted class labels given the input features. The goal of the tasks is to minimize this entropy loss to improve the accuracy and the model's prediction quality.

3 Proposed CNNRCoE

Motivated by the success of CoE in mitigating the effects of noisy labels and improving model sturdiness, this study proposed a novel CNN architecture, termed CNNRCoE, for MNIST handwritten digit classification. CNNRCoE integrates CoE as a regularization term into the training objective, thereby promoting the exploration of diverse solutions and reducing the sensitivity to outliers in the training data. By combining the expressive power of CNNs with the regularization capabilities of CoE, CNNRCoE aims to achieve state-of-the-art performance on the MNIST dataset while maintaining model simplicity and interpretability. Figure 5 visualizes the conceptual idea of CNNRCoE:

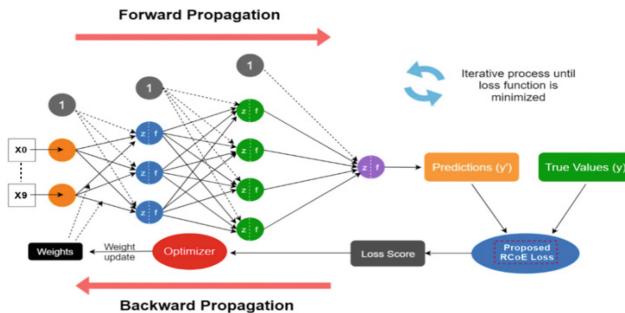


Fig. 5. A conceptual visualization of CNNRCoE

In Fig. 5, the visualization aligns the Input to represent the raw input data, such as an image. Convolutional Layers represent the performance of convolution operations on the input data to extract features. Pooling Layers down sample the feature maps generated by the convolutional layers. Dropout regularization is applied at pooling layers prevent overfitting. Flatten layer flattens the output from the previous layers into a vector. Fully Connected Layers take the flattened vector and learn to classify the input. Regularized Conditional Entropy Layer computes the regularized conditional entropy, which involves the conditional probability distribution of the network's predictions given the input data and is regularized by some entropy measure.

Output Layer produces the final output of the network, often representing class probabilities. Equation 2 represents the mathematical representations of the proposed RCoE:

$$RCoELos s_{w,b} = H(Y|X) = \frac{1}{n} \sum_{n=1}^N L(\sum P(y|x) * \log(P(y|x))) + \frac{\lambda}{n} \sum_j w_j^2 \quad (2)$$

Here, w_j^2 is essentially sums of squared weights, if the weights have larger positive or negative value that will make the regularized cost larger. While $\frac{\lambda}{n}$ normalization trump and the λ is a hyperparameter that has to be initialized during the process, it is the strength of regularization, the larger the better outcome. Typically, initialization starts with small value such as 0.1 or 0.01, can also initialize value larger than 1 that will be also a strong penalty to the cost, denotes the steps involved in training and evaluating the CNNRCoE model for MNIST handwritten digit classification. It includes the initialization of model parameters, training loop over multiple epochs with mini-batch gradient descent, and evaluation on the test set of CNNRCoE.

The bolded steps in Table 2 involves computing the conditional entropy loss between the predicted scores and the true labels. Conditional entropy measures the average uncertainty in predicting the true label given the input. In the context of MNIST handwritten digit classification, after passing an image through the CNN model, the output would be a set of probabilities assigned to each possible digit (0–9). The conditional entropy loss is computed based on these predicted probabilities and the true labels. It quantifies how much information is lost on average when predicting the true label from the predicted probabilities. In this step, regularization loss is calculated for each layer of the CNN. The regularization loss for each layer is typically calculated based on the weights of that layer. Once the conditional entropy loss and regularization loss for each layer are calculated, those are combined to compute the total loss.

Table 2. Algorithm of CNNRCoE MNIST Handwritten Digits.

Algorithm of CNNRCoE MNIST Handwritten Digits	
Input:	<ul style="list-style-type: none"> - X_train: Training images (features), y_train: Training labels - X_test: Test images, y_test: Test labels - learning_rate: Learning rate for optimization: 0.0001 - momentum rate: Momentum rate for optimization: 0.9 - num_epochs: Number of epochs for training: 100 - batch_size: Size of mini-batches for training: 64 - lambda_reg: Regularization parameter: 0.01 <p>Output:</p>
1. Initialize CNNRCoE model parameters:	<ul style="list-style-type: none"> - Convolutional layers - Fully connected layers - Regularization parameter (lambda_reg) - Learning rate (learning_rate)
2. For each epoch from 1 to num_epochs:	<ol style="list-style-type: none"> 2.1 Shuffle training data (X_train, y_train) and divide to mini batches 2.2 For each mini-batch: <ol style="list-style-type: none"> 2.3.1 Perform forward pass: <ul style="list-style-type: none"> - Apply convolutional layers, Apply activation function: SoftMax - Flatten the output. Pass through fully connected layers - Compute the output scores 2.3.2 Compute the Regularized Conditional Entropy Loss (RCoE): <ul style="list-style-type: none"> - Calculate Conditional-entropy loss between predicted scores and true labels - Calculate regularization loss for each layer - Compute the total loss as the sum of Conditional entropy loss and regularization loss. 2.3.3 Performance backward pass: <ul style="list-style-type: none"> - Compute gradients of the loss with respect to network parameters - Update parameters using gradient descent with the learning rate 2.4 Evaluate model performance: <ul style="list-style-type: none"> - Compute accuracy on the training set using the trained model - Print or log the loss and accuracy for the current epoch
3. Evaluate the trained model on the test set:	<ol style="list-style-type: none"> 3.1 Perform forward pass on the test images 3.2 Compute accuracy on the test set using the trained model
4. Output the trained CNNRCoE model and test accuracy	

4 Experimental Design and Results

Experiments are conducted to evaluate the proposed along with comparative model in providing minimum error and high accuracy for multiclass classification. Entirely the experiment was accomplished using Intel(R) Core (TM) i3-7100U CPU with 4GB of RAM having Windows 10 Pro operating system. The Python 3.6, Pytorch 1.4 and Jupyter notebook were used as language and compiler to develop and experimentally test these algorithms. TensorFlow library is an open-source library and machine learning that allows simple development that enables to support in both CPU and GPU. The architecture after ample trials it is fixed to similar setting in term of convolution layer, max pooling layer, fully connected layer, learning rate and other parameters in order to achieve optimum outcome. Since the datasets have parallel structure, the data segregated equally for both model with 80:20 ratio which suits large dimensional datasets, the model consist of 784 ($28 \times 28 \times 1$ pixels) input nodes and 10 nodes for output layer as that is the number of classes.

While conducting experiment, it is common to consider on several usual parameters which related to the model's nature. Hence, for this study the consideration of experimental design is aligned accordingly. Starting with 784 of input nodes and ends with 10 output nodes. The model consists of 2 fully connected layers with 256 hidden nodes. As per fine-tuned, maximum epoch measured for this model is 100. The model using SoftMax activation function with Adam Optimizer. Learning rate and is set to 0.00001 while Momentum to 0.9.

4.1 Result Analysis

The regularization provided by conditional entropy helped enhance the sturdiness of the CNN model, leading to better generalization and higher accuracy in digit classification tasks. The results highlight the effectiveness of incorporating regularization techniques with conditional entropy into CNN architectures for handwritten digits classification. Experiments of this study demonstrated that the CNNRCoE model achieved superior performance compared to baseline CNN models across various evaluation metrics, including accuracy, precision, recall and F1 score as shown in Fig. 6 and Fig. 7, results are in percentage.

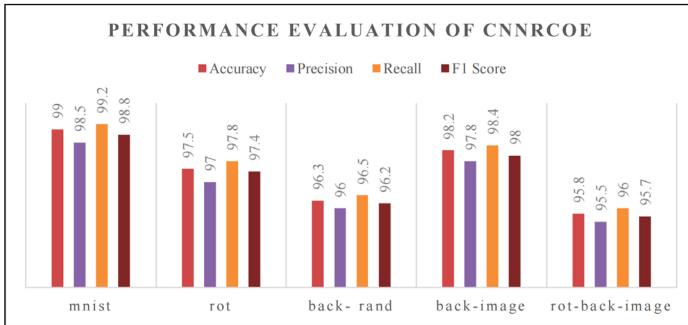


Fig. 6. Experiment results of CNNRCoE

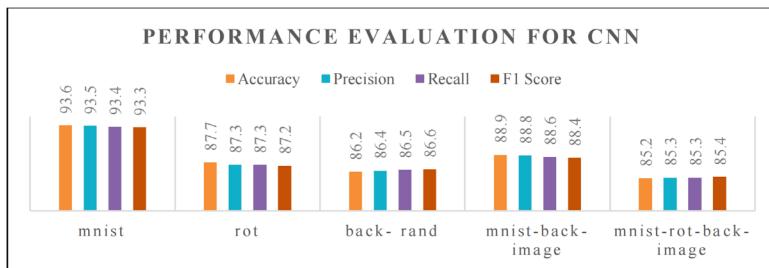


Fig. 7. Experiment results of CNN

From the figures above, it is evident CNNRCoE significantly outperforms the CNN on the *mnist* dataset across all metrics. This indicates that incorporating the RCoE into the training process improves the model's ability to classify handwritten digits accurately. Consistently, the CNNRCoE significantly beats the CNN on the *rot* dataset, it is clear that the proposed model helps in handling variations such as rotation in the input data. CNNRCoE also performed better on the *back-rand* dataset, hence the regularization also aids in mitigating the effects of noise in input data that resulting in better classification performance. CNNRCoE outperforms the CNN model on the complex *mnist-back-image* dataset, therefore proposed idea also capable in handling the complexity introduced by background images. On the *mnist-rot-back-image* dataset with rotated MNIST digits and complex background images once again CNNRCoE achieves significant performance compared to the CNN. This indicates the robustness of CNNRCoE in handling complex input scenarios. In summary, the CNNRCoE consistently outperforms the CNN across all datasets, demonstrating its effectiveness in improving classification performance, especially in handling variations and noise in the input data. The regularization provided by the RCoE contributes enhanced performance of the CNNRCoE model.

5 Conclusion and Future Works

In this paper, an approach for MNIST handwritten digits classification using a Convolutional Neural Network (CNN) with Regularized Conditional Entropy Loss (CNNRCoE) is implemented. This method controls regularized conditional entropy to improve the performance of CNNs on the MNIST dataset. While the proposed CNNRCoE model shows promising results, there are several avenues for future research and improvements. Exploration of different regularization techniques to investigate the effectiveness of other regularization techniques in combination with conditional entropy to further enhance model performance. Conduct hyperparameter tuning to optimize the CNNRCoE model for better performance on the MNIST dataset and potentially extend it to other datasets or domains, especially on violence contents. Explore transfer learning approaches to leverage pre-trained CNN models on larger datasets for improved feature extraction and classification accuracy. Investigate the use of data augmentation techniques to increase the diversity of the training dataset and improve the CNNRCoE model's sturdiness to variations in handwritten digit images. Deployment in different logarithm value of entropy of proposed CNNRCoE. Evaluate the CNNRCoE model's performance in real-world applications beyond digit classification, such as document analysis, signature verification, and postal automation systems. As a conclusion, the CNNRCoE model presents a promising framework for MNIST handwritten digits classification, and further research in the aforementioned areas could lead to even more significant advancements in the field of computer vision and deep learning-based image classification tasks.

References

1. Le, J.: The 4 Convolutional Neural Network Models That Can Classify Your Fashion Images (2018)
2. Saranya, P., Asha, P.: Survey on big data analytics in health care. 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT). IEEE (2019)
3. Song, A.-P., et al.: Similar face recognition using the IE-CNN model. *IEEE Access* **8**, 45244–45253 (2020)
4. Baldominos, A., Saez, Y., Isasi, P.: A survey of handwritten character recognition with mnist and emnist. *Appl. Sci.* **9**(15), 3169 (2019)
5. Ahlawat, S., et al.: Improved handwritten digit recognition using convolutional neural networks (CNN). *Sensors* **20**(12), 3344 (2020)
6. Rudraswamimath Vijayalaxmi R., Bhavanishankar, K.: Handwritten digit recognition using CNN. *International Journal of Innovative Science and Research Technology* **4**(6), 182–187 (2019)
7. LeCun, Y.: The MNIST database of handwritten digits. <http://yann.lecun.com/exdb/mnist/> (1998)
8. Chen, L., et al.: Review of image classification algorithms based on convolutional neural networks. *Remote Sensing* **13**(22), 4712 (2021)
9. Wei, H., et al.: Mitigating neural network overconfidence with logit normalization. *International Conference on Machine Learning*. PMLR (2022)
10. Nusrat, I., Jang, S.-B.: A comparison of regularization techniques in deep neural networks. *Symmetry* **10**(11), 648 (2018)

11. Xu, Y., et al. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. *Advances in neural information processing systems* **32** (2019)
12. Alom, M.Z., et al.: Handwritten bangla digit recognition using deep learning. arXiv preprint [arXiv:1705.02680](https://arxiv.org/abs/1705.02680) (2017)
13. Ahmed, S.S., et al.: A novel technique for handwritten digit recognition using deep learning. *Journal of Sensors* (2023)
14. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *nature* **521**(7553), 436–444 (2015)
15. Moradi, R., Berangi, R., Minaei, B.: A survey of regularization strategies for deep models. *Artif. Intell. Rev.* **53**(6), 3947–3986 (2020)
16. Alzubaidi, L., et al.: Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data* **8**, 1–74 (2021)
17. Garcin, C.: Loss functions for set-valued classification. Université de Montpellier, Diss (2023)



Optimizing Team Formation for Welfare Activities: A Study Using Four Metaheuristic Optimization Algorithms

Muhammad Akmaluddin and Rozlina Mohamed^(✉)

Universiti Malaysia Pahang Al-Sultan Abdullah, 26600 Pekan, Pahang, Malaysia
rozlina@umpsa.edu.my

Abstract. This study compares the effectiveness of Team Formation Problem (TFP) implementation in welfare activities using four metaheuristic optimization algorithms: Jaya (JA), Sine-Cosine (SCA), Firefly (FFA), and Particle Swarm Optimization (PSO). TFP often involves a large search space with numerous variables and constraints. These metaheuristic algorithms are efficiently exploring the search space and converge to promising solutions quickly. The purpose of this paper is to determine which algorithm performs best in assembling effective volunteer teams, taking into account factors such as skills, availability, and preferences. The study uses simulations to evaluate the algorithms' convergence speed, solution quality, and robustness. The experimental results show that SCA and PSO perform well in team formation, with SCA outperforming PSO and completing tasks faster. However, algorithm effectiveness may vary depending on the number of volunteer skills required.

Keywords: Metaheuristic · Jaya · Sine-cosine · Firefly · Particle Swarm Optimization · team formation

1 Introduction

Heterogeneity among volunteers poses a common challenge in team formation for welfare activities, encompassing differences in age, gender, education, skills, and motivation, which can impede successful collaboration and efficiency [1, 2]. To address this issue, various optimization strategies are available, and this study evaluates the efficacy of four algorithms: the Sine-Cosine Algorithm (SCA), Jaya Algorithm (JA), Firefly Algorithm (FFA), and Particle Swarm Optimization (PSO). Each algorithm offers unique approaches to optimizing team composition and synergy, drawing upon mathematical principles and mimicking natural phenomena to guide the search for optimal solutions that are required to solve the team formation problem (TFP).

The SCA leverages the mathematical functions sine and cosine to update the search space positions of solutions, demonstrating promising convergence speed and efficiency in addressing team formation and other optimization problems [3]. Likewise, the JA, renowned for its simplicity and implementation ease, iteratively relocates solutions

within the search space to tackle various optimization challenges, including team formation [4]. Meanwhile, the FFA emulates the behavior of fireflies, adjusting solution locations based on popularity within the search space [5]. This swarm-based approach has confirmed effective in optimizing team building and other optimization tasks [6].

Lastly, PSO, inspired by the social behavior of bird flocking or fish schooling, iteratively improves a population of candidate solutions to seek the optimal solution [7]. Developed in 1995, PSO has demonstrated versatility in solving a wide range of optimization problems, including those encountered in welfare activities. By evaluating the performance of these algorithms, stakeholders can gain insights into their suitability for addressing the complexities of volunteer team formation, informing decision-making to enhance the efficiency and impact of altruistic endeavors.

2 Existing Algorithms

2.1 Sine-Cosine Algorithm (SCA)

In 2016, Seyedali Mirjalili introduced the SCA, a population-based metaheuristic optimization technique inspired by the oscillation and convergence characteristics of sine and cosine functions. The SCA method is particularly adept at solving continuous optimization problems by effectively exploring solution spaces and converging on optimal solutions, even in challenging and multimodal scenarios. By combining exploration and exploitation strategies, SCA strikes a balance between global and local search capabilities, making it a versatile and efficient approach for finding the optimal solutions to complex optimization problems [8].

2.2 Jaya Algorithm (JA)

R. V. Rao introduced the Jaya algorithm in 2016, a population-based optimization method named after the Hindi word “Jaya,” denoting victory. Designed to address continuous optimization challenges, the JA iteratively improves candidate solution populations without the need for complex parameter tuning or extensive computational resources. Its simplicity and efficacy lie in striking a balance between exploration and exploitation, fostering development and cooperation principles. Notably, the JA’s deterministic nature ensures consistent results across various optimization problems, yet its lack of randomness may limit its ability to escape local optima in certain scenarios [9]. Nevertheless, it remains a valuable tool for small to medium-sized continuous optimization problems across diverse domains, including engineering design, machine learning, and data mining [10].

2.3 Firefly Algorithm (FA)

The FA introduced by Xin-She Yang in 2008, is a metaheuristic optimization technique inspired by the social interactions and lighting behaviors of fireflies. Primarily applied to continuous optimization problems, FA aims to find optimal solutions by efficiently exploring complex and multimodal solution spaces [11]. By integrating both exploration

and exploitation tactics, FA strikes a balance between global and local search capabilities, mimicking the attraction and movement patterns of fireflies. Since its inception, various modifications and enhancements have been proposed to enhance FA's performance and applicability across different optimization domains. These adaptations include adaptive parameter tuning, problem-specific adjustments, and hybridization with other optimization methodologies [12, 13]. Widely used in engineering design, image processing, data mining, clustering, and other fields, the FA offers a versatile and user-friendly approach to tackling a diverse range of optimization challenges.

2.4 Particle Swarm Optimization (PSO)

PSO mimics bird swarms' collective behavior to adjust particles' positions representing potential solutions iteratively. This involves particles moving towards the global best and local best positions, optimizing velocity based on factors like speed, distance from optimal position, and relation to global optimum [14, 15]. While PSO efficiently converges to global optima for various problems, premature convergence and insufficient search coverage may occur without proper parameter optimization. Parameters such as particle number, top speed, inertia mass, and acceleration coefficients significantly impact PSO's performance. Techniques like grid search, evolutionary algorithms, and machine learning are proposed for parameter optimization [16]. Despite scalability for solving massive problems through parallel implementations, extremely high-dimensional problems may face slow convergence and high computational costs [17].

PSO applies to function optimization, parameter estimation, and feature selection, aiding control systems, image processing, and machine learning. Its versatility in generating high-quality solutions highlights its utility. However, parameter tuning is crucial to address premature convergence and ensure effective search coverage. With ongoing research, PSO remains valuable for complex optimization challenges in various real-world applications.

3 Related Works

3.1 Data Synthesis

Malaysian Department of Social Welfare (*Jabatan Kebajikan Masyarakat – JKM*) volunteer registration is used as a baseline to create the dataset to the algorithms. Data synthesis methods are: volunteer profile, and performance metrics.

Volunteer Profiles

To start data synthesis, we'll gather comprehensive volunteer profiles from sources like registration forms, covering skills, experience, and preferences. Volunteers will be categorized by factors such as skills, availability, and task preferences, laying the groundwork for diverse and balanced volunteer groups in simulations.

Algorithm Performance Metrics

Precise metrics are vital for evaluating algorithm performance. Metrics will assess JA, SCA, and FFA effectiveness in team creation, focusing on forming diverse teams, optimizing resource allocation, enhancing coordination, and fostering member retention. Collected data on algorithm outputs for volunteer group formation will be systematically analyzed against these metrics, offering valuable insights into each algorithm's strengths and weaknesses in humanitarian efforts.

Metrics will gauge the effectiveness of JA, SCA, and FFA in team creation, focusing on forming diverse teams, optimizing resource allocation, enhancing coordination, and fostering member retention. Data collected on algorithm outputs for volunteer group formation scenarios will be systematically analyzed against these metrics, offering insights into the strengths and weaknesses of each algorithm in humanitarian efforts.

3.2 Development

Identify the Algorithms Attributes and Forms

Identify each algorithms' attributes and forms. Study its behavior and how it works for each event and how suitable it is for this case study and how to implement it. Key of attributes and forms is considered when identifying the algorithms: (i) Understand the problem, (ii) Research existing algorithms, (iii) Study algorithm complexity, (iv) Evaluate algorithm designs, (v) Compare and select, and (vi) Experiment and validate.

Analysis the Algorithms Attributes

Analyzing algorithm attributes involves examining various characteristics of an algorithm to understand its behavior, performance, and suitability for a particular task. Key attributes that being considered when analyzing the algorithm's attributes are: correctness, efficiency, scalability, optimality, and trade-offs.

3.3 Data Design

The JKM volunteer registration form is shown in Fig. 1. Dataset used in our experiment is developed based on this form. Data field name (attributes) is the skills owned by the volunteers in the dataset. This information is used as input that will be extracted by the algorithms. Example, their name, race, religion, district, region, and expertise will be extracted as a parameter to form the team when these is considered as the skill required for a specified event.

BORANG KEAHLIANSUKARELAWAN JKM

Butiran Permohonan :

1 Nama : _____

2 No. Kad Pengenalan : _____ Tarikh Lahir : _____ Umur : _____

3 Jantina : Lelaki Perempuan Regu

4 Bangsa : Melayu India Cina Lain-lain (nyatakan) : _____

5 Agama : Islam Budha Kristian Hindu Lain-Lain: _____

6 Taraf Penduduk : Warganegara Taraf Pendidikan : Sijil Diploma
Ijazah Lain-Lain : _____

Alamat Tetap :

7 Alamat Surat Menyurat :

Alamat Emel :

8 Poskod : _____ Daerah: _____ Dun: _____ Parlimen: _____

9 Pekerjaan / Jawatan : _____

10 No. Telefon : (Telefon Mudah Alih) :
(Rumah) :
(Pejabat) : _____

11 Maklumat Tambahan : Bidang Kemahiran :
Bahasa Pertuturan : Melayu Mandarin Tamil
Inggeris Kantonis Hokkien
Lain-lain : _____

12 Bidang Kerja Komasyarakatan yang diminati : Tandakan (✓) :
Kanak-Kanak Orang Miskin
Warga Tua Remaja / Juvana
OKU Wanita
Keluarga Komuniti

13 Pengalaman Bersama Pertubuhan Sukarelawan (Jika ada) : _____

14 Pihak yang perlu dihubungi jika berlaku kecemasan : _____

15 Alahan : _____

Tandatangan: _____ Tarikh : _____

Fig. 1. Jabatan Kebajikan Masyarakat Volunteer form

3.4 Time Complexity

The time complexity for algorithms used in this study are:

- SCA: $O(n \log(n))$
- JA: $O(m(n))$
- FFA: $O(n^2)$
- PSO: $O(n^2)$

3.5 Testing

For this study, the testing plan used is by a comparative analysis for all algorithms, JA SCA, FFA, and PSO to capture execution time, communication cost and the consistency of the output considering loading all attributes from JKM volunteers application form.

$$\text{Communication cost between person's A and B} = 1 - \frac{\text{Skills } A \cap \text{Skills } B}{\text{Skill } A \cup \text{Skill } B} \quad (1)$$

4 Results and Discussions

Jaya (JA), Sine-Cosine (SCA), Firefly (FFA), and Particle Swarm Optimization (PSO) algorithms are implemented in Java and executed on an Acer Swift 3 laptop running Windows 11. The laptop is equipped with a Ryzen 7 5700U CPU with a clock speed of 2.20 GHz and 16 GB of RAM. The laptop is not connected to a power source and is operating in balanced power mode. This method is tested for five sets of skills, with each set including 10, 20, or 30 distinct skills. The system can be tested for any number of skills between 1 and the entire number of desired unique skills in the dataset.

To ensure fairness in the trials, the JA, SCA, FFA, and PSO algorithms were each executed 10 times for the same population of candidate solutions. The average of the outcomes from each run was then calculated. In order to evaluate the efficiency of these algorithms, a moderately substantial data set that has been organized and purified is used. This dataset consists of 1000 volunteers and encompasses 75 distinct talents.

Table 1. Performance comparison of JA, SCA, FFA, and PSO (Best)

Test Skills	Team Size				Team Cost			
	Best				Best			
	JA	SCA	FFA	PSO	JA	SCA	FFA	PSO
10	7.1	7	7.6	7.2	21.18	20.48	24.41	21.87
20	10.6	10.7	11.1	11.1	50.17	51.07	55.07	55.08
30	14.4	14.3	14.7	14.7	94.92	93.69	99.55	98.97

Table 2. Performance comparison of JA, SCA, FFA, and PSO (Max)

Test Skills	Team Size				Team Cost			
	Max				Max			
	JA	SCA	FFA	PSO	JA	SCA	FFA	PSO
10	8.5	8.4	9.6	8.7	31.43	30.49	40.59	32.9
20	12.8	12.9	14.9	13.3	74.12	74.71	100.67	79.74
30	17.3	17.3	20.7	18	137.98	138.27	198.65	122.9

Table 3. Performance comparison of JA, SCA, FFA, and PSO (Mean)

Test Skills	Team Size				Team Cost			
	Mean				Mean			
	JA	SCA	FFA	PSO	JA	SCA	FFA	PSO
10	7.88	7.82	8.63	8	26.61	26.19	32.46	27.53
20	11.8	11.78	13.12	12.23	62.67	62.37	78.11	67.37
30	15.96	15.82	17.99	16.32	117.54	115.42	150.57	122.9

Table 4. Performance comparison of JA, SCA, FFA, and PSO (Running time)

Test Skills	Average Running Time (ms)			
	JA	SCA	FFA	PSO
10	107.1	69.6	109.1	69.1
20	101.4	65.5	97.4	70.2
30	105.1	63.5	105.5	71.3

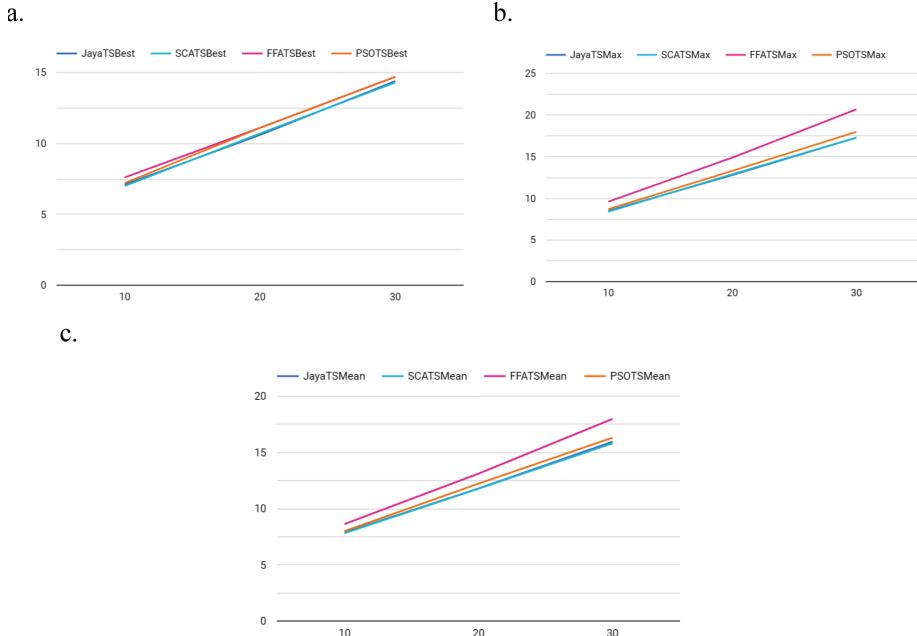
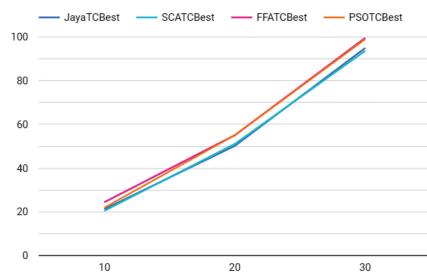


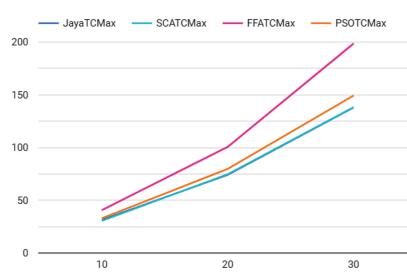
Fig. 2. **a.** Performance comparison of JA, SCA, FFA, and PSO in Team Size (Best), **b.** Performance comparison of JA, SCA, FFA, and PSO in Team Size (Max), **c.** Performance comparison of JA, SCA, FFA, and PSO in Team Size (Mean)

Tables 1, 2, 3, 4 present an empirical comparison of JA, SCA, FA, and PSO for specific talents, displaying average values of team size, team cost, and running time (in ms) for each algorithm with a single objective function: to form a team with the lowest communication cost. Overall, there is a noticeable improvement, with average results approaching optimal levels across all metrics. SCA and PSO typically perform best, with SCA slightly outperforming PSO in most cases. Figures 2, 3, 4 show that the SCA consistently outperforms other algorithms and has the shortest runtime.

a.



b.



c.

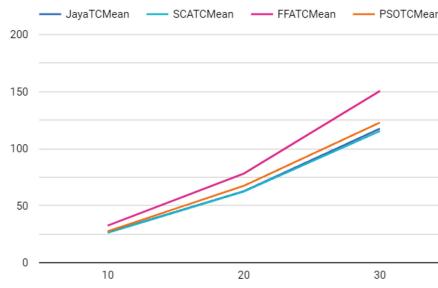


Fig. 3. **a.** Performance comparison of JA, SCA, FFA, and PSO in Team Cost (Best), **b.** Performance comparison of JA, SCA, FFA, and PSO in Team Cost (Max), **c.** Performance comparison of JA, SCA, FFA, and PSO in Team Cost (Mean)

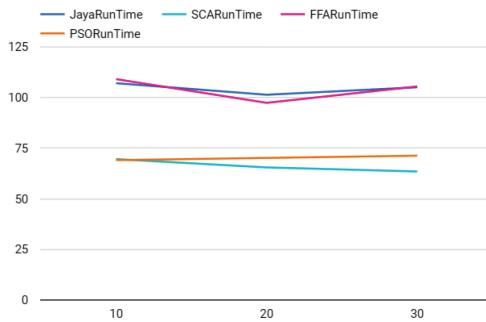


Fig. 4. Performance comparison (Running Time) of JA, SCA, FFA, and PSO

Tables 5, 6, 7, 8 provide an empirical comparison of JA, SCA, FFA, and PSO for specific talents, displaying average values of team size, team cost, and running time (in ms) for each algorithm with two objective functions: to form a team with the least communication cost and to find a suitable team leader with the most skills required.

Table 5. Performance comparison of JA, Sin-Cosine Algorithm, FFA, and PSO (Best) with team leader

Test Skills	Team Size				Team Cost			
	Best				Best			
	JA	SCA	FFA	PSO	JA	SCA	FFA	PSO
10	7.1	7	7.6	7.2	21.18	20.48	24.41	21.87
20	10.6	10.7	11.1	11.1	50.17	51.07	55.07	55.08
30	14.4	14.3	14.7	14.7	94.92	93.69	99.55	98.97

Table 6. Performance comparison of JA, Sin-Cosine Algorithm, FFA, and Particle Swarm Optimization (Max) with team leader

Test Skills	Team Size				Team Cost			
	Max				Max			
	JA	SCA	FFA	PSO	JA	SCA	FFA	PSO
10	8.5	8.4	9.6	8.7	31.43	30.49	40.59	32.9
20	12.8	12.9	14.9	13.3	74.12	74.71	100.67	79.74
30	17.3	17.3	20.7	18	137.98	138.27	198.65	122.9

Table 7. Performance comparison of JA, Sin-Cosine Algorithm, FFA, and Particle Swarm Optimization (Mean) with team leader

Test Skills	Team Size				Team Cost			
	Mean				Mean			
	JA	SCA	FFA	PSO	JA	SCA	FFA	PSO
10	7.88	7.82	8.63	8	26.61	26.19	32.46	27.53
20	11.8	11.78	13.12	12.23	62.67	62.37	78.11	67.37
30	15.96	15.82	17.99	16.32	117.54	115.42	150.57	122.9

Table 8. Performance comparison of JA, Sin-Cosine Algorithm, Firefly Algorithm, and Particle Swarm Optimization (Running time) with team leader

Test Skills	Average Running Time (ms)			
	Time (ms)			
	JA	SCA	FFA	PSO
10	107.1	69.6	109.1	69.1
20	101.4	65.5	97.4	70.2
30	105.1	63.5	105.5	71.3

5 Conclusion

This research explores team formation problems using Jaya (JA), Sine-cosine (SCA), Firefly (FFA), and Particle Swarm Optimization (PSO) methods, selected for their commonality as optimization algorithms. Comparative analysis shows SCA and PSO outperforming in team size and cost, with SCA consistently superior, even with 1 or 2 objective functions. SCA also demonstrates the shortest run times. These findings highlight SCA and PSO's effectiveness, although efficacy may vary based on volunteer skill requirements. This study provides valuable insights into algorithm effectiveness for resolving team formation problems, which can help with algorithm selection for various team formation scenarios.

References

1. Vetukuri, V.S., Sethi, N., Rajender, R.: Generic model for automated player selection for cricket teams using recurrent neural networks. *Evol. Intell.* **14**, 971–978 (2021)
2. Rapp, C.E., Wilson, R.S.: Factors that contribute to trustworthiness across levels of authority in wildland fire incident management teams. *Int. J. Disaster Risk Reduction* (2022)
3. Abualigah, L., Diabat, A.: Advances in sine cosine algorithm: a comprehensive survey. *Artif. Intell. Rev.* **54**, 2567–2608 (2021)
4. Kader, M.A., Zamli, K.Z.: Adopting Jaya algorithm for team formation problem. In: ACM International Conference Proceeding Series. Association Computing Machinery, pp. 62–66 (2020)
5. Yang, X.-S.: Firefly Algorithms. In: *Nature-Inspired Optimization Algorithms*. Elsevier, pp. 123–139 (2021)
6. Gharrad, H., Jabeur, N., Yasar, A.U.H., Galland, S., Mbarki, M.: A five-step drone collaborative planning approach for the management of distributed spatial events and vehicle notification using multi-agent systems and firefly algorithms. *Comput. Netw.* (2021)
7. Shami, T.M., El-Saleh, A.A., Alswaitti, M., Al-Tashi, Q., Summakieh, M.A., Mirjalili, S.: Particle swarm optimization: a comprehensive survey. *IEEE Access* **10**, 10031–10061 (2022)
8. Li, Y., Zhao, Y., Liu, J.: Dynamic sine cosine algorithm for large-scale global optimization problems. *Expert Syst. Appl.* (2021)
9. Wadood, A., Farkoush, S.G., Khurshaid, T., Yu, J.T., Kim, C.H., Rhee, S.B.: Application of the JAYA algorithm in solving the problem of the optimal coordination of overcurrent relays in single- and multi-loop distribution systems. *Complexity* (2019)
10. Zitar, R.A., Al-Betar, M.A., Awadallah, M.A., Doush, I.A., Assaleh, K.: An intensive and comprehensive overview of JAYA algorithm, its versions and applications. *Arch. Comput. Meth. Eng.* **29**, 763–792 (2022)
11. Liu, J., Mao, Y., Liu, X., Li, Y.: A dynamic adaptive firefly algorithm with globally orientation. *Math. Comput. Simul.* **174**, 76–101 (2020)
12. Kumar, V., Kumar, D.: A systematic review on firefly algorithm: past, present, and future. *Arch. Computat. Meth. Eng.* **28**, 3269–3291 (2021)
13. Rigakis, M., Trachanatz, D., Marinaki, M., Marinakis, Y.: Tourist group itinerary design: when the firefly algorithm meets the n-person Battle of Sexes. *Knowl. Based Syst.* **228**, 107257 (2021)
14. Wu, Y., Liu, Y., Li, N., Wang, S.: Hybrid multi-objective particle swarm optimization algorithm based on particle sorting. In: Proceedings of 2021 IEEE International Conference on Emergency Science and Information Technology, ICESIT 2021. Institute of Electrical and Electronics Engineers Inc., pp. 257–260 (2021)

15. Wang, D., Tan, D., Liu, L.: Particle swarm optimization algorithm: an overview. *Soft Comput.* **22**, 387–408 (2018)
16. Yu, V.F., Redi, A.A.N.P., Jewpanya, P., Gunawan, A.: Selective discrete particle swarm optimization for the team orienteering problem with time windows and partial scores. *Comput. Ind. Eng.* (2019)
17. Mapetu, J.P.B., Chen, Z., Kong, L.: Low-time complexity and low-cost binary particle swarm optimization algorithm for task scheduling and load balancing in cloud computing. *Appl. Intell.* **49**, 3308–3330 (2019)



Detection of Paddy Plant Diseases Using Google Teachable Machine

Nor Azuana Ramli¹ , Agus Pratondo² , Sahimel Azwal Sulaiman¹ , Wan Nur Syahidah Wan Yusoff¹ , and Noratikah Abu¹

¹ Centre for Mathematical Sciences, Universiti Malaysia Pahang Al-Sultan Abdullah, Lebuh Persiaran Tun Khalil Yaakob, 26300 Kuantan, Pahang, Malaysia
azuana@umpsa.edu.my

² Selaru lt. 3, School of Applied Science, Telkom University, Terusan Buah Batu, Jl. Telekomunikasi No. 1, Bandung, West Java 40257, Indonesia

Abstract. Malaysia faced a shortage of rice at the end of the previous year. Due to this shortage, the government had to import rice to meet the needs of the people, at the same time affecting the national economy. The shortage of rice can be caused by many factors and one of them is because of the decrease in rice production productivity due to the failure to prevent diseases that affect crop yields earlier. If no control measures are implemented after the infection begins, the disease may cause rice yield losses of up to half of the production. Hence, early detection of these diseases is important for effective management and control strategies. This study aims to develop a model that is able to detect paddy plant disease by using Google Teachable Machine. This model is compared to the model developed using You Only Look Once (YOLO) version 8. As the primary dataset has not been collected yet, this study utilized dataset from the Internet. Overall, our findings highlight the model developed using Google Teachable Machine outperforms the model developed using YOLOv8 in terms of accuracy, simplicity and performance time as it can be completed in half an hour compared to YOLOv8 which took 2.083 h to complete the training. For future study, the model from Teachable Machine will be deployed through mobile applications for disease monitoring using data from drones.

Keywords: paddy plant disease · computer vision · image processing

1 Introduction

Rice is the staple food of Malaysian people. Hence, it is very important for the government to ensure that the market supply of rice is sufficient; otherwise, the government will be forced to import rice from other nations, as happened in October last year [1]. The shortage of rice not only affects individuals but also exerts repercussions on the national economy. There are many factors that cause shortage of rice production and one of them is because of a decrease in productivity. Statistically, paddy production productivity has been decreasing since 2014 as shown in Fig. 1 and the production did not match the target set by the National Agrofood Policy 2021–2030 (NAP 2.0). One of the causes of

the decrease in rice production productivity is the failure to prevent diseases that affect crop yields earlier.

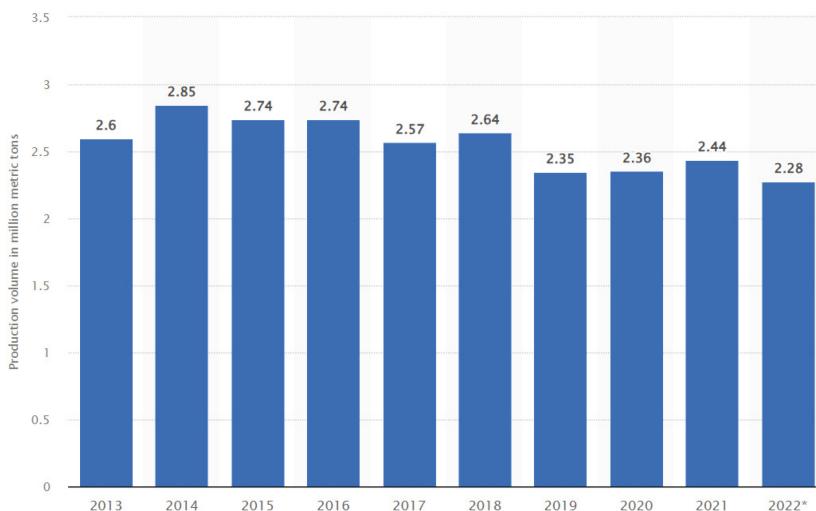


Fig. 1. Paddy production in Malaysia from 2013 until 2022 (Source: Statista.com)

There are many types of diseases involving paddy plants such as bacterial leaf streak (BLS), bacterial blight, bacterial panicle blight (BPB), blast, brown spot, dead heart, downy mildew, false smut, hispa, and tungro. These disease infections are caused by different types of bacteria, fungi, nematodes, and viruses [2]. Based on the interview with one of the farmers in Pahang, if no control measures are implemented after the infection begins, the disease may cause rice yield losses of up to half of the production. Hence, it is important to detect the diseases earlier before the infection begins. Currently, farmers in Kampung Lamir, Pahang are still using a traditional method to monitor the crop starting from seeding until harvesting the paddy. This method is not just impractical for large scale paddy fields but time consuming as well.

There are lots of studies that have been conducted recently involving the development of artificial intelligence (AI) technology in smart agriculture. The increase in the number of studies is not just because of the development of AI technology, but also because of the need for modernization of the agriculture sector to meet the demand. Various methodologies from machine learning have been applied in detecting paddy plant diseases such as support vector machine (SVM), Random Forest, k-nearest neighbor (k-NN), artificial neural network (ANN), convolutional neural network (CNN) and others deep learning methods. Some studies use datasets obtained from the Internet, while some others collected their own dataset using smartphone camera, Internet-of-Things (IoT) or drones. This study utilizes both datasets which are the Internet dataset and dataset collected from the paddy field in Kampung Lamir, Pahang. For this paper, the Internet dataset was used to develop the model since the dataset from the paddy field is not available yet.

The main objective of this study is to develop a model that can detect early signs of diseases involving paddy plants from the Internet dataset using Google Teachable Machine. In order to know whether the model proposed is better than the model developed using YOLOv8, a comparison is made by using accuracy, confusion matrix and computational time. All the methodologies applied in this study are discussed in Sect. 3, while the research gap for this study is highlighted in the next section. Finally, the results obtained are discussed in Sect. 4 before this paper ends with conclusion and some recommendations for the future study in the last section.

2 Literature Review

There is a rising trend in research focused on the early detection of rice diseases. The research area is getting more attention since food security is the main priority now due to the increasing population and shortage of food supply. Recent studies applied deep learning model since it shows a promising result in many fields of computer vision including text detection, medical image recognition, face recognition, image detection and so on. Study done by [3] used hybrid method combining convolutional neural network (CNN) and support vector machine (SVM). The study used dataset collected from the Internet through Mendeley, GitHub, Kaggle and UCI. Results from the study showed the proposed model predicted the paddy disease type and intensity with a 98.43% accuracy and concluded with a statement that the proposed model is reliable and effective in identifying the four levels of severity of bacterial blight, blast, and leaf smut infections in paddy crops.

Another study done by [4] used five machine learning techniques in detecting rice (*Oryza sativa*) disease. Results from this study showed CNN outperforms k- nearest neighbor (k-NN), SVM, Naïve Bayes and Random Forest with 93% accuracy. This study is among other published literature that proves deep learning outperforms traditional machine learning techniques in detecting plant diseases. Furthermore, image detection using deep learning does not need to extract specific features, and only through iterative learning can find appropriate features, which can acquire global and contextual features of images, and has strong robustness and higher accuracy [5]. Most of published studies applied the mature network structures in computer vision to detect and classify plant diseases such as GoogLeNet, AlexNet, VGGNet, ResNet, MobileNet, SqueezeNet, DenseNets and Inception V4. However, this study applies You Only Look Once (YOLO) as a main method since it is stated by [6] that YOLO is considered superior to CNN for certain applications.

This study is different from published literature in terms of its application since there is no study had been conducted using drone dataset in Malaysia. For this first phase, even though the drone dataset has not been applied yet, a comparison between YOLO and MobileNet is done using the Internet dataset, which highlights a contribution to this research area.

3 Methodology

The whole study will be done in two phases, where in the first phase, the proposed model is developed using the Internet dataset while in the second phase, the model will be tested with new data collected from the site visit in Kampung Lamir, Pahang using drones. Although the images will be different in terms of size, focus and quality, the model will be developed again if the model fails to correctly detect the diseases. The whole process in this study is captured as Fig. 2 below.

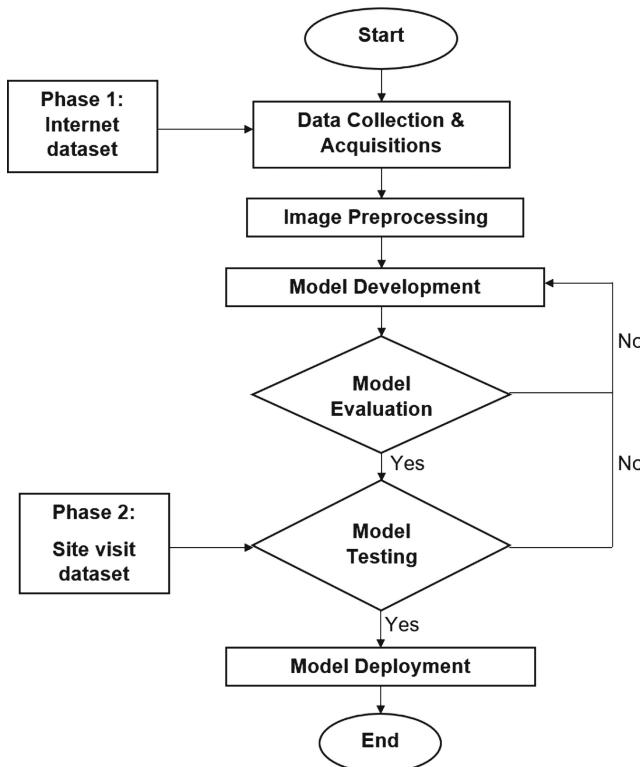


Fig. 2. Overall research framework.

3.1 Image Preprocessing

Image preprocessing is an important step in an image detection project or any other project involving computer vision. It needs to be done to ensure that all the images are formatted correctly, also to decrease the training time and increase the inference speed. There are several preprocessing options such as auto-orient, static crop, resize, grayscale, auto adjust contrast, tile, modify classes and filter null. This dataset applies resize where the image is widened to 1024 pixels (1024×1024). This is a common

practice in image preprocessing, so that the model performance can be improved and to optimize computational efficiency during model development phase later.

Next, the bounding box and labelling are done, where the bounding box is a hypothetical rectangle that creates a collision box for objects and serves as a point of reference for object detection, while labelling in bounding box have label with type of disease infected the paddy plant. There are ten labels for this dataset which are bacterial leaf streak, bacterial blight, bacterial panicle blight, blast, brown spot, dead heart, downy mildew, false smut, hispa, tungro, and normal rice leaves. After labelling, the dataset is partitioned into three parts which are train, test, and validation set. There are 44229 images for training, 5528 images for testing and 6000 images for validation [7].

3.2 Model Development

In model development phase, the data that has been partitioned as training dataset is used to be fed in the model by using YOLOv8 through Roboflow and MobileNet through Teachable Machine. A Google Teachable Machine was utilized in this research to detect the different between 9 different types of diseases in paddy plant and normal paddy plant. It is a free tool and an online platform developed by Google that allows user to generate a deep learning model without coding. According to the source [8], the tool utilizes TensorFlow.js to provide neural network training and inference from inside the browser. The underlying mechanism of the Teachable Machine relies on a widely used deep learning approach known as transfer learning. Although the architecture related to the Teachable Machine could not be found in research papers, a few online sources have proven that the Teachable Machine could be built by using MobileNet in TensorFlow.js through the transfer learning approach. One of the studies, [9] mentioned that Teachable Machine uses MobileNet as the foundation for its transfer learning process. MobileNet is a low-latency, lightweight neural network optimized for tiny devices. The training process is characterized by relatively fast execution times, and it may be started with a smaller number of pictures.

YOLOv8 has been chosen as the comparison method since it is known for its state-of-the-art performance and has been applied in published literature. YOLOv8 uses a complete loss function that combines several elements, such as classification loss, confidence loss, and localization loss to enhance the model's performance. Equation 1 shows the loss function. The loss function penalizes inaccurate predictions and promotes accurate detections by guiding the training process and optimizing the detection performance, where each term contributes to distinct parts of the detection job [10].

$$LOSS = L_{classification} + L_{confidence} + L_{CIoU} \quad (1)$$

There are several new key features that are added in YOLOv8 such as multiple backbone, adaptive training, advanced data augmentation, customizable architecture and pretrained models [11]. The backbone for YOLOv8 is CSPDarknet53, but it provides flexibility to user where user can choose the most suitable model based on the use cases since it has advanced backbone and neck architecture. On top of that, it also supports various backbones such as EfficientNet and ResNet. To increase the model's robustness and generalizability, YOLOv8 uses mosaic data augmentation methods including MixUp

and CutMix. It is highly customizable by allowing the user to modify the model's structure and parameters that suit different applications and transfer learning across several datasets.

In order to enhance generalization, YOLOv8 migrated to anchor-free detection. Pre-defined anchor boxes slow down the learning process for custom datasets, which is the drawback of anchor-based detection. With anchor-free detection, the model directly predicts an object's mid-point and lowers the number of bounding box predictions. This facilitates the acceleration of Non-max Suppression (NMS), a pre-processing stage that eliminates inaccurate forecasts [12]. The optimized balance between accuracy and speed makes YOLOv8 a great choice to be used as a comparison model for this study.

3.3 Model Evaluation

Model evaluation is the phase where the model developed will be assessed based on its performance through metrics like confusion matrix, accuracy, precision, recall, specificity, losses, and error. For this study, confusion matrix and accuracy are used to evaluate the models, while the learning curve is used to examine whether the model is overfitting or not.

4 Results and Discussion

For model development using Teachable Machine, the process started with uploading the dataset based on their classes. Then, the model was trained and tested before it was exported to Jupyter Notebook. Finally, the model was evaluated using accuracy and confusion matrix as shown in Fig. 3. The hyperparameters setting used for the model are similar to YOLOv8 which are epochs set to 5, batch size equal to 64 due to large dataset and learning rate set to 0.001. The results from model training showed that the model is not overfit since it can be seen from the learning curve in Fig. 4. A model that is overfit will have a large gap between the training and test curve.

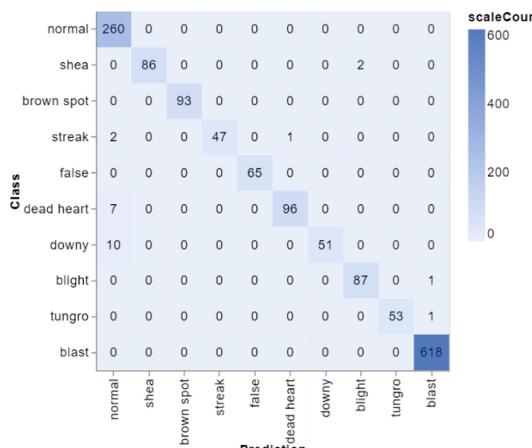


Fig. 3. Confusion matrix based on classes.

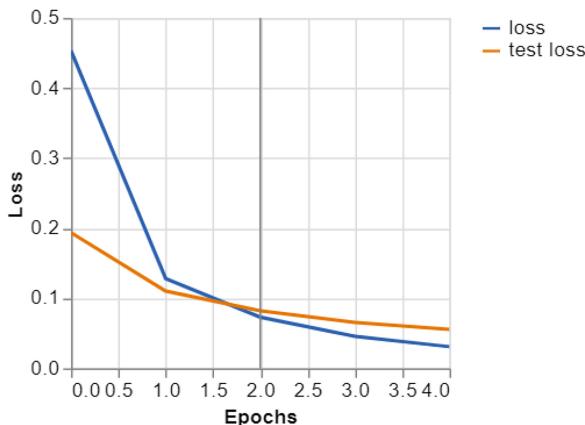


Fig. 4. Train and test loss.

From the accuracy per class tabulated in Table 1, it can be seen that most of the classes achieve high accuracy. In average, the model has high accuracy with 0.97. Besides easy to be implemented compared to YOLOv8, training model using Teachable Machine taking shorter time as it can be completed in half an hour.

Table 1. Accuracy per class using Teachable Machine.

Class	Accuracy	#SAMPLES
Normal	1.00	260
Shea	0.98	88
Brown spot	1.00	93
Streak	0.94	50
False	1.00	65
Dead heart	0.93	103
Downy	0.84	61
Blight	0.99	88
Tungro	0.98	54
Blast	1.00	618

For model development using YOLOv8, the model is not overfit as well as shown in Fig. 5 below. Both train/box loss and validation/box loss are decreasing over time, which indicates that the model is improving its performance as it is trained. Overall, the graph suggests that the model is training well and generalizing well to unseen data.

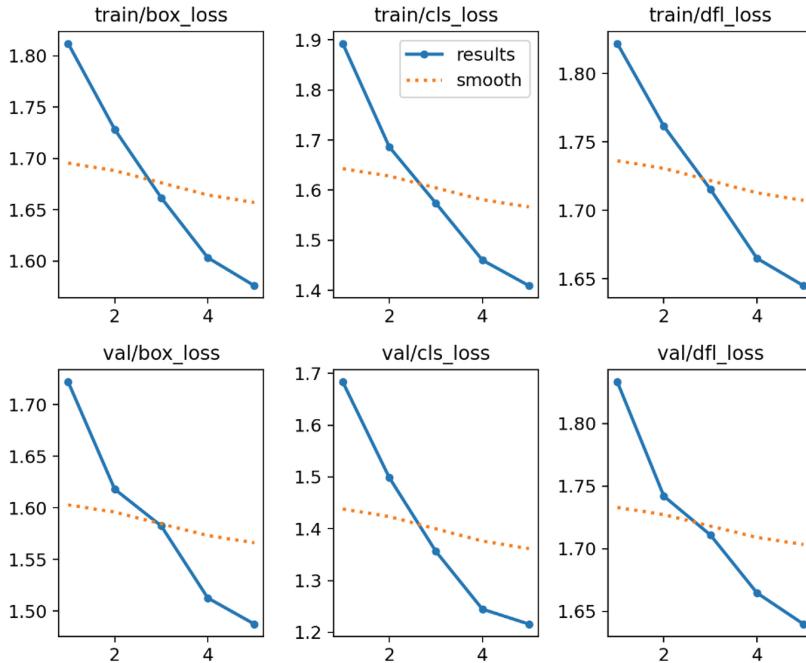


Fig. 5. Train and validation loss plots from YOLOv8.

However, the model was developed using two different classifications only which are normal and diseases. From the confusion matrix as shown in Fig. 6, it can be seen that the model has 100% accuracy. Further evaluation is done through model summary as shown in Fig. 7 and the model achieved a precision of 0.776 and recall of 0.676. The precision for the disease class is higher at 0.827 but with a lower recall of 0.548, while the model performs well in detecting instances of the normal class, with a high precision of 0.725 and recall of 0.804. Generally, the model performs reasonably well across all classes, with notable differences in performance between different categories, although it is not comparable to the model developed using Teachable Machine due to differences in classifications. Although with two classes only, the model still took longer time to complete the whole model training with 2.083 h using Google Collaboratory with extra GPU (Tesla V100-SXM2-16GB, 16151MiB).

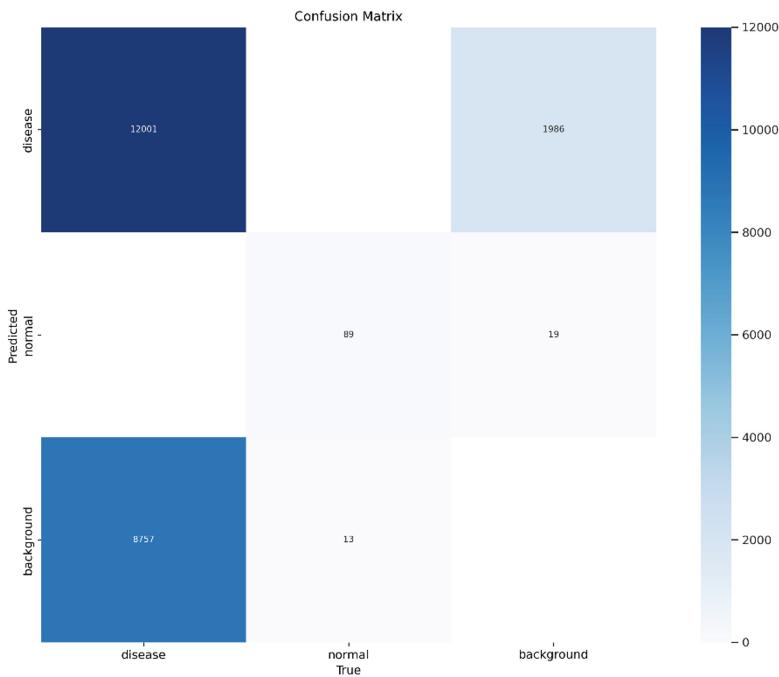


Fig. 6. Confusion matrix based on disease and normal only.

Class	Images	Instances	Box(P)	R
all	6000	20860	0.776	0.676
disease	6000	20758	0.827	0.548
normal	6000	102	0.725	0.804

Fig. 7. Model summary for YOLOv8.

5 Conclusion

The study conducted aimed to assist farmers in Malaysia, particularly in the area of Pekan, Pahang by leveraging modern technology to identify early signs of paddy plant diseases. Currently, disease monitoring relies on labor-intensive methods, which are both time consuming and impractical. The development of this model presented an opportunity to streamline the monitoring process through mobile applications, offering ease and convenience. The initial phase of this study was done in this paper which focused on model development resulting in the creation of two models: one utilizing Google Teachable Machine and the other employing YOLOv8. Through the evaluation of these models, it was determined that the Teachable Machine-based model yielded better results in terms of accuracy and computational efficiency. Based on the results obtained from this study, the next phase will involve collecting dataset from drones and integrating it into the Teachable Machine model. Since YOLOv8 can only detect disease

and normal paddy plants, we also plan to overcome this limitation for future study by detecting specific diseases using this model. Overall, this iterative approach demonstrates a commitment to refining and optimizing the technology for practical application in the field. By advancing this project, it is anticipated that the modernization of technology will not only benefit farmers but also contribute to increased rice production and address food security concerns in Malaysia. This study represents a step towards harnessing the power of technology to address real-world challenges and promote agricultural sustainability.

Acknowledgements. This paper was created within the project “Artificial Intelligence based drone in detecting paddy plant diseases”. Project registration number RDU232714 and UIC231524 under International Matching Grant.

References

1. The Straits Times: <https://www.straitstimes.com/asia/se-asia/malaysia-rice-shortage-will-recur-unless-govt-reforms-supply-chain-ups-crop-yields-analysts>. Last accessed 10 February 2024
2. Sabri, S., Ab Wahab, M.Z., Sapak, Z., Mohd Anuar, I.S.: A review of bacterial diseases of rice and its management in Malaysia. *Food Res.* **7**(Suppl. 2), 120–133 (2023)
3. Lamba, S., et al.: A novel fine-tuned deep-learning-based multi-class classifier for severity of paddy leaf. *Frontiers in Plant Science* **14** (2023)
4. Sobiya, P., Jayareka, K.S., Maheshkumar, K., Naveena, S.: Koppula Srinivas Rao: Paddy disease classification using machine learning technique. *Materials Today: Proceedings* **64**(1), 883–887 (2022)
5. Liu, J., Wang, X.: Plant diseases and pests detection based on deep learning: a review. *Plant Methods* **17**, 22 (2021)
6. ubiAI Website, <https://ubiai.tools/why-yolov7-is-better-than-cnns>. Last accessed 13 February 2024
7. Kaggle Dataset: <https://www.kaggle.com/competitions/paddy-disease-classification/data>. Last accessed 10 February 2024
8. Wiki by ST: https://wiki.st.com/stm32mcu/wiki/AI:How_to_use_Teachable_Machine_to_create_an_image_classification_application_on_STM32. Last accessed 13 February 2024
9. Siddiqui, N.: Creating Deep Convolutional Neural Networks for Image Classification. *Programming Historian* **12** (2023)
10. Hajjaji, Y., Alzahem, A., Boulila, W., Farah, I.F., Koubaa, A.: Sustainable palm tree farming: Leveraging IoT and multi-modal data for early detection and mapping of Red Palm Weevil. *Procedia Computer Science* **225**, 4952–4962 (2023)
11. Mehra, A.: Understanding YOLOv8 Architecture, Applications & Features. Labellerr, <https://www.labellerr.com/blog/understanding-yolov8-architectureapplications-features>. Last accessed 11 February 2024
12. Boesch, G.: A guide to YOLOv8 in 2024. Viso.ai, <https://viso.ai/deep-learning/yolov8-guide>. Last accessed 11 February 2024



Comparative Analysis of ResNet Models for Skin Cancer Diagnosis: Performance Evaluation and Insights

Razan Alharith¹, Ashraf Osman Ibrahim^{2(✉)}, Noorhaniza Wahid³, Rozaida Ghazali³, and Abubakar Elsaifi⁴

¹ School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 610014, Sichuan, China

² Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia
ashrafasman@gmail.com

³ Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn, Parit Raja, Malaysia

⁴ Department of Software Engineering, College of Computer Science and Engineering University of Jeddah, Jeddah, Saudi Arabia

Abstract. The performance of the ResNet-18, ResNet-34, and ResNet-50 models in identifying skin cancer is evaluated using a dataset of 2,357 photographs obtained from the International Skin Imaging Collaboration (ISIC). The photographs portray a range of skin conditions, including both malignant and benign diseases. The models are evaluated based on their capacity to classify the images into nine unique categories representing various skin issues. The dataset undergoes preprocessing procedures, such as resizing, cropping, and normalizing, prior to its utilization for training and assessment purposes. The Stochastic Gradient Descent (SGD) algorithm is employed for the purpose of model optimization, utilizing a learning rate of 0.001 and a momentum of 0.9. Furthermore, the implementation incorporates a learning rate scheduler known as StepLR, which systematically decreases the learning rate by 0.1 units every 7 epochs over a span of 10 epochs. The CrossEntropyLoss function is employed for the purpose of training the models on the dataset, wherein it quantifies the disparity between the predictor and true class probabilities. The findings indicate that ResNet-50 outperforms the other models, achieving an impressive accuracy of 95.3% and a loss rate of 18.5%. Nevertheless, it is important to acknowledge that these findings are derived from a limited representative sample. In order to enhance the credibility and use of forthcoming research endeavors, it is imperative to augment the magnitude and inclusiveness of the dataset. This study highlights the need of employing sophisticated deep learning models, such as ResNet-50, inside clinical environments to enhance the identification of skin cancer and improve patient outcomes. Furthermore, it emphasizes the significance of continuous study and development in the field of deep learning to enhance medical diagnosis.

Keywords: Convolutional neural networks (CNNs) · ResNet Models · Classification · skin cancer detection

1 Introduction

The field of medical image processing and interpretation has transitioned into the era of big data due to rapid advancements in medical imaging technology. This has generated significant interest among both industry professionals and academics, as it allows for the retrieval of crucial information from vast quantities of clinical imaging data. The presence of extensive datasets enhances the basis for clinical disease diagnosis and scientific study in the field of medicine.

Skin cancer is a frequently diagnosed form of cancer that is highly prevalent in the United States. Approximately 20% of the American population is projected to develop skin cancer during their lifetime [1]. Malignant lymphoma is a kind of skin cancer that is particularly worrisome because it causes a substantial number of deaths in the US [2]. Nevertheless, if skin cancer is identified in its first phases, it can frequently be effectively managed via uncomplicated surgical excision. Conversely, a diagnosis made in the later stages is linked to an increased likelihood of death. The 5-year survival rate is expected to be greater than 95% for early-stage detection, while it decreases to less than 20 for late-stage diagnosis [3].

Studies have demonstrated that deeper convolutional neural networks (CNNs) generally show enhanced performance in several image processing and identification tasks. This comprehension has resulted in a tendency to create more intricate and profound neural network structures in order to fulfill the growing requirements of image-related applications [4]. The benefit of deeper networks is in their capacity to acquire hierarchical representations of visual data. With an increase in network depth, each layer becomes capable of capturing and extracting increasingly abstract and high-level properties from the input data. The hierarchical representation facilitates the network's comprehension of intricate patterns and enhances its ability to make precise predictions. CNNs have greatly enhanced image classification tasks. Several alterations to the CNN architecture have been suggested, resulting in a growing number of layers. Notable architectural models in the field of computer vision include AlexNet, GoogLeNet Inception V3, Inception V4, VGG net, Microsoft ResNet, and DenseNets. Nevertheless, these deep networks encounter specific obstacles during the training process. An example of a challenge is the problem of gradients that either explode or vanish. This phenomenon arises when the gradients become too large or small, impeding the network's ability to efficiently update its weights. Consequently, the network may experience a failure to converge or require a prolonged duration for training [5].

ResNet refers to “Residual Networks” as a distinct artificial neural network (ANN) category. A novel architectural design and additional layers are added to classic neural networks [6]. And it uses residual or skip connections [7]. The network can accumulate residual mappings instead of anticipating the expected output since these connections bypass specific levels. This prevents gradients from fading and allows complicated network training. ResNet’s convolutional blocks include residual connections and activation layers like Batch Normalization. The activation layers normalize layer inputs, stabilizing the training process and increasing convergence. This architectural combination has helped ResNet outperform other computer vision algorithms in image categorization. ResNet addresses deep neural network training challenges by integrating residual connections and activation layers, optimizing accuracy and generalization [6].

2 Related Works

Convolutional neural networks (CNNs) can effectively represent visual input hierarchically. The authors demonstrate that CNNs trained holistically without manually created features surpass the current semantic segmentation leader. CNNs are trained to match input and output pixels, capturing complex visual patterns and spatial linkages. CNNs can learn directly from data, allowing them to effectively express semantic segmentation's complex properties. CNNs' hierarchical structure of many convolutional and pooling layers allows these models to collect picture details and context. CNNs' hierarchical structure enables pixel-level predictions that outperform human feature design and other computer vision methods [8].

This paper introduces a ResNet-18-based network model to improve grayscale face expression identification. The model does this by replacing ResNet-18's average pooling layer with a global one and two convolutional layers. This change improves accuracy. Data augmentation methods like random cropping, mirroring, and noise enhancement improve data robustness. The suggested strategy outperforms similar methods on the fer2013 dataset, according to empirical evidence. The paper also indicates areas that need further research and analysis to improve expression detection accuracy [9].

Deep learning employing ResNet architecture is used to identify colorectal cancer in colon gland pictures. The authors evaluate ResNet-18 and ResNet-50 on three data partitions. Evaluation indicators include accuracy, sensitivity, and specificity to aid categorization. The study found that ResNet-50 outperforms ResNet-18 in all areas. ResNet-50 excels when trained on 75% and 80% of data. The results show that the deeper ResNet-50 design detects colorectal cancer better from colon gland images.

It also shows the power of deep learning, specifically the ResNet framework, to detect colorectal cancer in biological images. Using these advanced methods, scientists and healthcare practitioners can improve cancer diagnosis and patient outcomes [10].

ResNet is a deep neural network design intended to address disappearing gradients and performance degradation in deep networks. ResNet uses residual blocks to bypass numerous tiers and directly transmit information. This feature helps ResNet train multi-layer networks efficiently. The study examines ResNet-34, ResNet-50, ResNet-101, and ResNet-152. The deepest architecture is ResNet-152. These algorithms are tested using a dataset to identify cats and dogs in images.

Accuracy and loss are evaluated. Accuracy is the percentage of images correctly identified, whereas loss measures the difference between anticipated and genuine labels. These measures are used to evaluate each ResNet variation on a validation set. The ResNet performance data show a pattern. In particular, the article shows that as ResNet architecture deepens, model accuracy increases. The authors report that ResNet-152 achieves the highest validation set accuracy of 95.29%. ResNet-152 has the lowest loss rate (18.5%), indicating its accuracy in label prediction for cats and dogs [11].

3 Methodology

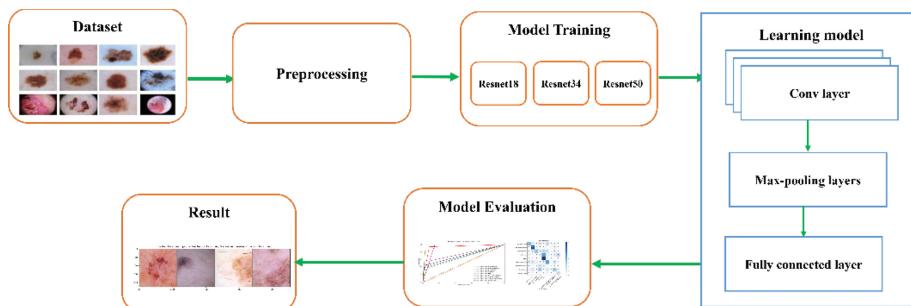


Fig. 1. Methodology

3.1 Datasets

Both malignant and benign oncological diseases are included in the dataset. Furthermore, it is worth noting that, apart from the melanomas and moles categories, which demonstrate a much greater number of photographs, all subgroups within the dataset encompass an equal quantity of images. The equitable allocation of images among various categories, with the exception of melanomas and moles, guarantees that each category is adequately represented in terms of sample size for the purposes of training and evaluation.

The dataset contains the following 9 diseases in both training and testing dataset:

Actinic Keratosis	Pigmented Benign Keratosis
Basal cell carcinoma	Seborrheic Keratosis
Dermatofibroma	Squamous Cell Carcinoma
Melanoma	Vascular Lesion
Nevus	

Each disease category is represented by a series of photographs that are used for training and evaluation inside the research.

3.2 Preprocessing

Preprocessing processes were performed on the dataset utilized in this investigation, involving the arrangement of the 2,357 photos obtained from The International Skin Imaging Collaboration (ISIC) based on the categories provided by ISIC. While explicit mention of specific preprocessing methods is absent, it is customary in deep learning projects to utilize a range of techniques including scaling, cropping, and normalization. These procedures are employed to standardize the photographs and convert them into a consistent format that can be effectively processed by the models (Fig. 1).

3.3 Learning Model

The performance of ResNet in the ILSVRC competition demonstrated its efficacy in the fields of deep learning and image recognition. The breakthrough in ResNet's ability to handle data degradation and maintain accuracy significantly influenced deep learning model development and facilitated the use of more intricate neural network structures. In 2016, ResNet was introduced by Microsoft researchers, marking a significant advancement. This architectural design achieved a remarkable accuracy of 96.4% in the ImageNet Large Scale Visual Recognition Competition (ILSVRC) [12]. ResNet was created to tackle the issues of degradation and precision in deep neural networks, addressing the problem of saturation and precision compromise when the depth of a model is too high [6].

3.4 ResNet Models

The ResNet models are a collection of deep convolutional neural networks, namely comprising ResNet18, ResNet34, and ResNet50. These models utilize a residual learning architecture, allowing for the training of more complex networks compared to earlier methods. The ResNet18 model employs a residual learning architecture, where the layers are redefined as learning residual functions with respect to the layer inputs [6].

The ResNet34 model, a successor to the ResNet series, uses the same residual learning methodology as ResNet18 but has a more complex structure with 35 layers, enhancing its depth and application in various tasks [13].

The ResNet50 model, with 50 layers and a residual learning framework, is an advanced version of the ResNet family used for tasks like remote sensing image classification and brain tumor identification. It achieved a remarkable accuracy of 95.3% when applied to diverse datasets [14, 15].

3.5 Evaluation

The models' effectiveness in detecting skin cancer was evaluated using metrics like validation set and accuracy statistic. Confusion matrices were used to verify accuracy, analyzing predictions for accurate positive, negative, inaccurate positive, and erroneous negative outcomes [16]. Analyzing these factors revealed categories that could cause problems or have high inaccuracy rates, allowing a complete evaluation of the models. ROC curves and confusion matrices assessed model effectiveness. ROC curves show the true positive-false positive ratio across classification criteria. These curves revealed the models' robustness in detecting positive cases and eliminating negative ones [17].

4 Results and Discussion

Table 1 presents the accuracy outcomes of different models, along with their respective training durations. This study examines three models, namely ResNet18, ResNet34, and ResNet50. The training duration for ResNet18 was recorded as 77 min and 3 s, resulting in a training accuracy of 0.6539 and a validation accuracy of 0.5678. The ResNet34

model necessitated a substantially extended duration of 133 min and 11 s for training, however it attained a comparable peak training accuracy of 0.6539. Nevertheless, the minimum validating accuracy achieved was marginally lower, measuring at 0.5339. Finally, it was observed that ResNet50 exhibited the lengthiest duration of training, amounting to 200 min and 13 s. Additionally, it displayed a superior training accuracy of 0.6717, accompanied by a validating accuracy of 0.5508.

Table 1. Model's accuracy result.

Model	Training time	Best training accuracy	Best validating accuracy
ResNet18	77 m and 3 s	0.6539	0.5678
ResNet34	133 m and 11 s	0.6539	0.5339
ResNet50	200 m and 13 s	0.6717	0.5508

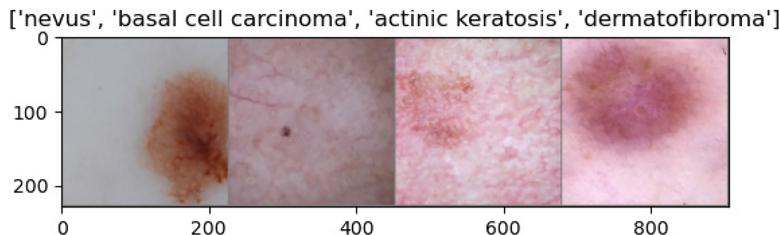
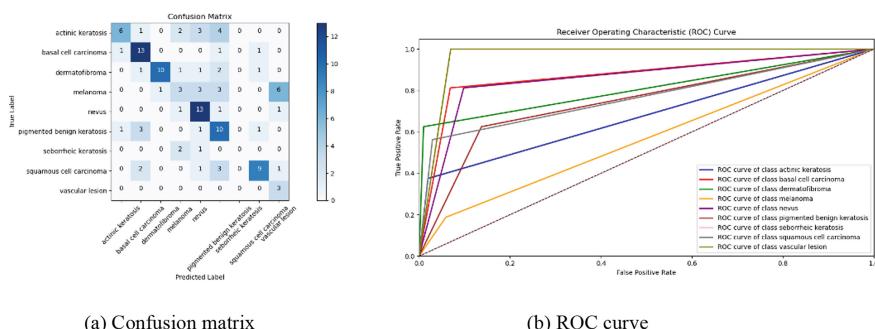


Fig. 2. Image classification



(a) Confusion matrix

(b) ROC curve

Fig. 3. Skin cancer diagnosis using ResNet-18

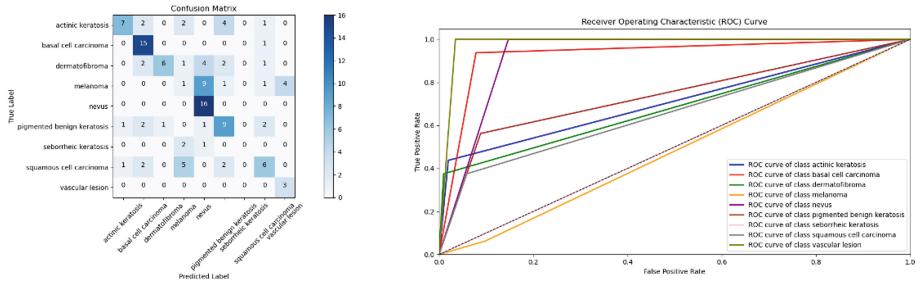


Fig. 4. Skin cancer diagnosis using ResNet-34

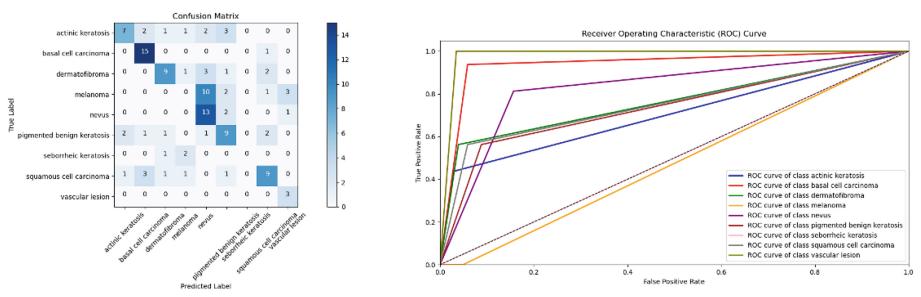


Fig. 5. Skin cancer diagnosis using ResNet-50

The data in our study offers a thorough examination of skin image categorization into nine distinct groups utilizing deep learning models. Figure 2 depicts the process of categorizing skin photos into their appropriate classifications. The objective is to precisely allocate each image to one of the eight categories, encompassing actinic keratosis, basal cell carcinoma, melanoma, and other classifications.

Regarding Figs. 3a, 4a, and 5a, we provide a confusion matrix that displays the actual labels and anticipated labels for a classification model. The rows in the table correspond to the actual labels, whereas the columns correspond to the predicted labels. The numbers in the table represent the frequency of occurrences in each category. The diagonal members of the matrix indicate accurate forecasts, whereas the off-diagonal elements indicate inaccurate predictions. Our investigation indicates that the model encounters difficulties in accurately differentiating between actinic keratosis, melanoma, and seborrheic keratosis, as it misclassifies certain occurrences of these classifications. Nevertheless, the model has high efficacy in accurately categorizing basal cell carcinoma, dermatofibroma, pigmented benign keratosis, and squamous cell carcinoma.

Each Figs. 3b, 4b, and 5b displays the Receiver Operating Characteristic (ROC) curves, which offer insights into the effectiveness of a binary classifier system. More precisely, we analyze the effectiveness of a system for categorizing skin lesions. The results of our study show that melanoma has the highest accuracy, with a True Positive

Rate (TPR) of 1.0 at a False Positive Rate (FPR) of around 0.15. This demonstrates an exceptional degree of precision in detecting cases of melanoma. The TPR for seborrheic keratosis is approximately 0.95, with an FPR of 0.2. Nevertheless, actinic keratosis and pigmented benign keratosis exhibit somewhat lesser effectiveness, with true positive rates (TPRs) of around 0.6 and 0.5, respectively, at a FPR of 0.2. This indicates a greater frequency of inaccurate negative outcomes. The ROC curve facilitates the identification of the best operating point by considering certain criteria, such as prioritizing a low false positive rate or a high true positive rate.

In the second part of the discussion, the table displays ROC curves that present TPR and FPR values at various discrimination levels. Examining the ROC curve specifically for the melanoma class, we note that when the discrimination threshold is set at 0.4, the TPR achieves a value of 1.0, while the FPR remains approximately 0.15. By establishing a threshold of 0.4, the classifier is able to accurately detect all instances of melanoma while maintaining a low proportion of false positives. Similarly, the ROC curve for seborrheic keratosis shows that when the discriminating threshold is set at 0.2, the TPR is roughly 0.95, whereas the FPR is around 0.2. Thus, by establishing a threshold of 0.2, the classifier can effectively detect the majority of seborrheic keratosis instances while maintaining a reasonable false positive rate.

5 Conclusion

The study showcases the efficacy of ResNet-18, ResNet-34, and ResNet-50 models in the field of dermatology, showcasing their ability to effectively detect skin cancer through the utilization of deep learning methodologies. ResNet-50 outperforms other models, with a remarkable accuracy rate of 95.3% and a loss rate of 18.5%. This method demonstrates notable efficacy in the identification of melanoma and seborrheic keratosis, which are prevalent dermatological disorders. Nevertheless, it is crucial to acknowledge that these findings are derived from a restricted dataset including 2,357 photographs. In order to further improve the precision of the ResNet-50 model, it is recommended that future investigations prioritize the expansion and diversification of the dataset. By implementing this approach, the performance of the model can be enhanced and rendered more dependable. The findings of this work indicate that the incorporation of machine learning methodologies, specifically ResNet models, within clinical environments has the potential to greatly improve the precision of skin cancer diagnosis. Consequently, this can enhance the process of decision-making and accelerate the delivery of medical care. The results highlight the capacity of artificial intelligence to revolutionize healthcare and enhance patient results when integrated with the proficiency of medical practitioners.

Acknowledgement. This research was supported by Universiti Tun Hussein Onn Malaysia.

References

1. Oliveira, R.B., Papa, J.P., Pereira, A.S., Tavares, J.M.R.: Computational methods for pigmented skin lesion classification in images: review and future trends. *Neural Comput. Appl.* **29**, 613–636 (2018)

2. Rogers, H.W., Weinstock, M.A., Feldman, S.R., Coldiron, B.M.: Incidence estimate of non-melanoma skin cancer (keratinocyte carcinomas) in the us population, 2012. *JAMA Dermatol.* **151**(10), 1081–1086 (2015)
3. Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M., Thrun, S.: Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **542**, 115–118 (7639) (2017)
4. Ruiz, P.: Understanding and visualizing densenets-towards data science (2018)
5. Too, E.C., Yujian, L., Njuki, S., Yingchun, L.: A comparative study of finetuning deep learning models for plant disease identification. *Comput. Electron. Agric.* **161**, 272–279 (2019)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
7. Raiko, T., Valpola, H., LeCun, Y.: Deep learning made easier by linear transformations in perceptrons. In: Artificial Intelligence and Statistics, PMLR, pp. 924–932.9 (2012)
8. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp. 3431–3440 (2015)
9. Zhou, Y., Ren, F., Nishide, S., Kang, X.: Facial sentiment classification based on resnet-18 model. In: 2019 International Conference on Electronic Engineering and Informatics (EEI), IEEE, pp. 463–466 (2019)
10. Sarwinda, D., Paradisa, R.H., Bustamam, A., Anggia, P.: Deep learning in image classification using residual network (resnet) variants for detection of colorectal cancer. *Procedia Comput.r Sci.* **179**, 423–431 (2021)
11. P. Nagpal, P., Bhinge, S.A., Shitole, A.: A comparative analysis of resnet architectures. In: 2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), IEEE, pp. 1–8 (2022)
12. Alotaibi, B., Alotaibi, M.: A hybrid deep resnet and inception model for hyperspectral image classification, PFG-Journal of Photogrammetry. *Remote Sens. Geoinf. Sci.* **88**(6), 463–476 (2020)
13. Gao, M., Qi, D., Mu, H., Chen, J.: A transfer residual neural network based on resnet-34 for detection of wood knot defects. *Forests* **12**(2), 212 (2021)
14. Sahaai, M.B., Jothilakshmi, G., Ravikumar, D., Prasath, R., Singh, S.: Resnet-50 based deep neural network using transfer learning for brain tumor classification. In: AIP Conference Proceedings, vol. 2463, AIP Publishing (2022)
15. Wang, L., Chen, Y., Wang, X., Wang, R., Chen, H., Zhu, Y.: Research on remote sensing image classification based on transfer learning and data augmentation. In: International Conference on Knowledge Science, Engineering and Management, Springer, pp. 99–111 (2023)
16. Fawcett, T.: An introduction to roc analysis. *Pattern Recogn. Lett.* **27**(8), 861–874 (2006)
17. Powers, D.M.: Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation, arXiv preprint [arXiv:2010.16061](https://arxiv.org/abs/2010.16061) (2020)



The Predictive Modelling of Student Academic Performance Using Machine Learning Approaches

Nurul Habibah Abdul Rahman^{1,2} , Sahimel Azwal Sulaiman² , and Nor Azuana Ramli²

¹ Department of Business, Faculty of Management and Informatics, Universiti Islam Pahang Sultan Ahmad Shah, Jalan Gambang, 25150 Kuantan, Pahang, Malaysia
sahimel@umpsa.edu.my

² Centre for Mathematical Sciences, Universiti Malaysia Pahang, Lebuh Persiaran Tun Khalil Yaakob, 26300 Kuantan Pahang, Malaysia
ahimel@umpsa.edu.my

Abstract. In the field of education, machine learning techniques have been applied in numerous studies covering a wide range of topics, including student enrollment, graduation forecasts, failure rates, retention, and academic performance. The use of predictive analytics in machine learning offers valuable insights to educators, potentially aiding in the improvement of students' outcomes through the analysis of historical data. However, based on a review of existing literature, research focusing on the use of machine learning and predictive analytics to enhance student performance in Malaysian higher education remains limited, specifically among Islamic Universities. The primary objective of this study is to create the most effective predictive model for forecasting students' final grades by employing machine learning methods such as Multinomial Logistic Regression, Decision Trees, Random Forest, K-Nearest Neighbors, Naïve Bayes, and Support Vector Machine. This research utilizes a dataset comprising student records from the Business Statistics course at Universiti Islam Pahang Sultan Ahmad Shah, spanning from 2013 to 2022. The findings reveal that the Decision Tree model is the most accurate, with a 0.60 accuracy rate in predicting students' performance levels. This optimal model is instrumental in enabling lecturers to identify students at risk of failing at an early stage.

Keywords: Machine learning · Predictive models · Students' performance · Education

1 Introduction

Predictive analytics is a subset of data engineering focused on forecasting future events or outcomes using historical data sets. This analytical process suggests the anticipation of future occurrences by constructing predictive models through the application of data mining (DM) and machine learning (ML) techniques. DM is dedicated to extracting insights from vast amounts of historical data, addressing the question of how to

effectively leverage past information to identify patterns and improve decision-making processes [1]. Meanwhile, ML is recognized as a scientific discipline that explores the methodology behind enabling machines to learn or acquire knowledge from experiences [2]. The integration of ML and DM methods in predictive analytics is vital for developing tools that can create precise predictive models. These models are shaped based on the data utilized and the aim of the predictive analysis, which involves either regression or classification strategies [3].

The field of predictive analytics has seen a surge in interest within the higher education sector due to its ability to furnish educators with valuable insights, potentially aiding in the enhancement of student achievement. Through predictive analytics, educators are empowered to develop effective strategies aimed at improving student performance, mitigating dropout rates, and bolstering student retention [4]. Common predictive targets in higher education include student performance, the likelihood of failing a course, dropout risks, grade forecasts, and graduation rates [5]. Universities maintain their own database systems, which provide lecturers and academic administrators access to student data. Thus, it is imperative for these institutions to leverage their databases optimally to inform decision-making processes, particularly in areas concerning academic enhancement for students.

Student dropout rates present a significant challenge for Higher Education Institutions (HEIs). A report by the New Straits Times on August 2, 2022, highlighted in the Dewan Rakyat that there was an increase in dropouts among undergraduate students in 2021. Over 4,000 students failed to finish their studies, bringing the total number of student dropouts in 2021 to 17,613. High dropout rates can negatively affect the reputation of the respective universities, as they may be perceived as less capable of retaining and graduating students. Moreover, students leaving their studies prematurely results in a considerable loss of human capital for the country. Public universities, in particular, face the consequence of producing a smaller quantity of professionals and experts, which can hinder national development and progress [6].

Evaluating the performance of various ML models is critical to identifying the most effective one for predicting students' performance [9]. Among the ML techniques applied for this purpose are the Random Forest (RF), decision tree (DT), support vector machine (SVM), Naïve Bayes (NB), k-nearest neighbor (k-NN), and logistic regression (LR). One study, [4] utilized DT, RF, SVM, and LR classifiers to predict student performance based on grades, finding that the DT classifier achieved the highest accuracy, at 99.6%. In another instance, [10] employed DT, NB, SVM, and k-NN classifiers to create a predictive model for student grades using a dataset of only 197 entries. This study determined that SVM was the superior model, boasting the highest accuracy.

The utilization of ML methods to forecast student performance is on the rise within both public and private universities in Malaysia. For instance, [7] conducted predictive analysis research on the GPA of 59 students from Universiti Teknologi MARA (UiTM), Terengganu. Similarly, [8] utilized data from 141 University of Technology Sarawak students to develop predictive models for grading using ML techniques. However, literature review suggests that this approach has yet to be employed in Islamic universities. Students attending Islamic universities and those at public universities typically possess slightly different cultural and educational backgrounds. Many students

at Islamic universities may not excel in subjects related to science and mathematics. Consequently, establishing a predictive model to identify potential academic struggles early on is crucial.

Initial predictions of students' grades, based on their cumulative marks, previous achievements, and other relevant factors, can assist both students and lecturers in taking proactive measures to enhance academic performance prior to final examinations. This facilitates students in preparing and improving their cumulative marks, potentially leading to enhancements in their final grades. Consequently, this study focused to suggest a predictive model for forecasting students' performance levels based on their final grades in chosen courses.

2 Methodology

This section outlines the methodology employed to fulfill the primary objective of the study. It is crucial to experiment with various ML models and come out with the best model. The study involved several phases of ML techniques, as depicted in the ML Pipeline (refer to Fig. 1). The dataset utilized was obtained from the Learning Management System (LMS) of Universiti Islam Sultan Ahmad Shah (UnIPSAS), and the variables were meticulously analyzed for their perceived relevance based on existing literature.

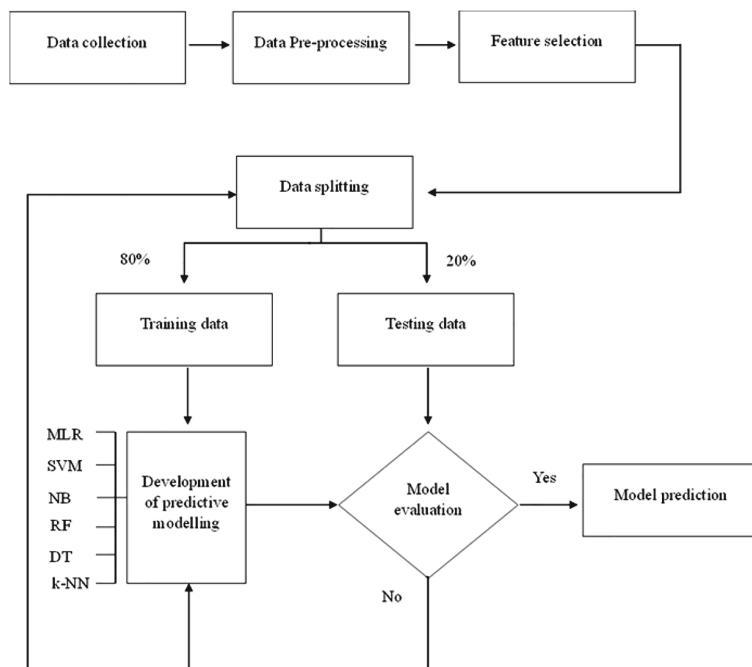


Fig. 1. Machine Learning Pipeline

2.1 Preparing and Preprocessing Dataset

The data preparation phase comprises two crucial steps namely data collection and data preprocessing. Initially, the necessary data was gathered and imported into the Python software. The dataset comprises real data from students enrolled in four Diploma programs, namely Accounting, Business Studies, Marketing, as well as Finance and Banking at UnIPSAS, spanning from June 2013 to June 2022. Specifically, the dataset focuses on 450 students who took the Business Statistics (BS) course during the fourth semester. As delineated in Table 1, the attributes include gender, students' intake programs, Cumulative Grade Point Average (CGPA) from the semester preceding the BS course, Business Mathematics (BM) grades, total cumulative/carry marks, with the target variable being students' performance levels according to their final grades in BS subject.

Table 1. List of attributes

Attributes	Types	Detail	Encode Value
Gender	Categorical	Male	1
		Female	0
Intake	Categorical	First intake	1
		Second intake	2
Students' program	Categorical	Diploma in Accounting	0
		Diploma in Business Studies	1
		Diploma in Finance and Banking	2
		Diploma in Marketing Management	3
CGPA	Numerical	1.0–4.00	-
Carry marks	Numerical	0–50	-
BM's grades	Categorical	A, A–, B+, B, B–, C+, C, C–, D+, D, E	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
Performance Levels	Categorical	Excellent (A, A–)	1
		Very good (B+, B)	2
		Good (B–, C+)	3
		Pass (C, C–)	4
		Weak (D+, D)	5
		Fail (E)	6

The preprocessing of data was carried out using Python, encompassing tasks such as data cleaning, encoding, and handling outliers. Outliers refer to data points that significantly deviate from the rest and can distort the distribution of data, potentially impacting the performance of the model and leading to inaccurate predictions.

In the realm of ML modelling, feature selection involves the steps of identifying relevant features or variables to be utilized in assessing the predictive models. Typically, for supervised ML techniques, the relevance of variables is assessed based on their correlation with the outcome, it could be either categorical or numerical [11]. As per Taylor (1990), the closer the correlation value is to ± 1.0 , the stronger the correlation between two variables. Hence, variables exhibiting a substantial correlation value (± 0.70 to ± 1.00) are identified as factors influencing students' performance levels.

The method for identifying correlation coefficients depends on the types of attributes utilized in ML model development. For categorical attributes, Cramer's V correlation proves valuable in assessing correlation strength, particularly when both attributes are significant, achieved through the Chi-square test. Conversely, Spearman's rank correlation coefficient is employed to ascertain the strength and direction of monotonic correlation between two attributes, especially when one or both are on an ordinal scale. Spearman's rank is particularly suitable for ordinal and numerical data, as it ranks attributes in preference order, providing valuable insights into their relationships.

2.2 Predictive Modelling

The foundation of all validation strategies lies in data partitioning, wherein the dataset is partitioned into training and testing sets. The model undergoes training using the training dataset, while evaluation is conducted on the test dataset. In this study, the Train Test Split method was utilized due to its simplicity and effectiveness in ML model development. Specifically, the initial 80% of the dataset was allocated for training data, while the remaining 20% was designated for testing data.

Subsequently, this research proposes the utilization of supervised models, namely multi-nomial logistic regression (MLR), SVM, RF, DT, k-NN, and NB. The performance of each model was assessed, including hyperparameter tuning, whereby the hyperparameters for each model were adjusted accordingly until optimal performance was attained.

- Multinomial logistic regression: Applicable to predict more than two classes of outputs using the Softmax function:

$$P(y = K|x) = \frac{e^{z_K}}{\sum_{i=1}^K e^{z_i}}$$

In this context, the logit (raw score) for K is represented as z_K , where e denotes the base of Euler's number, and the divisor is the summation of e^{z_i} where i ranges from 1 to K .

- Support vector machine: To identify a hyperplane in an N-dimensional space that effectively separates the data points into distinct classes. And trained using the cost function as follows:

$$\theta = C \sum_{i=1}^m \left[y^{(i)} \cos_1(\theta^T x^{(i)}) + (1 - y^{(i)}) \cos_0(\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{i=1}^n \theta_i^2$$

The functions cost_1 and cost_0 correspond to the cost associated with an example where $y = 1$ and $y = 0$ respectively.

- Decision tree: Depicts the various consequences of a set of choices. The hyperparameters that were adjusted or optimized were the Gini index and entropy:

$$\text{Gini Index: } I_G(t) = 1 - \sum_{i=1}^K P_i^2$$

$$\text{Entropy: } H(t) = - \sum_{i=1}^K P_i \log_2(P_i)$$

- Random Forest: An extension of the decision tree method involves creating a large number of trees for the model instead of just a single tree
- k-nearest neighbor: Computing the distance between an inquiry and each example in the dataset, selecting the k nearest to the query, and then determining the most frequent label through voting.
- Naïve Bayes: Probabilistic classifier based on Bayes' theorem:

$$P(y | x_1, x_2, \dots, x_n) = \frac{P(y) \times P(x_1 | y) \times P(x_2 | y) \times \dots \times P(x_n | y)}{P(x_1) \times P(x_2) \times \dots \times P(x_n)}$$

where:

$P(y | x_1, x_2, \dots, x_n)$ is the rear probability of class y with x_1, x_2, \dots, x_n as attributes. $P(y)$ is the previous probability of class y , $P(x_i | y)$ is the probability of attribute x_i with class y , and $P(x_i)$ is the previous probability of attribute x_i .

The model's performance was assessed using various performance metrics, including accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic (ROC) curve (AUC). In the case of multi-class models, the One-vs-Rest setting was employed to compute the AUC. Accuracy can be defined as the degree of similarity between expected and actual values of a quantity [11]. The interpretation of accuracy, precision, recall, F1-score, and AUC is as follows:

- A score below 0.5 suggests poor performance.
- Scores ranging from 0.5 to 0.7 indicate moderate to good performance.
- Scores exceeding 0.8 indicate excellent performance.
- A score of 1.0 indicates perfect performance.

3 Results and Discussion

This section elaborates on each result derived from the model development process conducted. Outlier values were identified through boxplot diagrams, following which extreme values were substituted with the nearest non-extreme value. Figure 2 illustrates the boxplot for carry marks values both before and after outlier treatment.

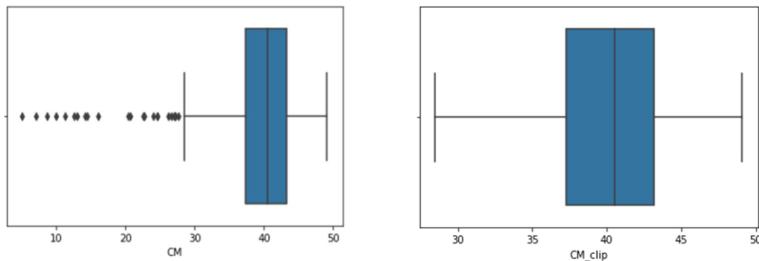


Fig. 2. The boxplot before and after treating the outliers.

According to the results from Cramer's V, three variables exhibited moderate associations with students' performance levels: gender, course, and mathematics grade. On the other hand, Spearman's Rank results indicate a strong negative correlation between carry marks and students' performance levels. This suggests that higher carry marks correspond to lower performance level codes, indicating higher student performance (refer to Table 1). Additionally, there was no correlation found between student intake and performance levels, leading to its exclusion from the list of attributes for predicting performance levels. Furthermore, all predictors or independent variables demonstrated weak and moderate relationships among themselves, indicating the absence of multicollinearity among attributes. Table 2 presents the correlation coefficient results between independent variables (predictors) and the target variable.

Table 2. Correlation coefficient between predictors and target variable

Predictors	Method	Results
Gender	Cramer's V	0.249
Course	Cramer's V	0.216
Intake	Cramer's V	0.0998
Math Grade	Cramer's V	0.239
CGPA	Spearman Rank	-0.543
Carry marks	Spearman Rank	-0.74

In Python, the accuracy, precision, recall, and F1-score of the models were obtained using classification reports. Since the proposed models were trained to predict six classes (levels) of performance, the AUC was evaluated by calculating the average AUC value across all classes. The ROC curve for each model is depicted in Fig. 3, and a summary of the results comparison is provided in Table 3.

Table 3. Comparison of all classification models' performances

ML Model	Accuracy	Precision	Recall	F1-score	Average AUC
MLR	0.56	0.50	0.48	0.49	0.865
DT	0.60	0.52	0.52	0.51	0.66
SVM	0.52	0.51	0.48	0.47	0.863
RF	0.56	0.63	0.46	0.40	0.833
NB	0.51	0.39	0.45	0.39	0.83
k-NN	0.54	0.48	0.36	0.36	0.763

Table 3 indicates that the DT model achieved the highest accuracy value of 0.60, outperforming other models, which attained moderate accuracy ranging between 0.51 and 0.56. Moreover, the DT model also demonstrated the highest recall and F1-score values compared to the other models. Consequently, the DT model is deemed the most suitable for predicting students' performance levels in Business Statistics (BS) subjects due to its superior accuracy. Additionally, the DT model consistently maintained moderate to good scores across other performance metrics. In contrast, by examining the AUC scores for each class, insights into the model's relative predictive performance for different performance levels can be gained. Figure 4 shows that there were four models, namely the MLR, DT, RF, and NB, have achieved 1.00 score of AUC for distinguishing class 6 (fail). This indicates that these models excel in accurately identifying instances of failing students.

Finally, a thorough analysis was performed on the new data of students who completed the BS course in the latest semester, and predictions were made based on their performance levels in the final exam. Visual inspection indicates that the trends from the predicted values and actual values are generally comparable (refer to Fig. 5). Upon examination, it was observed that eleven students out of 22 students received final grades as predicted, resulting in an accuracy score of 0.5.

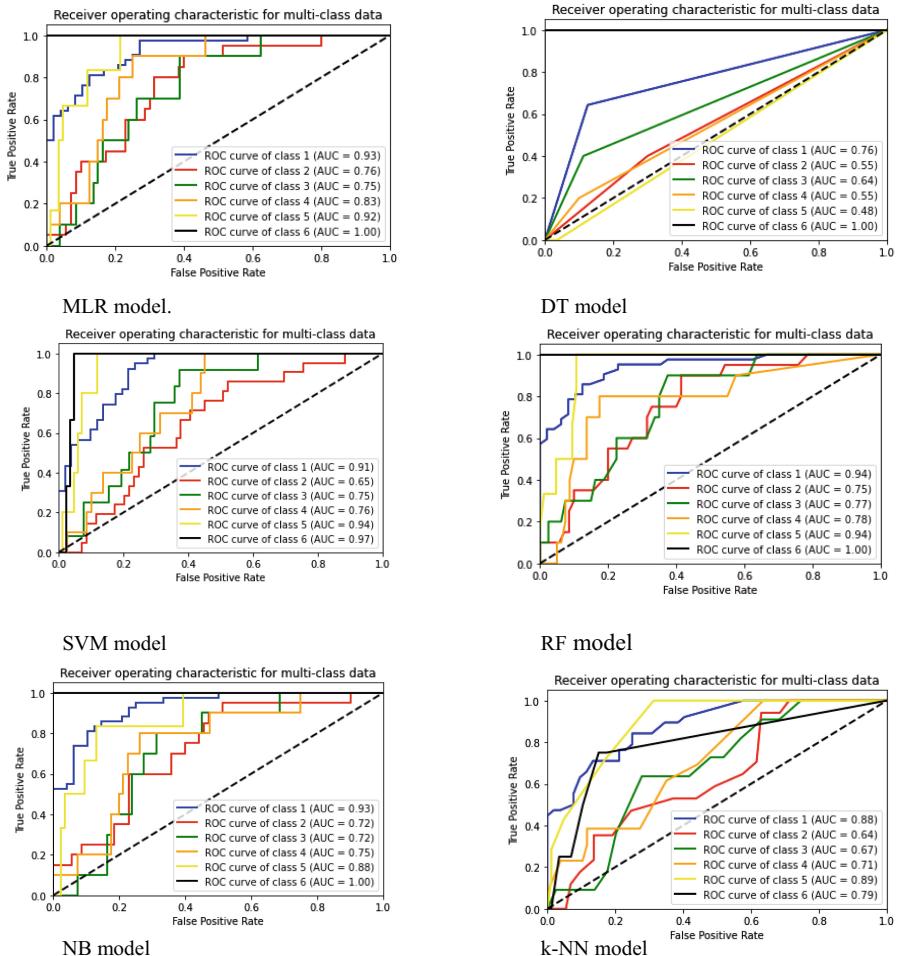


Fig. 3. The ROC curve for the proposed ML models



Fig. 4. The AUC of ROC curve for six performance level

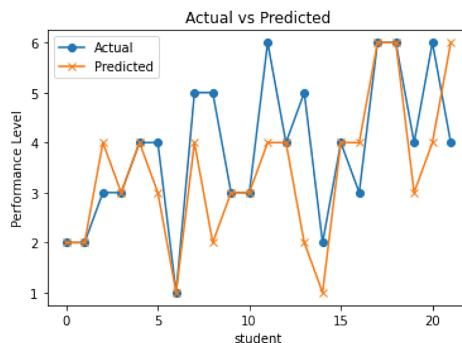


Fig. 5. Actual versus predicted data for students' performance levels

4 Conclusion

In conclusion, this research successfully achieved its primary objective, which was to evaluate the six machine learning models and choose the best model to predict students' performance levels in the Business Statistics (BS) subject at UnIPSAS. Based on the results, the Decision Tree (DT) emerged as the best predictive model with an accuracy of 0.60. Given the context of this study, which focuses on educational development rather than critical fields like medical or science, the DT model's performance is satisfactory. However, it's essential to acknowledge the limitations inherent in the data sources and attributes used in this study, which may have constrained the model's performance to some extent.

Each developed predictive model should serve a practical purpose for the relevant department. In this case, the predictive model for students' performance levels in the BS subject can be beneficial for lecturers at UnIPSAS, particularly those within the Faculty of Management and Informatics, where subjects like statistics and mathematics are taught. Students predicted to obtain an E grade (level 6) should receive special attention and intervention. Based on the findings, it's evident that students' carry marks are significantly related to their performance in the BS subject. Therefore, interventions such as retaking progress tests or completing additional coursework could be recommended to improve their carry marks and, subsequently, their overall performance.

In the future, it is recommended to augment the dataset with additional data from students taking the subject to further enhance the model's accuracy. Re-evaluation of the model, exploration of alternative approaches, and consideration of additional features or techniques are crucial steps to enhance model's accuracy and performance. Additionally, integrating the algorithm into the LMS along with a dashboard interface would facilitate easier analysis in the future. This would streamline the process of accessing and utilizing the predictive model, enabling educators and administrators to make data-driven decisions more effectively.

Acknowledgment. The authors would like to express their sincere gratitude to Universiti Malaysia Pahang Al-Sultan Abdullah for providing financial support for this research project. (RDU230378).

References

1. Mitchell, T.M.: Machine learning and data mining. *Commun. ACM* **42**(11), 30–36 (1999)
2. Kavakiotis, I., Tsavos, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., Chouvarda, I.: Machine learning and data mining methods in diabetes research. *Comput. Struct. Biotechnol. J.* **15**, 104–116. Elsevier B.V (2017)
3. Mishra, N., Silakari, D., ProudYogiki Vishwavidyalaya, G., Sc, C., Gandhi ProudYogiki Vishwavidyalaya, R.: Predictive analytics: a survey, trends, applications, opportunities and challenges (2012)
4. Abdul Bujang, S.D., Selamat, A., Krejcar, O.: A Predictive analytics model for students grade prediction by supervised machine learning. *IOP Conf. Ser.: Mater. Sci. Eng.* **1051**(1), 012005 (2021). <https://doi.org/10.1088/1757-899x/1051/1/012005>
5. Tatar, A.E., Düztegör, D.: Prediction of academic performance at undergraduate graduation: course grades or grade point average? *Appl. Sci. (Switz.)* **10**(14) (2020). <https://doi.org/10.3390/app10144967>
6. Sani, N.S., Nafuri, A.F.M., Othman, Z.A., Nazri, M.Z.A., Nadiyah Mohamad, K.: Drop-Out Prediction in Higher Education Among B40 Students. *Int. J. Adv. Comput. Sci. Appl.* **11**(11), 550–559 (2020). <https://doi.org/10.14569/IJACSA.2020.0111169>
7. Ahmad, N., Hassan, N., Jaafar, H., Enzai, N.I.M.: Students' performance prediction using artificial neural network. *IOP Conf. Ser.: Mater. Sci. Eng.* **1176**(1), 012020 (2021). <https://doi.org/10.1088/1757-899x/1176/1/012020>
8. Razali, M.N., Zakariah, H., Hanapi, R., Rahim, E.A.: Predictive model of undergraduate student grading using machine learning for learning analytics. In: Proceedings - 2022 4th International Conference on Computer Science and Technologies in Education, CSTE 2022, pp. 260–264 (2022). <https://doi.org/10.1109/CSTE55932.2022.00055>
9. Ofori, F., Gitonga, R.: Using Machine Learning Algorithms to Predict Students' Performance and Improve Learning Outcome: A Literature Based Review Cloud security View project Cloud security View project (2020). <https://www.researchgate.net/publication/340209478>
10. Venkat, N.: Predicting Student Grades using Machine Learning (2018)
11. Eman, E., Majeed, E.A., Junejo, K.N.: Grade prediction using supervised machine learning techniques (2016). <https://www.researchgate.net/publication/304689292>



Predictive Modeling of Gold Prices: Integrating Technical Indicators for Enhanced Accuracy

Noor Aida Husaini¹(✉), Yee Jing Gan¹, Rozaida Ghazali²,
Yana Mazwin Mohmad Hassim², Jie Shen Yeap¹, and Jerome Subash Joseph¹

¹ Faculty of Computing and Information Technology, Tunku Abdul Rahman University of Management and Technology, Jalan Genting Kelang, 53300 Setapak, Kuala Lumpur, Malaysia
nooraida@tarc.edu.my

² Faculty of Computer Science and Information Technology, Universiti Tun Hussian Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia

Abstract. This paper discusses the crucial requirement for reliable gold price prediction, which is necessary for financial market decision-making. We propose a comprehensive approach to develop a robust predictive model capable of predicting both the rise and fall of gold prices. For this, three (3) machine learning (ML) models—Decision Tree Regressor (DTR), Support Vector Regression (SVR), and Random Forest (RF); must be carefully chosen, and model parameters must be adjusted so that predicted values roughly match actual results. Given this, this paper investigates the influence and effectiveness of incorporating technical indicators in predicting fluctuations in gold prices, which might have an impact on the overall performance of the ML models. By achieving a prediction accuracy rate of at least 80%, the model becomes a favorable tool for informed decision-making and provides valuable insights to investors in the gold markets.

Keywords: Gold Price · Relative Strength Index · Bollinger Bands · Average Directional Index · Machine Learning

1 Introduction

Gold prices are influenced by economic or business cycles. For instance, in 2016, there was a downward trend in gold prices [1]. This could be explained as a result of global economic uncertainty, especially in the European Union. As noticed, gold prices have spiked exponentially during the first wave of the COVID-19 pandemic [2], which indirectly increases the uncertainty of the economic and financial market. Syahri & Robiyanto [3] investigated the relationship between gold, the foreign exchange rate, and the Indonesia Composite Stock Price Index. They discovered that during the pandemic, the volatility of stock prices has been greatly impacted by changes in the gold price. Furthermore, Atri et al. [4] and Yousef & Shehadeh [5] findings show a strong association between the gold price and the number of new infections of COVID-19.

Several studies have been conducted related to gold price prediction, including ML methods. The ML methods have witnessed remarkable progress in various domains, and

have sparked interest in applying these techniques to financial data analytics. In [6], they predict the movement of the gold market by the use of the widely used Long Short-Term Memory (LSTM) networks combined with Convolutional Neural Networks. The evolution of gold price prediction models has been studied by Ni et al. [7], who have also revealed the relationship between gold and oil prices by using an LSTM network to predict future gold prices.

To note, quantitative research on gold prediction often uses various calculated technical indicators [7–9], used to identify potential future price movements. Also, indicators like Bollinger Bands or Moving Averages can help identify support and resistance levels in gold prices [7–11]. For instance, Potoski [11] studied trendlines, rate of change, ratios, and stochastic oscillators. Meanwhile, Zhou et al. [12] addressed the volatility and complexity of the stock market by combining ML with technical analysis. Compared to benchmark methodologies, it yields satisfactory results for stock market prediction of 59.25% over the number of stocks. Furthermore, Hyndman & Athanasopoulos [13] investigated the use of technical indicators to predict the trend of security to help investors build their decision on when to buy or sell.

Therefore, to enhance the visibility of the research study and achieve our objectives, we evaluated three (3) technical indicators, and their effects were studied with three (3) ML methods namely Decision Tree Regressor (DTR), Support Vector Regression (SVR), and Random Forest (RF). The next section deals with the data exploration and experimental setup, Sect. 3 details the results by focusing on the influence of technical indicators and overall performance using those ML methods in alignment with our research objectives. Finally, Sect. 4 concludes the findings with appropriate achievements and future planning.

2 Data Exploration and Experimental Setup

This time-series dataset utilized for this project originates from Kaggle, donated by Manu Siddhartha in the year 2021, in CSV format, is comprises information gathered between 15th December 2011 (inclusive), and 31st December 2018 (inclusive) [14].

2.1 Data Overview

The dataset comprises a total of 1,718 rows and 81 columns. Within the 80 columns (excluding the ‘Date’ column), one specific column titled ‘Adj Close’ pertains to the adjusted closing price of gold. This column serves as the target variable, also referred to as the response variable or dependent variable. This is the value that we wish to predict. The remaining columns serve as the features, which are also known as independent variables [14].

2.2 Data Pre-Processing

In this section, we performed anomaly detection by considering three (3) primary categories of irregularities: null data, duplicated data, and outliers.

Null Data Detection. Null data might be caused by corruption of past data due to improper maintenance or other factors. Based on our observations, there are no missing values present. This outcome may result from careful and thorough data collection processes to ensure that all necessary information is collected without any omissions.

Duplicate Data Detection. The dataset is structured as a time-series dataset, where data points are recorded over time. In this dataset, it is expected that each date is unique, meaning that the same date should not appear more than once. Therefore, we have identified that there are no duplicate records in the ‘Date’ column.

Outliers Detection (r Related Columns). Figure 1 indicates the daily returns of the Gold ETF columns. From our observation, it is clear that the daily returns of the Gold ETF columns experienced exceptionally high fluctuations during June 2012, as indicated by a notably increased frequency of outliers during that period. For example, daily returns of values 0.0367, 0.0282, 0.0388, and 0.0388 are recorded in ‘High’, ‘Low’, ‘Close’, and ‘Adj Close’ on 1st June 2012. This can be attributed to the sovereign debt crisis in the Eurozone. Concerns about the stability of the euro and the financial health of European countries prompted investors to seek refuge in gold as a haven for their assets.

In a gold price prediction dataset, outliers can represent unique and important events or market conditions that have a significant impact on gold prices. For instance, a sudden and extreme spike in gold prices during a financial crisis. These historical patterns can provide valuable context for understanding how gold prices react to specific events. These outliers can be analyzed to gain a better understanding of market dynamics, including how quickly prices respond to shocks and how long-lasting the effects of certain events are. Therefore, keeping those outliers for future gold price predictions is important.

2.3 Data Selection

We used two (2) popular feature selection techniques which are (1): Correlation Coefficients Statistics (CC) and Mutual Information Statistics (MI) [12, 13]. This is to improve the quality of the model whilst making the process of modeling more efficient.

Data Selection by Correlation Coefficients Approach. We chose the correlation coefficient $|r \geq 0.7|$ for further consideration as it exhibits a significant linear association with the target.

Data Selection by Mutual Information Approach. The MI is a good alternative to Pearson’s correlation coefficient because MI can detect any kind of relationship, while correlation only detects linear relationships [12]. Therefore, in this study, we calculate the MI between the target variable and the features (Table 1).

Column: Open		Column: High		Column: Low		Column: Close		Column: Adj Close			
Outlier Date	Return	Outlier Date	Return	Outlier Date	Return	Outlier Date	Return	Outlier Date	Return		
0	2011-12-29	-0.032197	0	2012-03-01	-0.033815	0	2012-01-27	0.044420	0	2012-02-29	-0.053029
1	2012-01-27	0.039548	1	2012-03-14	-0.029200	1	2012-02-29	-0.049826	1	2012-06-01	0.038781
2	2012-03-01	-0.040938	2	2012-04-04	-0.031857	2	2012-06-01	0.028217	2	2013-04-12	-0.047004
3	2012-04-04	-0.033879	3	2012-06-01	0.036671	3	2012-06-29	0.029104	3	2013-04-15	-0.087808
4	2012-06-08	-0.030414	4	2013-04-15	-0.081290	4	2012-09-14	0.030126	4	2013-05-01	0.074633
5	2013-04-15	-0.085099	5	2013-05-01	0.035101	5	2013-04-12	-0.047799	5	2013-05-20	0.030899
6	2013-05-01	0.033750	6	2013-06-20	-0.050845	6	2013-04-15	-0.090079	6	2013-06-20	-0.053526
7	2013-06-20	-0.055514	7	2013-07-05	-0.029482	7	2013-05-01	0.067351	7	2013-07-22	0.029814
8	2013-07-01	-0.030004	8	2013-07-22	0.032212	8	2013-06-20	-0.054073	8	2013-09-12	-0.030600
9	2013-08-16	0.038383	9	2013-08-16	0.032540	9	2013-07-01	-0.028483	9	2013-09-18	0.043557
10	2013-09-19	0.053304	10	2013-09-18	0.040712	10	2013-07-05	-0.031525	10	2013-10-17	0.031407
11	2013-10-17	0.028255	11	2013-10-01	-0.030045	11	2013-08-16	0.029944	11	2014-06-19	0.034809
12	2013-11-01	-0.030345	12	2013-10-17	0.032045	12	2013-09-19	0.048502	12	2014-12-01	0.039872
13	2014-10-31	-0.034283	13	2013-11-01	-0.030864	13	2013-10-01	-0.030462	13	2016-02-11	0.040189
14	2014-11-17	0.028992	14	2014-06-19	0.036075	14	2013-10-17	0.035099	14	2016-02-16	-0.030331
15	2016-02-08	0.035340	15	2014-12-01	0.031011	15	2014-10-31	-0.028537	15	2016-06-24	0.049038
16	2016-02-11	0.040994	16	2016-02-11	0.055187	16	2016-02-08	0.034769	16	2016-10-04	-0.034711
17	2016-03-17	0.031852	17	2016-06-24	0.047839	17	2016-02-11	0.042397	17	2016-11-14	-0.030397
18	2016-06-24	0.050878	18	2016-11-14	-0.036202	18	2016-06-24	0.041146			
19	2016-11-14	-0.044437	19	2018-10-11	0.027266	19	2016-10-04	-0.032346			
						20	2016-11-14	-0.032585			

Fig. 1. Outliers with dates of daily returns for Gold ETF Columns.**Table 1.** Features to be chosen with respective r values.

Category	Feature Variable	Coefficient Correlation, r
Gold Exchange Traded Fund (ETF)	Open	0.998976
	High	0.999535
	Low	0.999532
	Close	1.000000
	Adj Close	1.000000

Table 2. Selected Features Based on MI Ranking.

Ranking	Feature Variable
1	Open
2	High
...	...
32	GDX_Adj Close

Based on the Table 2, the top 40%, which is equivalent to 32 features, are selected. Keeping 40% of features can lead to a significant reduction in the dimensionality of the dataset while still preserving a reasonable amount of information. Although the trend

did not show a significant relationship with the target variable when examining CC or MI, the dummy variable can still be handled within the framework of multiple regression models [12]. This is because predicting future value relies on understanding the trend. Therefore, it is important to include the trend columns, as neglecting them could lead to inaccurate predictions.

2.4 Data Construction

In addition to the selected data from the provided dataset, we employed the technical indicators, as we believe they will be beneficial for predicting future prices [7–9, 13]. Based on how investors have historically behaved, they can look for similar charts and data to predict what an asset will do next. We explore three (3) technical indicators – Relative Strength Index (RSI), Bollinger Bands (Upper Level), Bollinger Bands (Lower Level), and Average Directional Index (ADX).

2.5 Data Preparation

This section provides the step-by-step in preparing the data.

Data Normalization. Using min-max scaling, all features are transformed into the range $[0, 1]$, meaning that the minimum and maximum values of a feature/variable are going to be 0 and 1, respectively.

Hyperparameter Tuning. We employed GridSearchCV to search for the best working hyperparameters for our dataset to improve the model's performance.

Train-Test-Split. In time series analysis, we assume that previous values of the target variable play a role in causing future values. Shifting the target variable by one day ensures that the predictor variables are correctly aligned with the target variable in a manner that acknowledges this causal relationship. In summary, the data has been divided into 1,347 rows for the training set and 337 rows for the testing set.

2.6 Modeling

We focused on applying DTR, SVR, and RF. We opted for the DTR in our modeling because it is straightforward to interpret and understand. Moreover, their easy implementation and minimal need for hyperparameter tuning make them a practical choice compared to more intricate ML algorithms. We decided to employ SVR because it excels in high-dimensional spaces and can capture the nonlinearity within the data [15]. Given that our study employs a dataset with $\pm 1K$ rows and ± 80 columns, SVR is a fitting choice. Additionally, SVR's memory efficiency makes it well-suited for handling large datasets. Then, we have settled on RF due to its renowned capability to deliver highly accurate predictions [15, 16]. This aligns with our objective of constructing a precision-focused model. Additionally, RF's ability to effectively manage large datasets makes it a more fitting choice for this project.

3 Results

This section discussed the influence of technical indicators and overall performance.

3.1 Technical Indicators' Influence

We employed 3 technical indicators to create new attributes, which are the RSI [17], Bollinger Bands (Upper Level), Bollinger Bands (Lower Level), and ADX [7–9, 13].

Relative Strength Index. The RSI assesses whether a security's price is overpriced or undervalued by calculating the speed and magnitude of recent price fluctuations [17]. The RSI was introduced in a stepwise manner along with two (2) primary equations; the RSI and the Relative Strength (RS). By applying Eqs. (1) and (2), we then illustrate the RSI trend by generating a line chart for the target variable 'Adj Close'.

$$RS = \frac{\text{Avg.Gain}}{\text{Avg.Loss}} \quad (1)$$

$$RSI = 100 - \frac{100}{1 + RS} \quad (2)$$

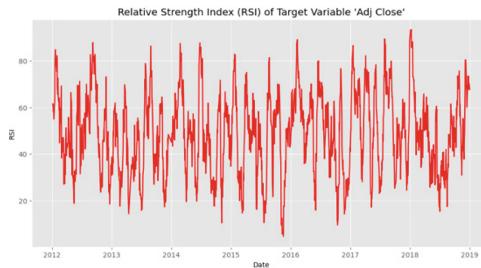


Fig. 2. Line Chart of RSI of Target Variable 'Adj Close'.

From Fig. 2, the line chart of RSI can be analyzed by looking at the RSI levels. RSI > 70 indicates that the gold may be overbought. Overbought refers to a condition where the price of an asset or an indicator has risen to an excessively high level. Overbought conditions may be caused by Fear of Missing Out (FOMO), where traders and investors fear missing out on a potentially profitable opportunity, they may rush to buy gold, driving up its price to unsustainable levels. On the other hand, the RSI < 30 indicates that gold may be Oversold. Oversold conditions may be caused by capitulation, where investors and traders feel hopeless or believe that the gold's price will continue to decline indefinitely, they may engage in panic selling or capitulation.

Bollinger Bands. Bollinger Bands (BB) are commonly used by traders in many markets, including stocks, futures, and currencies as they offer unique insights into price

and volatility. Because the bands' distance is determined by standard deviation, they can adapt to changes in the underlying price's volatility [13].

$$\text{UpperBB} = MA + D \sqrt{\frac{\sum_{i=1}^n (y_j - MA)^2}{n}} \quad (3)$$

$$\text{LowerBB} = MA - D \sqrt{\frac{\sum_{i=1}^n (y_j - MA)^2}{n}} \quad (4)$$

where *UpperBB* and *LowerBB* are the Upper Bollinger Band and Lower Bollinger Band, respectively , y_j represents the individual data points, *MA* is the moving average, *n* denotes the number of days in the smoothing period (typically 20), and *D* is the number of standard deviations (typically 2). Standard deviation is then calculated for the same period, reflecting price volatility [17].

Figure 3 reveals that the cyclical movements extend over longer time frames than seasonality and are influenced by many economic factors. Since BBs are better indicators of short-term trends, it will be better to analyze the graphs at interesting time intervals. Accordingly, a shift in price behavior occurred around October 2012. At this point, the price, which had been closely hugging the *UpperBB*, underwent a mean reversion and began to trend closer to the *LowerBB*, indicating a bearish move down. This sustained downward trend persisted until April 2013. These observations suggest a transition from a bullish phase to a bearish one, with the period around October 2012 being a turning point in the price dynamics. This may signal a period of oversold conditions which suggests that the asset's price may have declined too rapidly, and a potential price rebound or reversal might be approaching which eventually does occur.

Average Directional Index. The ADX can be used to measure if the price is trending strongly [13, 17]. It involves the calculation of several components, such as the positive directional movement (+DM), negative directional movement (-DM), true range (TR), positive directional index (+DI), and negative directional index (-DI). +DM represents an upward price movement, and -DM represents a downward price movement. If the previous high (up-move) is greater than the previous low (down-move), +DM is equal to the up-move, and -DM is set to zero, and vice versa. When comparing the TR, it shows that TR is the greatest among the three [13].

The average true range (ATR) is obtained by calculating a 14-period simple moving average of TR. +DI and -DI are calculated using their respective 14-period simple moving average of directional movement, divided by ATR and multiplied by 100. The directional index (DX) is obtained by the absolute difference of +DI and -DI, divided by the sum of +DI and -DI, and multiplied by 100 [13]. Finally, the average directional index (ADX) is obtained by calculating a 14-period simple moving average of DX. The value of ADX ranges from [0, 100]. Values of 0 to 25 indicate there is an absence of or weak trend, 25 to 50 indicates a strong trend, 50 to 75 indicates a very strong trend and 75 to 100 indicates a very strong trend. It is essential to note that a very strong trend often is not a good sign, as there might be possibilities of a trend reversal very soon.

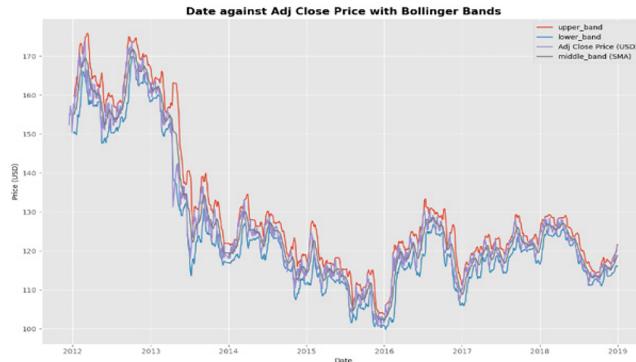


Fig. 3. Price with BB Line Plot.

As evident from Fig. 4, the line chart representing the ADX displays a very high peak in the second half of 2013. This peak indicates that the trend of the gold price during that period was exceptionally strong, characterized by a significant drop. As observed previously, this strong trend can be attributed to the strengthening of the US dollar. A stronger dollar typically makes gold more expensive and, consequently, reduces demand for the precious metal. Furthermore, there is another notable peak in early 2016 in the ADX chart, corresponding to a significant increase in the price of gold. This surge can be explained by global economic uncertainty, which prompted investors to seek safe-haven assets like gold as a hedge against uncertainty and market volatility.

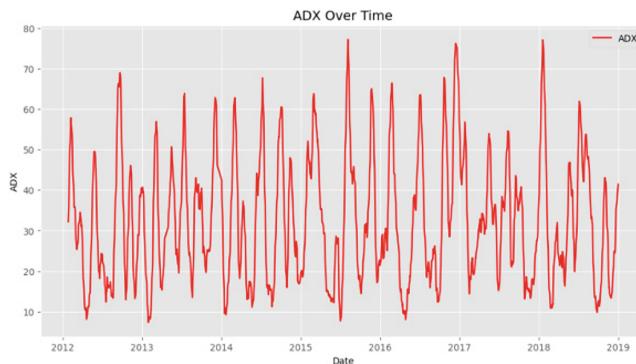


Fig. 4. Line chart of ADX.

3.2 Overall Model Performance

We assess by considering models that exhibit the highest predictive performance.

Table 3. Model Performances.

Models	MAPE (%)	R ²	RMSE	MAE
Decision Tree Regressor (DTR)	0.926	0.90469	1.44268	1.11804
Support Vector Regression (SVR)	0.490	0.97260	0.77350	0.59308
Random Forest (RF)	1.377	0.80218	2.07840	1.68893

Based on Table 3, DTR outperformed by giving the MAPE percentage of 0.926%. The model's performance is generally good and can be considered very accurate. R² quantifies the proximity of data points to the regression line created by a regression algorithm [18]. Besides, SVR indicates its superior performance by having a value of R² of 0.97260, indicating around 97.26% of the variability in the dependent variable. Therefore, the model performs exceptionally well in capturing the underlying patterns within the data and making accurate predictions.

A lower RMSE indicates better predictive performance, emphasizing the model's accuracy in making predictions [18]. Therefore, SVR shows the lowest RMSE value out of the other two models. This signifies that, on average, the SVR's predictions deviate by roughly 0.774 units from the actual values. Regarding MAE, an MAE of 0 means that the model is a perfect predictor of the outputs [18]. An MAE of 0.59308 for SVR indicates that the model's predictions deviate from the actual values by approximately 0.593 units in the same units as the target variable on average. Overall, the SVR consistently received the highest ranking across all evaluations, signifying its strongest performance among these models by considering having the lowest R², RMSE, MAE, and average MAPE, which is useful for predicting any commodity price [15, 18].

4 Conclusion and Future Improvement

We conducted a comprehensive comparison of three (3) models by examining their respective evaluation metric results. Our findings revealed that among DTR and SVR, SVR stood out as the top performer with a MAPE value of 0.490%. These models consistently demonstrated a MAPE value of less than 1.0%, indicating the accuracy rate exceeding 80% closely aligned with actual values. Together with technical indicators, we believe that it facilitates the identification of trends and possible market turning points by helping to discern patterns in gold price movements. In order to make wise judgments, this information will assist investors in determining whether the market is trending upward, downward, or sideways.

Future research could include experimenting with more sophisticated ML algorithms or ensemble approaches like deep learning techniques, to catch intricate patterns that typical models would overlook. Apart from that, we would like to investigate sentiment analysis of financial news or gold-related social media data. This is because market behavior can be influenced by public mood, and including this data could increase prediction accuracy.

Acknowledgements. This research was supported by Universiti Tun Hussein Onn Malaysia.

References

1. Ahir, H., Nicholas Bloom, N., Furceri, D.: Global Economic Uncertainty Remains Elevated, Weighing on Growth. IMF Blog (2023). <https://www.imf.org/en/Blogs/Articles/2023/01/26/global-economic-uncertainty-remains-elevated-weighing-on-growth>
2. World Gold Council. Gold Outlook 2021. Goldhub Professional Investors (2021). <https://www.gold.org/goldhub/research/outlook-2021>
3. Syahri, A., Robiyanto, R.: The correlation of gold, exchange rate, and stock market on Covid-19 pandemic period. Jurnal Keuangan dan Perbankan **24**(3), 350–362 (2020)
4. Atri, H., Kouki, S., imen Gallali, M.: The impact of COVID-19 news, panic and media coverage on the oil and gold prices: An ARDL approach. Resour. Policy **72**, 102061 (2021)
5. Yousef, I., Shehadeh, E.: The impact of COVID-19 on gold price volatility. Int. J. Econ. Bus. Adm. **8**(4), 353–364 (2020)
6. Liveris, I.E., Pintelas, E., Pintelas, P.: A CNN–LSTM model for gold price time-series forecasting. Neural Comput. Appl. **32**, 17351–17360 (2020)
7. Ni, Y., Day, M.Y., Huang, P., Yu, S.R.: The profitability of Bollinger Bands: Evidence from the constituent stocks of Taiwan 50. Physica A **551**, 124144 (2020)
8. Agrawal, M., Khan, A.U., Shukla, P.K.: Stock price prediction using technical indicators: a predictive model using optimal deep learning. Learning **6**(2), 7 (2019)
9. Vaiz, J.S., Ramaswami, M.: A study on technical indicators in stock price movement prediction using decision tree algorithms. Am. J. Eng. Res. (AJER) **5**(12), 207–212 (2016)
10. Surendra, J., Rajyalakshmi, K., Apparao, B.V., Charankumar, G., Dasore, A.: Forecast and trend analysis of gold prices in India using auto regressive integrated moving average model. J. Math. Comput. Sci. **11**(2), 1166–1175 (2021)
11. Potoski, M.: Predicting gold prices. CS229, Autumn **2** (2013)
12. Zhou, H., Wang, X., Zhu, R.: Feature selection based on mutual information with correlation coefficient. Appl. Intell. 1–18 (2022)
13. Hyndman, R.J., Athanasopoulos, G.: Forecasting: principles and practice. OTexts (2018)
14. Siddhartha, M.: Gold Price Prediction Using Machine Learning. Kaggle (2021). <https://www.kaggle.com/code/sid321axn/gold-price-prediction-using-machine-learning>
15. Makala, D., Li, Z.: Prediction of gold price with ARIMA and SVM. In Journal of Physics: Conference Series, vol. 1767, no. 1, p. 012022. IOP Publishing (2021)
16. Singh, S.K., Gupta, N., Baliyan, S., Mishra, P.K.: Gold price prediction using machine learning algorithm. NeuroQuantology **20**(20), 2998 (2022)
17. Lento, C., Gradojevic, N., Wright, C.S.: Investment information content in Bollinger Bands? Appl. Finan. Econ. Lett. **3**(4), 263–267 (2007)
18. Chicco, D., Warrens, M.J., Jurman, G.: The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Comput. Sci. **7**, e623 (2021)



Portfolio Optimization with Percentage Error-Based Fuzzy Random Data for Industrial Production

Mohammad Haris Haikal Othman¹(✉), Nureize Arbaizy¹,
Muhammad Shukri Che Lah¹, and Pei-Chun Lin²

¹ Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Malaysia

harishaikal940322@gmail.com

² Department of Information Engineering and Computer Science, Feng Chia University, No. 100, Wenhwa Road, Taichung, Taiwan

Abstract. Data-driven decision-making processes are pervasive in various domains, yet the inherent uncertainties within observational and measurement data can lead to misleading outcomes, particularly in portfolio selection where randomness may seem ambiguous. While existing methodologies recognize the significance of data preprocessing in managing uncertainties such as fuzziness and randomness, a systematic framework to effectively address these challenges is currently lacking. This study aims to bridge this gap by presenting a comprehensive framework tailored to efficiently handle uncertainty during the preprocessing stage. The proposed framework not only acknowledges the importance of data preprocessing but also offers a systematic approach to processing fuzzy random data, thus providing a robust foundation for portfolio selection algorithms. Leveraging fuzzy integers to manage fuzziness and probability distributions to address randomness, our methodology ensures the construction of reliable portfolio selection strategies. The main objective is to optimize selection based on industrial production, effectively managing uncertainty in traditional portfolio selection models. In this proposed approach, fuzziness is handled using fuzzy numbers, and randomness is addressed through probability distributions. The efficacy of this approach is demonstrated in agricultural planning, evaluating five distinct industrial production types: Agriculture, Mining, Manufacturing, Electricity, and Water. The findings underscore the effectiveness of the proposed methodology in managing uncertainties, reducing errors in model development stages, and providing a robust framework for optimal portfolio selection tailored to industrial production contexts, thereby enhancing decision-making processes in uncertain environments.

Keywords: Fuzzy Random Variable · Fuzzy Random Data · Data Pre-processing · Mean-Variance

1 Introduction

Real-world data is rarely flawless, and uncertainties might come in at any point, reducing modelling and predicting accuracy. Uncertainties in collected data include measurement errors, data quality, representativeness, and bias. Uncertainties inherent in real-world scenarios apply a significant influence on forecasting and decision-making processes, complicating decision [1] parameters across various industries. Data contains fuzziness and randomness because of the ambiguities and vagueness that exist in the real world. Randomness refers to physical uncertainty, while fuzziness is caused by human cognition. Fuzzy random data can be used to describe all the information associated with a measurement result, including systematic and random components to the overall uncertainty.

Portfolio selection models often face challenges in uncertain environments, where security returns are difficult to predict based on historical data alone. To address this, various models have been proposed, such as uncertain portfolio selection models with background risk [2], those based on VaR minimization [3], and multi-period portfolio models considering expert evaluations. These models aim to enhance decision-making by incorporating uncertainty into the portfolio selection process.

The evaluation of the usefulness of portfolio selection models across diverse industrial sectors serves multiple important purposes, including validating the methodology's applicability, assessing its performance comprehensively, and demonstrating its practical relevance. By examining sectors such as Agriculture, Mining, Manufacturing, Electricity, and Water, the study ensures that the proposed methodology addresses sector-specific challenges and is effective in various real-world scenarios. This approach helps validate the robustness and generalizability of the model, making it applicable beyond specific industries.

Data containing fuzzy random uncertainties reduces the accuracy of portfolio selection models by introducing uncertainty and ambiguity into the decision-making process. In modeling uncertain phenomena [4] across different production sectors, the concepts of random variables and fuzzy theory play pivotal roles. While probability theory addresses random events, fuzzy theory offers solutions for handling fuzzy data. However, existing models often treat these uncertainties separately, overlooking their simultaneous occurrence in real data.

2 Literature Review

This section provides a background on portfolio selection through the mean-variance model and the utilization of fuzzy numbers.

2.1 Mean-Variance Model

Portfolio selection entails the process of identifying an optimal portfolio. Markowitz introduced a method aimed at determining the optimal portfolio that maximizes returns while minimizing risks, framing such scenarios as portfolio selection problems. Portfolio

selection finds wide application in real-world contexts and has been expanded to address uncertainties [5, 6].

Equation (1) defines the portfolio selection model [7].

$$\begin{aligned} & \min \left\{ \sum_{i=1}^n \sum_{j=1}^n x_i \sigma_{ij} x_j \right\} \\ & \text{s.t. } \sum_{i=1}^n E(r_i) x_i = \rho \\ & \quad \sum_{i=1}^n x_i = 1 \\ & \quad x_i \geq 0, i = 1, \dots, n \end{aligned} \tag{1}$$

In this context, r_i represents the random variable denoting the return, $E(r_i)$ signifies the expected value associated with r_i , σ_{ii} stands for the variance of r_i , σ_{ij} represents the covariance between r_i and r_j , x_i denotes the proportion of capital allocated to security i and ρ signifies the parameter related to the expected return.

2.2 Fuzzy Set

The concept of fuzzy sets is as follows:

Definition 1: Within the context of this analysis, let U represent the universal set of discourse. We define a fuzzy set A on U using a membership function m_A . This function assigns a real value between 0 and 1 to each element x in U , indicating the degree to which x satisfies the characteristic property of A . Formally, we can express a fuzzy set A in U as a collection of ordered pairs $A = \{(x, m_A(x)) : x \in U\}$, where $m_A : U \rightarrow [0, 1]$ represents the membership degree of x in A . Formally, a fuzzy number x is described as:

$$m_A(X) = \sum_{i=1}^n m_i(X) I_{A_i}(X) \tag{2}$$

where $I_{A_i}(X) = 1$ when $x \in A_i$ and $I_{A_i}(X) = 0$ if $x \notin A_i$.

Definition 2: A fuzzy set A on the universal set U , defined by the membership function $y = m(x)$, is regarded normal if there is at least one element x in U with $m(x) = 1$.

2.3 Fuzzy Random Variables

Data hybridization, which combines fuzzy data with randomness, appears as a major advance in the extension of fuzzy sets [11]. The concept of fuzzy random variables [12] was improved by [13] with a new notion of exclusion, allowing the generalization of integrals for set-valued functions.

Theorem 1. Central Limit Theorem.

Let X_1, \dots, X_n be the random variables with mean m and variance σ^2 . Let

$$Z_n = \frac{n^{\frac{1}{2}}(\bar{X}_n - m)}{\sigma} = \frac{W_n - nm}{n^{\frac{1}{2}}\sigma} \quad (3)$$

X_1, X_2, \dots, X_n are independent, identically distributed random variables with mean μ and standard deviation σ ; Z_n is the standardised sample mean. The sample mean, or \bar{X}_n , is calculated by dividing the sum of the random variables X_i by the sample size, n is the sample size, σ is the standard deviation of each individual random variable, and m is the population mean, or expected value. Z_n converges in distribution to Z as $n \rightarrow \infty$. $Z_n \rightarrow Z \sim N(0, 1)$ is denoted as $n \rightarrow \infty$ where Z distribute according to a standard normal distribution function $N(0, 1)$.

3 Portfolio Selection Model Using Fuzzy Random Data Based on Percentage Error on Industrial Production Index

3.1 Fuzzy Random Data Pre-processing

This section describes the methods for pre-processing data with fuzzy random variables. To account for real-world uncertainty, we use an approach based on measurement error ranges, as established in previous publications [14–16].

In this analysis, we depict the minimum potential value (A_i) and the highest potential value (B_i). Equation (4) describes the exact procedure used to generate fuzzy data from the initial single-value data points, where p is the percentage of error and ip is index production.

$$\begin{aligned} A_i &= ip - (ip * p\%) \\ B_i &= ip + (ip * p\%) \end{aligned} \quad (4)$$

The original data is transformed into fuzzy intervals $F_i = [A_i, B_i]$ to account for real-world uncertainties. The intervals represent the potential range of each data point. We also define each fuzzy interval by its center point (c_i) and width (w_i). The central point ($c_i = (A_i + B_i)/2$) represents the most likely outcome, while the width ($w_i = B_i - A_i$) indicates the level of uncertainty. Specific formulas for determining these numbers in the Eq. (5).

$$\begin{aligned} c_i &= \frac{A_i + B_i}{2} \quad \forall i = 1, 2, \dots, n \\ w_i &= \frac{B_i - A_i}{2} \quad \forall i = 1, 2, \dots, n \end{aligned} \quad (5)$$

This study addresses randomness in the data using four probability distributions: Normal, Weibull, Gamma, and Logistic. Each distribution generates a p -value, which indicates how well it fits the data. The distribution with the highest p -value, indicating the best fit, is selected for further analysis. After pre-processing with fuzzy random variables, the input is translated into interval form $Y_i = [C_i, W_i]$, representing potential changes around a central point. Within each interval, C_i represents the central point and W_i indicates the width of the interval.

3.2 Fuzzy Random Based Portfolio Selection Model for Industrial Production Index

Let $Y_i = (C_i, W_i)$ represent the set of interval fuzzy numbers with $\forall i = 1, 2, \dots, n$. The expected value and variance of the fuzzy interval data are determined as follows:

Expected value:

$$E(A_i) = (E(c_i), E(w_i)) \quad (6)$$

Variance:

$$\sigma^2(A_i) = (\sigma^2(c_i), \sigma^2(w_i)) \quad (7)$$

E is the expected value of interval fuzzy number and σ^2 is the variance. Portfolio formulation is given as follows:

$$\left. \begin{array}{l} \max \sum_{i=1}^n E(c_i)x_i \\ \min \sum_{i=1}^n E(w_i)x_i \\ s.t. \sum_{i=1}^n \sigma(c_i)x_i \\ \sum_{i=1}^n \sigma(w_i)x_i \\ \sum_{i=1}^n x_i \leq 1 \\ \sum_{i=1}^n \sigma(w_i)x_i \\ x_i \geq 0 \forall i = 1, 2, \dots, n \end{array} \right\} \quad (8)$$

$\max \sum_{i=1}^n E(c_i)x_i$ and $\min \sum_{i=1}^n E(w_i)x_i$ in (8) uses expected value in center and width. Based on the findings in Sects. 3.1, 3.2, we provide an optimized approach to this stage within the method:

1. **Data Collection:** Gather relevant data.
2. **Determine Measurement Error:** Before using the data for fuzzification, it is critical to identify any potential collection inaccuracies. Quantifying this inaccuracy, such as a 5% margin, directs the fuzzification process by establishing acceptable limits for portraying data variability.
3. **Fuzzification Process:**
 - For each data point x_i , calculate the maximum and minimum potential values based on the measurement error. Let's denote these as A_i and B_i respectively.
 - Formulate the fuzzy data interval $F_i = A_i, B_i$.
4. **Calculate Central Point and Width:**
 - Determine the central point c_i of the fuzzy interval F_i .
 - Calculate the width w_i of the fuzzy interval F_i .

5. **Repeat for Each Data Point:** Perform steps 3 and 4 for each data point in the dataset.
6. **Compile Results:** Present the fuzzified data in the form of interval parameters F_i , where each F_i represents a fuzzy data interval with its respective central point and width.
7. **Calculate Probability Distribution Function (PDF):** The central point and width provide clues into distribution types, but further investigation is required for identification. While a distribution's center point and width provide indications about its basic shape (symmetric, skewed), determining the underlying PDF requires more information or analysis, such as higher-order moments or goodness-of-fit tests.
8. **Compute Portfolio Selection Model:** Use the PDF result to identify the risk level to get the most optimize result using Eq. (8).

This strategy provides an efficient method for decision-makers to create selection models while handling data uncertainty.

4 Numerical Experiment

The analysis incorporates data from five distinct production sectors: Agriculture, Mining, Manufacturing, Electricity, and Water. This data is obtained from Malaysia's official data catalog (<https://data.gov.my/ms-MY/data-catalogue/pi>) and spans a three-year period, from 2015 to 2023. The primary goal of this study is to identify an investment portfolio that chooses one of these five production sectors.

Table 1. Single form data

Data/Production	Agriculture	Mining	Manufacturing	Electricity	Water
1/1/2010	92.7	99.4	98.8	99.8	99.2
1/2/2010	93.3	97.8	99	99.4	99.1
1/3/2010	96.1	108.6	99.2	100.5	99.2
1/4/2010	95.3	107.7	99.3	99.7	99.9
1/5/2010	94.1	91.4	99.8	100.3	101.5
...
1/12/2023	123.2	97.4	120	117	118.3

Fuzzy random data pre-processing starts here. The raw data obtained in Table 1 is then fuzzified by using 5% percentage measurement error based on Eq. (4). This resulted in fuzzy interval data in a form of $[a, b]$ where a is minimum data and b is the maximum data. Table 2 shows the percentage error data.

Table 2. Percentage error 5% data $[a_i, b_i]$

Data/Production	Agriculture	Mining	Manufacturing	Electricity	Water
1/1/2010	[88.07, 92.47]	[94.43, 99.15]	[93.86, 98.55]	[94.81, 99.55]	[94.24, 98.95]
1/2/2010	[88.64, 93.07]	[92.91, 97.56]	[94.05, 98.75]	[94.43, 99.15]	[94.15, 98.85]
1/3/2010	[91.30, 95.86]	[103.17, 108.33]	[94.24, 98.95]	[95.47, 100.25]	[94.24, 98.95]
1/4/2010	[90.54, 95.06]	[102.32, 107.43]	[94.34, 99.05]	[94.72, 99.45]	[94.91, 99.65]
...
1/12/2023	[117.04, 122.89]	[92.53, 97.16]	[114.00, 119.70]	[111.15, 116.71]	[112.39, 118.00]

After the data been processed using the percentage error, the data should be fuzzy data to o and l . The data should be processed such as Table 3.

Table 3. Fuzzy data, center point, and width $[o_i, l_i]$

Date/Production	Agriculture	Mining	Manufacturing	Electricity	Water
1/1/2010	[-2.43, 2.20]	[-2.61, 2.36]	[-2.59, 2.35]	[-2.62, 2.37]	[-2.60, 2.36]
1/2/2010	[-1.85, 2.22]	[-4.17, 2.32]	[-2.40, 2.35]	[-3.01, 2.36]	[-2.70, 2.35]
1/3/2010	[0.88, 2.28]	[6.35, 2.58]	[-2.20, 2.36]	[-1.94, 2.39]	[-2.60, 2.36]
1/4/2010	[0.10, 2.26]	[5.47, 2.56]	[-2.11, 2.36]	[-2.72, 2.37]	[-1.92, 2.37]
...
1/12/2023	[27.27, 2.93]	[-4.56, 2.31]	[18.05, 2.85]	[14.13, 2.78]	[15.99, 2.81]

After collecting the data, the center point and width of the fuzzy data is identified as in Eq. (8). Table 4 shows the center point and width of the fuzzy data. The probability distribution function is then performed to treat the randomness. Each of the production data will provide 4 types of data which each of them will generate Normal, Log, Gamma, and Weibull distribution. Each of the distributions will provide the p-value. In this paper, the biggest p-value indicates as the best result to treat the randomness.

Table 4. Interval number of the probability distribution function

	center, C	width, W
Agriculture	N(14.7736, 18.5711)	γ (33.4912, 0.0783)
Mining	N(3.0240, 24.6628)	γ (17.2472, 0.1448)
Manufacturing	γ (1.5832, 4.4442)	W(358.2068, 0.0072)
Electricity	W(2.8662, 13.0576)	W(24.6696, 2.7416)
Water	W(1.7513, 10.2081)	W(28.1013, 2.6715)

Table 5 shows 5 types of production are considered and represents in the form of $x_n = (x_1, x_5, \dots, x_5)$ respectively. Table 4 shows the probability distribution function that has been selected based on the highest p-value. Note that the LOG denotes logistic distribution, was the Weibull Distribution and γ as gamma distribution. The moment estimator is utilized to approximate the expected value and variance each of the vegetable.

Table 5. Expected value and variance

Production		center, C		width, W	
		expected value	variance	expected value	variance
Agriculture	x_1	14.77360	344.88390	2.62131	0.20517
Mining	x_2	3.02401	608.25370	2.49815	0.36184
Manufacturing	x_3	7.03619	488.69613	2.58137	0.01860
Electricity	x_4	2.75506	0.06620	21.95005	74.78710
Water	x_5	1.66756	0.03871	24.98108	101.46823

Table 6 shows the result. Finally, the expected value and variance for each of the production indexes is computed. The data has now completed the pre-processing phase. This pre-processed data is then presented to the portfolio selection model to identify the best portfolio.

The portfolio selection model in Eq. (8) is used to build the Model (9).

$$\begin{aligned}
 & \max 14.77360x_1 + 3.02401x_2 + 7.03619x_3 + 2.75506x_4 + 1.66756x_5 \\
 & \min 2.62131x_1 + 2.49815x_2 + 2.58137x_3 + 21.95005x_4 + 24.98108x_5 \\
 & s.t. \sqrt{344.88390x_1} + \sqrt{608.25370x_2} + \sqrt{488.69613x_3} + \sqrt{0.06620x_4} + \sqrt{0.03871x_5} = k \\
 & \sqrt{0.20517x_2} + \sqrt{0.36184x_2} + \sqrt{0.01860x_3} + \sqrt{74.78710x_4} + \sqrt{101.46823x_5} \leq k \\
 & x_1 + x_2 + x_3 + x_4 + x_5 \leq 1 \\
 & x_i \geq 0 \forall i = 1, 2, \dots, n
 \end{aligned} \tag{9}$$

Equation (9) is solved using a linear programming approach. Here, we assume that k represents the risk level, and the optimal solution is achieved with $x_n = (1, 0, 0, 0, 0)', x_n = (0, 1, 0, 0, 0)', x_n = (0, 0, 1, 0, 0)', x_n = (0, 0, 0, 1, 0)',$ or $x_n = (0, 0, 0, 0, 1)'$. The model's computation halts upon reaching the optimal solution.

5 Result and Discussion

Table 6 shows the optimal solution results. From the results table, the risk of $k = 5.591$ indicates the optimal solution where the expected return is $(7.03, 2.58)$ for a 5% percentage error. The optimum result based on the five industrial production data is manufacturing production.

Table 6. The result with optimal solution

Risk, k	3	4	5	5.591	6
x_n^*	[0,0.6,0,0,0]	[0,0.81,0,0,0]	[0,0.95,0.05,0,0]	[0,0,1,0,0] *	[0, –0.65, 1.65, 0, 0]
Expected Value	(1.82, 1.34)	(2.4, 1.8)	(3.24, 2.3)	(7.03, 2.58)	[9.66, 2.77]

Risk level k is critical in helping management make portfolio selection decisions. This parameter, shown by the optimal value $x^* 5.591$ in Table 6, denotes the amount of risk associated with a specific portfolio. In this situation, it indicates that manufacturing output can create the largest return compared to other production sectors, but at a greater risk level.

The fundamental goal of this research is to find industrial production solutions with high potential for returns while controlling associated risks. While risk levels $k = 3, 4$, and 5 often produce positive expected returns, $k = 5.591$ stands out as having the greatest potential return based on our findings in Table 6. As a result, this model can assist management in prioritizing industries with optimal risk-return profiles to maximize prospective profitability.

A total allocation of 1 is required while aiming for a particular risk level, $k = 6$, based on the portfolio outcome that has been provided, which is the allocation of resources represented by x_1 to x_5 . To achieve a risk level of 1.6541, it is specifically necessary to decrease the allocation of x_2 by roughly 0.651 units and raise the allocation of x_3 . Within the context of this structure, the modifications seek to reallocate resources across the portfolio to satisfy the target degree of risk while abiding by the limitations set forth by the optimization issue. By using this reallocation approach, resources are distributed fairly and in accordance with the designated risk tolerance level.

6 Conclusions

In conclusion, this study introduces a portfolio selection approach designed to effectively address the inherent uncertainties present in real-world data, particularly in industrial production planning. Employing a two-stage strategy, our approach demonstrates resilience and adaptability in navigating uncertainty. Firstly, through the innovative integration of fuzzy random variables, our methodology rigorously cleans and prepares data,

capturing both unpredictability and ambiguity inherent in real-world datasets. Second, using this pre-processed data in a portfolio selection model based on the standard mean-variance paradigm, we optimise portfolio strategies across multiple production sectors while accounting for uncertainty. Particularly, our approach enables more resilient and adaptive portfolio selection strategies, which are critical in the face of industrial production planning uncertainties. Specifically, the first stage utilizes a measurement error method to transform crisp data into fuzzy sets, enriching the dataset and capturing potential variances. Subsequently, key parameters such as fuzzy center and width are derived from these fuzzy sets. To address randomness, our method incorporates probability distributions alongside the preprocessed fuzzy data, providing a comprehensive approach to managing uncertainty. Application of this technique across five distinct industrial sectors—agriculture, mining, manufacturing, electricity, and water—yields promising results, showcasing its ability to identify optimal production yields for each sector. These findings offer valuable insights for strategic planning and decision-making processes across industries, underscoring the resilience and adaptability of our portfolio selection tactics in uncertain environments.

Acknowledgments. This research was supported by Universiti Tun Hussein Onn Malaysia (UTHM) through Tier 1 (Vot Q507) and Ministry of Higher Education (MOHE) through Fundamental Research Grant Scheme (FRGS) (FRGS/1/2019/ICT02/UTHM/02/7).

References

1. Dorsey, A.H.: Active Alpha: A Portfolio Approach to Selecting and Managing Alternative Investments, vol. 356. Wiley, Hoboken (2011)
2. Wang, S., Xia, Y.: Portfolio Selection and Asset Pricing, vol. 514. Springer, Heidelberg (2012)
3. Tarasi, C.O., Bolton, R.N., Hutt, M.D., Walker, B.A.: Balancing risk and return in a customer portfolio. *J. Mark.* **75**(3), 1–17 (2011)
4. Markowitz, H.M.: Portfolio selection. *J. Financ.* **7**(60), 77–91 (1952)
5. Shapiro, A., Dentcheva, D., Ruszczynski, A.: Lectures on Stochastic Programming: Modeling and Theory. SIAM-Society for Industrial and Applied Mathematics (2009)
6. Vakarchuk, R.N., Mäntyniemi, P., Tatevossian, R.E.: On the effect of synthetic and real data properties on seismic intensity prediction equations. *Pure Appl. Geophys.* **176**, 4261–4275 (2019)
7. Re, C., Suciu, D.: Management of data with uncertainties. In: Proceedings of the 16th ACM Conference on Information and Knowledge Management, pp. 3–8 (2007)
8. Krause, P., Clark, D.: Representing Uncertain Knowledge: An Artificial Intelligence Approach. Springer, Heidelberg (2012)
9. Hasuike, T., Katagiri, H., Ishii, H.: Portfolio selection problems with random fuzzy variable returns. *Fuzzy Sets Syst.* **160**(18), 2579–2596 (2009)
10. Rubinstein, M.: Markowitz's portfolio selection: a fifty-year retrospective. *J. Financ.* **57**(3), 1041–1045 (2002)
11. Zhang, Y., Li, X., Guo, S.: Portfolio selection problems with Markowitz's mean–variance framework: a review of literature. *Fuzzy Optim. Decis. Making* **17**, 125–158 (2018)
12. Amiri, A., Tavana, M., Arman, H.: An integrated fuzzy analytic network process and fuzzy regression method for bitcoin price prediction. *Internet Things* **25**, 101027 (2024)

13. Uusipaikka, E.: Confidence Intervals in Generalized Regression Models. CRC Press, Boca Raton (2008)
14. Li, B., Teo, K.L.: Portfolio optimization in real financial markets with both uncertainty and randomness. *Appl. Math. Model.* **100**, 125–137 (2021)
15. Qin, Z.: Mean-variance model for portfolio optimization problem in the simultaneous presence of random and uncertain returns. *Eur. J. Oper. Res.* **245**(2), 480–488 (2015)
16. Markowitz, H.M.: Foundations of portfolio theory. *J. Financ.* **46**(2), 469–477 (1991)



The Football Matches Outcome Prediction for English Premier League (EPL): A Comparative Analysis of Multi-class Models

Nur Amirah Adnan¹, Luqman Al Hakim Mohd Asri¹, Aida Mustapha^{1(✉)}, and Muhammad Nazim Razali²

¹ Faculty of Applied Sciences and Technology, Universiti Tun Hussein Onn Malaysia,
KM 1, Jalan Panchor, 84600 Panchor, Johor, Malaysia
aidam@uthm.edu.my

² Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia,
86400 Parit Raja, Batu Pahat, Johor, Malaysia

Abstract. Football match outcome prediction has evolved into a dynamic field of research, driven by the integration of machine learning models to achieve precise forecasts. This paper embarks on a comprehensive exploration, presenting a comparative analysis of four prominent multiclass machine learning (ML) models-Multiclass Logistic Regression, Multiclass Neural Network, Multiclass Decision Forest, and Multiclass Decision Jungle. Furthermore, it explores the application of Azure Machine Learning for predicting English Premier League (EPL) match results in terms of win, draw, and lose, underscoring the league's significance in the global football landscape through multiclassification ML models. The paper aims to contribute valuable insights to the field, emphasizing the importance of ML model selection for enhanced predictive performance. As the result, Multiclass Logistic Regression emerges as the most accurate model with a 57.08% success rate, followed closely by the multiclass decision forest at 56.71%. The multiclass neural network exhibits a respectable accuracy of 55.59%, while the multiclass decision jungle performs similarly with an identical accuracy of 55.59%. These findings show critical role of a choosing suitable and well-performing ML model in achieving accurate football match outcome predictions.

Keywords: Football Matches · Prediction · Multi-class Classification · Machine Learning · Logistic Regression · Neural Network · Decision Forest · Decision Jungle

1 Introduction

Since the FIFA World Cup began in 1930, football has grown into a multifaceted global phenomenon. The abundance of football leagues worldwide, including the English Premier League, French Ligue 1, Spanish La Liga, Italian Serie A,

and German Bundesliga, has spurred extensive research into predictive models [2, 11]. Pioneering efforts, such as the Least Squares Model in 1977, laid the groundwork for more sophisticated approaches, including Bayesian networks, complex frameworks like the FRES system, and hierarchical models [12].

According to [7], the existing studies in football matches outcomes prediction models can be broadly categorized into three groups which are Statistical models, which assume that the goals scored adhere to a specific parametric probability distribution. The Machine learning models, which typically involve deriving various complex features from the data and feeding them into a ready-made learning algorithm. Finally, the Rating systems, assigning a real-valued rating to each team to represent its strength.

Although the football matches outcomes prediction models are categorized into three clusters, machine learning models possesses the capability and potential to integrate alternative approaches into its functioning, while also incorporating more complex and sophisticated features which suggested by [7] to conclude their works on statistical and rating models. Thus, the scope of this research will cover machine learning model that provided by library of Microsoft Azure Machine Learning Studio for multiclass classification to predict football matches outcomes in term of win, draw or lose.

The structure of this paper is as follows: Sect. 2 reviews the relevant literature. Section 3 outlines the research methodology, encompassing details on the dataset, algorithms employed, and evaluation metrics utilized. Comparative results are discussed in Sect. 4, followed by the conclusion in Sect. 5.

2 Literature Review

Concurrently, the realm of sports analytics has witnessed a surge in interest and research focused on predicting football match results. Football, being a globally celebrated sport, has been at the forefront of outcome prediction research. The fusion of machine learning with predictive modeling has given rise to sophisticated tools vying to provide accurate forecasts for match results [6]. Leveraging advanced technology and machine learning techniques, this study explores the application of Azure Machine Learning in forecasting the outcomes of English Premier League (EPL) matches [2]. By harnessing historical data and employing state-of-the-art algorithms, the research aims to uncover insights into the factors contributing to successful match predictions in one of the world's most competitive and widely followed football leagues [9].

The popularity of machine learning algorithms in sports analytics stems from their ability to analyze vast datasets and discern patterns that might elude human analysts [14]. The EPL, with its rich dataset encompassing team performance indicators, player statistics, historical match results, and other contextual factors, provides an ideal landscape for predictive modeling. This paper undertakes a detailed comparative analysis of four multiclass models, which are Multiclass Logistic Regression, Multiclass Neural Network, Multiclass Decision Forest, and Multiclass Decision Jungle with a specific emphasis on their utility in predicting football match outcomes based on Azure platform.

Multiclass classification pertains to machine learning tasks where there are more than two classes involved. Various metrics are available to assess the performance of a multiclass classifier. These metrics are valuable throughout the development stages, particularly when comparing the performance of different models [5]. The use of multiclass classification in predicting football match outcomes is a popular strategy, and Azure Machine Learning offers various components and resources for this purpose. Azure Machine Learning Studio calculates performance metrics for each classification model generated, including multiclass classification metrics for datasets with two or more classes [10]. [1] employed Multiclass Neural Network and the Multiclass Decision Forest using Microsoft Azure Machine Learning Studio to study football match outcomes prediction (win, lose and draw) for English Premier League.

Studies for predicting football match outcomes have been conducted in two-ways either by binary classification which the target class is win or not win or multiclass classification which is win, draw or lose [4, 13]. In [4], Azure machine learning was employed to predict the outcomes of football matches during the 2022–2023 English Premier League (EPL) season. The performance of random forest, logistic regression, linear support vector classifier, and extreme gradient boosting models was evaluated for both binary and multiclass classification tasks. The findings indicated that the models achieved improved performance when the training dataset utilized a balanced sampling technique for binary classification. However, these studies reported an average prediction accuracy of 53.7% obtained using a Random Forest.

In this study, the effectiveness of machine learning models are assessed using a football dataset obtained from Microsoft Azure Machine Learning Studio. Then, the performance of Multiclass Logistic Regression, Multiclass Neural Network, Multiclass Decision Forest, and Multiclass Decision Jungle for multiclass classification tasks are compared, focusing on predicting win, draw, and lose outcomes.

3 Research Methodology

Research methodology is a structured and scientific approach used to collect, analyze, and interpret data to answer research questions. The research methodology involves training the model on historical EPL match data and assessing its performance through cross-validation techniques to gauge its accuracy in forecasting future match outcomes. This section presents the classification model for a prediction which team wins whether the home team wins (H), a draw (D), or the away team wins (A). This study aims to build the best machine learning model to predict a match outcome with the highest possible accuracy and it deals with several algorithms of multiclass classification.

In this study, Azure Machine Learning serves as the chosen platform for developing a predictive model. Azure's comprehensive set of tools, including prebuilt machine learning algorithms and robust data processing capabilities, positions it as an ideal choice for real-time analysis of complex datasets. The workflow for Azure Machine Learning and the Azure pipeline for experiments is shown in Figs. 1 and 2 respectively.

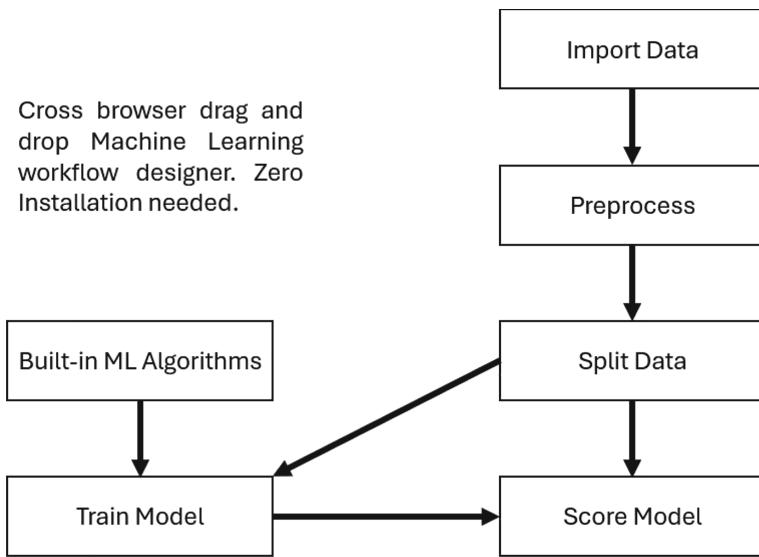


Fig. 1. The Azure Workflow for Machine Learning

The Azure workflow for machine learning simplifies the machine learning process, allowing users to visually design and implement their models without the need for extensive coding or programming expertise [15]. Each step in the workflow are described as following:

- **Import Data.** In this step, the raw data are imported into the Azure Machine Learning Studio. The data can be imported from various sources, like Azure Blob Storage, SQL databases, or upload files.
- **Preprocess.** This step involves preparing and cleaning the data. In this phase where the missing values are handled, features transformation as well as other data preprocessing tasks taken place in order to ensure the data is cleaned and suitable for modeling.
- **Build-in ML Algorithm.** Azure ML Studio Classic provides a range of built-in machine learning algorithms. In this step, the selected algorithm that fits the problem statement is connect to the preprocessed data.
- **Split Data.** To evaluate the performance of selected model, the data are split into training and testing sets. This step allows the data be reserved as portion/batch for model evaluation and validation.
- **Train Model.** Here, where the selected machine learning model are trained using the training dataset. The model learns from the patterns in the data to make predictions or classifications based on the input features.
- **Score Model.** Once the model is trained, the testing dataset is applied in this steps to see how well it generalizes to new, unseen data. This step evaluates the model's performance and gives insights into its accuracy.

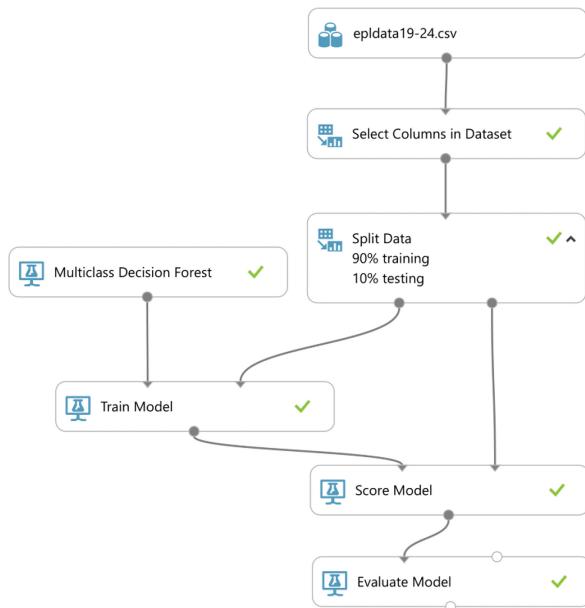


Fig. 2. Azure Pipeline for Football Match Outcome Classification

The experiments were carried out with an 80:20 hold-out validation method for training and testing, respectively. The 80% training portion of the dataset is used to train the machine learning model. Meanwhile, 20% testing portion of the dataset is reserved for evaluating the model's performance. After the model has been trained on the training set, it will be tested on the unseen testing set to assess how well it generalizes to new unseen data.

3.1 Dataset

The dataset for the experiment is sourced from the [Football-Data.co.uk](https://www.football-data.co.uk) website. The dataset contains all information related to football match results for the English Premier League (EPL) from season 2019/2020 to season 2023/2024. The league is made of several $T = 20$ teams and each team has a chance as a home team and away team. There are three possible outcomes of a football match (dependent variables) which are the home team win (H), a draw (D), or the away team win (A). Table 1 provides a list of the types of attributes that take place which are the factors that will manipulate the outcomes of football matches. The excerpt of the dataset is shown in Table 2, where FTR denotes the Full Time Results.

Table 1. Main Factors in Football Match Prediction

No.	Attributes	Notation
1	Home Team Shots	HS
2	Away Team Shots	AS
3	Home Team Shots on Target	HST
4	Away Team Shots on Target	AST
5	Home Team Fouls Committed	HF
6	Away Team Fouls Committed	AF
7	Home Team Corners	HC
8	Away Team Corners	AC
9	Home Team Yellow Cards	HY
10	Away Team Yellow Cards	AY
11	Home Team Red Cards	HR
12	Away Team Red Cards	AR

Table 2. The First 10 Entries of the EPL Dataset

Home Team	Away Team	FTR	HS	AS	HST	AST	HF	AF	HC	AC	HY	AY	HR	AR
Fulham	Arsenal	A	5	13	2	6	12	12	2	3	2	2	0	0
Crystal Palace	Southampton	H	5	9	3	5	14	11	7	3	2	1	0	0
Liverpool	Leeds	H	22	6	6	3	9	6	9	0	1	0	0	0
West Ham	Newcastle	A	15	15	3	2	13	7	8	7	2	2	0	0
West Brom	Leicester	A	7	13	1	7	12	9	2	5	1	1	0	0
Tottenham	Everton	A	9	15	5	4	15	7	5	3	1	0	0	0
Brighton	Chelse	A	13	10	3	5	8	13	4	3	1	0	0	0
Sheffield United	Wolves	A	9	11	2	4	13	7	12	5	2	1	0	0
Everton	West Brom	H	17	6	7	4	9	11	11	1	1	0	0	1
Leeds	Fulham	H	10	4	7	6	13	18	5	3	1	2	0	0

3.2 Algorithms

There are four different classification algorithms used in this study and all of them are available in the Azure Machine Learning Studio.

- **Multiclass Logistic Regression.** Used for multiclass classification problems. It is an extension of binary logistic regression that adds native support for multiclass classification problems [3].
- **Multiclass Neural Network.** A neural network model that is specifically designed to handle multiclass classification tasks. It is used to predict outcomes of football matches by learning complex patterns and relationships within the data [16].
- **Multiclass Decision Forest.** The classification learning method operates by constructing numerous decision trees, which subsequently vote on the most

prevalent output. Each tree within the decision forest generates a frequency histogram without normalization, and the aggregation process involves summing these histograms and normalizing the outcomes to obtain probabilities. [1,8].

- **Multiclass Decision Jungle.** A machine learning algorithm resembling decision forest constructs multiple decision trees and combines their outcomes to enhance predictive accuracy [2].

3.3 Evaluation Metrics

The evaluation metrics used in the experiments are accuracy, precision, and recall. More information related to evaluation metrics is available in the literature [1,5]. The building blocks of these evaluation metrics are as follows:

- **True Positive (TP):** Represents correct predictions for each class.
- **False Positive (FP):** Represents instances where the model predicts a class incorrectly.
- **True Negative (TN):** Represents incorrect predictions for each class.
- **False Negative (FN):** Represents instances where the model fails to predict a class.

Accuracy is the overall correctness of the model which is the ratio of several correct predictions to the total number of input samples. The formula for calculating accuracy is shown in Eq. 1.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Population}} \quad (1)$$

Next, Precision can be defined as the proportion of predicted positive instances that were correctly predicted. The formula for calculating precision is shown in Eq. 2.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (2)$$

Finally, Recall refers to a proportion of actual negative instances that were correctly predicted. The formula for calculating recall is shown in Eq. 3.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (3)$$

4 Result and Discussion

The experiments compared the performance of Multiclass Logistic Regression, Multiclass Neural Network, Multiclass Decision Forest, and Multiclass Decision Jungle algorithm in the football matches outcomes prediction dataset. Figure 3 show the excerpt of detailed results for output of ML algorithm in Azure ML Studio.

Metrics	
Overall accuracy	0.570896
Average accuracy	0.71393
Micro-averaged precision	0.570896
Macro-averaged precision	NaN
Micro-averaged recall	0.570896
Macro-averaged recall	0.485169

Fig. 3. Excerpt of detailed results for output of ML algorithm in Azure ML Studio

The summarized results are shown in Table 3. The results showed that Multiclass Logistic Regression has the highest accuracy (57.08%) followed by multiclass decision forest (56.71%), multiclass neural network (55.59%), and multiclass decision jungle with the same accuracy (55.59%).

Table 3. Overall Prediction Results Across Five Seasons In EPL

Algorithm	Accuracy	Macro-Averaged Precision	Macro-Averaged Recall
Multiclass Logistic Regression	0.5709	NaN	0.4852
Multiclass Neural Network	0.5560	0.4514	0.4822
Multiclass Decision Forest	0.5672	0.5046	0.5149
Multiclass Decision Jungle	0.5560	0.4949	0.4969

Note that Azure measured both micro-averaged and macro-averaged precision and recall. In Table 3, the results for macro-averaged for precision and recall are reported because it computes both precision and recall independently for each class and averages the results. A macro-averaged metric is also desirable for this experiment because the dataset has a balanced class of win, draw, or lose.

The results showed that Multiclass Decision Forest has the advantage of having the highest calculated macro-averaged precision (50.46%) and recall (51.49%) making it more informative in terms of precision across different classes. However, the decision should also consider factors like interpretability, computational complexity, and potential for future improvements. If interpretability is a priority, Logistic Regression might be favored. If the emphasis is on overall performance and the ability to handle imbalances, Decision Forest may be an excellent choice. This models are employed in default without any tuning and if there any tuning, maybe Neural Network can challenge the Logistic Regression and Decision Forest.

In summary, both Multiclass Logistic Regression and Multiclass Decision Forest demonstrate robust performance, and the choice between them should align with the specific requirements and priorities for this task.

5 Conclusion and Future Work

This research presented an analysis of football match prediction results of England Premier League from season 2019/2020 to 2023/2024 based on several multi-class classifications. Several multi-class classifications that take place are multiclass logistic regression, multiclass neural network, multiclass decision forest, and multiclass decision jungle. The performance of the techniques is measured by accuracy, precision, and recall. The results showed that multiclass logistic regression has the highest accuracy (57.08%) followed by multiclass decision forest (56.71%), multiclass neural network, and multiclass decision jungle have the same accuracy (55.59%).

This research holds significant implications not only for football enthusiasts but also for various stakeholders such as coaches, team managers, betting agencies, and sports analysts. Accurate predictions derived from this study can inform strategic decisions and enhance the decision-making process in the dynamic landscape of football competitions. For future works, this study can be broadened by discovering more algorithms, incorporating more complex and sophisticated features of data, leveraging the advantages of machine learning to include statistical models and rating systems as part of data features or even as integrated components within the machine learning model itself. Moreover, exploration on application of ensemble methods and deep learning may offer additional avenues to enhance the model's performance.

Acknowledgements. This research is supported by the University Tun Hussein Onn Malaysia.

References

1. Azeman, A.A., Mustapha, A., Razali, N., Nanthaamomphong, A., Abd Wahab, M.H.: Prediction of football matches results: decision forest against neural networks. In: 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 1032–1035 (2021). <https://doi.org/10.1109/ECTI-CON51831.2021.9454789>
2. Baboota, R., Kaur, H.: Predictive analysis and modelling football results using a machine learning approach for the English Premier League. Int. J. Forecast. **35**(2), 741–755 (2019). <https://doi.org/10.1016/j.ijforecast.2018.01.003>
3. Brownlee, J.: Multinomial logistic regression with python (2021). <https://machinelearningmastery.com/multinomial-logistic-regression-with-python/>
4. Choi, B.S., Foo, L.K., Chua, S.L.: Predicting football match outcomes with machine learning approaches. Mendel **29**(2), 229–236 (2023). <https://doi.org/10.13164/mendel.2023.2.229>

5. Grandini, M., Bagli, E., Visani, G.: Metrics for multi-class classification: an overview, pp. 1–17 (2020)
6. Hubáček, O., Šourek, G., Železný, F.: Learning to predict soccer results from relational data with gradient boosted trees. *Mach. Learn.* **108**(1), 29–47 (2019). <https://doi.org/10.1007/s10994-018-5704-6>
7. Hubáček, O., Šourek, G., Železný, F.: Forty years of score-based soccer match outcome prediction: an experimental review. *IMA J. Manag. Math.* **33**(1), 1–18 (2021). <https://doi.org/10.1093/imaman/dpab029>
8. Learning, A.M.: Multiclass decision forest component more about decision forests how to configure multiclass decision forest (2021). <https://docs.microsoft.com/en-us/azure/machine-learning/algorithim-module-reference/multiclass-decision-forest>
9. Pantzalis, V.C., Tjortjis, C.: Sports analytics for football league table and player performance prediction. In: 11th International Conference on Information, Intelligence, Systems and Applications (IISA) (2020). <https://doi.org/10.1109/IISA50023.2020.9284352>
10. Price, E., Masood, A., Aroraa, G.: Azure machine learning. In: Hands-on Azure Cognitive Services, pp. 321–354 (2021). https://doi.org/10.1007/978-1-4842-7249-7_10
11. RaginiSingla, D.A.S.: Sports prediction using machine learning. *JETIR* **2020**(10), 3862–3866 (2020). <https://doi.org/10.1190/segam2020-w13-04.1>
12. Razali, N., Mustapha, A., Yatim, F.A., Ab Aziz, R.: Predicting football matches results using Bayesian networks for English Premier League (EPL). *IOP Conf. Ser. Mater. Sci. Eng.* **226**(1) (2017). <https://doi.org/10.1088/1757-899X/226/1/012099>
13. Rodrigues, F., Pinto, Â.: Prediction of football match results with machine learning. *Procedia Comput. Sci.* **204**, 463–470 (2022). <https://doi.org/10.1016/j.procs.2022.08.057>
14. Sjöberg, F.: Football match prediction using machine learning (2023)
15. Developer Support: Exploring feature weights using R and Azure machine learning studio. Blog (2019). <https://devblogs.microsoft.com/premier-developer/exploring-feature-weights-using-r-and-azure-machine-learning-studio/>
16. You, Y.J., Wu, C.Y., Lee, S.J., Liu, C.K.: Intelligent neural network schemes for multi-class classification. *Appl. Sci.* **9**(19) (2019). <https://doi.org/10.3390/app9194036>



An Automated Quasi-Identification (QID) for Re-identification

Saida Nafisah Roslan¹, Isredza Rahmi A Hamid¹(✉), Abdulbasit A. Darem², and Nordiana Rahim¹

¹ Department of Information Security and Web Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, Johor, Malaysia

saidanafisahroslan@gmail.com, {rahmi, nordiana}@uthm.edu.my

² Department of Computer Science, Northern Border University, Arar, Saudi Arabia

basit.darem@nbu.edu.sa

Abstract. The rapid growth of data generation and technology has led to concerns about security and privacy. Personal and sensitive information can be disclosed by unauthorized individuals, necessitating the use of privacy-preserving techniques. In this study, an automated quasi-identification (QID) approach for privacy preservation is proposed. The approach aims to select appropriate QID based on re-identification risk in each attribute, reducing subjective judgment and minimizing data loss. The methodology involves data processing, frequency calculation, re-identification risk score computation, QID selection, and evaluation using data anonymization tool used for privacy-preserving data processing and analysis. The selection process considers the likelihood of information being disclosed through a single record in each attribute based on an attack where attackers are aware of the victim's background or a randomly selected victim whose information is available in a published or external dataset. The effectiveness of the approach is evaluated on two datasets: the adult dataset and the bank marketing dataset. The results demonstrate a significant reduction in re-identification risks and success rates of attacker models after preserving QID attributes. The proposed approach offers a valuable solution for privacy preservation while minimizing data loss and maintaining data utility.

Keywords: Quasi-identifier · re-identification risk · privacy preserving

1 Introduction

A significant concern in data security is the inadvertent or deliberate exposure of private and sensitive information, which can be exploited by unauthorized parties or attackers, resulting in serious repercussions for individuals and organizations. To address this challenge, various methods have been devised to uphold data integrity and protect personal privacy. Among these methods, de-identification, anonymization, and perturbation have emerged as essential tools for minimizing the risk of privacy breaches. A fundamental aspect of data protection involves categorizing data elements into distinct groups, such as direct identifier (DID), quasi-identifier (QID), sensitive attribute (SA), and non-sensitive

attributes (NSA), each playing a crucial role in formulating effective privacy-preserving measures [1].

Direct identifiers, like names or social security numbers, directly identify individuals, while quasi-identifiers, such as age or ZIP codes, may indirectly identify individuals when combined with other information. Sensitive attributes include confidential data like medical conditions or salary, whereas non-sensitive attributes do not reveal or harm individuals' identities. Re-identification risk refers to the potential for individuals in a dataset to be re-identified or linked to their personal information, even without direct identifiers. This risk arises from certain combinations of quasi-identifiers or other data elements, posing a threat to privacy by disclosing sensitive information or enabling unauthorized tracking and profiling of individuals. Data anonymization techniques offer sophisticated methodologies to mitigate this threat.

However, the selection of suitable quasi-identifiers for anonymization raises concerns, as manually choosing them carries inherent risks. This study introduces an automated quasi-identification framework to enhance privacy protection, aiming to improve the effectiveness of the de-identification process while minimizing re-identification risks. Unlike traditional approaches [2], which calculate re-identification risk based on equivalence class size, this paper introduces a formula for assessing risk in singleton attributes, providing a more sophisticated methodology. Additionally, it presents a method to compute the frequency of unique values within each attribute, enabling an evaluation of their uniqueness and impact on re-identification risk.

The remainder of this paper is structured as follows: Sect. 2 discusses previous work in privacy preservation in data mining and publishing, while Sects. 3 and 4 delve into the research work and experimental implementation in detail.

2 Related Work

Privacy-preserving techniques like k-anonymity, I-diversity, t-closeness, and perturbation are commonly used to safeguard sensitive data. The k-anonymity model, as per [3], ensures that at least k individuals share the same quasi-identifier value to reduce re-identification risk. Later, I-diversity and t-closeness models were introduced to address limitations of the k-anonymity model [4] and [5]. Meanwhile, [6] focuses on incrementally preserving data using anonymization methods but does not delve into QID selection. They assume pre-defined QID based on dataset attributes. On the other hand, [7] proposes a determination scheme that prioritizes selecting quasi-identifier based on uniqueness and influence metrics, aiming to identify attributes posing higher re-identification risk. The scheme's effectiveness is evaluated using real clinical datasets, assessing impact on re-identification risk and data utility. However, it may vary based on dataset characteristics and domain relevance, and does not explicitly consider other factors like data linkage or background knowledge. Various methods exist for QID selection, like selecting primary QID based on unique value of power set element [8]. It is described as simple and straightforward but lacks extensive discussion on specific limitations. Considering limitations or trade-offs is crucial. The method's effectiveness may vary based on dataset characteristics and may require tailored approaches for different datasets. As per [9], an approach proposes an algorithm to identify the best quasi-identifier by counting unique

combinations of singleton values in the dataset. However, it is computationally inefficient and relies heavily on choice of QID and generalization strategies. The research in [10] proposes a privacy-preserving collaborative data anonymization approach with sensitive quasi-identifiers, emphasizing the need for careful consideration when choosing QID attributes to strike a balance between data utility and privacy preservation during collaborative data sharing.

3 Methodology

Quasi-identification is a widely used method for safeguarding sensitive data by replacing DID with QID that can still be used to re-identify individuals. However, choosing the right QID is challenging, as it involves balancing privacy preservation with data usefulness. Our proposed automated QID methodology as the following steps:

1. Data Processing: This initial step involves standardizing the dataset to correct inconsistencies in values, ensuring uniformity, and eliminating redundancy. For instance, variations like “male” and “Male” representing the same gender are standardized.
2. Frequency Calculation: Next, the frequency of distinct tuples in each attribute is computed to gauge their uniqueness. This information helps in assessing the risk of re-identification and understanding the sensitivity of attributes.
3. Re-identification Risk Score Computation: Each tuple’s re-identification score is calculated based on the frequency of distinct tuples within each attribute. Higher scores indicate a greater risk to privacy and assist in prioritizing protection efforts.
4. QID Selection: Quasi-identifier are chosen by identifying the attribute value with the highest risk score, prioritizing privacy protection. This stage helps measure the probability of re-identification for the entire dataset and guides selection by considering the maximum risk score across all records.
5. ARX Evaluation: The effectiveness of QID selection in preserving privacy while maintaining data utility is evaluated using the ARX tool, which offers various techniques and algorithms for privacy-preserving data anonymization. It ensures sensitive information protection while retaining dataset utility for analysis and research.

3.1 Selection of QID

In real-world situations, the chance of re-identification for direct identifiers is typically assumed to be 1. However, in clinical trial data collections, the presence of direct identifiers poses a significant risk of re-identification, and regulatory medical requirements mandate their deletion [2]. As a result, it is often necessary to focus on quasi-identifiers, which offer greater data utility, and assume a conservative estimate of re-identification likelihood as 1.

The standard approach to calculating the risk of re-identification for a record in a data collection involves dividing the size of its equivalence class by 1. However, this paper proposes a formula to identify re-identification risk in singletons of attributes.

Algorithm 1 Re-identification risk of each attribute

```

Input: Dataset
Output: Classified QID
// Step 1: Calculate frequency of distinct records in each attribute
1. Start
    // Declaration and initialization
2. n <- record to check
3. m <- column size
4. CountFreq(n, m) // Call the function to count frequencies
5. count <- 1
    // Inside loop, count frequency in column size
6. for i = 1; i < m; i++
7.   for j = i + 1; j <= m; j++
8.     if (n[i] == n[j])
9.       visited[j] = true
10.    count++
11. y <- count // corrected assignment
// Step 2: Calculate re-identification risk in each attribute
    // Declaration and initialization
12. Rz <- 0
    // Calculate re-identification risk score of attributes
    // Perform calculation for each frequency distinct record
13. for Ri = 1; Ri <= y; Ri++ // corrected loop initialization
14.   Rz <- Rz + 1/y
// Step 3: Determine attribute based on max value of probability of re-identification
15. If Rz > T
16. Then classify as QID
17. Else END
18. END

```

This adapted algorithm for identifying quasi-identifiers in a dataset follows a systematic process to calculate frequencies, assess re-identification risk, and classify attributes as quasi-identifiers based on a predetermined threshold. In this algorithm, if the re-identification risk score (Rz) is higher than the specified threshold (T) but still less than or equal to 1, the attribute is classified as a quasi-identifier. The algorithm systematically calculates the frequency of distinct records within each attribute by tallying occurrences of unique values. Then, it computes the re-identification risk score (Rz) by summing the inverses of the frequency for each distinct record. If the computed risk score exceeds the threshold, the attribute is classified as a quasi-identifier. This decision-making process recognizes that a higher risk score indicates a greater potential for re-identification. Once a classification decision is reached, the algorithm concludes its processing for the given attribute. The algorithm identifies an attribute as a quasi-identifier if its re-identification risk score surpasses the specified threshold (T), even if the risk score falls within the range of $(0, 1)$.

4 Experiment Evaluation

The proposed experiment will be conducted using Windows 7 Ultimate with 4.00 GB RAM, ARX Tool version 3.8.0 for data anonymization. The experiment involves implementing simulation attack model which are prosecutor, journalist, and marketer on selected QID attributes within the dataset.

In the prosecutor model [11], the attacker targets specific individuals with prior knowledge of their presence in the dataset, aiming to identify or re-identify their information. The journalist model also targets specific individuals but without prior knowledge of their inclusion in the dataset. The goal remains to re-identify the individuals' information, though without access to background information. Lastly, the marketer model focuses on re-identifying a larger number of individuals within the dataset without targeting specific individuals. Success in this model is determined by the ability to re-identify a significant fraction or percentage of the records in the dataset, without prior knowledge of the individuals.

In an experiment using the ARX tool, users typically select attributes and apply anonymization methods. In the ARX tool, users manually identify and choose attributes to preserve in the dataset. The selected QID attributes, derived from re-identification risk calculation, are preserved, while the rest are classified as non-sensitive attributes. In this experiment, the chosen anonymization method is k-anonymization, a widely-used privacy protection technique. K-anonymity ensures that each record in the dataset is indistinguishable from at least $k-1$ other records concerning the selected QID attributes. Here, $k = 5$ is applied, following common settings found in previous studies [12, 13]. Other common parameterizations for k-anonymity are $k = 2, 3, 5$, and 10 , corresponding to thresholds for prosecutor re-identification risk of 50% , 33% , 20% , and 10% , respectively.

Our proposed method is tested on two publicly available datasets: the adult and bank marketing datasets, both sourced from the UCI Machine Learning repository [14]. Previous research studies have also utilized these datasets in their experiments [4] and [15]. It is important to note that merely examining the different attribute values in tables doesn't automatically reveal attributes that could lead to re-identification. Although tables provide an overview of diversity within each attribute, assessing re-identification risk requires a deeper analysis, considering factors like uniqueness, frequency, and contribution to overall risk. Identifying attributes with re-identification potential typically involves a more detailed evaluation than just counting distinct values.

Table 1. Adult dataset attribute with distinct value and re-id risk value

Attribute	Distinct value	Re-id risk
Condition	2	0.000167
Gender	2	0.001387
Race	6	0.008220

(continued)

Table 1. (*continued*)

Attribute	Distinct value	Re-id risk
Relationship	7	0.023410
Education	17	0.043110
Education num	17	0.043110
Marital status	8	0.048240
Occupation	16	0.125419
Native country	30	0.592650
Age	73	3.501600
Hours per week	95	12.664850
Capital gain	119	26.254900
Capital loss	93	27.471000
Final weight	21647	17957.153

Table 2. Bank marketing dataset attribute with distinct value and re-id risk value

Attribute	Distinct value	Re-Id Risk
Default	2	0.0060
Day	31	0.1023
Marital	3	0.0012
Loan	2	0.0008
Education	4	0.0031
Job	12	0.0336
Housing	2	0.0036
Month	12	0.0272
Age	76	6.5190
Balance	3806	2464.6790
Duration	1428	558.3430

Table 1 displays the re-identification risk values for each attribute in the adult dataset. The highest risk value is linked to the final weight attribute, indicating a greater likelihood of re-identification due to numerous unique records. A threshold of 1 is used to identify quasi-identifying attributes. Attributes with risk scores above this threshold, namely age, hours per week, capital loss, capital gain, and final weight, are potential QIDs. Meanwhile, Table 2 provide a list of attributes from bank marketing dataset along with their re-identification risk scores. Attributes like default, marital, loan, education, housing, and month have relatively low risk scores. Attributes like day and job have

slightly higher risk scores but still remain below the threshold. However, attributes such as age, balance, and duration exhibit significantly elevated risk scores surpassing the threshold, indicating a high risk of facilitating re-identification and potential privacy vulnerabilities. Attributes with risk scores below 1 are generally considered safe, while those exceeding this threshold require special attention due to their heightened potential for re-identification risk. This evaluation helps determine which attributes could serve as quasi-identifiers and informs privacy protection strategies.

5 Result

In this study, we introduce a method to improve privacy and reduce the risk of re-identification by generalizing selected quasi-identifiers, as outlined in [3]. Our goal is to minimize subjective judgments and data loss by choosing appropriate quasi-identifiers based on re-identification risk in each attribute. We evaluate the effectiveness of our approach using various metrics within the ARX tool.

Table 3 displays the analysis of the adult dataset. Initially, the prosecutor and journalist models pose a risk to 99.98% of records, with a highest risk score of 100% for a single record. However, after preserving the Quasi-Identifier (QID) attributes, the percentage of at-risk records drops to 0%, and the highest risk score decreases to 5.882%. Moreover, the success rate of both models significantly decreases from 97.7% to 0.025%. Meanwhile, the Marketer model's success rate drops from 97.7% to 0.025% after QID preservation. The information loss score for this dataset is calculated as 0.765, indicating the amount of information lost due to QID preservation.

The same three attacker models are applied to the bank marketing dataset. Initially, all three models put 100% of records at risk, with a highest risk score of 4.762%. After preserving the QID attributes, the percentage of at-risk records decreases to 0%, and the highest risk score decreases to 4.762%. At the same time, the success rate of all three models experiences a substantial drop from 99.8% to 0.045%. The information loss score for this dataset is calculated as 0.731, indicating a relatively lower level of information loss compared to the adult dataset. These findings demonstrate the effectiveness of our approach in reducing re-identification risks and the success rates of the attacker models. By selecting appropriate QID attributes based on re-identification risk, the approach successfully preserves privacy while minimizing the loss of valuable data.

Table 3. Comparison of preservation techniques for different attacker models of adult dataset

Dataset	Attacker Model		Before Preserve (%)	After Preserve (%)	Information loss score
Adult dataset	Prosecutor	Record At Risk	99.98	0	0.765
		Highest Risk	100	5.882	
		Success Rate	97.70	0.025	

(continued)

Table 3. (*continued*)

Dataset	Attacker Model		Before Preserve (%)	After Preserve (%)	Information loss score
Bank marketing dataset	Journalist	Record At Risk	99.98	0	0.731
		Highest Risk	100	5.882	
		Success Rate	97.70	0.025	
	Marketer	Success Rate	97.70	0.025	
	Prosecutor	Record At Risk	100	0	
		Highest Risk	100	4.762	
		Success Rate	99.80	0.045	
	Journalist	Record At Risk	100	0	
		Highest Risk	100	4.762	
		Success Rate	99.80	0.045	
	Marketer	Success Rate	99.80	0.045	

Table 4 compares our QID selection method with commonly used ones in literature [9, 16] and [17]. It shows the risk of re-identification for a single record before and after data preservation, along with the information loss score. In the adult dataset, the initial risk for the first set of QID attributes, Age and Sex, was 100%. After preserving these attributes in ARX, the risk dropped significantly to 12.5%, indicating a major privacy improvement. The information loss score decreased to 0.045, suggesting effective preservation of data utility. For the second set of QIDs, including age, work per week, native country, education, education num, occupation, marital status, relationship, and race, and the initial highest risk was 100%. After preservation in ARX, the highest risk dropped to 7.692%, still showing a significant privacy improvement. However, the information loss score slightly increased to 0.787, indicating a bit more loss in data utility compared to the first set of QIDs. The third set, an “Automated QID,” comprising Age, hours per week, capital loss, capital gain, and final weight, saw the highest risk drop to 5.882% after preservation in the ARX tool. This indicates a meaningful improvement in privacy while maintaining a reasonable level of data utility. The information loss score for this set of QIDs was 0.765, indicating a balanced compromise between privacy and data utility.

Table 4. Comparison of risk in single record before and after privacy preservation adult datasets

Dataset	Work	Attribute	Highest risk of single record		Information loss score
			Before Preserve (%)	After Preserve (%)	
Adult dataset	[9, 16]	Age, Sex	100	12.5	0.045
	[17]	Age, work per week, native country, education, education num, occupation, marital status, relationship and race	100	7.692	0.787
	Automated QID	Age, hours per week, capital loss, capital gain, final weight	100	5.882	0.765
Bank Marketing dataset	[9, 16]	Age	100	14.286	0.0909
	[17]	Age, job, education and marital status	100	11.111	0.757
	Automated QID	Age, balance, duration	100	4.762	0.731

For the bank marketing dataset in Table 4, the initial highest risk of re-identification for the first set, which includes only the age attribute, was 100%. After preserving the selected QID in ARX, the highest risk decreased to 14.286%, showing a substantial improvement in dataset privacy. Simultaneously, the information loss score significantly improved to 0.0909, indicating a relatively favorable trade-off between enhanced privacy and data utility. The second set of QIDs, including age, job, education, and marital status, initially had the highest risk of 100%. After preservation, the highest risk decreased to 11.111%, demonstrating a meaningful enhancement in privacy. The information loss score increased to 0.757, indicating a moderate compromise between privacy and data utility compared to the first set. The third set, an “Automated QID,” comprising age, balance, and duration, saw the highest risk drop to 4.762% after preservation in the ARX tool. This indicates a significant improvement in privacy while maintaining a reasonable level of data utility. The information loss score for this set of QIDs was 0.731, showing a balanced trade-off between privacy and data utility.

6 Conclusion

In conclusion, this research tackles the pressing issue of preserving privacy amidst increasing data volume and technological advancements. The proposed automated Quasi-identification method offers an objective and systematic way to select Quasi-identifiers based on re-identification risk in each attribute. By calculating re-identification risk scores and using a threshold-based approach, the study notably reduces the risk of individual re-identification while minimizing data loss. Unlike traditional methods that rely on subjective judgments, the automated approach eliminates human bias and enhances the consistency of privacy-preserving decisions. Lastly, several promising directions for future work include exploring and developing more sophisticated risk score formulations that consider not only the frequency of distinct attribute values but also the interdependence between different attributes. This could lead to a more precise estimation of re-identification risk and better support for complex data structures. Additionally, addressing scalability issues when applying the proposed approach to larger datasets and developing techniques to handle big data efficiently while ensuring manageable computational costs without compromising privacy preservation effectiveness are vital considerations. Develop techniques that can handle big data efficiently, ensuring that the computational cost remains manageable without sacrificing the effectiveness of privacy preservation.

Acknowledgements. This research was supported by Universiti Tun Hussein Onn Malaysia (UTHM) through TIER 1 (Vot. Q167). The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number “NBU-FFR-2023-xxxx”.

References

1. Zigomitros, A., Casino, F., Solanas, A., Patsakis, C.: A survey on privacy properties for data publishing of relational data. *IEEE Access* **8**, 51071–51099 (2020). <https://doi.org/10.1109/ACCESS.2020.2980235>
2. Committee on Strategies for Responsible Sharing of Clinical Trial Data, Board on Health Sciences Policy, and Institute of Medicine, *The Roles and Responsibilities of Stakeholders in the Sharing of Clinical Trial Data* (2015)
3. Sweeney, L.: k -anonymity: a model for protecting privacy 1, vol. 10, no. 5, pp. 1–14 (2002)
4. Machanavajjhala, J.G.A.: ℓ -Diversity : Privacy Beyond k -Anonymity, vol. 1
5. Li, N.: t -Closeness : Privacy Beyond k -Anonymity and -Diversity,” no. 2
6. Nadagoudar, R.: A scalable two-phase top-down specialization approach for data anonymization using mapreduce on cloud. *Ijarcce* **4**(5), 541–546 (2015). <https://doi.org/10.17148/ijarcce.2015.45116>
7. Jung, J., Park, P., Lee, J., Lee, H., Lee, G.K., Cha, H.S.: A determination scheme for quasi-identifiers using uniqueness and influence for de-identification of clinical data. *J. Med. Imaging Heal. Inf.* **10**(2), 295–303 (2019). <https://doi.org/10.1166/jmhi.2020.2966>
8. Omer, A.M., Bin Mohamad, M.M.: Simple and effective method for selecting quasi-identifier. *J. Theor. Appl. Inf. Technol.* **89**(2), 512–517 (2016)
9. Pastore, M., Pellegrino, M.A., Scarano, V.: Detecting and generalizing quasi-identifiers by affecting singletons. *CEUR Workshop Proc.* **2797**, 327–335 (2020)

10. Wong, K.S., Tu, N.A., Bui, D.M., Ooi, S.Y., Kim, M.H.: Privacy-preserving collaborative data anonymization with sensitive quasi-identifiers. In: 2019 12th C. Conference Cybersecurity Privacy, C. 2019 [8962140] (2019 12th C. Conf. Cybersecurity Privacy, C. (2019). <https://doi.org/10.1109/CMI48017.2019.8962140>
11. B. and C. Supported by TUM, “ARX – Data Anonymization Tool.” <https://arx.deidentifier.org/anonymization-tool/analysis/>. Accessed 10 Jul. 10 2023
12. Prasser, F., Eicher, J., Spengler, H., Bild, R., Kuhn, K.A.: Flexible data anonymization using ARX—Current status and challenges ahead. *Softw. - Pract. Exp.* **50**(7), 1277–1304 (2020). <https://doi.org/10.1002/spe.2812>
13. Tomás, J., Rasteiro, D., Bernardino, J.: Data anonymization: an experimental evaluation using open-source tools. *Futur. Internet* **14**(6) (2022). <https://doi.org/10.3390/fi14060167>
14. Dua, C., Dheeru, G.: UCI Machine Learning Repository (2017)
15. Khan, S., Iqbal, K., Faizullah, S., Fahad, M., Ali, J., Ahmed, W.: Clustering based privacy preserving of big data using fuzzification and anonymization operation. *Int. J. Adv. Comput. Sci. Appl.* **10**(12), 282–289 (2019). <https://doi.org/10.14569/ijacsa.2019.0101239>
16. Touhidul Hasan, A.S.M., Jiang, Q.: A general framework for privacy preserving sequential data publishing. In: Proceedings 31st International Conference on Advanced Information Networking and Applications Workshops WAINA 2017, no. June, pp. 519–524 (2017). <https://doi.org/10.1109/WAINA.2017.18>
17. Mansour, H.O., Siraj, M.M., Ghaleb, F.A., Saeed, F., Alkhammash, E.H., Maarof, M.A.: Quasi-identifier recognition algorithm for privacy preservation of cloud data based on risk reidentification. *Wirel. Commun. Mob. Comput.* **2021** (2021). <https://doi.org/10.1155/2021/7154705>



Correction to: Predicting Undergraduate Academic Success with Machine Learning Approaches

Yuan-Zheng Li, Keng-Hoong Ng , Kok-Chin Khor , and Yu-Hsuen Lim

Correction to:

Chapter 15 in: R. Ghazali et al. (Eds.): *Recent Advances on Soft Computing and Data Mining*, LNNS 1078,
https://doi.org/10.1007/978-3-031-66965-1_15

In the original version of the book, the following belated corrections have been incorporated: The author name “Juan-Cheng Li” has been changed to “Yuan-Zheng Li” in the Frontmatter, Backmatter and in Chapter 15.

The updated version of this chapter can be found at
https://doi.org/10.1007/978-3-031-66965-1_15

Author Index

A

- A Hamid, Isredza Rahmi 421
Abdul Rahman, Nurul Habibah 379
Abirami, K. Rama 53
Abu, Noratikah 360
Abualhaj, Mosleh M. 244, 254
Abu-Shareha, Ahmad Adel 254
Adeleke, Abdullahi O. 196
Adnan, Nur Amirah 411
Agagu, Modupe 1, 11
Akmaluddin, Muhammad 349
Al Hakim Mohd Asri, Luqman 411
Al-Allawee, Ali 244, 254
Alawatugoda, Janaka 105
Alharith, Razan 370
Ali, Ashikin 337
Al-Khatib, Sumaya N. 244
Al-Rimy, Bander Ali Saleh 186
Anbar, Mohammed 244, 254
Arbain, Siti Hajar 307
Arbaiy, Nureize 94, 136, 296, 400
Aziz, Noor Azeera Abdul 186

B

- Batool, Tuba 74, 307

C

- Che Lah, Muhammad Shukri 400
Chia, Kai Lin 175
Chua, Chi Log 23
Chuah, Chai-Wen 105

D

- Darem, Abdulbasit A. 421
Dewi, Deshinta Arrova 207

E

- Efendi, Riswan 64, 94, 196
Elsafi, Abubakar 370

F

- Fadhel, Abdulrahman Sharaf Mohammed 266

G

- Gan, Yee Jing 390
Ghaleb, Fuad Abdulgaleel Abdoh 186
Ghazali, Rozaida 1, 74, 125, 136, 266, 307, 370, 390
Goh, Pei-Jin 43

H

- Hariz, Hussein Muhi 125
Harun, Nur Ziadah 33
Hassan, Abdullahi Abdi Abubakar 266
Hassan, Mustafa Hamid 125
Hassan, Norlida 74
Hassim, Yana Mazwin Mohammad 74, 136, 266, 390
He, WanXian 105
Hoo, Meei-Hao 43
Huang, De-Shuang 105
Husaini, Noor Aida 74, 125, 390

I

- Ibrahim, Ashraf Osman 370
Ismail, Lokman Hakim 266, 307
Ismail, Mohd Tahir 115
Ismail, Shahrinaz 219

J

- Jafri, Syed Irteza Hussain 74
Javid, Irfan 74, 136, 307
Joseph, Jerome Subash 390
Jubair, Mohammed Ahmed 125

K

- Kasim, Shahreen 207
Kasinathan, Vinothini 233
Khalid, Shamsul Kamal A. 196

- K**
 Khant, Kyi Lin 219
 Khor, Kok-Chin 43, 144, 165
 Krishnan, Nor Farisha Binti Muhamad 317
 Kurniawan, Tri Basuki 207
- L**
 Lasisi, Ayodele 1, 11
 Li, Yuan-Zheng 144
 Liang, Zhuqin 115
 Liew, Jun-Yen 165
 Lim, Tong Ming 23
 Lim, Yu-Hsuen 144
 Lin, Pei-Chun 296, 400
- M**
 Mausor, Farahida Hanim Binti 317
 Mohamed, Rozlina 349
 Mohamed, Rozlini 276
 Mohd Zin, Nur Ariffin 1
 Mostafa, Salama A. 125
 Muda, Muhammad Zaiyad 84
 Munther, Alhamza 244, 254
 Murli, Norhanifah 337
 Mustapa, Nur Qasheeh 207
 Mustapha, Aida 125, 233, 411
- N**
 Nawi, Nazri Mohd 154
 Nawi, Rosmamalmi Mat 33
 Ng, Keng-Hoong 144, 165
- O**
 Ogunbiyi, Ibrahim Abayomi 11
 Omorogiuwa, Osaremwinda 11
 Osman, Nurul Aida 33
 Othman, Mohammad Haris Haikal 400
 Othman, Muhamaini 286
- P**
 Panoutsos, George 84
 Praditya, Farrel Yuda 53
 Pratiwi, Puspa Setia 53
 Pratondo, Agus 360
- Q**
 Qu, Huimin 115
- R**
 Rachmawati, Ummi Azizah 53
- Rahim, Nordiana 421
 Rahmi, Izzati 64, 94
 Rajendran, Piraviendran a/l 286
 Ramli, Nor Azuana 360, 379
 Razali, Muhammad Nazim 411
 Rejab, Mazidah Mat 94
 Roslan, Saida Nafisah 421
- S**
 Saba, Noushin 154
 Salikon, Mohd Zaki Mohd 296
 Salleh, Syahrizal 327
 Samat, Nor Azah 64
 Samsuddin, Noor Azah 276
 Samsudin, Noor A. 196
 Saringat, Mohd Zainuri 125
 See, Kwee Teck 23
 Senan, Norhalina 337
 Shahab, Muhammad 136
 Shahabudin, Mohd Safuwan Bin 317
 Shen Yeap, Jie 390
 Sulaiman, Sahimel Azwal 360, 379
 Suleiman, Mohsin 154
- T**
 Tee, Kai-Yau 165
 Tomari, Mohd Razali Md 266
- W**
 Wahid, Noorhaniza 370
 Wahyudi, Muhammad 64
 Widyawati, Sri R. 94
- X**
 Xuan, Yui Chee 33
- Y**
 Yau, Jia Xin 175
 Yaziz, Siti Roslindar 327
 Yofi, Widya T. 94
 Yozza, Hazmira 64, 94
 Yu, Chen-Yu 296
 Yusnita, 53
 Yusof, Munirah Mohd 276
 Yusoff, Wan Nur Syahidah Wan 360
- Z**
 Zafar, Afia 154
 Zafar, Kainat 154

Zafar, Shahneer [154](#)
Zainal, Anazida [186](#)

Zakaria, Mohd Zaki [207](#)
Zakaria, Roslinazairimah [327](#)