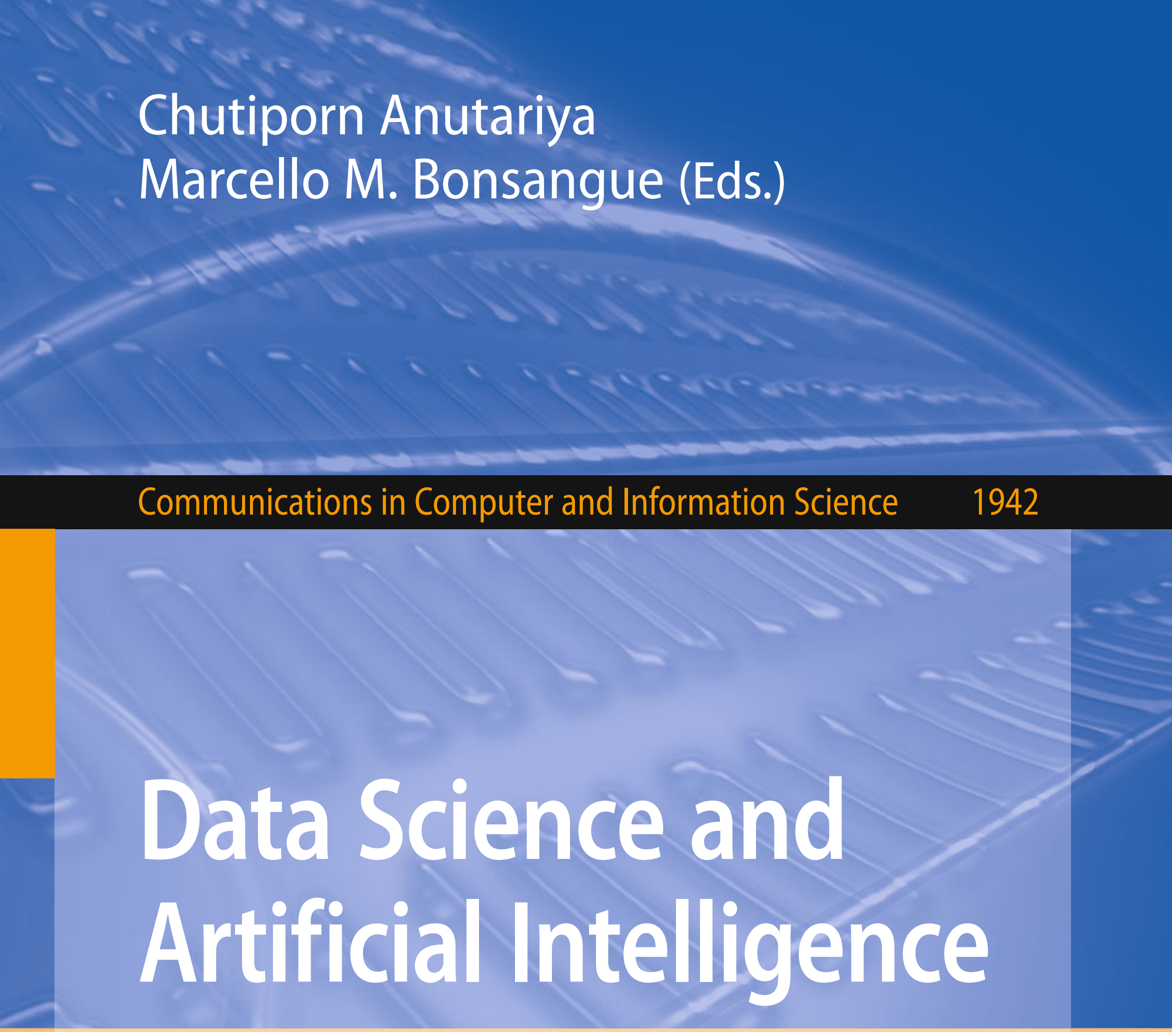


Chutiporn Anutariya
Marcello M. Bonsangue (Eds.)

Communications in Computer and Information Science 1942

Data Science and Artificial Intelligence

First International Conference, DSAI 2023
Bangkok, Thailand, November 27–29, 2023
Proceedings

Springer

DS&AI

Communications in Computer and Information Science 1942

Editorial Board Members

Joaquim Filipe, *Polytechnic Institute of Setúbal, Setúbal, Portugal*
Ashish Ghosh, *Indian Statistical Institute, Kolkata, India*
Raquel Oliveira Prates, *Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil*
Lizhu Zhou, *Tsinghua University, Beijing, China*

Rationale

The CCIS series is devoted to the publication of proceedings of computer science conferences. Its aim is to efficiently disseminate original research results in informatics in printed and electronic form. While the focus is on publication of peer-reviewed full papers presenting mature work, inclusion of reviewed short papers reporting on work in progress is welcome, too. Besides globally relevant meetings with internationally representative program committees guaranteeing a strict peer-reviewing and paper selection process, conferences run by societies or of high regional or national relevance are also considered for publication.

Topics

The topical scope of CCIS spans the entire spectrum of informatics ranging from foundational topics in the theory of computing to information and communications science and technology and a broad variety of interdisciplinary application fields.

Information for Volume Editors and Authors

Publication in CCIS is free of charge. No royalties are paid, however, we offer registered conference participants temporary free access to the online version of the conference proceedings on SpringerLink (http://link.springer.com) by means of an http referrer from the conference website and/or a number of complimentary printed copies, as specified in the official acceptance email of the event.

CCIS proceedings can be published in time for distribution at conferences or as post-proceedings, and delivered in the form of printed books and/or electronically as USBs and/or e-content licenses for accessing proceedings at SpringerLink. Furthermore, CCIS proceedings are included in the CCIS electronic book series hosted in the SpringerLink digital library at http://link.springer.com/bookseries/7899. Conferences publishing in CCIS are allowed to use Online Conference Service (OCS) for managing the whole proceedings lifecycle (from submission and reviewing to preparing for publication) free of charge.

Publication process

The language of publication is exclusively English. Authors publishing in CCIS have to sign the Springer CCIS copyright transfer form, however, they are free to use their material published in CCIS for substantially changed, more elaborate subsequent publications elsewhere. For the preparation of the camera-ready papers/files, authors have to strictly adhere to the Springer CCIS Authors' Instructions and are strongly encouraged to use the CCIS LaTeX style files or templates.

Abstracting/Indexing

CCIS is abstracted/indexed in DBLP, Google Scholar, EI-Compendex, Mathematical Reviews, SCImago, Scopus. CCIS volumes are also submitted for the inclusion in ISI Proceedings.

How to start

To start the evaluation of your proposal for inclusion in the CCIS series, please send an e-mail to ccis@springer.com.

Chutiporn Anutariya · Marcello M. Bonsangue
Editors

Data Science and Artificial Intelligence

First International Conference, DSAI 2023
Bangkok, Thailand, November 27–29, 2023
Proceedings

Springer

Editors
Chutiporn Anutariya
Asian Institute of Technology
Pathum Thani, Thailand

Marcello M. Bonsangue
Leiden University
Leiden, The Netherlands

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-981-99-7968-4 ISBN 978-981-99-7969-1 (eBook)
https://doi.org/10.1007/978-981-99-7969-1

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Singapore Pte Ltd. 2023

This work is subject to copyright. All rights are reserved 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 Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Paper in this product is recyclable.

Preface

This volume contains the papers presented at the first edition of the International Conference on Data Science and Artificial Intelligence (DSAI 2023), held on November 27–29, 2023 in Bangkok, Thailand. Data science and artificial intelligence are shaping the way we perceive, analyze, and harness data to address complex challenges in various domains, from healthcare and finance to transportation and education. The main goal of the DSAI conference series is to contribute to the growth of data science and artificial intelligence knowledge in developing countries in the Southeast Asia region. DSAI 2023 provided a platform for researchers and practitioners to exchange ideas, methodologies, and cutting-edge research that can drive progress and foster interdisciplinary collaboration in the fields of data science and artificial intelligence. By raising awareness and interest directly in the local research community, we believe that DSAI 2023 stimulated the development of data science and artificial intelligence research and application.

The success of the DSAI 2023 inaugural edition is a demonstration of the interest and enthusiasm surrounding the conference theme. We received an overwhelming response, with a total of 70 submissions. After a rigorous single-blind review process, 22 regular papers and 4 short papers were selected for presentation and inclusion in these proceedings. Each submission was reviewed by at least 3 program committee members or their subreviewers. The program also includes 2 keynote talks, one invited presentation, and a workshop. The papers included in this volume span a wide spectrum of topics within data science and artificial intelligence. They cover areas such as machine learning, natural language processing, computer vision, data mining, and interdisciplinary applications that demonstrate the practical impact of these technologies.

We want to thank the authors and speakers for their contributions. We are also indebted to the members of the Program Committee and external reviewers who dedicated their expertise and time to ensure the quality of the papers presented here. We acknowledge the support of Springer in publishing this volume and the invaluable financial support from the Leiden Institute of Advanced Computer Science (LIACS) from the University of Leiden and the Asian Institute of Technology (AIT). Thanks to the student volunteers, and the broader academic community for their commitment to making this conference successful.

We hope that DSAI 2023 fostered sustainable connections among researchers, inspired new ideas, and ultimately contributed to the advancement of the fields of data science and artificial intelligence.

November 2023

Chutiporn Anutariya
Marcello M. Bonsangue

Organization

Steering Committee

Taufik F. Abidin	Universitas Syiah Kuala, Indonesia
Chutiporn Anutariya (Chair)	Asian Institute of Technology, Thailand
Marcello M. Bonsangue (Chair)	Leiden University, The Netherlands
Matthew Dailey	Asian Institute of Technology, Thailand
Vatcharaporn Esichaikul	Asian Institute of Technology, Thailand
Joao Monteiro	University of Minho, Portugal
Opim Salim Sitompul	Universitas Sumatera Utara, Indonesia
Chitsutha Soomlek	Khon Kaen University, Thailand
Amalka Pinidiyaarachchi	University of Peradeniya, Sri Lanka
Hakim Usoof	University of Peradeniya, Sri Lanka
Chitraka Wickramarachchi	University of Sri Jayewardenepura, Sri Lanka

Publicity Chair

Chitsutha Soomlek	Khon Kaen University, Thailand

Local Organizer

Siriporn Nanthasing	Asian Institute of Technology, Thailand

Program Committee

Taufik Abidin	Universitas Syiah Kuala, Indonesia
Sonali Agarwal	IIIT Allahabad, India
Chutiporn Anutariya (Chair)	Asian Institute of Technology, Thailand
Marcello Bonsangue (Chair)	Leiden University, The Netherlands
Erna Budhiarti-Nababan	Universitas Sumatera Utara, Indonesia
Jose Luis Calvo-Rolle	University of A Coruña, Spain
Rathachai Chawuthai	King Mongkut's Institute of Technology Ladkrabang, Thailand
Xueqin Chen	Delft University of Technology, The Netherlands
Matthew Dailey	Asian Institute of Technology, Thailand

Vatcharaporn Esichaikul	Asian Institute of Technology, Thailand
Chu-Hsuan Hsueh	Japan Advanced Institute of Science and Technology, Japan
Natthawut Kertkeidkachorn	Japan Advanced Institute of Science and Technology, Japan
Yanika Kongsorot	Khon Kaen University, Thailand
José Machado	University of Minho, Portugal
Pedro Melo-Pinto	Universidade de Trás-os-Montes e Alto Douro, Portugal
Phan Minh Dung	Asian Institute of Technology, Thailand
João Monteiro	University of Minho, Portugal
Paulo Novais	University of Minho, Portugal
Millie Pant	Indian Institute of Technology Roorkee, India
Hai Pham	Hanoi University of Science and Technology, Vietnam
Amalka Pinidiyaarachchi	University of Peradeniya, Sri Lanka
Chantri Polprasert	Asian Institute of Technology, Thailand
Teeradaj Racharak	Japan Advanced Institute of Science and Technology, Japan
Mariam Rehman	Government College University Faisalabad, Pakistan
Pusadee Seresangtakul	Khon Kaen University, Thailand
Chaklam Silpasuwanchai	Asian Institute of Technology, Thailand
Prarinya Siritanawan	Japan Advanced Institute of Science and Technology, Japan
Opim Salim Sitompul	Universitas Sumatera Utara, Indonesia
Chitsutha Soomlek	Khon Kaen University, Thailand
Meng Sun	Peking University, P.R. of China
Khamron Sunat	Khon Kaen University, Thailand
Thepchai Supnithi	NECTEC, Thailand
Hakim Usoof	University of Peradeniya, Sri Lanka
Darshana Wickramarachchi	University of Sri Jayewardenepura, Sri Lanka
Ting Zhong	University of Electronic Science and Technology of China, P.R. of China

Organizers and Sponsors

Asian Institute of Technology
Leiden Institute of Advanced Computer Science Leiden University

Additional Reviewers

Cheng, Zhangtao
Janyoi, Pongsathon
Jearanaiwongkul, Watanee
Kaewyotha, Jakkrit
Liu, Nan
Long, Jingmin
Xie, Yuanlun
Xue, Xiaoyong
Yu, Liu
Zhang, Yihao
Zhang, Zhiyang

Keynote Presentations, Invited Talk, and Workshop

Explainable Artificial Intelligence, Explained (Keynote Talk)

Peter Flach

School of Computer Science, University of Bristol, UK
`Peter.Flach@bristol.ac.uk`

Abstract. Explainable Artificial Intelligence (XAI for short) aims at giving insight in the behaviour of AI models in general and machine learning models in particular. In this talk I will give an overview of this growing field, using some recent results from my group as examples. These include explainability for time series (LIMEsegment); actionable counterfactuals (FACE); explainability fact sheets; as well as the fat-forensics.org toolkit for evaluating Fairness, Accountability and Transparency of AI systems. Finally, I will discuss the importance of properly treating probabilities in feature attribution methods such as LIME and SHAP through log-linear models.

Machine Speech Chain: A Deep Learning Approach for Modeling Human Speech Perception and Production with Auditory Feedback Mechanism for Low-Resource Languages (Keynote Talk)

Sakriani Sakti

Japan Advanced Institute of Science and Technology (JAIST), Japan
`ssakti@jaist.ac.jp`

Abstract. The development of automatic speech recognition (ASR) and text-to-speech synthesis (TTS) has enabled computers to learn how to listen or speak, imitating the capability of human speech perception and production. However, computers still cannot hear their own voice, as the learning and inference to listen and speak are made separately and independently. Consequently, the separate training of ASR and TTS in a supervised fashion requires a large amount of paired speech-text data—furthermore, there is no ability to grasp the situation and overcome the problem during inference. On the other hand, humans learn how to talk by constantly repeating their articulations and listening to the sounds produced. By simultaneously listening and speaking, the speaker can monitor her volume, articulation, and the general comprehensibility of her speech. Therefore, a closed-loop speech chain mechanism with auditory feedback from the speaker's mouth to her ear is crucial.

In this talk, I will introduce a machine speech chain framework based on deep learning. First, I will describe the training mechanism that learns to listen or speak and to listen while speaking. The framework enables semi-supervised learning in which ASR and TTS can teach each other given unpaired data. Applications of multilingual and multimodal machine speech chains to support low-resource ASR and TTS will also be presented. After that, I will also describe the inference mechanism that enables TTS to dynamically adapt ("listen and speak louder") in noisy conditions, given the auditory feedback from ASR.

Pedagogy-Infused AI for Precision Education (Invited Talk)

Patcharin Panjaburee

Faculty of Education, Khon Kaen University, Thailand
patchapan@kku.ac.th

Abstract. Artificial Intelligence (AI) has been recognized by scholars as one of the challenging research issues of education. Several researchers have indicated the challenges and importance of considering individual students when designing AI in education based on their learning status, preferences, or personal characteristics. This talk aims to review the current state of AI in education and proper pedagogies. To this end, various implementations of AI-based systems and harmonization of learning strategies are introduced in this talk, such as their implementations in mathematics, science, computer science, and information literacy courses for elementary school students to undergraduate students. Moreover, the experience and thinking of conducting these studies of pedagogy-infused AI for precision education and the promotion of the research results across Thai school and university settings are presented. With my experience in conducting research on pedagogy-infused AI for precision education, this talk is expected to ignite researchers and school or university teachers devoted to AI in education to reach interesting research issues and design effective AI-based learning approaches relevant to your context.

The Teaching Algorithm: How to Design Educational Units Like a Master of Education (Workshop)

Alexandra Blank

Leiden Institute of Advanced Computer Science,
Leiden University, The Netherlands
a.blank@liacs.leidenuniv.nl

Abstract. Ever wondered if there is an algorithm to transform a target function into a teaching/learning process? In other words: Is there a manual for successful transfer of knowledge?

Since I started as Master and PhD Student Coach at LIACS (Leiden Institute of Advanced Computer Science), both students and professors approached me with questions and issues around course content, structure and evaluation. Why do group assignments fail? Why do students drop courses? Why is no one actively participating in discussions?

In this tutorial, I will introduce you to the 6 Principles of Successful Instruction according to Instructional Psychology. I will present the principles as a manual which you can use for designing your own educational units.

Contents

Foundations of Data Science and AI

SecureQNN: Introducing a Privacy-Preserving Framework for QNNs at the Deep Edge . . . 3
Miguel Costa, Tiago Gomes, Jorge Cabral, João Monteiro, Adriano Tavares, and Sandro Pinto

Chaotic Mountain Gazelle Optimizer (CMGO): A Robust Optimization Algorithm for K-Means Clustering of Diverse Data Types . . . 18
Tanatip Watthaisong, Khamron Sunat, and Nipotepat Muangkote

Hybridization of Modified Grey Wolf Optimizer and Dragonfly for Feature Selection . . . 35
Said Al Afghani Edsa and Khamron Sunat

A Modified Hybrid RBF-BP Network Classifier for Nonlinear Estimation/Classification and Its Applications . . . 43
Po-Chai Wong and Jeff Chak-Fu Wong

Data Mining and Machine Learning

Comparison of Machine Learning Models to Classify Documents on Digital Development . . . 59
Uvini Ranaweera, Bawun Mawitagama, Sanduni Liyanage, Sandupa Keshan, Tiloka De Silva, and Supun Hewawalpita

Exploration of the Feasibility and Applicability of Domain Adaptation in Machine Learning-Based Code Smell Detection . . . 74
Peeradon Sukkasem and Chitsutha Soomlek

Web Usage Mining for Determining a Website's Usage Pattern: A Case Study of Government Website . . . 88
Panunsiya Rawira and Vatcharaporn Esichaikul

Deep Learning

Deep-Learning-Based LSTM Model for Predicting a Tidal River's Water Levels: A Case Study of the Kapuas Kecil River, Indonesia . . . 103
Kanada Kurniawan, Joko Sampurno, Riza Adriat, Randy Ardianto, and Arie Antasari Kushadiwijayanto

Data Augmentation for EEG Motor Imagery Classification Using Diffusion Model 111
Nutapol Soingern, Akraradet Sinsamersuk, Itthi Chatnuntawech, and Chaklam Silpasuwanchai

Thai Conversational Chatbot Classification Using BiLSTM and Data Augmentation 127
Nunthawat Lhasiw, Tanatorn Tanantong, and Nuttapong Sanglerdsinlapachai

DeepFake Detection Using Deep Learning 142
Mariam Rehman, Mehran Rasool, and Sadaf Safder

Image Analysis and Object Recognition

Using the New YoLo Models in Detecting Small-Sized Objects in the Case of Rice Grains on Branche 157
Khang Nguyen Quoc, Anh Nguyen Quynh, Hoang Tran Ngoc, and Luyl-Da Quach

Segmentation of NKX2.5 Signal in Human Pluripotent Stem Cell-Derived Cardiomyocytes 170
Siem Jongsma, Verena Schwach, Simone A. Ten Den, Robert Passier, Fons J. Verbeek, and Lu Cao

Deep Learning Implementation for Pineapple Sweetness Classification 185
Sayed Muchallil, Silvia Roza, Muhammad Hafez Al-Assad, Yudi Candra, Maya Fitria, and Rahmad Dawood

Improving Low Light Object Detection Using Image Enhancement Models 198
Muhammad Omer Farooq Bhatti and Matthew N. Dailey

Unsupervised Segmentation of High-Throughput Zebrafish Images Using Deep Neural Networks and Transformers 213
Shima Javanmardi, Xiaoqin Tang, Mehrdad Jahanbanifard, and Fons. J. Verbeek

Exploring Deep Learning Techniques for Vision-Based Vehicle Re-Identification: A Traffic Intersection Case Study 228
Biplav S. Regmi, Matthew N. Dailey, and Mongkol Ekpanyapong

Large Language Models

ChatGPT Knows Your Attacks: Synthesizing Attack Trees Using LLMs 245
Olga Gadyatskaya and Dalia Papuc

BERT Fine-Tuning the Covid-19 Open Research Dataset for Named Entity Recognition 261
Shin Thant, Teeradaj Racharak, and Frederic Andres

Data Science and AI in Healthcare

Optimizing Heart Disease Classification: Exploring the Impact of Feature Selection and Performance of Machine Learning Algorithms 279
Aga Maulana, Farrasa Rani Faisyal, Faris Khowarizmi Tarmizi, Taufik Fuadi Abidin, and Hammam Riza

Enhancing Data Science Interoperability: An Innovative System for Managing OpenEHR Structures 288
Miguel Dias, Regina Sousa, Júlio Duarte, Hugo Peixoto, António Abelha, and José Machado

A Machine Learning Approach to Evaluating the Relationship Between Dental Extraction and Craniofacial Growth in Adolescents 300
Guillermo Hernández, Alfonso González-Briones, José Machado, Pablo Chamoso, and Paulo Novais

AI and Creativity

Generative Design Optimization for Tourism-Oriented Souvenir Development: Blending Art and Science into Tourism Industry 317
Somlak Wannarumon Kielarova and Prapasson Pradujphongphet

A Computer Scientist's View on Machine Learning and Creativity 325
Maarten H. Lamers

Data Science and AI in Education

Performance Analysis of Models Used to Predict Failure in Secondary School 339
Sofia Jordão, Dalila Durães, and Paulo Novais

Navigating the Implications of AI in Indonesian Education: Tutors, Governance, and Ethical Perspectives 349
Daphne Wong-A-Foe

Author Index 361

Foundations of Data Science and AI

SecureQNN: Introducing a Privacy-Preserving Framework for QNNs at the Deep Edge

Miguel Costa(✉), Tiago Gomes, Jorge Cabral, João Monteiro,
Adriano Tavares, and Sandro Pinto

ALGORITMI Research Centre, Universidade do Minho, PT, Guimarães, Portugal
`miguel.costa@dei.uminho.pt`

Abstract. Recent concerns about real-time inference and data privacy are making Machine Learning (ML) shift to the edge. However, training efficient ML models require large-scale datasets not available for typical ML clients. Consequently, the training is usually delegated to specific Service Providers (SP), which are now worried to deploy proprietary ML models on untrusted edge devices. A natural solution to increase the privacy and integrity of ML models comes from Trusted Execution Environments (TEEs), which provide hardware-based security. However, their integration with heavy ML computation remains a challenge. This perspective paper explores the feasibility of leveraging a state-of-the-art TEE technology widely available in modern MCUs (TrustZone-M) to protect the privacy of Quantized Neural Networks (QNNs). We propose a novel framework that traverses the model layer-by-layer and evaluates the number of epochs an attacker requires to build a model with the same accuracy as the target with the information disclosed. The set of layers whose information makes the attacker spend less training effort than the owner training from scratch is protected in an isolated environment, i.e., the secure-world. Our framework will be evaluated in terms of latency and memory footprint for two ANNs built for the CIFAR-10 and Visual Wake Words (VWW) datasets. In this perspective paper, we establish a baseline reference for the results.

Keywords: Machine Learning · Artificial Neural Networks · Quantized Neural Networks · ML Model Privacy · TEE · TrustZone-M · Armv8-M

1 Introduction

Advances in computing processing and the availability of vast amounts of data have propelled Machine Learning (ML) to the forefront of various domains,

---

We would like to thank the anonymous reviewers for their valuable feedback and suggestions. This work is supported by FCT - Fundação para a Ciência e Tecnologia within the R&D Units Project Scope UIDB/00319/2020. Miguel Costa was supported by FCT grant SFRH/BD/146780/2019.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 3–17, 2023.
https://doi.org/10.1007/978-981-99-7969-1_1

including healthcare [1], finance [2], and mobility [3,4]. Among the wide range of algorithms used in ML, Artificial Neural Networks (ANNs) have emerged as a prominent choice. However, training ANNs often requires access to large-scale private datasets, which are not typically held by ML clients. To address this challenge, big data companies introduced Machine Learning as a Service (MLaaS), enabling clients to use proprietary models for inference tasks using personal data.

Typical MLaaS follow a centralized computing paradigm where the client sends its data to the cloud to get the results. However, heavy reliance on the cloud induces unpredictable latency due to network overload which may render ML services useless in real-time scenarios [5,6]. This combined with rising concerns about data privacy is making ML shift to the edge of the network [5,6]. If this increases privacy for the user as its data is not transferred over the network, the same does not apply to Service Providers (SPs) as their models would be deployed in untrusted edge devices. Unauthorized access to these models can lead to the exposure of proprietary algorithms [7], reverse engineering [7], or even the extraction of sensitive training data [7,8]. Moreover, recent attacks have demonstrated that malicious manipulation of ANN parameters can compromise the integrity of decision-making processes [9,10]. All these issues combined gave rise to a still open research question: “Is it possible to transfer a proprietary ML model to the edge while providing privacy guarantees to the SP?”.

A primary response to this concern comes from hardware-based security mechanisms, such as Trusted Execution Environments (TEEs). TEEs enforce memory isolation between applications running in a Rich Execution Environment (REE) and smaller critical applications isolated by hardware [11]. Consequently, researchers have started to adapt off-the-shelf TEEs, such as Intel-SGX [12] and ARM TrustZone [13,14], to protect ML computations at the edge of the network. However, TEEs are designed to execute small critical operations and tend to pose severe memory constraints to resource-hungry ML computation [15–20]. Furthermore, they do not extend to accelerators such as GPUs or ASICs [15,17–19,21]. As a consequence, developing privacy-enhanced ML frameworks on top of TEEs remains an open challenge.

Prior works [15,18,22,23] propose to encrypt the model and store it in unprotected memory. When the inference process starts, the model is loaded and decrypted into the enclave, typically layer-by-layer to overcome the memory limits. If the activations can not be stored within the enclave to serve as input to the next layer, they are encrypted before being stored into an unprotected memory region. Despite the novelty, these approaches suffer from increased latency disrupted by the decryption of all ANN parameters and the frequent context switch from the secure-world to the normal-world. To counteract these issues, some works propose to (i) perform loading in parallel with prediction [20]; (ii) execute only sensitive layers within the enclave [16,19,21]; (iii) obfuscate only relevant ANN weights [17]; or (iv) refactor a model in a bident structure [24].

Despite their novelty, prior works target x86 and Arm Cortex-A processors, letting out Arm Cortex-M processors, which are at the forefront development of IoT applications. The recent support for friendly ML APIs (CMSIS-NN and TensorFlow Lite Micro) is pushing Arm Cortex-M processors even further [5,25]. As some Arm Cortex-M processors are now being shipped with TrustZone-M technology, a new question arises: *can TEEs be leveraged to protect the privacy of ANNs on resource-constrained Arm Cortex-M MCUs?* Although previous works show impressive results for the Intel-SGX and Arm Cortex-A architectures, they may not hold to Arm Cortex-M MCUs as these devices only hold a few megabytes of memory and much lower throughput. The integration of Quantized Neural Networks (QNNs), which operate with reduced precision and memory requirements, and TrustZone-M may hold great promise for advancing ML capabilities on these devices while ensuring the privacy and security of sensitive ML models.

In this perspective paper, we dig into this topic by providing an initial overview of the feasibility of leveraging the Arm TrustZone-M technology to protect the privacy of a QNN. Previous work [26] has found that the last layers of a neural network are the ones that reveal more sensitive information about the classification task, while the first ones reveal more information about the input itself. We borrow this finding to develop a novel framework that splits the execution of a QNN between the secure and normal worlds while minimizing the number of layers delegated to the secure-world to the bare minimum. As the last layers tend to be smaller than the first ones, we believe this approach could be a good starting point to tackle the severe memory constraints imposed by TrustZone-M. To abstract the implementation details of TrustZone-M, our framework works on top of Trusted Firmware-M (TF-M). By combining the built-in capabilities of TF-M for secure storage and by minimizing the number of layers within the TEE, we envision reducing the need for frequent context switches, while optimizing the access to critical data.

To evaluate our framework, we consider the most challenging scenario where the adversary has access to everything in the (untrusted) normal-world. In such a scenario, the attacker can freeze the non-protected layers, while iteratively designing and training possible adjacent layers till he gets a substitute model with an accuracy equal to or greater than the target model. In this context, we consider the privacy of the target model protected when the effort required to build the substitute model is, at least, equal to the effort required to train the target model from scratch. We measure the training effort in training epochs. We evaluate our framework (SecureQNN) in terms of TEE memory footprint and inference latency. The evaluation suite includes two QNNs designed and trained for the CIFAR-10 and Visual Wake Words (VWW) datasets. In this perspective paper, we establish the baseline reference for future evaluation. To the best of our knowledge, we are the first to propose this ANN splitting strategy to enhance privacy while utilizing a TEE. Furthermore, no previous works have specifically targeted Arm Cortex-M MCUs.

2 Background

2.1 Arm TrustZone-M

Arm TrustZone-M is a technology designed to enhance the security of MCUs following the Armv8-M architecture [11,13]. TrustZone-M employs hardware-based isolation to establish two distinct execution states: (i) the secure-world and (ii) the normal-world. The secure-world is the host for software components that are critical from the integrity or privacy perspective, while the normal-world accommodates less sensitive applications. The division between the two worlds is memory-based, meaning that when the processor is running from secure memory, the processor state is secure, and when the processor is running from non-secure memory, the processor state is normal. The secure-world has access to every memory region, while the normal-world only has access to the non-secure memory. For this purpose, TrustZone-M relies on two attribution units. The Security Attribution Unit (SAU) is programmable, allowing dynamic address partitioning. The Implementation-Defined Attribution Unit (IDAU) partitions memory statically and is defined at design time by the chip maker. To determine if a given address is secure or not, TrustZone-M performs a logical OR between the output of SAU and IDAU. By preventing unauthorized access to secure resources, TrustZone-M prevents malicious applications in the normal-world from impacting the integrity and privacy of the applications running in the secure-world.

2.2 Trusted Firmware-M (TF-M)

TF-M is an open-source firmware implementation that uses the TrustZone-M hardware capabilities, providing trusted firmware for Arm Cortex-M devices. It supports the Armv8-M and Armv8.1-M architectures and incorporates various security features, including (i) secure boot, (ii) software component isolation, (iii) protected storage of sensitive information, and (iv) secure firmware update.

TF-M uses MCUboot, as its secure bootloader, enabling the verification and authentication of applications running in the secure and normal worlds during the system startup. This prevents the execution of malicious code and helps establish a trusted foundation for the entire system. Furthermore, TF-M enforces isolation between the secure and normal worlds, as well as between applications within the secure-world if required. To partition the memory as secure and non-secure, TF-M relies on the hardware-enforced isolation provided by TrustZone-M through the SAU and IDAU units, as well as system-wide protections units, such as Peripheral Protection Controllers (PPCs), used to partition peripherals. To partition the secure memory between different Trusted Applications (TAs), TF-M relies on the secure memory protection unit.

Protected storage and firmware update are two important services provided by TF-M. The former relies on the hardware isolation provided by TrustZone-M to isolate a given fraction of the flash memory from the normal-world. This mechanism enables the secure storage and management of cryptographic keys, certificates, and other confidential data. Additionally, TF-M enables secure firmware updates, allowing devices to securely receive and install new firmware binaries. More specifically, it provides the device with the ability to verify digital signatures of received binaries, preventing unauthorized modifications or malicious code injection.

3 Related Work

Several mechanisms to ensure the confidentiality of a proprietary ML model have been proposed over the last few years. Table 1 reviews and puts into perspective the most relevant works in this area. In this paper, we restrict the scope to works that use TEEs to protect the privacy of the ML model. Works that employ TEEs to protect the confidentiality of user input data or enable collaborative learning from multiple-edge devices are out of the scope of this review. For a review including these subjects, we refer to the work [27].

Table 1. Gap Analysis

	Architecture TEE	**Trusted Execution**	**Needs to fit on TEE?**	**Integrity Check**	**Description**
VanNostrand *et al.* [15] 2019	Arm Cortex-A OP-TEE	All layers	Largest layer	No	The computation of every layer is performed on the TEE. Loads the model layer-by-layer, requiring frequent context switches.
OMG [22] 2020	Arm Cortex-A SANCTUARY	All layers	Full model	No	The computation of every layer is performed on the TEE. Loads the model in a single iteration, not requiring context switches during inference.
SecDeep [18] 2021	Arm Cortex-A OP-TEE	All layers Conf. functions	Largest layer	No	Splits Arm-NN into confidential and non-confidential functions. Confidential functions deal with private user and model data and are delegated to the TEE.
Nakai *et al.* [20] 2021	Arm Cortex-A OP-TEE	All layers	Largest layer	Yes	Uses the shared memory between the REE and the TEE to load the model in parallel with prediction.
MLCapsule [23] 2021	x86 Intel-SGX	All layers	Largest layer	No	The computation of every layer is performed on the TEE. Loads the model layer-by-layer, requiring frequent context switches.
Slalom [21] 2019	x86 Intel-SGX	Selected layers	Largest layer	Yes	Outsources the computation of linear layers from TEE to GPU without revealing weights and user data. Non-linear layers are executed within the TEE.
DarkneTZ [16] 2020	Arm Cortex-A OP-TEE	Selected layers	Largest layer	No	Delegates the execution of the layers most sensitive to MIAs to a TEE.
ShadowNet [19] 2023	Arm Cortex-A OP-TEE	Selected layers	Largest layer	Yes	Outsources the computation of linear layers from TEE to GPU without revealing weights and user data. Non-linear layers are executed within the TEE.
Hou *et al.* [17] 2022	x86 Intel-SGX	Selected neurons	Selected neurons Largest layer I/O	No	Obfuscates strategic weights to outsource most of the inference to the REE. The output of each layer is denoised within the TEE.
Lin *et al.* [24] 2020	-	Small asymetric model	Small asymetric model	No	Refactors the model into two asymmetric models, one on the TEE and the other on the REE. The model on the TEE has more weight in the final prediction.
Costa et al. 2023	**Arm Cortex-M TF-M**	**Selected layers**	**Selected layers or largest layer**	**No**	**Delegates the execution of the layers containing the most sensitive information about the classification task and training data to the TEE.**

3.1 All Layers Running Inside the TEE

A first approach consists in embedding all layers inside the TEE. This is the most basic but also the most effective technique to protect the privacy of a neural network. Nevertheless, it is the least optimized in terms of latency as it involves frequent context switches between the secure and normal worlds and typically requires cryptography to be executed over all ANN parameters.

OMG [22] is the only work relying on the SANCTUARY security architecture developed for Arm Cortex-A processors. OMG follows a very simple approach: after the initialization phase, the full model is loaded and decrypted in a single iteration within the enclave and the inference process starts. Given the limited memory footprint of TEEs, OMG does not scale to large models.

The works [15,23] tackle this issue by splitting the inference process layer-by-layer. For each layer, the weights are loaded from the non-secure to the secure memory and decrypted afterward. The inference process is fully confined to the TEE. However, if the outputs of a given layer do not fit in the secure memory, they are encrypted and offloaded to the non-secure region being loaded and decrypted within the TEE when the computation of the following layer starts. The process is repeated till the output layer, where the result of the classification is returned in plain text to the normal-world. Despite the similar ANN splitting process, these works have some differences. While MLCapsule [23] targets x86 processors with support for Intel-SGX, the work [15] targets Arm Cortex-A processors with OP-TEE. In addition to layer-based partitioning, the work [15] proposes sub-layer and branched partitioning. In the former technique, the layer is partitioned into sub-layers, which helps to deal with the memory limits of the enclave; however, at the cost of increased decision latency. Sub-layer partitioning requires the inputs of the current layer to be loaded and decrypted more than once to calculate the full set of layer outputs. Branched partitioning solves this problem by splitting the last layers of a neural network into independent layers with independent connections to the forward layers. However, this involves changes to the model architecture. Despite the novelty and potential of these splitting strategies, VanNostrand *et al.* [15] does not evaluate them.

The work [20] improves over the previous works by reducing the need for context switches anytime the parameters of a new layer have to be loaded into the TEE. For this purpose, Nakai *et al.* [20] uses the shared memory between the secure and normal worlds to load the parameters and execute the decryption and the prediction in parallel. The work targets Arm Cortex-A processors with support for OP-TEE.

When analyzing the source code of Arm-NN, an API tailored for the execution of ANNs in Arm Cortex-A processors, Liu *et al.* [18] found that more than 90% of the code implementing layer functions is dedicated to tensor preparation or performance optimization, while only less than 10% is dedicated to mathematical tensor computation with sensitive user and model data. Based on this finding, Liu *et al.* [18] proposed SecDeep, a framework that reformulates the Arm-NN functions implementing ANN layers into two types: (i) confidential functions and (ii) non-confidential functions.

3.2 Selected Layers Running Inside the TEE

A second approach reported in the literature resorts to running a set of selected layers only inside the TEE. When compared to the previous approach, this strategy is intended to reduce the memory footprint of TEEs, while also reducing the number of context switches between the secure and normal worlds.

Slalom [21] was one the first works to explore this technique by outsourcing the computation of linear layers from the enclave to an untrusted GPU. The inference process starts and is managed from the enclave. However, whenever Slalom finds a linear layer on the pipeline, Slalom encrypts both the inputs and the layer parameters with a pre-computed pseudorandom stream and outsources the computation to a co-located GPU. The result is then decrypted within the enclave and the integrity is verified using the Freivalds' algorithm. Slalom targets x86 processors with support for Intel-SGX. Sun *et al.* [19] borrowed the idea behind this work and developed ShadowNet, a framework similar to Slalom but for mobile devices powered by Arm Cortex-A MCUs with support for OP-TEE.

DarkneTZ [16] proposes a different approach to speed up the inference latency on Arm Cortex-A processors. Nevertheless, DarkneTZ does not protect the model parameters like the previous works. Instead, DarkneTZ protects the privacy of the training data by offering robustness against membership inference attacks (MIAs). DarkneTZ traverses the model layer-by-layer and uses all the information available till the layer under analysis as input to a strong white-box MIA. The set of layers to which the attack succeeds is delegated to the TEE.

3.3 Selected Neurons

A third approach reported in the literature proposes to delegate the computation of critical neurons to the TEE. Hou *et al.* [17] propose a framework to design a secure version of the ML model with crafted random weights. More specifically, the framework adds crafted values to obfuscate strategic weights within the ANN. The crafted noise must change the predicted label of the query sample, rendering the accuracy of the obfuscated model useless. In this framework, the inference process is mostly carried out in the normal-world, being the TEE responsible for denoising the outputs of each layer. The framework distorts the top percentage weights with the highest absolute value. The distortion must force the attackers to pay at least the same machine training hours on retraining the crafted model with high accuracy as the model owner trains from scratch. The framework targets x86 processors with support for Intel-SGX.

3.4 Model Refactoring

Lin *et al.* [24] addressed the limitations of TEEs with a different strategy. The authors propose to refactor the model to be protected into two asymmetric models, one to be executed within a TEE and another within an untrusted computation unit. The output of both models is then fused by a few public layers. The protected model is expected to have a smaller memory footprint and contribute the most to the final prediction. The model deployed in the normal-world must not enable the attacker to reconstruct the model within the TEE or the original model. Despite reducing the memory footprint of the secure application, the examples and evaluation given in the original work suggest that the strategy may increase the memory footprint of the overall ML system, as the sub-model deployed in the normal-world can be bigger than the original one.

3.5 Gap Analysis

When compared to works that run all layers inside the TEE, SecureQNN has the potential to reduce the number of context switches between the secure and normal worlds. To reduce the TEE memory footprint, most works in this category encrypt the model parameters and save them in non-secure memory. During inference, the parameters are decrypted and loaded layer-by-layer to the secure-world. This strategy reduces the TEE memory footprint to the size of the largest layer but requires a context switch every time a new layer is processed.

Compared to other works in the same category, DarkneTZ also reduces the computation within the TEE to sensitive layers. However, DarkneTZ protects ANNs against MIAs, which has different requirements from protecting the privacy of model parameters. Slalom and ShadowNet work by offloading linear layers to a GPU. However, as the linear layers compose most of the layers of an ANN and the parameters used by the GPU are obfuscated, the outputs of linear layers always need to be denoised within the TEE, requiring frequent context switches between the normal and secure worlds. These two works are not easily portable to devices powered by Arm Cortex-M as they do not feature GPUs.

Works that delegate the computation of selected neurons to the TEE are the most efficient in terms of TEE memory footprint. The work [17] only requires space to save the selected neurons and the output of the layer under process. Despite performing most of the computation outside the TEE, this work still requires the outputs of each layer to be denoised within the TEE, which significantly increases the number of context switches compared to our framework.

The only work that relies on model refactoring [24] requires the same number of context switches as we envision for our framework. In terms of memory footprint, the results and the description of the model refactoring strategy are not deep enough to hold strong assumptions.

SecureQNN is the only work targeting Arm Cortex-M MCUs. Other works either target Arm Cortex-A or x86 processors, which are far more powerful and frequently coupled with GPUs, which can be used to overcome the performance bottleneck imposed by frequent context switches and cryptography operations. In addition, works that apply fine-grain techniques to select the layers (DarkneTZ [16]) or the neurons [17] to be performed within the TEE do not consider the impact of quantization. It is worth noting that while previous works target ANNs with floating-point precision, our work targets QNNs.

4 SecureQNN

4.1 Scope

SecureQNN targets IoT devices powered by Arm Cortex-M MCUs following the latest Armv8-M architecture and supported by TF-M. These tiny and low-power devices typically have a single central processing unit and feature a wide range of peripherals. However, GPU acceleration is not supported. The most advanced Armv8-M MCUs support the TrustZone-M technology, which enables the programmer to create two execution environments: secure and normal.

4.2 Threat Model

Our threat model considers two agents: (i) an SP and a (ii) client. The SP uses its big data infrastructure to provide the client with a highly accurate ANN. The SP is concerned about the privacy of its model as it constitutes its intellectual property. Training an accurate model may require years of data storage and treatment and weeks to months of training. Furthermore, recent studies have shown that ANNs are vulnerable to MIAs [16], attacks specifically designed to understand if a data point is part of the training data. As ANNs are usually trained on private data belonging to a myriad of users, the SP wants to maintain the training data private, as leaking data would undermine the users' trust in the SP. Our work focuses on maintaining the privacy of ANN critical layers to reduce the risk of these threats.

The client gets the ANN trained by the SP on its local device, which is powered by an Arm Cortex-M MCU and must support TruztZone-M and TF-M. The client is an honest but curious adversary. He follows the protocol required by the SP but tries to learn about the ANN. In this context, we consider the client has access to everything in the normal-world, including memory and operating system. We consider the model provided by the SP as a gray-box, where part of it is executed and stored within the normal-world and the other part in the secure-world. The user has full access to the public part of the model and can use it to try to disclose the private part. We consider the user has no access to the training data. As data is the most valuable asset of an ML model, there is no interest in stealing the model if the attacker already has access to data.

4.3 Model Partitioning

Previous research [26] discovered that the initial layers of ANNs contain crucial information for breaking down the input data, while the subsequent layers play a more significant role in the classification task. We leverage this insight to divide a given ANN into secure and normal worlds. Figure 1 illustrates the overall process for model partitioning. The model partitioning framework will be built in Python using the Tensorflow v2 API as the basis. Tensorflow v2 provides seamless conversion from floating-point to integer precision models compatible

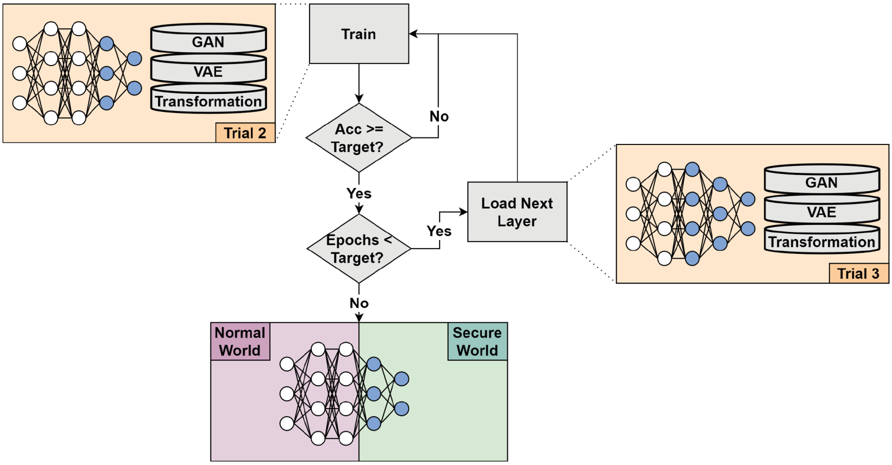


Fig. 1. Overview of the model partitioning technique. The example describes the second trial, where it is evaluated the criticality of the last two layers.

with the TensorFlow Lite Micro and CMSIS-NN, the most prominent APIs for ML development in Arm Cortex-M MCUs.

Our framework systematically traverses the model layer-by-layer, starting from the final layer. At each layer, we expose the parameters of the current and forward layers to a training process. During this training process, we freeze the parameters of the disclosed layers and attempt to reconstruct the unknown ones. We consider the most challenging scenario, assuming that the attacker possesses complete knowledge of the model's architecture and training hyperparameters. For every layer, we measure the number of epochs needed for the reconstructed model to achieve an accuracy equal to or higher than the model to be safeguarded. The minimum set of frozen layers that require, at least, the same epochs as those needed by the model owner to train from scratch will be allocated to the secure-world.

Since the attacker lacks access to the original training dataset, they must construct a substitute dataset. Our framework emulates this situation by employing data augmentation techniques on the original test dataset. Specifically, it utilizes three data augmentation methods: (i) image transformations, (ii) Generative Adversarial Networks (GANs), and (iii) Variational Auto-Encoders (VAEs). The augmented dataset is then used to simulate the training attempts conducted by a potential attacker.

4.4 Trusted Execution

Figure 2 details the architecture of an edge device running an ANN provisioned by SecureQNN. As depicted, we leverage the TrustZone-M technology to create two distinct execution environments: (i) normal and (ii) secure. Both environments feature privileged and non-privileged execution. The blocks running at the

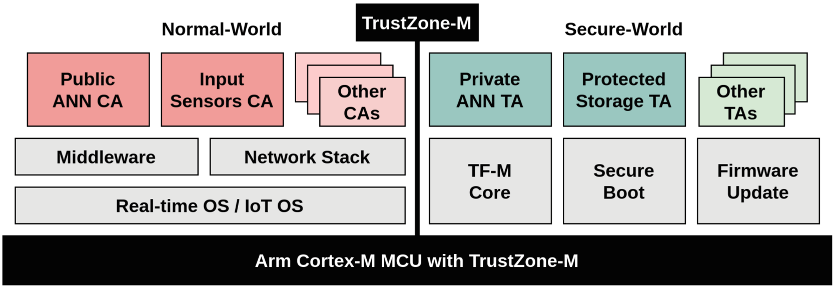


Fig. 2. Overview of the ANN trusted execution

privileged level are depicted in gray, while the non-privileged blocks are depicted in green (secure-world) and red (normal-world).

Within the privileged level of secure-world, the Secure Boot is essential to detect any attempt of tampering with the firmware image and therefore with the ANN itself. For this purpose, the Secure Boot process verifies the digital signature of the firmware during the device startup. If the firmware is found modified, the boot process is aborted. For a reference implementation of secure firmware, we use TF-M. From the panoply of secure services offered by TF-M, the Firmware Update is the most relevant for our framework, as it enables secure over-the-air updates of the ANN. Whenever the SP wants to deploy a new ANN, it wraps the updates in a new firmware and sends it to the edge device. As the firmware is downloaded and stored in a secure memory space, the authenticity and integrity of the new firmware are verified. If the digital signature is valid, the new firmware starts to run. The Firmware Update service also features an anti-rollback mechanism to prevent an attacker from installing an older firmware image, and consequently, an outdated ANN.

Regarding the non-privileged environment of the secure-world, SecureQNN relies on two main TAs. The Protected Storage is a service implemented by TFM and is responsible for storing in a secure memory region the parameters of the private ANN layers. The Private ANN is the TA responsible for computing the output of the set of private layers. Whenever this TA is called, it interacts with the Protected Storage to load the ANN parameters and then proceeds with the inference process. The final output is sent in plain text to the Public ANN CA.

Within the normal-world, most of the software blocks are optional, except for the Input Sensors and the Public ANN CAs. The former is responsible for collecting the ANN input data, while the latter is responsible for starting and coordinating the inference process until a private layer is found.

4.5 QNN Execution Flow

As the inference process starts, the Input Sensors CA transfers the collected user/sensor data to the Public ANN CA, which controls the overall inference process. We do not make assumptions about the confidentiality and integrity of the user data, as this is out of the scope of our work. Although protecting the privacy of an ANN reduces the vulnerability to white-box adversarial attacks, our system would still be vulnerable to gray-box and black-box attacks. As the Public ANN CA finds a private layer, it uses the TF-M communication channels to send the output of the last public layer to the Private ANN TA.

As the RAM available in the secure-world is allocated during the boot and maintained static afterward, optimizing the amount of RAM required by the Private ANN TA is fundamental to not undermine the applications running in the normal-world. For this purpose, the ANN parameters will be stored in the flash memory and loaded layer-by-layer to the RAM. The Protected Storage TA handles the memory read process, while maintaining the privacy of the ANN parameters. In this context, we estimate that the maximum amount of RAM required by the Private ANN TA equals the parameter size of the largest private layer plus the size of the auxiliary buffer required in the *im2col* process of convolutional layers. The output of the last private layer (prediction vector) is sent in plain text to the Public ANN CA using the TF-M communication channel.

5 Preliminary Results

We evaluate SecureQNN in terms of latency and TEE memory footprint for two distinct QNNs. To provide a fair evaluation of our framework, we first need to establish the baseline reference values for the scenario where the full computation of the QNN is delegated to the secure-world.

SecureQNN pretends to address current or future applications of ML in IoT. Consequently, we evaluate our framework on two QNNs borrowed from the TinyML benchmark [28] - an open-source benchmark for ML workloads on emerging low-power hardware. These QNNs are trained to perform inference on the CIFAR-10 and VWW datasets. CIFAR-10 comprises 60,000 RGB images measuring 32 × 32 pixels. These images are categorized into 10 distinct classes, each representing a different object. The dataset is divided into 50,000 training images and 10,000 testing images. On the other hand, the VWW dataset is binary and contains 115,000 RGB images measuring 96 × 96 pixels. Its purpose is to identify whether a person is present in the image or not. The training subset of VWW consists of 98,568 images, while the testing subset contains 10,961 images. The results are extracted for an STM Nucleo-L552ZE-Q board, powered by an Arm Cortex-M33 MCU running at 120 MHz.

5.1 Baseline Reference Values

Table 2 shows the baseline reference of latency and TEE memory footprint for the two QNNs described above. As expected, the memory footprint of the TEE when running a full QNN is incredibly high. Regarding RAM, when the TEE runs the full CIFAR-10 and VWW QNNs, it requires 55.51% and 69.43% of the total RAM available on the board, respectively. Regarding flash memory, CIFAR-10 and VWW QNNs take 22.49% and 55.51% of the space available. Under this

Table 2. Baseline reference values

	TEE Memory Footprint (kB)		Latency	
	RAM	Flash	Clock Cycles	Milliseconds
CIFAR-10	142.09 (55.51%)	125.36 (24.49%)	99483061	829.03
VWW	177.90 (69.43%)	284.21 (55.51%)	75017684	625.15

umbrella, we can securely argue that fully delegating the computation of a QNN to the secure-world is unfeasible in a real-world scenario, as this puts severe restrictions on the memory available in the normal-world. TF-M allocates the secure and non-secure memory regions during the boot process and the amount of memory allocated is maintained statically afterward. Consequently, allocating too much memory for the secure-world severely limits the number of applications that can be deployed in the normal-world, which might jeopardize the purpose of having ML privacy mechanisms in IoT and put the model owner at risk. Our framework tackles this issue by reducing the computation performed within the TEE to the bare minimum while maintaining the privacy of the critical layers.

Regarding latency, the values in Table 2 already consider two context switches between the normal and secure worlds, one before and one after the inference process. We envision our framework will keep up with these values, as the QNN splitting strategy that we propose also requires no more than two context switches.

6 On the Road

SecureQNN's development is still at an embryonic stage. As of this writing, we are developing the Python framework that evaluates the privacy leaking of each layer in a QNN. Our framework requires the following inputs: (i) the QNN to be protected, (ii) the number of training epochs, and (iii) the test accuracy. It then identifies which layers should operate within the secure-world. In the first stage of development, we will perform the evaluation layer-by-layer, starting from the last layer. As our framework targets resource-constrained Arm Cortex-M MCUs, we don't expect QNNs to have a large number of layers. Nevertheless, to speed up the process, in the second stage of development, we might consider the use of group-layer sorting strategies - instead of evaluating each layer individually, we evaluate a given set of layers at each iteration.

The last stage of development addresses the implementation of a mechanism to trade off privacy and TEE memory footprint. More specifically, SecureQNN will return the TEE memory footprint for different sets of private layers along with the number of epochs they require to achieve the same accuracy as the target model. This will allow the user to evaluate if it is worth increasing the privacy at the cost of memory footprint. Whenever the minimum privacy guarantee is achieved before the memory footprint reaches the limit, SecureQNN will continue to evaluate the additional privacy guarantee by iteratively extending the set of protected layers till the maximum memory footprint is met.

7 Conclusion

In this paper, we present SecureQNN, a framework to protect the privacy of QNNs in Arm Cortex-M MCUs featuring TrustZone-M technology. Although development is still embryonic, we already outline the testing scenario and the baseline reference values for the TEE memory footprint (RAM and flash) and decision latency. We outline SecureQNN's development roadmap, which includes two fundamental stages to determine which layers are more privacy-critical and to trade off TEE memory footprint and privacy. One additional step will attempt to speed up the QNN splitting strategy. Once completed, we will open-source the SecureQNN framework.

References

1. Miotto, R., et al.: Deep learning for healthcare: review, opportunities and challenges. Briefings Bioinf. **19**(6), 1236–1246 (2017)
2. Ahmed, S., et al.: Artificial intelligence and machine learning in finance: a bibliometric review. Res. Int. Bus. Financ. **61**, 101646 (2022)
3. Costa, M., et al.: Detecting driver's fatigue, distraction and activity using a non-intrusive AI-based monitoring system. J. Artif. Intell. Soft Comput. Res. **9**(4), 247–266 (2019)
4. Li, Y., et al.: Deep learning for lidar point clouds in autonomous driving: a review. IEEE Trans. Neural Netw. Learn. Syst. **32**(8), 3412–3432 (2021)
5. Costa, D., et al.: Train me if you can: decentralized learning on the deep edge. Appl. Sci. **12**(9), 4653 (2022)
6. Murshed, M.G.S., et al.: Machine learning at the network edge: a survey. ACM Comput. Surv. **54**(8), 1–37 (2021)
7. Rigaki, M., et al.: A survey of privacy attacks in machine learning, CoRR (2020)
8. Hu, H., et al.: Membership inference attacks on machine learning: a survey. ACM Comput. Surv. **54**(11s), 1–37 (2022)
9. Li, G., et al.: TensorFI: a configurable fault injector for tensorflow applications. In: 2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW) (2018)
10. Chen, Z., et al.: BinFI: an efficient fault injector for safety-critical machine learning systems. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (2019)
11. Oliveira, D., et al.: uTango: an open-source TEE for IoT devices. IEEE Access **10**, 23913–23930 (2022)
12. Costan, V., et al.: Intel SGX explained (2016). https://eprint.iacr.org/2016/086
13. Pinto, S., et al.: Demystifying ARM TrustZone: a comprehensive survey. ACM Comput. Surv. **51**(6), 1–36 (2019)
14. Cerdeira, D., et al.: ReZone: disarming TrustZone with TEE privilege reduction. In: 31st USENIX Security Symposium (USENIX Security 22) (2022)
15. VanNostrand, P.M., et al.: Confidential deep learning: executing proprietary models on untrusted devices, CoRR (2019)
16. Mo, F., et al.: DarkneTZ: towards model privacy at the edge using trusted execution environments. In: Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services (2020)

17. Hou, J., et al.: Model protection: real-time privacy-preserving inference service for model privacy at the edge. IEEE Trans. Dependable Secure Comput. **19**(6), 4270–4284 (2022)
18. Liu, R., et al.: SecDeep: secure and performant on-device deep learning inference framework for mobile and IoT devices. In: Proceedings of the International Conference on Internet-of-Things Design and Implementation (2021)
19. Sun, Z., et al.: ShadowNet: a secure and efficient on-device model inference system for convolutional neural networks. In: 2023 IEEE Symposium on Security and Privacy (SP) (2023)
20. Nakai, T., et al.: Towards trained model confidentiality and integrity using trusted execution environments. In: Applied Cryptography and Network Security Workshops (2021)
21. Tramer, F., et al.: Slalom: fast, verifiable and private execution of neural networks in trusted hardware. In: International Conference on Learning Representations (2019)
22. Bayerl, S.P., et al.: Offline model guard: Secure and private ML on mobile devices. In: Proceedings of the 23rd Conference on Design, Automation and Test in Europe (2020)
23. Hanzlik, L., et al.: MLCapsule: guarded offline deployment of machine learning as a service. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2021)
24. Lin, H.-Y., et al.: Bident structure for neural network model protection. In: ICISSP (2020)
25. Costa, M., et al.: Shifting capsule networks from the cloud to the deep edge. ACM Trans. Intell. Syst. Technol. **13**(6), 1–25 (2022)
26. Yosinski, J., et al.: How Transferable are Features in Deep Neural Networks? MIT Press, Cambridge (2014)
27. Babar, M.F., et al.: Trusted deep neural execution - a survey. IEEE Access **11**, 45736–45748 (2023)
28. Banbury, C.R., et al.: Benchmarking TinyML systems: challenges and direction, CoRR (2020)

Chaotic Mountain Gazelle Optimizer (CMGO): A Robust Optimization Algorithm for K-Means Clustering of Diverse Data Types

Tanatip Watthaisong[1](✉), Khamron Sunat[1], and Nipotepat Muangkote[2]

[1] Department of Computer Science, College of Computing, Khon Kaen University, Khon Kaen, Thailand
`tanatip.w@kkumail.com, skhamron@kku.ac.th`
[2] Department of Business Computer, Mahasarakham Business School, Mahasarakham University, Mahasarakham, Thailand

Abstract. Addressing challenges in data clustering for diverse data types, we introduce the Chaos Mountain Gazelle Optimizer (CMGO). This enhanced Mountain Gazelle Optimizer (MGO) is tailored for K-means clustering solutions. Noticing a skew in MGO's strategy distribution, we integrated a chaotic map into the Territorial Solitary Males strategy and omitted the Migration to Search for Food strategy. This adjustment increases exploration and curtails exploitation, improving CMGO's effectiveness in clustering complex datasets. We implemented the Gower distance technique to navigate K-means clustering's limitations with categorical and binary data. Tests on numeric, binary, categorical, and mixed data underscore the clustering's versatility. We evaluated CMGO against 14 algorithms on 28 UCI and OpenML datasets using the F-Measure metric and the tied rank test for statistical significance ranking. CMGO outperforms the original MGO and other tested algorithms in clustering pure numeric and categorical data, securing first place, and third for mixed data. Thus, CMGO emerges as a robust, efficient K-means optimizing method for complex, diverse datasets.

Keywords: Data clustering · K-Means Clustering · Mountain Gazelle Optimizer · Mixed-type data · Chaos map · Nature-Inspired Optimization

1 Introduction

Data science is vital in numerous industries, enabling informed decision-making through in-depth data analysis and interpretation. Through the application of techniques like machine learning, businesses can effectively use predictive analytics to anticipate future outcomes and meet customer needs [1]. Clustering algorithms automatically reveal data patterns and relationships. They analyze data to identify similarities, detect patterns, and group data points based on the characteristics and desired clustering techniques [2]. It is an unsupervised learning method that uncovers natural clusters in a dataset, facilitating data exploration and comprehension [3]. However, clustering faces challenges in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 18–34, 2023.
https://doi.org/10.1007/978-981-99-7969-1_2

selecting suitable data representatives, handling diverse data, and dealing with distribution complexities. It's a computationally complex task in the class of NP-complete problems, aiming to minimize dissimilarity measures for identifying clusters in varied datasets [4]. Two fundamental approaches to data clustering include hierarchical clustering, which entails a tree-like division of data, and partition clustering. The objective in data clustering is to determine cluster centers (centroids) and improve the partitioning through iterative relocation, with examples of partition clustering algorithms like K-means [5].

Several nature-inspired optimization algorithms have gained attention in search clustering approaches. These algorithms aim to optimize an objective function considering the sum of intra-cluster distances to find centroids. Examples from the literature include the Gray Wolf Optimization (GWO) [6], the Jaya Algorithm (JAYA) [7], Chaotic League Championship Algorithm (KSCLCA) [8], Salp Swarm Algorithm (SSA) [9], Dandelion Optimizer (DO) [10], Leader Slime Mould Algorithm (LSMA) [11], Flow Direction Algorithm (FDA) [12], Artificial Gorilla Troops Optimizer (GTO) [13], Mountain Gazelle Optimizer (MGO) [14], Prairie Dog Optimization Algorithm (PDO1) [15], Chimp Optimization Algorithm (CHIMP) [16] and Opposition African Vultures Optimization Algorithm (OAVOA) [17].

The Mountain Gazelle Optimization (MGO) algorithm proposed [14], mimics the social behaviors of mountain gazelles and utilizes factors like male herds, maternity herds, territorial males, and migration for food exploration. While MGO excels in benchmark functions and engineering problems, its application to data clustering remains challenging. We propose a variation, Chaos Mountain Gazelle Optimizer (CMGO), which integrates a chaotic map into the distribution strategy to address this issue. Furthermore, the Migration to Search for Food strategy in MGO is unsuitable for clustering problems. To address this, we modify the strategy distribution by integrating a chaotic map into the Territorial Solitary Males strategy while excluding the Migration to search for Food strategy. In data clustering, K-means clustering is commonly used, but it encounters challenges when computing distances between objects and centroids, especially with categorical and binary data. To address this, the Gower distance technique is integrated into K-means clustering. According to [18] using Gower's similarity coefficients improved the accuracy of the K-means algorithm in experiments with various datasets. We selected a total of 28 real datasets from the UCI and OpenML repositories to assess the performance of the proposed algorithms on three data types: numerical, categorical, and mixed. The effectiveness of the algorithm was compared against 14 state-of-the-art approaches. The evaluation employed the F-Measure metric to assess performance, and statistical significance ranking was conducted using the tied rank test. Results revealed that CMGO exhibited lower intra-cluster distance and higher F-Measure values, outperforming both the original Mountain Gazelle Optimization (MGO) algorithm and other tested algorithms. Specifically, CMGO secured the first position in clustering numeric and categorical data, while ranking third for mixed data.

The remainder of the paper is organized as follows: Sect. 2 provides an overview of K-means clustering problems and the traditional MGO algorithm. Section 3 introduces the proposed method, CMGO. In Sect. 4, we discuss the performance evaluation experiments. Section 5 presents the discussion. Finally, Sect. 6 conclusion.

2 Related Works

In this section, we will discuss the K-means algorithm for data clustering and introduce a traditional Mountain Gazelle Optimization (MGO) to enhance its capabilities.

2.1 K-means Clustering

The K-means clustering algorithm has received significant attention in the literature. Especially in nature-inspired optimization approaches, a large number of researchers employ optimization algorithms to search for cluster centers. These optimization algorithms aim to discover cluster centers by minimizing the objective function, which takes into account the sum of intra-cluster distances. The K-means algorithm partitions the dataset into K distinct clusters. The K-means algorithm operates through unsupervised learning. Based on the data points $X = [x_{1,}x_2x_3, \ldots x_N]$ and the positions of K cluster centroids $C = \{c_1, c_2, c_3, \ldots, c_k | \forall i = 1, \ldots, K : c_i \neq \varnothing \; and \; \forall i \neq j : c_i \cap c_j = \varnothing\}$. In clustering, each data point in set X is assigned to one of the K clusters in a manner that minimizes the objective fitness function. The sum of the squared Euclidean distance between data points x_N and the center of the cluster c_j is used as the objective function, as presented in Eq. (1).

$$f(k) = \sum_{k=1}^{K} \sum_{i=1}^{N_k} (x_i - c_k)^2, \tag{1}$$

where $k = 1, 2, ...K$ is the number of clusters, x_i, $i = 1, 2, ...N_k$ are the patterns in the kth cluster, c_k is center of the kth cluster. In this context, the cluster centers are depicted as:

$$c_k = \frac{1}{n_k} \sum_{i=1}^{n_k} x_i. \tag{2}$$

In this research, nature-inspired algorithms are employed for the purpose of identifying cluster centers within the dataset. The primary objective of the K-means algorithm is to determine optimal centers for each of the K clusters in a partitioned cluster.

2.2 Traditional Mountain Gazelle Optimization (MGO)

This section provides a brief explanation of the main inspiration behind the traditional MGO algorithm [14], followed by a description of the mathematical model.

Territorial Solitary Males

Male mountain gazelles establish solitary territories through intense territorial battles and competition for females, with adult males vigorously defending their boundaries.

$$TSM = male_{gazelle} - |(ri_1 \times BH - ri_2 \times X(t)) \times F| \times Cof_r \tag{3}$$

The $male_{gazelle}$ is the position vector of the best global solution, representing an adult male. ri_1 and ri_2 are random integer 1 or 2. The coefficient vector BH is the young male herd. Cof_r is randomly selected in each iteration.

$$BH = X_{ra} \times \lfloor r_1 \rfloor + M_{pr} \times \lfloor r_2 \rfloor, ra = \left(\left\lfloor \frac{N}{3} \right\rfloor \ldots N \right) \tag{4}$$

where X_{ra} is a random young male within the interval ra. M_{pr} is the average number of randomly selected search agents from a pool of N, based on the ceiling division of N by 3. N is the total number of gazelles. r_1 and r_2 are random values (0, 1].

$$F = N_1(D) \times exp\left(2 - Iter \times \left(\frac{2}{MaxIter} \right) \right) \tag{5}$$

N_1 is a randomly generated number from the standard distribution, exp is the exponential function, $MaxIter$ is the total iterations, and $Iter$ is the current iteration.

$$Cof_i = \begin{cases} (a+1) + r_3, \\ a \times N_2(D), \\ r_4(D), \\ N_3(D) \times N_4(D)^2 \times \cos((r_4 \times 2) \times N_3(D)), \end{cases} \tag{6}$$

r_3, r_4, and $rand$ are random numbers (0, 1). N_2, N_3 and N_4 are randomly generated numbers from a normal distribution, and cos represent the cosine function.

$$a = -1 + Iter \times \left(\frac{-1}{MaxIter} \right) \tag{7}$$

$MaxIter$ is the total number of iterations, while $Iter$ is the current iteration count.

Maternity Herds

Maternity herds facilitate robust male gazelle births, with active male participation in delivery and young males competing for dominance over females.

$$MH = \left(BH + Cof_{1,r} \right) + (ri_3 \times male_{gazelle} - ri_4 \times X_{rand}) \times Cof_r \tag{8}$$

ri_3 and ri_4 are random integers, either 1 or 2. The $male_{gazelle}$ is the global solution in the current iteration. X_{rand} is the position of a gazelle randomly selected from the entire population.

Bachelor Male Herds

Male gazelles establish territories and engage in intense battles for female possession, demonstrating dominance and control. This behavior is computed as follows:

$$BMH = (X(t) - D) + \left(ri_5 \times male_{gazelle} - ri_6 \times BH \right) \times Cof_r, \tag{9}$$

where $X(t)$ is the position vector of the gazelle in the current iteration. ri_5 and ri_6 are randomly selected integers, either 1 or 2.

$$D = \left(|X(t)| + \left| male_{gazelle} \right| \right) \times (2 \times r_6 - 1), \tag{10}$$

the parameter r_6 is a random value between 0 and 1.

Migration to Search for Food
The mathematical formulation representing the foraging and migratory behavior of mountain gazelles incorporates their ability to cover long distances and engage in migration, as well as their exceptional running speed and jumping abilities.

$$MSF = (ub - lb) \times r_7 + lb \tag{11}$$

ub and lb is upper and lower limits. r_7 is a randomly selected integer within the range of 0 and 1.

The mechanisms (TSM, MH, BMH, and MSF) are applied to all gazelles, generating new generations, and adding to the population. High-quality gazelles are preserved, while weak or old ones are removed, with the adult male gazelle considered the best among them.

3 Proposed method Chaotic Mountain Gazelle Optimizer

3.1 Motivation

The Mountain Gazelle Optimizer (MGO) algorithm draws inspiration from the social structure of wild mountain gazelles. While MGO demonstrates strong search capabilities in benchmark functions and engineering problems [14], its application to NP-complete real-world problems like data clustering remains challenging. To address this, we enhance MGO by incorporating a chaotic map into the Territorial Solitary Males strategy and excluding the Migration to Search for Food strategy. Additionally, we introduce the Gower distance technique to overcome challenges in computing distances for categorical and binary data in K-means clustering.

Chaotic Territorial Solitary Males Strategy
In our proposed CMGO algorithm, the Territorial Solitary Males strategy is enhanced by incorporating a chaotic map. The updated mathematical expression for the territory of adult male $TSMC^{t+1}$ is given by the following equation.

$$TSMC^{t+1} = male_{gazelle} - \left| \left(\left(\frac{C^{t+1}}{ri_1} \right) \times BH - ri_2 \times X(t) \right) \times F \right| \times Cof_r, \tag{12}$$

$male_{gazelle}$ is the position vector of the best global solution. ri_1 and ri_2 are random integers, either 1 or 2. The coefficient vector BH corresponds to the young male herds from the original MGO algorithm. F and Cof_r Similar to the original MGO.

The Chaotic Parameter
The parameters ri_1 and ri_2 serve as controls for updating the territory of the adult male $TSMC^{t+1}$ in our CMGO algorithm. The parameter ri_1 is a random integer, taking a value of either 1 or 2, and directly in fluences the search solution. If ri_1 is 1, the coefficient vector BH remains unchanged. However, when incorporating a chaotic map

into the computation of ri_1, the coefficient vector BH undergoes changes throughout the entire evolution process. Previous studies have demonstrated the seamless and effective integration of a chaotic map with the biogeography-based optimization (BBO) algorithm [19]. In our proposed CMGO algorithm, we introduce the use of the Piecewise map within the Territorial Solitary Males strategy.

The iterative form of the Piecewise map is defined as:

$$C^{t+1} = \begin{cases} \frac{C^t}{P}, & 0 \le C^t < P \\ \frac{C^t - P}{0.5 - P}, & P \le C^t < 0.5 \\ \frac{1 - P - C^t}{0.5 - P}, & 0.5 \le C^t < 1 - P \\ \frac{1 - C^t}{P}, & 1 - P \le C^t < 1 \end{cases}, P = 0.4 \tag{13}$$

where the parameter P is set to 0.4. The visualization of the Piecewise map is depicted in Fig. 1.

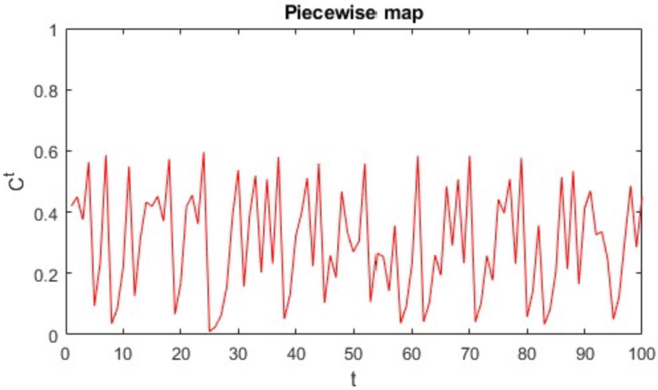


Fig. 1. The behavior of the Piecewise maps employed in our CMGO algorithm.

The Gower Similarity Coefficient

To improve the performance of K-means clustering when dealing with categorical and binary data, a similarity measure, such as the Gower coefficient [20] or the Gower distance technique, is used instead of the squared Euclidean distance to calculate the dissimilarity measure $D_{Gow}(X_n, C_j)$ during the clustering process. The Gower distance (D_{Gow}), is employed in this context. The computation of the Gower distance is as follows:

$$D_{Gow}(X_n, C_j) = \frac{\sum_{k=1}^{N_k} S_{njk}\delta_{njk}w_k}{\sum_{k=1}^{N_k} \delta_{njk}w_k} \tag{14}$$

In the case of binary and categorical attributes, $S_{njk} = 0$ if $X_{nk} = C_{jk}$ otherwise $S_{njk} = 1$. For continuous attributes, $S_{njk} = |X_{nk} - C_{jk}|/(max_l X_{lk} - min_l X_{lk})$, where

l run for all non-missing values for the attribute k. If we can compare X_n and C_j for the attribute k then $\delta_{njk} = 1$, zero otherwise. w_k is the weight for the attribute k. For simplicity, we will set $w_k = 1$. N_k the total number of species recorded across both units.

3.2 The Main Process of Proposed CMGO Algorithm

The Chaos Mountain Gazelle Optimizer (CMGO) algorithm is developed to tackle challenges in data clustering for various data types. This enhanced version of the Mountain Gazelle Optimizer (MGO) is specifically designed for K-means clustering solutions.

The relationship between the CMGO optimizer and K-means clustering can be explained as follows: We utilize the CMGO to optimize the cluster centers in K-means clustering. First, it initializes the cluster centers with random positions and then proceeds to perform the K-means algorithm from each of these random positions. Secondly, during the evolutionary process, the CMGO iteratively updates the position of the optimal cluster center. The process continues until it reaches the desired optimal position (the best cluster center). Lastly, all positions are assigned to the cluster centers, resulting in the output of the clustering results. The pseudo-code of the CMGO algorithm is also shown in Algorithm 1.

Algorithm 1 Pseudo-code of K-mean clustering based on CMGO

```
Input the population size $N$ and maximum number of iterations $T$
Initialize cluster centers on a random population using $X_i (i = 1, 2, \ldots, N)$
For i = 1 to $N$ Do
    Perform the K-means algorithm for each Gazelle formation, or row, of $X_i$
End for
Evaluate the cluster center (Gazelle's fitness) on $X_i$ by Eq. (1)
Generate the Chaos sequence $C^{t+1}$
While (stopping condition is not met)
    For (each Gazelle ($X_i$)) Do
        Calculate $TSMC^{t+1}$ using Eq. (12)
        Calculate $MH$ using Eq. (8)
        Calculate $BMH$ using Eq. (9)
        Calculate the fitness values of $TSMC^{t+1}$, $MH$, $BMH$, Then add to habitat
    End for
    Sort the entire population in ascending order
    Update $best_{gazelle}$
    save the $N$ Best Gazelle in Max number of population
End while
Return $X_{bestGazelle}$, best Fitness (the best cluster center)
```

4 Experimental Results and Analysis

The experiments were conducted using MATLAB R2022a 64-bit on a desktop computer with an AMD Ryzen 9 5950X 16-Core Processor (3.40 GHz), 32.00 GB RAM, SSD M.2 500 GB, and Microsoft Windows 11 Professional 64-bit operating system.

4.1 Performance Evaluation

The CMGO algorithm was evaluated against competing algorithms using UCI and OpenML datasets, employing the F-Measure metric and tied rank test for statistical significance rankings. The F-Measure, which integrates both precision and recall, served as the metric for evaluating performance, can be calculated by a confusion matrix as follows:

$$F - Measure(x) = \frac{2 \times precision \times recall}{precision + recall} \times 100, \tag{17}$$

where, precision and recall are calculated using the following equations based on a confusion matrix:

$$precision = \frac{TP}{TP + FP}, \tag{15}$$

$$recall = \frac{TP}{TP + FP}, \tag{16}$$

where, TP represents true positives, FP corresponds to false positives, and FN signifies false negatives. Higher precision values indicated superior algorithm performance, while greater recall captured more true positives, thereby indicating improved performance in correctly identifying positive instances. The evaluation of the F-Measure occurs upon termination of the optimization algorithm. A higher F-Measure leads to increased clustering accuracy. In [8] stressed the importance of a low objective fitness value for accurate cluster formation, guiding our adoption of the F-measure metric to evaluate and achieve precise clusters.

The Benchmark Dataset for Clustering

We partitioned the benchmark dataset into three distinct groups: numerical datasets, categorical datasets, and mixed-data type datasets. We utilized a total of 28 well-known datasets taken from the UCI [21] and OpenML [22] repositories. The numerical dataset consisted of: Iris, Glass, Breast-Cancer-Wisconsin, Wine, Thyroid, Synthetic-Control-Charts, Ionosphere, Sonar, Diabetes, Ecoli, and Banknote-Authentication. The categorical dataset included: Balance Scale, Hayes-Roth, Monks, SPECT-Heart, asnd Nursery. The mixed-data type dataset encompassed the following datasets: Acute-Inflammations, Analcatdata-Seropositive, Churn, Cloud, Fruitfly, Haberman, Newton-Hema, Sleuth-Case2002, Socmob, Tae, Heart-Disease, and ACA. By incorporating diverse datasets representing different data types, we aimed to comprehensively evaluate the performance of our algorithm across various scenarios. The characteristics of the three datasets are presented in Table 1.

4.2 Experimental Results

To verify the proposed CMGO, we compared it against 14 algorithms on 28 UCI and OpenML datasets. The algorithms used for comparison included: Opposition African Vultures Optimization Algorithm (OAVOA) [17], Salp Swarm Algorithm (SSA) [9],

Table 1. The characteristics of the three datasets.

Numeric	Categorical	Mixed-type dataset
1. Iris (N=150, D=4, k=3) 2. Glass (N=214, D=9, k=6) 3. Breast-Cancer-Wisconsin (N=699, D=9, k=2) 4. Wine (N=1178, D=13, k=3) 5. Thyroid (N=215, D=5, k=3) 6. Synthetic-Control-Chart (N=600, D=60, k=6) 7. Ionosphere (N=351, D=34, k=2) 8. Sonar (N=208, D=60, k=2) 9. Diabetes (N=768, D=8, k=2) 10. Ecoli (N=336, D=7, k=4) 11. Banknote-Authentication (N=1370, D=4, k=2)	1. Balance-Scale (N=625, D=4, k=3) 2. Hayes-Roth (N=160, D=4, k=3) 3. Monks (N=432, D=6, k=2) 4. SPECT-Heart (N=267, D=22, k=2) 5. Nursery (N=12,960, D=8, k=2)	1. Acute-Inflammations (N=120, D=6(c=5, η=1), k=2) 2. Analcatdata-Seropositive (N=132, D=3(c=1, η=2), k=2) 3. Churn (N=5000, D=20(c=4, η=16), k=2) 4. Cloud (N=108, D=7(c=1, η=6), k=2) 5. Fruitfly (N=125, D=4(c=2, η=2), k=2) 6. Haberman (N=306, D=3(c=1, η=2), k=2) 7. Newton-Hema (N=140, D=3(c=1, η=2), k=2) 8. Sleuth-Case2002 (N=147, D=6(c=4, η=2), k=2) 9. Socmob (N=1156, D=5(c=4, η=1), k=2) 10. Tae (N=151, D=5(c=2, η=3), k=2) 11. Heart-Disease (N=1025, D=13(c=8, η=5), k=2) 12. ACA (N=690, D=14(c=8, η=6), k=2)

* N = Number of data, D = Number of Dimension(c = categorical, η = numerical), k = Number of Center

Artificial Gorilla Troops Optimizer (GTO) [13], Jaya Algorithm (JAYA) [7], Dandelion Optimizer (DO) [10], Gray Wolf Optimization (GWO) [6], modified particle swarm optimization (MPSO) [23], Leader Slime Mould Algorithm (LSMA) [11], Flow Direction Algorithm (FDA) [12], Mountain Gazelle Optimizer (MGO) [14], Prairie Dog Optimization Algorithm (PDO1) [15], Time-varying Acceleration Coefficients Particle Swarm Optimization algorithm (TACPSO) [23], Chimp Optimization Algorithm [16], and Chaotic League Championship Algorithm (KSCLCA) [8].

Table 2, the analysis encompasses numeric, categorical, and mixed data types. The average tied rank of 15 algorithms is determined based on the F-Measure. The average tied rank (Avg. tied rank) displayed in Table 2 represents the mean scores of tied rank scores for the F-Measure of each algorithm when implemented on datasets belonging to their respective data types. For numeric data, CMGO achieves the top rank with an average score (avg.sc) of 4.27, outperforming the original MGO ranking which holds the 6 positions with an avg.sc of 6.55. When considering Categorical data, the algorithm ranking first obtains an avg.sc of 4.40, demonstrating superior performance compared to the original MGO ranking at the 8 positions with an avg.sc of 7.70. In the case of Mixed data, the algorithm secures rank 3 with an avg.sc of 6.46, surpassing the original MGO ranking at the 6 positions with an avg.sc of 6.88.

Table 3 presents a thorough evaluation of data clustering performance, comparing the MGO and the proposed CMGO algorithms across three groups of datasets. The findings consistently indicated that CMGO outperformed the original MGO algorithm across the datasets, with a ratio of 18:9 in favor of CMGO. The performance measures utilized for evaluation were the F-Measure and tied rank.

Table 2. Compares the tied rank of 15 algorithms across three datatypes.

Methods	Tied rank of average F-measure					
	Numeric		Categorical		Mixed data type	
	Avg. tied rank	Rank	Avg. tied rank	Rank	Avg. tied rank	Rank
OAVOA	6.86	7	8.80	10	7.38	7
SSA	9.18	10	7.20	7	8.21	8
GTO	4.82	3	6.60	5	9.92	13
JAYA	4.32	2	9.40	11	8.88	10
DO	7.91	9	8.70	9	6.38	2
GWO	9.55	12	10.20	13	9.54	12
MPSO	12.73	14	9.80	12	10.33	14
LSMA	9.50	11	5.60	2	8.29	9
FDA	5.91	4	6.40	4	6.54	4
MGO	6.55	6	7.70	8	6.88	6
PDO1	11.18	13	7.00	6	11.08	15
TACPSO	7.73	8	6.10	3	6.58	5
CHIMP	13.36	15	10.60	14	9.04	11
KSCLCA	6.14	5	11.50	15	**4.50**	**1**
Proposed CMGO	**4.27**	**1**	**4.40**	**1**	6.46	3

Table 4 presents the average ranking, tied rank, and average tied rank for all 3 types of information based on the F-Measure. The rankings are associated with a set of 15 algorithms and all datasets. The algorithms are ranked as follows: CMGO, FDA, KSCLCA, MGO, TACPSO, JAYA, GTO, DO, OAVOA, LSMA, SSA, GWO, PDO1, CHIMP, and MPSO, respectively. CMGO achieved the first rank. In contrast, MGO obtained a rank of 4. These results highlight the capability of the proposed CMGO approach to enhance the performance of the original MGO algorithm for all datasets.

Table 3. The comparative of the tied rank between the MGO and proposed CMGO algorithms.

Types	Datasets	F-Measure		Rank	
		MGO	Proposed CMGO	MGO	Proposed CMGO
Numerical datasets	Iris	85.81717	**89.80941**	2	**1**
	Glass	**43.67836**	38.16387	**1**	2
	Breast-Cancer-Wisconsin	**95.12551**	95.11379	**1**	2
	Wine	72.46922	**72.74876**	2	**1**
	Thyroid	67.37993	**72.45515**	2	**1**
	Synthetic-Control-Chart	42.79285	**43.47290**	2	**1**
	Ionosphere	**68.18605**	68.13166	**1**	2
	Sonar	**53.93852**	49.75967	**1**	2
	Diabetes	59.02891	**59.55515**	2	**1**
	Ecoli	72.21830	**74.65103**	2	**1**
	Banknote-Authentication	59.07605	**59.30723**	2	**1**
Categorical datasets	Balance-Scale	50.10709	**50.11917**	2	**1**
	Hayes-Roth	36.78197	**37.66778**	2	**1**
	Monks	49.69362	**49.71152**	2	**1**
	SPECT-Heart	62.98250	**63.10270**	2	**1**
	Nursery	38.50658	**40.71493**	2	**1**
Mixed-data type datasets	Acute-Inflammations	**77.41939**	74.90621	**1**	2
	Analcatdata-Seropositive	**82.81519**	82.79930	**1**	2
	Churn	57.12235	**59.17942**	2	**1**
	Cloud	**56.49374**	**56.49374**	**1**	**1**
	Fruitfly	52.90640	**52.94304**	2	**1**
	Haberman	50.53389	**51.79722**	2	**1**
	Newton-Hema	**63.48013**	63.43048	**1**	2
	Sleuth-Case2002	56.90424	**59.32102**	2	**1**
	Socmob	85.28709	**86.42502**	2	**1**
	Tae	62.99309	**63.16615**	2	**1**
	Heart-Disease	**75.16155**	66.17032	**1**	2
	ACA	**67.83375**	56.90297	**1**	2

Table 4. The comparative analysis of the tied rank of 15 algorithms, utilizing the F-Measure metric computed for all 28 datasets.

Types	Datasets	Methods														
		OAVOA	SSA	GTO	JAYA	DO	GWO	MPSO	LSMA	FDA	MGO	PDO1	TACPSO	CHIMP	KSCLCA	Proposed
	Iris	12.0	6.0	8.0	**2.5**	10.0	10.0	7.0	10.0	**2.5**	13.0	14.0	**2.5**	15.0	**2.5**	5.0
Numerical datasets	Glass	4.0	13.0	5.0	3.0	7.0	11.0	12.0	9.0	6.0	2.0	15.0	8.0	14.0	**1.0**	10.0
	Breast-Cancer-Wisconsin	4.0	3.0	8.5	6.0	6.0	12.0	15.0	10.0	11.0	2.0	**1.0**	13.0	14.0	8.5	6.0
	Wine	6.0	9.0	7.0	5.0	3.0	4.0	13.0	11.0	10.0	8.0	15.0	12.0	14.0	**1.0**	2.0
	Thyroid	3.0	14.0	5.0	4.0	6.0	7.0	12.0	10.0	2.0	9.0	15.0	8.0	11.0	13.0	**1.0**
	Synthetic-Control-Chart	6.0	7.0	**1.0**	8.0	14.0	15.0	13.0	11.0	5.0	4.0	9.0	10.0	12.0	3.0	2.0
	Ionosphere	6.5	9.0	**1.5**	3.0	11.0	12.0	15.0	10.0	6.5	**1.5**	14.0	8.0	13.0	5.0	4.0
	Sonar	5.0	7.0	2.0	3.0	9.0	13.0	15.0	11.0	**1.0**	4.0	14.0	10.0	12.0	6.0	8.0
	Diabetes	10.0	10.0	7.0	10.0	10.0	2.0	14.0	4.0	10.0	5.5	15.0	3.0	13.0	5.5	**1.0**
	Ecoli	7.0	10.0	4.0	2.0	6.0	13.0	14.0	11.0	**1.0**	8.0	9.0	3.0	15.0	12.0	5.0
	Banknote-Authentication	12.0	13.0	4.0	**1.0**	5.0	6.0	10.0	7.5	10.0	15.0	2.0	7.5	14.0	10.0	3.0
Categorical datasets	Balance-Scale	14.0	6.0	7.0	4.0	10.0	8.0	12.0	11.0	3.0	2.0	9.0	5.0	13.0	15.0	**1.0**
	Hayes-Roth	13.0	6.0	**1.0**	7.0	12.0	9.0	4.0	5.0	8.0	15.0	3.0	10.0	2.0	14.0	11.0
	Monks	**1.0**	9.0	11.0	13.0	6.5	10.0	12.0	2.0	4.0	6.5	14.0	6.5	15.0	6.5	3.0
	SPECT-Heart	5.0	11.0	6.0	13.0	3.0	10.0	12.0	9.0	14.0	2.0	4.0	7.0	8.0	15.0	**1.0**
	Nursery	11.0	4.0	8.0	10.0	12.0	14.0	9.0	**1.0**	3.0	13.0	5.0	2.0	15.0	7.0	6.0
Mixed-data type datasets	Acute-Inflammations	5.0	**1.0**	9.0	15.0	12.0	13.0	11.0	8.0	6.0	2.0	3.0	10.0	14.0	7.0	4.0
	Analcatdata-Seropositive	11.0	5.0	14.0	12.0	3.0	6.0	13.0	2.0	9.0	7.0	15.0	4.0	10.0	**1.0**	8.0
	Churn	3.0	6.0	9.0	2.0	13.0	7.0	12.0	10.0	4.0	14.0	**1.0**	8.0	5.0	15.0	11.0
	Cloud	8.5	8.5	**1.0**	8.5	8.5	8.5	2.0	8.5	8.5	8.5	15.0	8.5	8.5	8.5	8.5

(continued)

Table 4. (*continued*)

Types	Datasets	Methods														
		OAVOA	SSA	GTO	JAYA	DO	GWO	MPSO	LSMA	FDA	MGO	PDO1	TACPSO	CHIMP	KSCLCA	Proposed
	Fruitfly	4.0	13.0	8.0	9.0	7.0	12.0	10.0	11.0	3.0	6.0	15.0	**1.0**	14.0	2.0	5.0
	Haberman	14.0	7.0	10.0	5.0	8.0	9.0	**1.0**	3.0	4.0	15.0	2.0	11.0	6.0	13.0	12.0
	Newton-Hema	6.0	5.0	10.0	9.0	4.0	15.0	11.0	13.0	7.0	**1.5**	12.0	8.0	14.0	**1.5**	3.0
	Sleuth-Case2002	5.0	11.0	10.0	12.0	6.0	9.0	14.0	13.0	7.0	8.0	15.0	3.0	**1.0**	2.0	4.0
	Socmob	10.0	3.0	13.0	12.0	8.0	9.0	15.0	4.0	7.0	6.0	14.0	5.0	11.0	**1.0**	2.0
	Tae	15.0	9.0	13.0	11.0	3.0	3.0	12.0	3.0	10.0	7.5	14.0	7.5	6.0	**1.0**	5.0
	Heart-Disease	3.0	15.0	10.0	6.0	2.0	14.0	12.0	11.0	7.0	4.0	13.0	5.0	9.0	**1.0**	8.0
	ACA	4.0	15.0	12.0	5.0	2.0	9.0	11.0	13.0	6.0	3.0	14.0	8.0	10.0	**1.0**	7.0
Average Rank		7.43	8.41	7.32	7.18	7.39	9.66	11.18	8.29	6.27	6.89	10.39	6.95	11.02	6.39	**5.23**
		9	11	7	6	8	12	15	10	2	4	13	5	14	3	**1**

5 Discussion

In this section, we compare the exploitation and exploration abilities of the original MGO algorithm and the proposed CMGO algorithm. The evaluation employed strategies such as TSM, MH, BMH, and MSF, revealing differences between the two algorithms (Fig. 2). The Socmob dataset, representing mixed data types, was used to compare their performance. Results indicated that in the original MGO algorithm, there was an initial emphasis on exploitation (as observed in the TSM strategy on the red line), which gradually increased until reaching the final iteration. While its exploration capability (as seen in the MH strategy on the green line) isn't heavily emphasized in the initial stages, it also gradually diminishes until reaching the final iteration. On the contrary, the CMGO algorithm intentionally reduced exploitation (as observed in the TSM strategy on the red line) to prevent premature convergence. However, it experienced a slight reduction that continued until the final iteration. To achieve a more effective balance between exploration and exploitation, the focus on the exploration capability (MH strategy on the green line) begins with less emphasis in the initial stages, increases rapidly during the intermediate stages, and then remains almost constant until the final iteration. The same behavioral curves can be observed for both BMH and MSF strategies. Note that the MSF strategy was removed from the CMGO algorithm due to the negligible changes observed throughout the evaluation process.

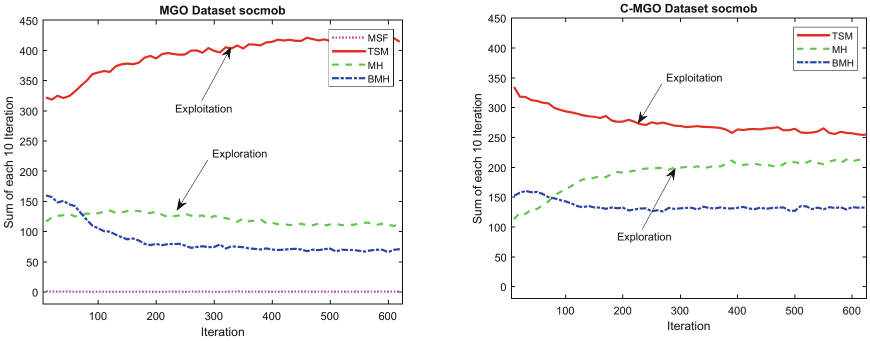


Fig. 2. The behaviors of the 4 strategies of the MGO and CMGO for Mixed data type.

It is worth noting, as shown in Table 2, that the mixed data type in all datasets exhibits only two classes (cluster center) and generally has a relatively small number of dimensions, while the other two data types, numerical and categorical, differ. It can be assumed that finding a solution for the problem of mixed data types with fewer classes is not challenging. Some algorithms with a high degree of exploitation ability, such as the KSCLCA algorithm, perform exceptionally well on this problem, ranking first. In contrast, the proposed CMGO algorithm dropped to third place in the context of mixed data types. It can be inferred that the CMGO algorithm aims to enhance the balance between exploration and exploitation abilities by reducing exploitation and increasing exploration, albeit to a lesser degree than the KSCLCA algorithm. Nevertheless, the

CMGO's performance ranks it among the top three algorithms, securing the third position, with a slight variation from the second-ranked DO algorithm. In summary, it can be inferred that our proposed CMGO algorithm demonstrates its effectiveness particularly when dealing with problems having more than two classes. Further exhaustive investigations will be pursued in future work.

6 Conclusion

We anticipate that the findings and techniques presented in this study will prove valuable to individuals and researchers who have a keen interest in advancing the field of data clustering. The CMGO algorithm, along with the integration of the Gower distance technique, offers novel insights and solutions for addressing challenges in clustering diverse data types. By introducing the CMGO algorithm, we have expanded the capabilities of the traditional MGO for K-means clustering. The incorporation of a chaotic map into the Territorial Solitary Males strategy and the exclusion of the Migration to Search for Food strategy have enhanced CMGO's exploration and exploitation abilities. This adjustment allows CMGO to effectively handle complex datasets by striking a balance between thorough exploration and efficient exploitation of the solution space. Furthermore, our utilization of the Gower distance technique has overcome the limitations of K-means clustering when dealing with categorical and binary data. This technique has enabled CMGO to accurately compute distances between objects and cluster centers, ensuring reliable clustering results across a wide range of data types. We believe that the comprehensive evaluation of CMGO against 14 other state-of-the-art algorithms using 28 diverse datasets adds significant value to the field. The use of the F-Measure metric and the tied rank test for statistical significance ranking provides robust and reliable measures of CMGO's performance. The results clearly demonstrate CMGO's superiority over the original MGO and other tested algorithms, particularly in clustering pure numeric and categorical data.

In summary, we are confident that the insights and innovations presented in this study will inspire further developments in the field of data clustering. The CMGO algorithm, along with the integration of the Gower distance technique, offers a promising avenue for researchers and practitioners to tackle the challenges posed by diverse datasets. We hope that our contributions will serve as a foundation for future advancements in the field and encourage further exploration and experimentation in this area of study.

Acknowledgments. The authors extend their gratitude to Tanachapong Wangchamhan, Ph.D. (tanachapong.wa@ksu.ac.th) for generously sharing his intellectual resources throughout the duration of this project.

References

1. Provost, F., Fawcett, T.: Data science and its relationship to big data and data-driven decision making. Big Data **1**(1), 51–59 (2013). https://doi.org/10.1089/big.2013.1508
2. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer, New York

3. Bahmani-Firouzi, B., Shasadeghi, M., Niknam, T.: A new hybrid algorithm based on PSO, SA, and K-means for cluster analysis. Int. J. Innov. Comput. Inform. Control **6**, 3177–3192 (2010)
4. Niknam, T., Amiri, B.: An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. Appl. Soft Comput. J. **10**(1), 183–197 (2010). https://doi.org/10.1016/j.asoc.2009.07.001
5. Krishnasamy, G., Kulkarni, A.J., Paramesran, R.: A hybrid approach for data clustering based on modified cohort intelligence and K-means. Expert Syst. Appl. **41**(13), 6009–6016 (2014). https://doi.org/10.1016/j.eswa.2014.03.021
6. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014). https://doi.org/10.1016/j.advengsoft.2013.12.007
7. Venkata Rao, R.: Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int. J. Indust. Eng. Comput. **7**(1), 19–34 (2016). https://doi.org/10.5267/j.ijiec.2015.8.004
8. Wangchamhan, T., Chiewchanwattana, S., Sunat, K.: Efficient algorithms based on the k-means and Chaotic League Championship Algorithm for numeric, categorical, and mixed-type data clustering. Expert Syst. Appl. **90**, 146–167 (2017). https://doi.org/10.1016/j.eswa.2017.08.004
9. Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M.: Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems. Adv. Eng. Softw. **114**, 163–191 (2017). https://doi.org/10.1016/j.advengsoft.2017.07.002
10. Zhao, S., Zhang, T., Ma, S., Chen, M.: Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications. Eng. Appl. Artif. Intell. **114** (2022). https://doi.org/10.1016/j.engappai.2022.105075
11. Naik, M.K., Panda, R., Abraham, A.: Normalized square difference based multilevel thresholding technique for multispectral images using leader slime mould algorithm. J. King Saud Univ. Comput. Inform. Sci. **34**(7), 4524–4536 (2022). https://doi.org/10.1016/j.jksuci.2020.10.030
12. Karami, H., Anaraki, M.V., Farzin, S., Mirjalili, S.: Flow Direction Algorithm (FDA): a novel optimization approach for solving optimization problems. Comput. Ind. Eng. **156** (2021). https://doi.org/10.1016/j.cie.2021.107224
13. Abdollahzadeh, B., Gharehchopogh, F.S., Mirjalili, S.: Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Int. J. Intell. Syst. **36**(10), 5887–5958 (2021). https://doi.org/10.1002/int.22535
14. Abdollahzadeh, B., Gharehchopogh, F.S., Khodadadi, N., Mirjalili, S.: Mountain gazelle optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Adv. Eng. Software **174**, 103282 (2022). https://doi.org/10.1016/j.advengsoft.2022.103282
15. Ezugwu, A.E., Agushaka, J.O., Abualigah, L., Mirjalili, S., Gandomi, A.H.: Prairie dog optimization algorithm. Neural Comput. Appl. **34**(22), 20017–20065 (2022). https://doi.org/10.1007/s00521-022-07530-9
16. Khishe, M., Mosavi, M.R.: Chimp optimization algorithm. Expert Syst. Appl. **149**, 113338 (2020). https://doi.org/10.1016/j.eswa.2020.113338
17. Jena, B., Naik, M.K., Panda, R., Abraham, A.: A novel minimum generalized cross entropy-based multilevel segmentation technique for the brain MRI/dermoscopic images. Comput. Biol. Med. **151**, 106214 (2022). https://doi.org/10.1016/j.compbiomed.2022.106214
18. Ben Ali, B., Massmoudi, Y.: K-means clustering based on gower similarity coefficient: a comparative study. In: 2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO), pp. 1–5 (2013). https://doi.org/10.1109/ICMSAO.2013.6552669
19. Saremi, S., Mirjalili, S., Lewis, A.: Biogeography-based optimisation with chaos. Neural Comput. Appl. **25**(5), 1077–1097 (2014). https://doi.org/10.1007/s00521-014-1597-x

20. Gower, J.C.: A general coefficient of similarity and some of its properties. Biometrics **27**(4), 857–871 (1971). https://doi.org/10.2307/2528823
21. Markelle, K., Rachel, L., Kolby, N.: UCI Dataset. The UCI Machine Learning Repository. https://archive.ics.uci.edu. Accessed 24 Jun 2023
22. Vanschoren, J., van Rijn, J.N., Bischl, B., Torgo, L.: OpenML: networked science in machine learning. SIGKDD Explor. **15**(2), 49–60 (2013). https://doi.org/10.1145/2641190.2641198
23. Mirjalili, S., Lewis, A., Sadiq, A.S.: Autonomous particles groups for particle swarm optimization. Arab. J. Sci. Eng. **39**(6), 4683–4697 (2014). https://doi.org/10.1007/s13369-014-1156-x

Hybridization of Modified Grey Wolf Optimizer and Dragonfly for Feature Selection

Said Al Afghani Edsa and Khamron Sunat(✉)

College of Computing, Khon Kaen University, Khon Kaen 40000, Thailand
saidal.a@kkumail.com, skhamron@kku.ac.th

Abstract. There are numerous techniques designed to enhance the performance of machine learning models, with feature selection being one of the key strategies. Although many feature selection methods exist, our study presents a novel hybrid approach that merges two metaheuristic techniques: the Modified Grey Wolf Optimizer (MGWO) and the Dragonfly Algorithm (DA). This innovative method not only boosts the model's performance but also emphasizes the most pertinent features. Our experimental results showcase robust model performance, achieving an F1-score of 90% on our experimental dataset, surpassing other approaches. Further results and discussions are provided in this paper, .

Keywords: Modified Grey Wolf Optimizer · Dragonfly Algorithm · Support Vector Classifier · Feature · Selection · Model Performance

1 Introduction

There are several fundamental strategies for improving the performance of machine learning models. These strategies encompass adding more predictor features, enlarging the training dataset, adjusting or updating model parameters, enhancing feature engineering, and data preprocessing, among other techniques.

Numerous researchers have developed methods to enhance model performance [3, 8], with feature selection emerging as a leading strategy. While various feature selection techniques are available, the adoption of metaheuristic approaches has been on the rise. In light of this, our study aims to formulate an effective feature selection method that optimizes model outcomes. Our primary goal is to identify and prioritize crucial features that substantially contribute to optimal model results.

It's worth noting that previous research [11] has advocated for feature selection using the Modified Grey Wolf Optimization, especially for high-dimensional data. Yet, they encountered challenges, particularly in the evaluation of the fitness function. From their findings, it became evident that while the objective was to achieve both outstanding model performance and a concise set of selected features, the enhancement of the fitness function was lacking. To address this gap, we incorporated a rapid fitness function, designed to boost accuracy during the training phase and consequently cut down on training time. Simultaneously, we adopted a sampling technique tailored for large datasets and maintained model consistency using cross-validation throughout both the

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 35–42, 2023.
https://doi.org/10.1007/978-981-99-7969-1_3

feature selection and training stages. Our analysis further delves into a performance comparison, emphasizing the role of the Support Vector Machine in classification tasks.

We present a nature-inspired feature selection approach that combines the Modified Grey Wolf Optimization (MGWO) with the Dragonfly Algorithm. Our ambition with this hybrid technique is to pinpoint the most relevant features and achieve unparalleled model performance.

2 Research Method

This research uses a literature study and experimental approach. Other researchers such as Seyedali Mirjalili in his research [8] developed GWO hybridization for the case of feature selection with a binary approach, inspired by that we try to develop a combination of GWO and DA which is used as an indicator of feature selection, which aims to enrich the feature selection method, moreover this principle can also be used for other optimization cases because this method is derived from the meta heuristic method as well.

The process flow carried out in this experiment is given below (Fig. 1):

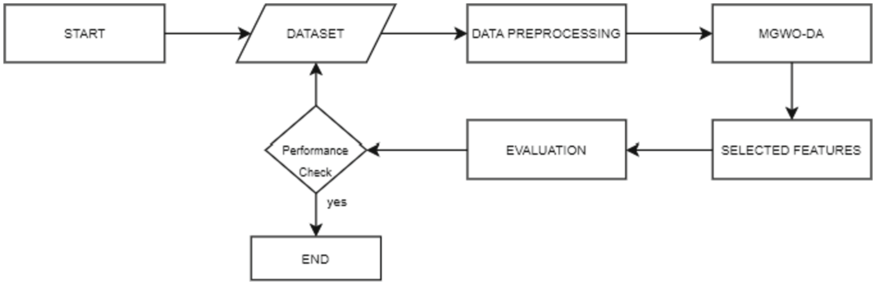


Fig. 1. Process flow.

A. Grey Wolf Optimizer (GWO)

Grey wolf Optimization (GWO) is the swarm intelligence optimization technique which was first introduced in [3]. It is inspired by the leadership hierarchy and hunting process of the grey wolf in nature. The simple mechanism of GWO makes it easy to implement over other NIAs. Also, it has fewer decision variables, less storage required, and does not possess any rigorous mathematical equations of the optimization problem. Muro [5] explained the hunting behavior of wolf into three stages as:

1. Social hierarchy: The social hierarchy of grey wolf has four levels: alpha α, beta β, delta δ and omega ω. The leaders are responsible for decision making and is denoted as the alpha wolf. The second level, called beta wolf works as a helping hand to the alpha for any activity. In the third level, the delta wolf is placed, which plays the role of scapegoat in grey-wolf packing. The rest of the wolves are categorized as omega wolf and is dominated by all other wolves.

2. Encircling the Prey: The encircling process is given by the following mathematical equation:

$$\boldsymbol{D} = \boldsymbol{C}.\boldsymbol{X}_{p,t} - \boldsymbol{X}_t, \tag{1}$$

$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_{p,t} - \boldsymbol{A}.\boldsymbol{D} \tag{2}$$

   where **A** and **C** are coefficient vectors, $\boldsymbol{X_{p,t}}$ denotes the position vector of the prey at current iteration $\boldsymbol{t}$, and $\boldsymbol{X_{t+1}}$ denotes the position vector of a grey wolf at next iteration. The vectors are determined as:

$$\boldsymbol{A} = 2\boldsymbol{a}.\boldsymbol{r}_1 - \boldsymbol{a}, \tag{3}$$

$$\boldsymbol{C} = 2\boldsymbol{r}_2 \tag{4}$$

   with condition vector **a** is a linearly decreasing parameter from 2 to 0 and $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are random vectors in [0,1].
3. Hunting To encircle the position of prey, and the wolf position is approximated by the average of the position guided by alpha ($\boldsymbol{X}_1$), beta ($\boldsymbol{X}_2$), and gamma ($\boldsymbol{X}_3$) wolves [3]. The position of prey is estimated as:

$$\boldsymbol{X}_{t+1} = \frac{\boldsymbol{X}_1 + \boldsymbol{X}_2 + \boldsymbol{X}_3}{3}. \tag{5}$$

Many researchers have attempted modifications of the GWO, ranging from aspects of movement mutation to hunting strategy alterations by changing existing formulas, as seen in [4, 8, 11]. In this paper, we draw inspiration from [11] due to its thorough modifications, which span from the Initialization strategy, competitive strategy, multi-convergence factor, to the incorporation of differential evolution. Given the extensive modifications made by these researchers, we've endeavored to further modify it in terms of position updates by integrating the Dragonfly Algorithm.

B. Dragonfly Algorithm (DA)

The Dragonfly algorithm is inspired by the behavior of those who form a flock that has static and dynamic properties. Static and dynamic properties are then raised into a meta heuristic method because the meta heuristic has the main properties of exploration (static in terms of flocks) and exploitation (when it has found an optimal). In the simulation for the mathematical model, the dragonfly's behavior is divided into several parts, namely separation, alignment, cohesion, attraction for food, and distraction from enemies. In summary, the position of all dragonflies in a swarm is determined by the following equation, which $\boldsymbol{X_t}$ describes the position of the dragonfly by the follwong formula:

$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t + \Delta\boldsymbol{X}_{t+1}, \tag{6}$$

where

$$\Delta\boldsymbol{X}_{t+1} = (sS_i + aA_i + cC_i + fF_i + w\Delta\boldsymbol{X}_t). \tag{7}$$

The position vector (6) gives the position of the dragonfly. However, when there are no neighboring solutions, the dragonflies are required to fly in random search space, and their position is updated using the modified equation for the position vector (see [6]):

$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t + \boldsymbol{X}_t.levy(d). \tag{8}$$

C. Hybridization: MGWO – DA (Proposed Method).

Similarly to GWO - PSO, a vector a is used which has a value from 2 and 0 and can be calculated by:

$$a_{da} = 2 - l(\frac{2}{Max\_iter}), \tag{9}$$

where l is the iteration factor. To combine MGWO [11] and DA we try to use the generalized position by using the levy multiplier in the case of DA with GWO which is formed in the following equation:

$$\boldsymbol{X}_{\boldsymbol{i}}^{d} = (\boldsymbol{X}_{\boldsymbol{j}}^{d} - \boldsymbol{A}_{\boldsymbol{k}}\boldsymbol{D}_{\boldsymbol{l}}) * Levy + \boldsymbol{X}_{\boldsymbol{j}}^{d} \tag{10}$$

with $\boldsymbol{i}, \boldsymbol{k} = 1, 2, 3; \boldsymbol{j}, \boldsymbol{l}$ represents for α, β and δ. The updated position of the wolf is given by:

$$\boldsymbol{X}_{t+1} = \boldsymbol{X}_t + r.(\boldsymbol{X}_\alpha - \boldsymbol{X}_t) + levy(d).\boldsymbol{X}_t, \tag{11}$$

where r comes from random number from 0 to 1.

This concept is rooted in the idea that, in the absence of prey or during the search for prey, wolves exhibit random movement, directing themselves either toward potential prey or other destinations. We aim to integrate this behavior with that of dragonflies, which also move randomly in the absence of neighboring individuals in their vicinity [10]. This combination can provide wolves with a potential advantage in exploration (Fig. 2).

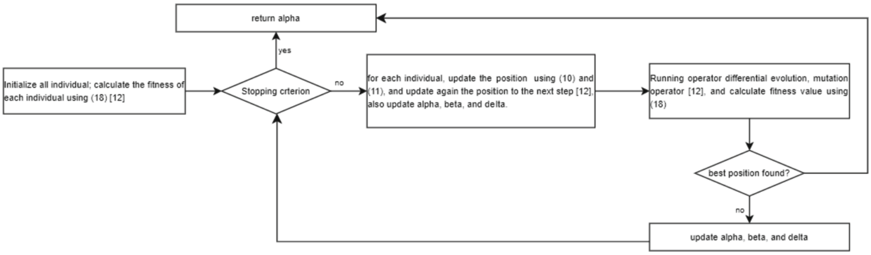

return alpha
yes
Initialize all individual; calculate the fitness of each individual using (18) [12]
Stopping crterion
no
for each individual, update the position using (10) and (11), and update again the position to the next step [12], also update alpha, beta, and delta.
Running operator differential evolution, mutation operator [12], and calculate fitness value using (18)
best position found?
no
update alpha, beta, and delta

Fig. 2. Overview of the Procedure.

D. Support Vector Machine: Classifier

For details related to the support vector machine (SVM) model see [7]. The following is the general procedure related to this SVM:

Variables and parameters:

$X = \{x_1, x_2, \ldots, x_n\}$: training sample
$Y = \{y_1, y_2, \ldots, y_n\} \subset \{\pm 1\}$: label from sample (binary case)

Kernel : type of kernel used

Par : parameter of the kernel
C : cost slak (penalty)

$\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]$: lagrange multiplier
b : bias

Procedure:

1. Calculate the kernel K matrix
2. Determine constraints for quadratic programming (QP)
3. Determine objective function of (2) using:

$$\underset{\alpha}{argmax} \sum \alpha_i - \frac{1}{2} \sum\nolimits_{i=1}^{l} \sum\nolimits_{j=1}^{l} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{12}$$

Subject to:

$$\sum\nolimits_{i=1}^{l} \alpha_i y_i = 0; C_i \geq \alpha_i \geq 0, \; for \; i = 1, 2, \ldots, l \tag{13}$$

4. Find the solution of QP, and α_i, b
5. Use the output from (4) to perform predictions:

$$y_{prediction} = sign\left(w^T + b\right) \tag{14}$$

$$y_{prediction} = sign(\sum\nolimits_{i=1}^{l} \alpha_i y_i (x_i x) + b) \tag{15}$$

The hybridization of MGWO-DA is employed for feature selection in the model. To evaluate the model's performance, accuracy and F1-score metrics are utilized. Accuracy can be calculated as follows:

$$Accuracy = \frac{True\ Negatives + True\ Positive}{True\ Positive + False\ Positive + True\ Negative + False\ Negative} \tag{16}$$

For F1-score, it can be calculated as follows:

$$F1 - score = \frac{2 \times True\ Positive}{2 \times True\ Positive + False\ Positive + False\ Negative} \tag{17}$$

E. MGWO-DA Feature Selection Process (Proposed Method).

In the domain of feature selection, two primary principles are often emphasized: enhancing the performance of a model and minimizing the number of features used, ideally fewer than the original number of features. Several studies in the literature, such as [10, 11], have employed these principles. In our paper, we propose a novel principle that places an emphasis on assessing the robustness and efficacy of the selected features. After this selection phase, we conduct a further evaluation of our model's performance using various metrics like accuracy, AUC, and the F1 score. For increased efficiency during evaluation, we utilize a subset of the data, say 90%. This approach substantially trims down computational time. To ensure consistent and unbiased results, we incorporate stratified cross-validation. This ensures both rapid and reliable performance estimations. Furthermode, we use the following formula [11] as the objective function of our proposed algorithm for feature selection:

$$fitness = \beta \times \frac{d}{D} + \alpha \times error\ rate \tag{18}$$

with

$$errorrate = 1 - correct\ rate, \tag{19}$$

where d as number selected feature, D as total features, α as control for performance and number of selected features, and β defined as $1 - \alpha$ [11].

In our methodology, accuracy assessment is anchored on the outcomes of cross-validation. In the evaluation phase of the features we've selected, we apply stratified cross-validation to a subset of the entire dataset. This approach not only ensures a balanced representation of each class but also promotes computational efficiency (Fig. 3).

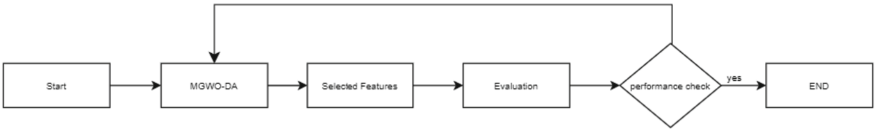


Fig. 3. Feature selection process.

3 Result and Discussion

In this experiment, we made use of the Sonar dataset, previously referenced in other research as experimental data [9]. The dataset contains 208 rows, 60 columns, and has a binary target class distribution: class 1 at 53% and class 0 at 47%. During the modeling phase, we adhered to standard procedures, partitioning the data into training and testing subsets. We assessed the model's effectiveness using the accuracy, and F1-score metrics and also analyzed its susceptibility to overfitting (Table 1).

To guarantee consistent model performance, cross-validation was employed. Although the Support Vector Machine (SVM) was our primary choice for this experiment, our selection was not exclusively tied to SVM. The main aim was to evaluate our novel feature selection algorithm across various methodologies utilizing the same

Table 1. Experimental Result

Evaluation metric	Metric	MGWO-DA	GWO	MGWO	PSO	WOA	GA
Accuracy	AVG	**0.8339**	0.7704	0.8148	0.7605	0.5929	0.7915
	STD	**0.0282**	0.0430	0.0050	0.0514	0.0775	0.0391
	BEST	**0.9038**	0.8309	0.8123	0.8451	0.7887	0.8450
F1 Score	AVG	**0.8326**	0.7915	0.8109	0.7854	0.7121	0.8082
	STD	**0.0287**	0.0402	0.0107	0.0464	0.0495	0.0448
	BEST	**0.9031**	0.8461	0.8420	0.8607	0.8235	0.8607

Table 2. Index of Features by MGWO-DA

Metrics	Accuracy	F1-score	Index of Selected Features
AVG	**0.8339**	**0.8326**	[0, 2, 5, 8, 27, 33, 35, 39, 40, 45, 48, 53, 54, 59]
STD	**0.0282**	**0.0287**	
BEST	**0.9038**	**0.9031**	

model. Moreover, our algorithm is compatible with other machine learning models, like the k-Nearest Neighbors (KNN) and Naive Bayes (Table 2).

Experimental results demonstrate consistent performance, indicating that this hybrid method for feature selection holds significant promise for application in modeling and other optimization problems. Its value lies not only in enhancing accuracy but also in selecting features that significantly influence predictions.

For comparative purposes, we utilized the same dataset and model (Naive Bayes) but incorporated different metaheuristics, as discussed in [12]. In that reference, the proposed metaheuristic method, Leopard Seal Optimization (LSO), achieved a peak accuracy of 97.62%. However, using the method we introduced, namely GWO-DA, the accuracy, AUC, and F1-score reached 100% (with a minimum of 100 iterations). This result is particularly significant given the unbalanced class distribution in the dataset, where accuracy alone might not sufficiently capture model performance. Our findings suggest that the GWO-DA hybridization technique offers compelling advantages in terms of accuracy enhancement.

Regarding the algorithm's complexity, it's worth noting that the required iterations (k) combined with the number of agents (in this case, wolves) (n) lead to a big-O notation of O(kn) for the method we've developed.

4 Conclusion

Based on the study's findings, the hybridization of MGWO-DA demonstrates competitive outcomes. Notably, configuring particle count and iteration numbers during training significantly impacts model performance. Hence, parameter selection warrants careful consideration for optimal results. Future research exploring binary optimization or BGWO-DA principles on high-dimensional datasets holds promise, enriching our understanding of feature selection and metaheuristic methods.

Acknowledgments. The author would like to thank the College of Computing and Khon Kaen University through the KKU Scholarship for ASEAN and GMS Countries' Personal Academic Year 2023 program.

References

1. Yang, Q., Zhang, C., Zhang, S.: Data preparation for data mining. Appl. Artif. Intell. **17**(5–6), 375–381 (2003)
2. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 3rd edn. Morgan Kaufmann Publishers, USA (2000)
3. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
4. Preeti, Deep, K.: A modified Lévy flight grey wolf optimizer feature selection approach to breast cancer dataset. In: Thakur, M., Agnihotri, S., Rajpurohit, B.S., Pant, M., Deep, K., Nagar, A.K. (eds.) Soft Computing for Problem Solving. LNNS, vol. 547. Springer, Singapore (2023). https://doi.org/10.1007/978-981-19-6525-8_31
5. Muro, C., Escobedo, R., Spector, L., Coppinger, R.P.: Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations. Behav. Proc. **88**(3), 192–197 (2011)
6. Mirjalili, S.: Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput. Appl. **27**(4), 1053–1073 (2015). https://doi.org/10.1007/s00521-015-1920-1
7. Cristianini, N., Ricci, E.: Support vector machines. In: Kao, MY. (eds.) Encyclopedia of Algorithms. Springer, Boston, MA (2008)
8. Al-Tashi, Q., Abdul Kadir, S., Rais, H., Mirjalili, S., Alhussian, H.: Binary optimization using hybrid grey wolf optimization for feature selection. IEEE Access **7**, 39496–39508 (2019)
9. Altay, O., Varol Altay, E.: A novel hybrid multilayer perceptron neural network with improved grey wolf optimizer. Neural Comput. Applic. **35**, 529–556 (2023)
10. Al-Tashi, Q., Rais, H., Jadid, S.: Feature selection method based on grey wolf optimization for coronary artery disease classification. In: Saeed, F., Gazem, N., Mohammed, F., Busalim, A. (eds.) Recent Trends in Data Science and Soft Computing. IRICT 2018. AISC, vol. 843, pp. 257–266. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-99007-1_25
11. Pan, H., Chen, S., Xiong, H.: A high-dimensional feature selection method based on modified Gray Wolf Optimization. Appl. Soft Comput. **135**, 110031 (2023)
12. H. Rabie, A., Nehal, A.M.I., Saleh, M.: Leopard seal optimization (LSO): a natural inspired meta-heuristic algorithm. Commun. Nonlinear Sci. Num. Simul. **125**(1), 107338 (2023)

A Modified Hybrid RBF-BP Network Classifier for Nonlinear Estimation/Classification and Its Applications

Po-Chai Wong(✉) and Jeff Chak-Fu Wong

Department of Mathematics, Chinese University of Hong Kong, Shatin, Hong Kong, China
`{pcwong,jwong}@math.cuhk.edu.hk`
`https://www.math.cuhk.edu.hk/∼jwong/`

Abstract. In this work, a modified hybrid radial basis function-back-propagation (RBF-BP) supervised neural network classifier based on the works of Wen et al. [11,12] is proposed. The modified hybrid RBF-BP network is formulated as an adaptive incremental learning algorithm for a single-layer RBF hidden neuron layer. The algorithm uses a density clustering approach to determine the number of RBF hidden neurons and it maintains the self-learning process of updating the neural network's weights using back-propagation. For the last step of the BP neural network in the modified hybrid classifier, the centers and the width parameters of the basis functions are iteratively updated by the stochastic gradient descent algorithm. As a comparative study, some artificial and real-life datasets, for example, Double Moon, Concentric Circle, No Structure and UCI datasets, are used to test the effectiveness of our homemade implementation strategies. The experimental results showed that the implemented algorithm has significant accuracy improvement and reliability.

Keywords: Radial basis function · Back-propagation · Adaptive hybrid algorithm · Classification

1 Introduction

A combination of two neural network models, the radial basis function network (RBFN) and the back-propagation network (BPN) (e.g., [2,5,7]), are most widely used in nonlinear estimation/classification. RBFN is a local approximation network and the main advantage of the RBFN is that it has only one hidden layer that uses RBF as the activation function. In addition, RBFN maps nonlinearly separable problems in low-dimensional spaces to high-dimensional spaces

---

Supported by Department of Mathematics at CUHK and the Teaching Development and Language Enhancement Grant 2022-25 at CUHK.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 43–56, 2023.
https://doi.org/10.1007/978-981-99-7969-1_4

through RBFs, making them linearly separable in high-dimensional spaces. Its disadvantage is that the classification is slow in comparison to BPN since every node in the hidden layer must compute the RBF function for the input during the classification. Another problem is the number of RBF units. Too many RBF units may result in over-fitting while too few may lead to under-fitting. This number has been manually selected on a trial-and-error basis. Some attempts have been made to decide this number adaptively [11,12]. BPN is a global approximation neural network. During the training process, the error is propagated backward layer by layer to the input layer, and the ownership value and threshold appearing in the network are corrected. For each training sample, there is only a small number of weights and thresholds to be updated. Other hybrid classifiers, for example, the radial basis function-extreme learning machine classifier for a mixed data type and medical prediction, [6,10,13], perform better than the BP classifier. Further extensions of the hybrid RBF-BP network classifier (the Hybrid classifier for short) using pre-RBF kernels were found in [14,15].

In this paper, a modified version of the hybrid classifier (the mHybrid classifier for short) is proposed based on the works of Wen et al. [11,12]. It is formulated as an adaptive incremental learning algorithm for a single-layer RBF hidden neuron layer by fixing/shifting the center, adjusting the width parameter of the basis functions and updating the number of RBF hidden neurons, and it maintains a self-learning process of tuning a single-layer BP neural network's weights to improve classification accuracy. In the mHybrid classifier, the center and the width parameter of the RBFs are iteratively updated by the stochastic gradient descent (SGD) algorithm.

The rest part of the paper is outlined as follows. Section 2 presents the architecture of the mHybrid network and the centers, widths and number of RBF hidden neurons interacting with the multi-layer perceptron (MLP) hidden neurons. In Algorithm 1, with optimally determined centers and width parameters, the coverage effect of each hidden neuron can be guaranteed. In Sect. 3, by passing the output of the RBF hidden neurons into an MLP neural network, backpropagation (BP) is used to update the weights of the MLP. Bridging between RBF-BP networks is shown in Algorithm 2 and their implementations are highlighted. In Algorithm 3, two SGD iterative steps are proposed for updating the centers and the width parameters of the RBF units. Section 4 examines the numerical performance of the mHybrid classifier on artificial datasets, e.g., Double Moon, Concentric Circle (e.g., [4]). No Structure [9] and real-life UCI datasets [3]. Section 5 summarizes our findings and concludes the paper.

2 Structure of the Incremental Learning Algorithm

This section describes the three sequential steps used to first find the centers for the RBFs, then find the widths for the RBFs, and to determine a new center as needed.

2.1 Finding the Centers for the Radial Basis Functions

Our aim here is to introduce an efficient technique of density clustering for balanced data. For the RBF networks, the data $\{\boldsymbol{x}\}$ of each class y is covered by circles of different sizes. To decide the optimal number of circles, a pre-selected discriminant function is designed. Then the locations and the widths of the circles are determined from the repulsive force exerted by nearby heterogeneous members, so that each circle contains many homogeneous members and few heterogeneous members. Formally, we define a set of discriminant functions ρ_i, one for each class i [1],

$$\rho_i(\boldsymbol{x}) > \rho_j(\boldsymbol{x}) \text{ for any } j \neq i \implies \boldsymbol{x} \text{ belongs to class } i. \tag{1}$$

The higher the value of ρ_i, the more likely that $\boldsymbol{x}$ belongs to class i. Another natural assumption is that the more members of the same class there are around $\boldsymbol{x}$, the more likely $\boldsymbol{x}$ belongs to the same class. To obtain the discriminant function, one therefore can define a potential function γ as

$$\gamma(\boldsymbol{x}, \boldsymbol{z}) = \frac{1}{1 + T\|\boldsymbol{x} - \boldsymbol{z}\|^2}, \tag{2}$$

where $T > 0$ is a distance weighting factor and $\|\cdot\|$ is the Euclidean norm. The potential γ is a particular example of the general inverse multi-quadratic function, $(1 + \epsilon^2\|\boldsymbol{x} - \boldsymbol{z}\|^2)^{-p/2}$ when $T = \epsilon^2$ and $p = 2$. The potential γ is proportional to the closeness between two points $\boldsymbol{x}, \boldsymbol{y}$.

Given a data set S that consists of N training samples $\{(\boldsymbol{x}_k, y_k)\}_{k=1}^N$, where $\boldsymbol{x}_k \in \mathbb{R}^n$ and $y_k \in \mathbb{R}^H$, where n is the dimension of $\boldsymbol{x}_k$ and H is the dimension of y_k. Let $S^i = \{\boldsymbol{x}_k^i : k = 1, \cdots, N_i\}$ be the set of training samples in the ith pattern class, N_i the number of samples in class i and $S^i \cap S^j = \emptyset$ for $i \neq j$. For $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{x}_k^i \in S^i$, a discriminant function ρ_i can be constructed by the superposition of such potential functions $\gamma(\cdot, \boldsymbol{x}_k^i)$:

$$\rho_i(\boldsymbol{x}) = \sum_{k=1}^{N_i} \gamma(\boldsymbol{x}, \boldsymbol{x}_k^i) = \sum_{k=1}^{N_i} \frac{1}{1 + T\|\boldsymbol{x} - \boldsymbol{x}_k^i\|^2}. \tag{3}$$

As shown in Fig. 1, $T > 0$ controls the width of the region of influence and the sharpness of the distribution, e.g., the larger the factor T, the sharper ρ_i and the distribution of the samples will have a better shape.

It is worth mentioning that the ρ_i defined here is different from that in Wen et al. [11,12], where when evaluated at sample point $\boldsymbol{x}_l^i$, the summation does not consider that point (i.e., $\widetilde{\rho}_i(\boldsymbol{x}_l^i) = \sum_{k=1,k\neq l}^{N_i} \gamma(\boldsymbol{x}_l^i, \boldsymbol{x}_k^i)$). Then we have $\rho_i(x) = \widetilde{\rho}_i(x) + 1$ for $x^i \in S^i$. This difference is not critical in the end (except for the choice of δ) since we will only compare the values within the same class. Here are a few properties of γ [8]:

1. $\gamma(\boldsymbol{x}, \boldsymbol{z})$ attains maximum at $\boldsymbol{x} = \boldsymbol{z}$.
2. $\gamma(\boldsymbol{x}, \boldsymbol{z})$ tends to 0 as $\|\boldsymbol{x} - \boldsymbol{z}\|$ increases.

3. $\gamma(\boldsymbol{x}, \boldsymbol{z})$ is smooth and decreases monotonically as $\|\boldsymbol{x} - \boldsymbol{z}\|$ increases.
4. $\gamma(\boldsymbol{x}_1, \boldsymbol{z}) = \gamma(\boldsymbol{x}_2, \boldsymbol{z})$ if $\|\boldsymbol{x}_1 - \boldsymbol{z}\| = \|\boldsymbol{x}_2 - \boldsymbol{z}\|$.

These properties allow ρ_i to capture the local influence of S^i at $\boldsymbol{x}$ and to correlate to the likeliness for the input $\boldsymbol{x}$ to belong to S^i. Finally, all that remains to be checked is whether this ρ_i can classify correctly, as stated in the theorem below.

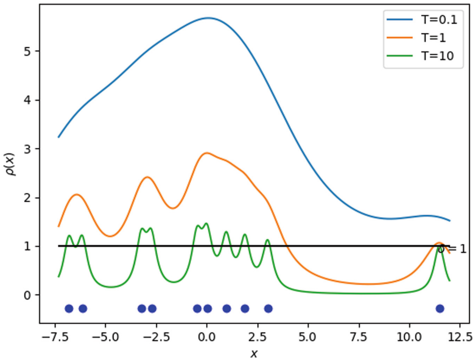


Fig. 1. Illustration of the effect of T in a 1D example using 10 data samples.

Theorem 1. Suppose $\boldsymbol{x}_l^i \in S^i := \{\boldsymbol{x}_1^i, \cdots, \boldsymbol{x}_{N_i}^i\}$. Let N be the number of classes. Let $D = \min\{\|\boldsymbol{x}^i - \boldsymbol{x}^j\|^2 : \boldsymbol{x}^i \in S^i, \boldsymbol{x}^j \in S^j \text{ for any } i \neq j,\ i, j \leq N\} > 0$. Then, there exists a continuously differentiable discriminant function ρ_i such that

$$\rho_i(\boldsymbol{x}_l^i) > \rho_j(\boldsymbol{x}_l^i) \text{ for any } j \neq i.$$

Proof. Suppose $S^j = \{\boldsymbol{x}_1^j, \cdots, \boldsymbol{x}_{N_j}^j\}$. Let $T > \frac{1}{D}(N_j - 1)$. Define ρ_i as stated before using this T. Then,

$$\begin{aligned}
\rho_j(\boldsymbol{x}_l^i) &= \sum_{k=1}^{N_j} \frac{1}{1 + T\|\boldsymbol{x}_l^i - \boldsymbol{x}_k^j\|^2} \leq N_j \max_{k=1,\cdots,N_j} \frac{1}{1 + T\|\boldsymbol{x}_l^i - \boldsymbol{x}_k^j\|^2} \\
&\leq N_j \frac{1}{1 + T \min\limits_{k=1,\cdots,N_j} \|\boldsymbol{x}_l^i - \boldsymbol{x}_k^j\|^2} \leq \frac{N_j}{1 + TD} < 1 \\
&= \frac{1}{1 + T\|\boldsymbol{x}_l^i - \boldsymbol{x}_l^i\|^2} \leq \sum_{k=1}^{N_i} \frac{1}{1 + T\|\boldsymbol{x}_l^i - \boldsymbol{x}_k^i\|^2} = \rho_i(\boldsymbol{x}_l^i).
\end{aligned}$$

$\square$

The value T can be selected as $T > \frac{1}{D}(N_j - 1)$ for any j such that the theorem holds.

Now, using the discriminant function ρ_i, we can select an initial center by taking the maximum of the discriminant function over class i since the samples are concentrated/localized around it. Define the initial center

$$\boldsymbol{\mu}_k := \arg\max\{\rho_i(\boldsymbol{x}_l^i) : \ \boldsymbol{x}_l^i \in S^i\}. \tag{4}$$

We now circle it with a given predetermined radius σ with a predefined threshold $\sigma_{\min} < \sigma$. Our next goal is to move the center $\boldsymbol{\mu}_k$ and resize the circle so as to reduce the number of heterogeneous members around $\boldsymbol{\mu}_k$.

A repulsive force $\boldsymbol{F}$ from each heterogeneous member $\boldsymbol{x}^j$ in the ball $\mathcal{B}(\boldsymbol{\mu}_k, \lambda\sigma) := \{\boldsymbol{x} : \ \|\boldsymbol{x} - \boldsymbol{\mu}_k\| \leq \lambda\sigma\}$, where $\lambda > 1$ is a width covering factor, is given by

$$\boldsymbol{F} = \exp\left(-\alpha\|\boldsymbol{\mu}_k - \boldsymbol{x}^j\|\right) \frac{\boldsymbol{x}^j - \boldsymbol{\mu}_k}{\|\boldsymbol{x}^j - \boldsymbol{\mu}_k\|}, \tag{5}$$

where α is the repulsive force control factor. The exponential scale factor term is used to control the variation of the center drift due to any heterogeneous members within the ball.

The center position is updated to move away from heterogeneous members as follows:

$$\boldsymbol{\mu}_k \leftarrow \boldsymbol{\mu}_k + \boldsymbol{F}. \tag{6}$$

The preassigned value $\alpha > 0$ should not be too small otherwise it loses the purpose of maximizing the discriminant function ρ_i. But a very small α may not reduce the number of heterogeneous members in the ball. A big center shift may also result in covering new heterogeneous members. Therefore, this choice of α is highly dependent on the dataset. In some cases, center drifts are not sufficient to reduce the number of heterogeneous members. We, therefore, fix the maximum number of iterations Epo allowed for testing for center drifts.

We count the number $M_{\neq i}$ of heterogeneous samples $\boldsymbol{x} \in S \backslash S^i$ in the current ball $\mathcal{B}(\boldsymbol{\mu}_k, \lambda\sigma)$ by $\|\boldsymbol{x} - \boldsymbol{\mu}\| \leq \lambda\sigma$. Define M_i as the number of samples in S^i in $\mathcal{B}(\boldsymbol{\mu}_k, \lambda\sigma)$. The center for each ball can be adjusted by the sum of the resultant forces:

$$\boldsymbol{\mu}_k \leftarrow \boldsymbol{\mu}_k + \frac{1}{M} \sum_{p=1}^{M_{\neq i}} \exp\left(-\alpha\|\boldsymbol{\mu}_k - \boldsymbol{x}_p\|\right) \frac{\boldsymbol{x}_p - \boldsymbol{\mu}_k}{\|\boldsymbol{x}_p - \boldsymbol{\mu}_k\|}, \tag{7}$$

where M is the preassigned value to average all the resultant forces.

After an iteration, check the number of heterogeneous members in the ball again. If the new number $M'_{\neq i}$ of heterogeneous members in the ball is reduced, the iterative process continues, otherwise the center is fixed. Once the center is positioned, the algorithm readjusts the size of the ball to mostly cover only homogeneous members. An RBF center is then selected.

2.2 Finding the Widths for the Radial Basis Functions

If heterogeneous members exist in the ball (i.e., $M'_{\neq i} > 0$), we shrink the ball so that it almost covers the closest heterogeneous member $\boldsymbol{x}^j$ from $\boldsymbol{\mu}_k$ by the formula below. To avoid over-shrinking, $\beta > 1$ acts as a relaxation factor and $\sigma_{\min}$ as the lower bound of the size.

$$\sigma_k \leftarrow \begin{cases} \max\{\min\{\|\boldsymbol{\mu}_k - \boldsymbol{x}\|/\beta : \ \boldsymbol{x} \in S\backslash S^i\}, \sigma_{\min}\} & \text{if } M'_{\neq i} > 0 \\ \sigma & \text{if } M'_{\neq i} = 0 \end{cases}. \tag{8}$$

This β should be slightly smaller than λ to avoid setting too large a value on λ. Suppose λ/β is less than but close to 1. Then for any $\boldsymbol{x}^{\neq i} \in (S\backslash S^i) \cap \mathcal{B}(\boldsymbol{\mu}_k, \lambda\sigma)$,

$$\sigma_{\min} \le \sigma_k \le \min\{\|\boldsymbol{\mu}_k - \boldsymbol{x}\|/\beta : \ \boldsymbol{x} \in S\backslash S^i\} \le \|\boldsymbol{\mu}_k - \boldsymbol{x}^{\neq i}\|/\beta \le (\lambda/\beta)\sigma \le \sigma. \tag{9}$$

Since the last inequality is "slightly less than", if $M'_{\neq i} > 0$, the updated circle is not too large when controlled by β.

2.3 Updating the Discriminant Function

To decide if a new center is needed, the discriminant function ρ_i is updated to remove the influence of the centers found so far. The processes of shifting the center and resizing the RBF balls repeat if some updated potential is above the threshold δ, i.e.,

$$\max\{\rho_i^{\text{new}}(\boldsymbol{x}_1^i), \cdots, \rho_i^{\text{new}}(\boldsymbol{x}_{N_i})\} > \delta, \tag{10}$$

where

$$\rho_i^{\text{new}}(\boldsymbol{x}) = \rho_i(\boldsymbol{x}) - \rho_i(\boldsymbol{\mu}_k) \exp\left(-\frac{1}{2\sigma_l^2}\|\boldsymbol{x} - \boldsymbol{\mu}_k\|\right). \tag{11}$$

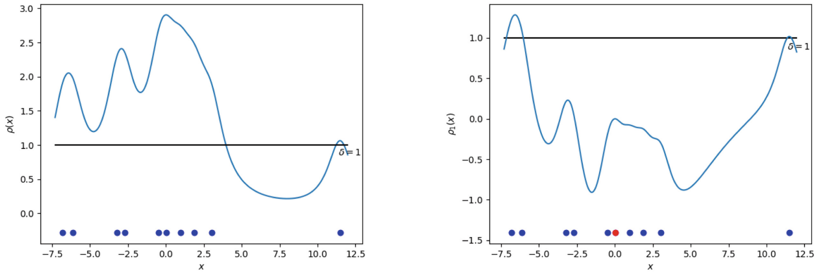
If all of the new ρ_i^{new} are less than δ, the remaining data are not dense enough to form an effective cluster center. It is important to note that since all the terms in the discriminant function ρ_i are positive, this δ is dependent on the size N_i and the scaling parameter T, as shown in Fig. 1, and thus can only be determined on an empirical basis. For the overall effect of δ, see Figs. (2a)–(2d). An alternative method is to normalize a new discriminant function $\widetilde{\rho}_i := \frac{1}{N_i}\sum_k \gamma(\cdot, \boldsymbol{x}_k^i)$. The T in the proof of Theorem 1 is then dependent on N_i. Future studies can then be done on the suggested selection of δ for every class i.

The whole center selection process is summarised in Algorithm 1.

3 Moving from the RBFN to the BPN

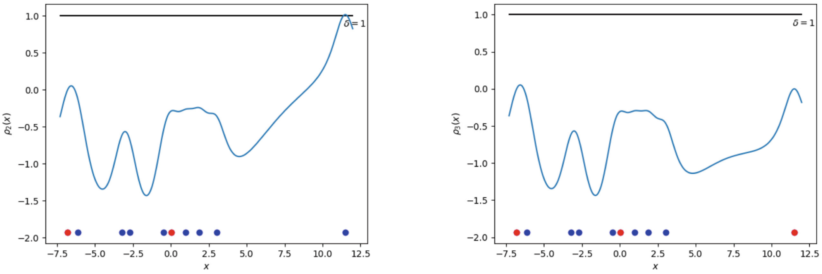
The algorithm follows the original purpose of imposing the RBF layer, which captures the relation of the classes with the points of high relative density. However, it is natural to question the relation of such a doctrine with the ultimate purpose of reducing classification error.

In MLP, a gradient-search algorithm is applied to iteratively update the weights using backpropagation in order to attain a local minimum of the error function. This simple idea comes from the fact that the infimum of a normed space is not bigger than the infimum of a subspace. The motivation for our extension comes from questioning whether extending the parameter space in the gradient-search algorithm would improve the classification. By combining the hybrid structure, we apply backpropagation to both MLP and RBF layers with the initial guess of RBF centers selected by the incremental algorithm above.



(a) Step 1

(b) Step 2



(c) Step 3

(d) Step 4

Fig. 2. This shows an example of 10 homogeneous data points (blue). The black horizontal line marks the threshold $\delta = 1$. The blue curve is the value of $\rho(x)$ with $T = 1$. We pick the data point with the highest value of ρ in Figure (2a) and mark it as our first cluster point $\boldsymbol{\mu}_1$ (red). (2b) The function ρ_1 is updated, penalizing the region around $\boldsymbol{\mu}_1$. Since some points are above the threshold, the selection continues. (2c) $\boldsymbol{\mu}_2$ is found and marked in red. Note that this showcases the importance of δ since the data point almost does not pass the test. (2d) $\boldsymbol{\mu}_3$ is found. The graph lies below the threshold and the selection process terminates. (Color figure online)

The main concern regarding this approach is the problem of hitting local minima. Here, we mainly focus on the extent to which the classifier gets stuck at the local minima after the extension of the parameter space. Therefore, we use the common stochastic gradient descent method.

Let K be the number of RBF centers. Define the output of RBF at each center $\boldsymbol{\mu}_k$ by

$$\phi_k(\boldsymbol{x}; \boldsymbol{\mu}_k, \sigma_k) = \exp\left(-\frac{1}{2\sigma_k^2}\|\boldsymbol{x} - \boldsymbol{\mu}_k\|^2\right) \text{ for } k = 1, 2, \cdots, K. \tag{12}$$

Algorithm 1: Incremental Learning Algorithm for Constructing RBF Hidden Neurons

```
Data: $S = \cup_{i=1}^{H} S^i$, where $S^i = \{\boldsymbol{x}_1^i, \cdots, \boldsymbol{x}_{N_i}^i\}$ the set of training samples
      labeled $i$ and $H$ is the total number of classes
Result: $\{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K\}$, $\{\sigma_1, \cdots, \sigma_K\}$
let
    $\sigma > \sigma_{\min} > 0, k \leftarrow 0, T, M, \alpha, \beta, \lambda, \mathsf{Epo} > 0$
for $i = 1$ to $H$ do
    Define the discriminant function $\rho_i$ for class $i$ by $\rho_i(\boldsymbol{x}) := \sum_{k=1}^{N_i} \gamma(\boldsymbol{x}, \boldsymbol{x}_k^i)$
    repeat
        $\boldsymbol{\mu}_k \leftarrow \arg\max\{\rho_i(\boldsymbol{x}_k^i) : \ \boldsymbol{x}_k^i \in S^i\}$
        Define $M_i$ and $M_{\neq i}$
        for each sample $\boldsymbol{x}_p \in S \backslash S^i$ in $\mathcal{B}(\boldsymbol{\mu}_k, \lambda\sigma)$ do
            $\boldsymbol{\mu}_k \leftarrow \boldsymbol{\mu}_k + \exp\left(-\alpha\|\boldsymbol{\mu}_k - \boldsymbol{x}_p\|\right) \frac{\boldsymbol{x}_p - \boldsymbol{\mu}_k}{\|\boldsymbol{x}_p - \boldsymbol{\mu}_k\|}$
        end
        Define the updated $M_i'$ and $M_{\neq i}'$
        $m \leftarrow 0$
        while $M_{\neq i}' > 0$ and $m \leq \mathsf{Epo}$ do
            if $M_i' \geq M_i$ and $M_{\neq i}' \leq M_{\neq i}$ then
                Update with $M_i \leftarrow M_i'$; $M_{\neq i} \leftarrow M_{\neq i}'$
                $\mu_k \leftarrow \mu_k + \frac{1}{M} \sum_{p=1}^{M_{\neq i}} \exp\left(-\alpha\|\boldsymbol{\mu}_k - \boldsymbol{x}_p\|\right) \frac{\boldsymbol{x}_p - \boldsymbol{\mu}_k}{\|\boldsymbol{x}_p - \boldsymbol{\mu}_k\|}$
                $m \leftarrow m + 1$
            else
                $\sigma_k \leftarrow \begin{cases} \max\{\min\{\|\boldsymbol{\mu}_k - \boldsymbol{x}\|/\beta : \boldsymbol{x} \in S \backslash S^i\}, \sigma_{\min}\} & \text{if } M_{\neq i}' > 0 \\ \sigma_{\min} & \text{if } M_{\neq i}' = 0 \end{cases}$
            end
        end
        Define $\rho_i^{\text{new}}(\boldsymbol{x}) := \rho_i(\boldsymbol{x}) - \rho_i(\boldsymbol{\mu}_k) \exp\left(-\frac{1}{2\sigma_k^2}\|\boldsymbol{x} - \boldsymbol{\mu}_k\|\right)$
        $\rho_i \leftarrow \rho_i^{\text{new}}$
        $k \leftarrow k + 1$
    until
        $\max\left\{\rho_i(\boldsymbol{x}_1^i), \cdots, \rho_i(\boldsymbol{x}_{N_i}^i)\right\} \leq \delta$
end
```

The output of $\Phi(\boldsymbol{x}) := [\phi_1(\boldsymbol{x}; \boldsymbol{\mu}_1, \sigma_1), \cdots, \phi_K(\boldsymbol{x}; \boldsymbol{\mu}_K, \sigma_K)]$ is nonnegative. To facilitate computational speed, we polarise them by a mapping $\boldsymbol{x} \mapsto 2\boldsymbol{x} - 1$, which becomes the input of a feedforward network with the activation function $\boldsymbol{x} \mapsto a \tanh(b\boldsymbol{x})$, where a and b are constants. They are updated using back-propagation, passing the error term in the output to each hidden neuron.

In Algorithm 2, $\mathbf{W} = [W^{(1)}, \cdots, W^{(L)}]$ is a tensor with each $W^{(l)}$ being the weight matrix between the $(l-1)$- and l-layers, where L is the number of BP hidden layers (including the output layer). $\boldsymbol{\delta}^{(l)} := [\delta_1^{(l)}, \cdots, \delta_{n_l}^{(l)}]$ is a vector recording the error passed to the n_l hidden neurons of the l-th layer, $l = 1, \cdots, L$. $\boldsymbol{\delta}^{(0)}$ is the error vector for the RBF units. The error term of the first BP layer $\boldsymbol{\delta}^{(1)}$ can further be passed to the RBF layer, since $\boldsymbol{x} \mapsto \Phi(\boldsymbol{x})$ is smooth with

Algorithm 2: Hybrid RBF-BP Network Architecture

```
Data: $S = \cup_{i=1}^{H} S^i$, where $S^i = \{\boldsymbol{x}_1^i, \cdots, \boldsymbol{x}_{N_i}^i\}$, and Initialized weights $\mathbf{W}$
Result: Updated weights $\mathbf{W}$
Define constants $a, b$, learning rate $\eta$
Apply Algorithm 1 to obtain centers $\{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K\}$ and widths $\{\sigma_1, \cdots, \sigma_K\}$
repeat
    $\widetilde{\boldsymbol{y}}^{(\mathbf{0})} \leftarrow 2\Phi(\boldsymbol{x})^T - 1$
    for $l = 1$ to $L$ do
        $\boldsymbol{v}^{(l)} \leftarrow W^{(l)} \widetilde{\boldsymbol{y}}^{(l-1)}$
        $\widetilde{\boldsymbol{y}}^{(l)} \leftarrow a \tanh(b\boldsymbol{v}^{(l)})$
    end
    $\boldsymbol{E} \leftarrow$ the error term $(\boldsymbol{y} - \widetilde{\boldsymbol{y}}^{(L)})$
    for $l = L$ to $l = 1$ do
        if $l = L$ then
            $\boldsymbol{\delta}^{(L)} \leftarrow ab\boldsymbol{E} \odot \operatorname{sech}^2(b\boldsymbol{v}^{(L)})$        $\odot$ is elementwise multiplication
        else
            $\boldsymbol{\delta}^{(l)} \leftarrow ab \operatorname{sech}^2(b\boldsymbol{v}^{(l)}) \odot \left(W^{(l+1)}\boldsymbol{\delta}^{(l+1)}\right)$
        end
        $W^{(l)} \leftarrow W^{(l)} + \eta\boldsymbol{\delta}^{(l)}(\widetilde{\boldsymbol{y}}^{(l-1)})^T$
    end
until
    $e <$ Tolerance
```

respect to $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ (for strictly positive $\boldsymbol{\sigma}$). Suppose $e = \sqrt{\sum_{j=1}^{H}(y_j - \widetilde{y}_j^{(L)})^2}$ is the error of our predicted output $\widetilde{\boldsymbol{y}}^{(L)}$ of the L-th layer. By the Chain rule, we obtain

$$\begin{cases} \dfrac{\partial e}{\partial \boldsymbol{\mu}_k} = \dfrac{\partial e}{\partial \widetilde{\boldsymbol{y}}_k^{(0)}} \dfrac{\partial \widetilde{\boldsymbol{y}}_k^{(0)}}{\partial \boldsymbol{\mu}_k} = \delta_k^{(0)} \dfrac{\partial \widetilde{\boldsymbol{y}}_k^{(0)}}{\partial \boldsymbol{\mu}_k} = \delta_k^{(0)} \dfrac{\partial}{\partial \boldsymbol{\mu}_k} \phi_k(\boldsymbol{x}; \boldsymbol{\mu}_k, \sigma_k) \\ \qquad = \delta_k^{(0)} \exp\left(\dfrac{-\|\boldsymbol{x} - \boldsymbol{\mu}_k\|^2}{2\sigma_k^2}\right) \dfrac{1}{\sigma_k^2}(\boldsymbol{x} - \boldsymbol{\mu}_k) \\ \dfrac{\partial e}{\partial \sigma_k} = \dfrac{\partial e}{\partial \widetilde{\boldsymbol{y}}_k^{(0)}} \dfrac{\partial \widetilde{\boldsymbol{y}}_k^{(0)}}{\partial \sigma_k} = \delta_k^{(0)} \dfrac{\partial \widetilde{\boldsymbol{y}}_k^{(0)}}{\partial \sigma_k} = \delta_k^{(0)} \dfrac{\partial}{\partial \sigma_k} \phi_k(\boldsymbol{x}; \boldsymbol{\mu}_k, \sigma_k) \\ \qquad = \delta_k^{(0)} \exp\left(\dfrac{-\|\boldsymbol{x} - \boldsymbol{\mu}_k\|^2}{2\sigma_k^2}\right) \dfrac{1}{\sigma_k^3}\|\boldsymbol{x} - \boldsymbol{\mu}_k\|^2 \end{cases}. \tag{13}$$

Then, the update rules of $\boldsymbol{\mu}_k$ and σ_k follow from SGD with learning rate η:

$$\boldsymbol{\mu}_k \leftarrow \boldsymbol{\mu}_k + 2\eta\delta_k^{(0)} \frac{\partial \widetilde{\boldsymbol{y}}^{(0)}}{\partial \boldsymbol{\mu}_k}(\boldsymbol{x}) \quad \text{and} \quad \sigma_k \leftarrow \sigma_k + 2\eta\delta_k^{(0)} \frac{\partial \widetilde{\boldsymbol{y}}^{(0)}}{\partial \sigma_k}(\boldsymbol{x}). \tag{14}$$

Let $\frac{\partial \widetilde{\boldsymbol{y}}^{(\mathbf{0})}}{\partial \boldsymbol{\mu}} = \left[\frac{\partial \widetilde{\boldsymbol{y}}^{(0)}}{\partial \boldsymbol{\mu}_1}, \cdots, \frac{\partial \widetilde{\boldsymbol{y}}^{(0)}}{\partial \boldsymbol{\mu}_K}\right]$ and $\frac{\partial \widetilde{\boldsymbol{y}}^{(\mathbf{0})}}{\partial \boldsymbol{\sigma}} = \left[\frac{\partial \widetilde{\boldsymbol{y}}^{(0)}}{\partial \sigma_1}, \cdots, \frac{\partial \widetilde{\boldsymbol{y}}^{(0)}}{\partial \sigma_K}\right]$. The investigated algorithm is shown as Algorithm 3, where $diag(\boldsymbol{x})$ is the diagonal matrix with the diagonal element being $\boldsymbol{x}$.

Algorithm 3: Modified Hybrid RBF-BP Network Architecture

```
Data: $S = \cup_{i=1}^{H} S_i$, where $S_i = \{\boldsymbol{x}_1^i, \cdots, \boldsymbol{x}_{N_i}^i\}$ and Initialized weights $\mathbf{W}$
Result: updated weights $\mathbf{W}$, centers $\{\boldsymbol{\mu}_k\}$ and widths $\{\sigma_k\}$
Define constants $a, b$, learning rate $\eta$
Apply Algorithm 1 to obtain centers $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K]$ and widths $\boldsymbol{\sigma} = [\sigma_1, \cdots, \sigma_K]$
repeat
    Lines $4 - 17$ in Algorithm 2
    $\boldsymbol{\delta}^{(0)} \leftarrow W^{(1)} \boldsymbol{\delta}^{(1)}$
    $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + 2\eta \frac{\partial \widetilde{\boldsymbol{y}}^{(\mathbf{0})}}{\partial \boldsymbol{\mu}}(\boldsymbol{x}) diag(\boldsymbol{\delta}^{(0)})$
    $\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma} + 2\eta \frac{\partial \widetilde{\boldsymbol{y}}^{(\mathbf{0})}}{\partial \sigma}(\boldsymbol{x}) diag(\boldsymbol{\delta}^{(0)})$
until
    $e <$ Tolerance
```

4 Numerical Results

As a comparative study, some artificial and real-life datasets were used to test the effectiveness of our implementation strategies and our homemade MATLAB codes.

4.1 Comparison of MLP and mHybrid Classifiers on Different Datasets

We used 8 datasets with classes as shown in Table 1 and all the parameters used for the mHybrid classifier are in Table 2. We set Tolerance $= 10^{-6}$. Figure 3 shows the patterns for Double Moon with 2 classes, Concentric Circles (CC), Concentric Circles with 2 extra layers (CC2) and Double Moon with 6 classes (DM6) (Each crescent is split into three sections). For each dataset, we first obtained the theoretical value $\max\{(N_j - 1)/D\}$ (the last column in Table 2), then we adjusted the T value descendingly to obtain the best result. Table 1 shows the results of the MLP and mHybrid classifiers. To assess their performance, accuracy, precision, recall, and F-score were used. Table 1 shows that the mHybrid classifier outperformed the MLP classifier.

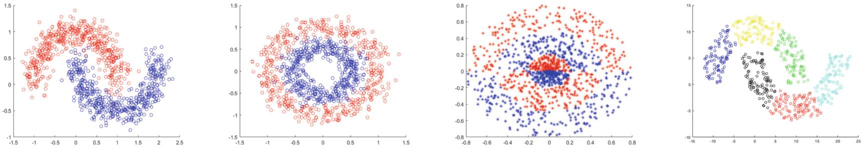
We further examined the pendigit dataset with 10 classes. Different network architectures of MLP and mHybrid classifiers are shown in Fig. 4, where the first label refers to the number of hidden neurons of the two layers for MLP, while the second label refers to the number of RBF centers and hidden neurons used in the mHybrid classifier. According to the results, the accuracy of the mHybrid classifier is significantly better than that of the MLP classifier when the numbers of RBF centers and hidden neurons increase.

Table 1. Performance comparison of MLP and mHybrid classifiers on different datasets.

Dataset	Classes	MLP				mHybrid			
		Accuracy	Precision	Recall	F Score	Accuracy	Precision	Recall	F Score
Pendigit	10	0.9780	0.8938	0.8906	0.8906	0.9819	0.9126	0.9109	0.9089
Letter	26	0.9781	0.7127	0.7121	0.7070	0.9821	0.8000	0.7660	0.7680
Ozone	2	0.9448	0.7101	0.6135	0.6354	0.9540	0.9259	0.6741	0.7371
Sonar	2	0.8428	0.8533	0.8468	0.8424	0.8809	0.8820	0.8813	0.8807
Iono	2	0.9277	0.9099	0.9357	0.9166	0.9277	0.9282	0.9175	0.9214
DM	2	0.9983	0.9955	0.9960	0.9957	0.9933	0.9790	0.9826	0.9803
CC	2	0.6694	0.6700	0.6691	0.6688	0.9008	0.8992	0.8992	0.8992
CC2	2	0.9240	0.9239	0.9243	0.9240	0.9557	0.9556	0.9564	0.9557

Table 2. Parameters of the mHybrid classifier used on different datasets.

Dataset	Parameters										
	a	b	T	α	β	σ_{min}	σ_{init}	λ	Epo	δ	$\max{(N_j - 1)/D}$
Pendigit	1.2	0.8	50	25	1.2	1e−4	0.5	1.3	10	0.001	68.94
Letter	1.2	0.8	100	25	1.2	1e−4	5	1.3	10	0.001	813
Ozone	1.2	0.8	50	25	1.2	1e−4	5	1.3	10	0.01	281.78
Sonar	1.2	0.8	100	25	1.2	1e−4	5	1.3	10	0.001	216.83
Iono	1.2	0.8	100	25	1.2	1e−4	5	1.3	10	0.001	479.98
DM	1.2	0.8	50	25	1.2	1e−4	0.5	1.3	10	0.001	7.15e+04
CC	1.2	0.8	100	25	1.2	1e−4	0.5	1.3	10	0.001	5.86e+04
CC2	1.2	0.8	200	25	1.2	1e−4	0.5	1.3	10	0.001	6.53e+03
DM6	1.2	0.8	500	25	1.2	1e−4	0.7	1.3	10	0.01	151.65
No Structure	1.2	0.8	200	25	1.2	1e−4	See Fig. 6	1.3	10	0.001	9.23e+03



(a)

(b)

(c)

(d)

Fig. 3. (3a) Double Moon with 2 classes. (3b) Concentric Circles (CC). (3c) Concentric Circles with 2 extra layers (CC2). (3d) Double Moon with 6 classes (DM6).

4.2 Comparison of Hybrid and mHybrid Classifiers Using DM6 and No Structure Datasets

In what follows, we numerically analyze the effect of the center and the width parameter of the RBFs, which are iteratively updated by the SGD algorithm, referred to as Algorithm 3. Using the DM6 dataset, we observed that the mHybrid classifier outperformed the MLP and Hybrid classifiers as shown in Table 3.

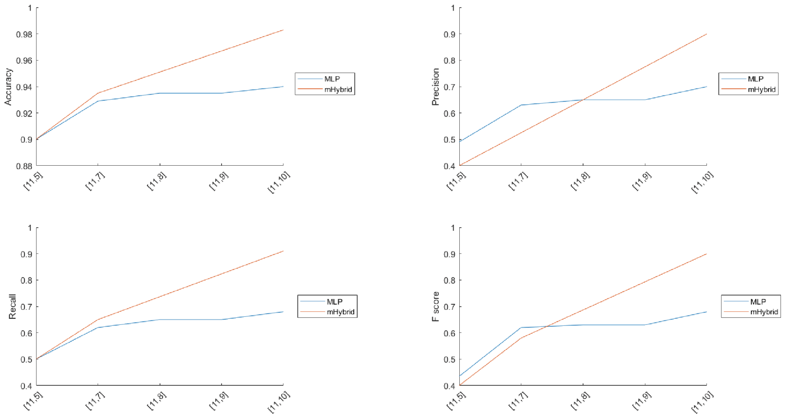


Fig. 4. Comparison of the different network architectures of two classifiers using the pendigit dataset. For MLP, the two numbers are the numbers of hidden neurons of the two hidden layers. For mHybrid, the first number is the number of RBF units and the second is the BP units.

Table 3. Comparison of three classifiers using DM6.

Classifier	Accuracy	Precision	Recall	F Score
Modified Hybrid RBF-BP	0.9933	0.9790	0.9826	0.9803
Hybrid RBF-BP	0.9811	0.9532	0.9624	0.9703
MLP	0.9552	0.8816	0.8849	0.8567

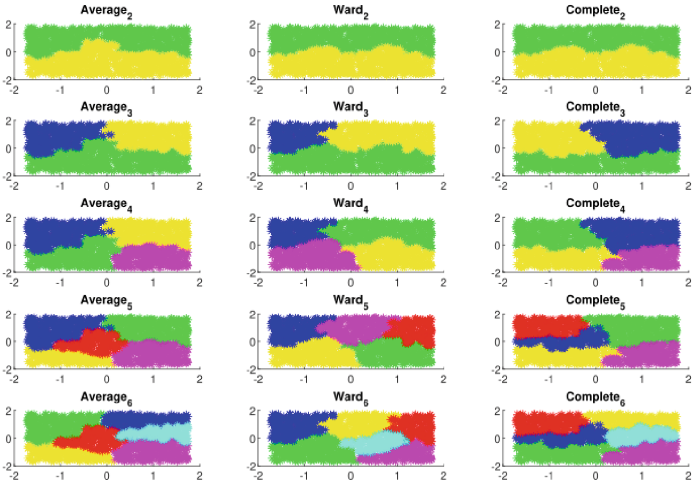


Fig. 5. Distribution of No Structure datasets.

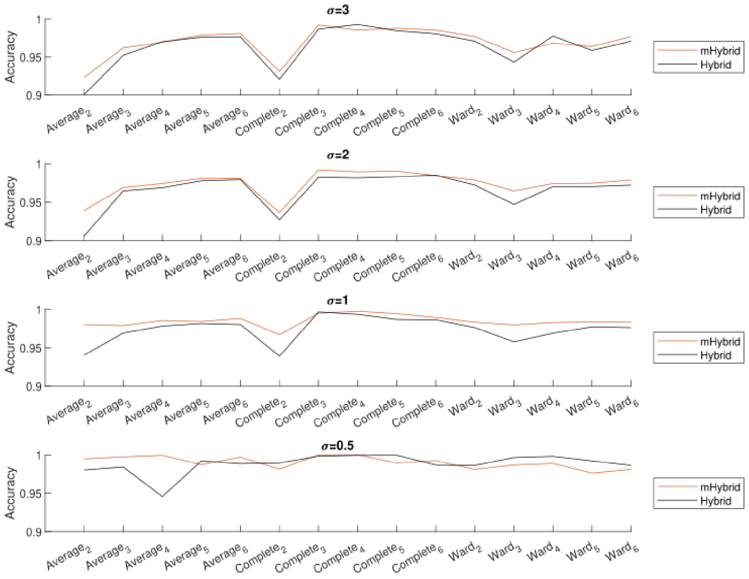


Fig. 6. Comparison of two classifiers on different No Structure datasets.

Figure 5 shows the pattern of No Structure datasets, where data is labeled using agglomerative clustering algorithms with different linkages ('Ward', 'Average', and 'Complete') with different numbers of classes. To be more precise, 'Ward' minimizes the variance of each cluster, 'Average' uses the average of the distances of each sample pair from two clusters, and 'Complete' uses the maximum distances between all samples of the clusters. The datasets are tested on different initial σ values with identical configurations. Figure 6 shows that the mHybrid classifier outperformed the hybrid classifier. In addition to that, the smaller the σ value, the more RBF centers generated, and thus accuracy increased.

5 Conclusions

The performance of the modified hybrid RBF-BP classifier has been tested using artificial and real-life datasets. Based on the numerical results, we concluded that when using the modified classifier, including the lines for updating the last step yielded better accuracy than not including them. Although it performs better than the MLP classifier, higher computational efficiency will be required if more testing is to be done. In the future, tests may be done to fine-tune the parameters to find a set of all optimal parameters for the modified hybrid classifier. Moreover, the efficiency of combining different classifiers with the proposed classifier requires more work.

References

1. Babu, C.C., Kalra, S.N.: On the application of Bashkirov, Braverman, and Muchnik potential function for feature selection in pattern recognition. Proc. IEEE **60**(3), 333–334 (1972)
2. Deng, Y., et al.: New methods based on back propagation (BP) and radial basis function (RBF) artificial neural networks (ANNs) for predicting the occurrence of haloketones in tap water. Sci. Total Environ. **772**, 145534 (2021) Epub 2021 Feb 2. PMID: 33571763. https://doi.org/10.1016/j.scitotenv.2021.145534
3. Dua, D., Graff, C.: UCI Machine Learning Repository (2017). http://archive.ics.uci.edu/ml
4. Haykin, S.: Neural Networks and Learning Machines. 3rd edn. Pearson Education (2009)
5. Hu, P.H., Lu, Z.X., Zhang, Y.Q., Liu, S.L., Dang, X.M.: A new modeling method of angle measurement for intelligent ball joint based on BP-RBF algorithm. Appl. Sci. **9**(14), 2850 (2019). https://doi.org/10.3390/app9142850
6. Li, Q., Xiong, Q., Ji, S., Yu, Y., Wu, C., Yi, H.: A method for mixed data classification base on RBF-ELM network. Neurocomputing **431**, 7–22 (2021)
7. Markopoulos, A.P., Georgiopoulos, S., Manolakos, D.E.: On the use of back propagation and radial basis function neural networks in surface roughness prediction. J. Ind. Eng. Int. **12**(3), 389–400 (2016). https://doi.org/10.1007/s40092-016-0146-x
8. Meisel, W.S.: Potential functions in mathematical pattern recognition. IEEE Trans. Comput. **18**(10), 911–918 (1969)
9. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
10. Siouda, R., Nemissi, M., Seridi, H.: Diverse activation functions based-hybrid RBF-ELM neural network for medical classification. Evol. Intel. (2022). https://doi.org/10.1007/s12065-022-00758-3
11. Wen, H., Xie, W., Pei, J.: A structure-adaptive hybrid RBF-BP classifier with an optimized learning strategy. PLoS ONE **11**(10), e0164719 (2016). https://doi.org/10.1371/journal.pone.0164719
12. Wen, H., Xie, W., Pei, J., Guan, L.: An incremental learning algorithm for the hybrid RBF-BP network classifier. EURASIP J. Adv. Sig. Process. **2016**(1), 1–15 (2016). https://doi.org/10.1186/s13634-016-0357-8
13. Wen, H., Fan, H., Xie, W., Pei, J.: Hybrid structure-adaptive RBF-ELM network classifier. IEEE Access **5**, 16539–16554 (2017). https://doi.org/10.1109/ACCESS.2017.2740420
14. Wen, H., Yan, T., Liu, Z., Chen, D.: Integrated neural network model with pre-RBF kernels. Sci. Progress **104**(3) (2021). https://doi.org/10.1177/00368504211026111
15. Wen, H., Li, T., Chen, D., Yang, J., Che, Y.: An optimized neural network classification method based on kernel holistic learning and division. Math. Probl. Eng. (2021). https://doi.org/10.1155/2021/8857818

Data Mining and Machine Learning

Comparison of Machine Learning Models to Classify Documents on Digital Development

Uvini Ranaweera(✉), Bawun Mawitagama, Sanduni Liyanage, Sandupa Keshan, Tiloka De Silva, and Supun Hewawalpita

University of Moratuwa, Katubedda, Moratuwa 10440, Sri Lanka
{ranaweeraraua.19,mawitagamabc.19,liyanagesc.19,keshanams.19,tilokad,supungs}@uom.lk

Abstract. Automated document classification is a trending topic in Natural Language Processing (NLP) due to the extensive growth in digital databases. However, a model that fits well for a specific classification task might perform weakly for another dataset due to differences in the context. Thus, training and evaluating several models is necessary to optimise the results. This study employs a publicly available document database on worldwide digital development interventions categorised under twelve areas. Since digital interventions are still emerging, utilising NLP in the field is relatively new. Given the exponential growth of digital interventions, this research has a vast scope for improving how digital-development-oriented organisations report their work. The paper examines the classification performance of Machine Learning (ML) algorithms, including Decision Trees, k-Nearest Neighbors, Support Vector Machine, AdaBoost, Stochastic Gradient Descent, Naive Bayes, and Logistic Regression. Accuracy, precision, recall and F1-score are utilised to evaluate the performance of these models, while oversampling is used to address the class-imbalanced nature of the dataset. Deviating from the traditional approach of fitting a single model for multiclass classification, this paper investigates the One vs Rest approach to build a combined model that optimises the performance. The study concludes that the amount of data is not the sole factor affecting the performance; features like similarity within classes and dissimilarity among classes are also crucial.

Keywords: Natural Language Processing · Document Classification · Class-imbalanced · Multiclass Classification · Machine Learning

1 Introduction

With the increased integration of technology in operations, electronic documents or e-documents have seen significant prominence. E-documents usually include Portable Document Format (PDF), text files, e-mails, HTML, and PostScript [6]. The use of such documents has grown due to properties like accessibility and

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 59–73, 2023.
https://doi.org/10.1007/978-981-99-7969-1_5

convenience pushing towards the information explosion era [13]. Because of the growing use of e-documents, databases are developed with predefined categories, and these documents are classified based on their content. This classification enhances the document management mechanism within the database and makes it convenient for users to locate and refer to the documents based on their specific requirements.

In most databases, the categories of documents are determined by human annotators. However, given the exponential increase in e-documents available, manual classification has become a strenuous task and continuing the process has been challenging. Though it is time-consuming, the task is an easy decision for human beings to make [5]. As a result of continuous experiments, e-document classification has now been automated with user-defined categories and is widely used for databases [5]. Automated document classification uses computer programs to map documents into one or more predefined classes [8]. Many attempts have been introduced in the literature to classify domain-specific documents [4,19,26,28].

Along the line, digital development initiatives are a rising domain to be embedded in Knowledge Management Systems. The field of digital development has seen a leap in its significance in the world with digital transformation and the growing reliance on digital technology. This surge has led to increased reports on the topic, broadly covering areas like digital infrastructure, literacy, finance, services, data systems and education. Effective classification of documents in the digital development field is important as it facilitates stakeholders to engage in knowledge sharing, informed decision making and targeted analysis with the help of insights, trends and other useful information extracted from these documents. Since the field is still expanding, the documents available are limited, making it possible for human annotators to classify the available documents manually. As a result, the world today has many documents accumulated on databases specified for digital development. This does not translate to millions of reports but several thousands of digital development reports accessible online. However, over time, the reports on digital development will grow significantly, making the manual classification process challenging.

This paper presents the results of automated document classification in the digital development domain. To address all the domain-specific issues mentioned, the study employs a dataset with twelve predefined classes and is class imbalanced due to fewer data points. A range of ML algorithms like Decision Trees, k-Nearest Neighbors (k-NN), Support Vector Machine (SVM), AdaBoost, Stochastic Gradient Descent (SGD), Naive Bayes and Logistic Regression [13] are employed in classifying the documents.[1]

---

[1] At the beginning of the study, the performance of Deep Learning (DL) algorithms such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and Transformers were measured. However, training DL algorithms was challenging, given the limitations in computational resources and the number of data points available. With low performance recorded, it was unanimously decided to drop the DL algorithms and proceed with the ML algorithms for this study.

Another unique aspect of this study is that the dataset used is not specifically designed for a research purpose and exhibits thousands of features per record, increasing the practicality of the research.

Since going ahead with a single ML model to perform multiclass classification on a limited number of documents is challenging, the study adopts the One vs Rest (OvR) approach, assuming that the same model might not perform equally well for all the classes. It identifies the best classifier for each class individually, after which an overall model is fitted by combining the classifiers. The dataset being class-imbalanced (the number of records among categories is unequal) is another challenge to handle, which could result in inaccurate classifications if left unaddressed [13].

The rest of the research paper is structured as follows. Section 2 reviews the literature on text document classification using various ML algorithms. Section 3 describes the methodology, which includes an outline of the conceptual framework followed in the study, an introduction to the dataset, the preprocessing techniques applied, the two phases of the distinct models developed for classification, and the performance measures used for model evaluation. Section 4 analyses the results and carries out an extensive discussion of the findings of the study. Section 5 gives the conclusion at the end of the study and suggestions for future works to enhance the functionality of the research.

2 Literature Review

Natural Language Processing (NLP) has a wide range of applications that improve word processing significantly. Church and Rau [7] suggested that NLP includes advanced data organisation features such as spelling correction, dictionary access, and categorisation. Furthermore, they stated that NLP improves information management and retrieval systems by automating data classification and extraction. NLP contributes significantly to accurate and organised content creation by improving word processing capabilities. Sebastiani [21] showed that automated document classification classifies text documents using computational techniques into predefined categories. It entails labelling or tagging documents according to their content using tools in ML and NLP.

Many studies have been carried out worldwide to evaluate and suggest methodologies to automate the process of text classification. Kowsari et al. [16] have surveyed algorithms for text classification. The study concluded that algorithms such as Rocchio, boosting, bagging, logistic regression, Naïve Bayes, k-NN and SVM have their own pros and cons in text classification. Ting et al. [23] examined Naive Bayes' suitability as a document classifier. The findings demonstrated that Naive Bayes outperforms other classifiers in precision, recall, and F-measure, achieving a high classification accuracy. The performances of the k-NN and Naive Bayes algorithms for classifying text documents were compared by Rasjid and Setiawan [20] using the TREC Legal Track dataset. The study concluded that Naive Bayes outperforms k-NN with $k = 1$ in terms of recall. However, as k rises, k-NN outperforms Naive Bayes in terms of F-measure, recall, and precision. k = 13 is found to be the ideal value for the dataset, producing stable and precise results. The lack of specific content in the text documents contributes to the overall poor performance, but the study suggested using k-NN with k values greater than 10 for better classification.

Moreover, studies have been carried out in the recent past on domain-specific document classification. Wei et al. [26] focused on applying DL algorithms such as CNN in the legal field, where large sets of electronically saved data are available. The study found that ML algorithms like SVM are generally suited for smaller datasets, whereas CNN performs significantly for larger datasets. Pandey et al. [19] talked about how ML algorithms like Naïve Bayes, Linear Discriminant Analysis (LDA), K-NN, SVM with different kernels, Decision Trees and Random Forests can be used to classify issue reports in the field of software development. Random Forests and SVM with certain kernels performed well under the metrics F-measure, Average accuracy, and weighted F-measure on 7401 issue reports extracted from five open-source projects; three of these projects were tracked through Jira and two through Bugzilla. Behera et al. [4] described how DL algorithms CNN, RNN, Deep Neural Network (DNN) and Ensemble Deep Learning models are used in the biomedical domain and how the performance of these models outweighs that of ML algorithms. Using three datasets: Ads dataset, TREC 2006 Genomics Track dataset and BioCreative Corpus III (BC3), it was found that, with the increase of the dataset size, the use of DL algorithms is much more successful than ML algorithms. Zhang et al. [28] discussed how ML algorithms such as SVM, K-NN, Linear regression, Decision Trees, Naïve Bayes, and ensemble methods could analyse reports on construction site accidents. The weighted F1 score indicated that the optimised ensemble method performs the best.

As stated above, though there is much literature on diverse fields, less work can be found on classifying digital development-related reports. Further, it is rare to find research conducted for an imbalanced dataset, even though it is less likely to find a perfectly balanced dataset in reality.

3 Methodology

3.1 Defining the Dataset and Preprocessing

As indicated in Fig. 1, a conceptual framework was designed to address the research aim of optimising the model performance in digital development report classification. Accordingly, to carry out the study, first, it is necessary to get a dataset centred around digital development initiatives. The United States Agency for International Development - Digital Ecosystem Evidence Map (USAID DEEM) [24] was chosen as the primary data source to study as it is a searchable database that stores worldwide digital development evidence, making it a world map for such data (Fig. 2).

The USAID DEEM is an evidence map compiled by the United States Agency for International Development (USAID) as a part of its digital strategy. USAID continually adds reports to this evidence map to cover the worldwide efforts in

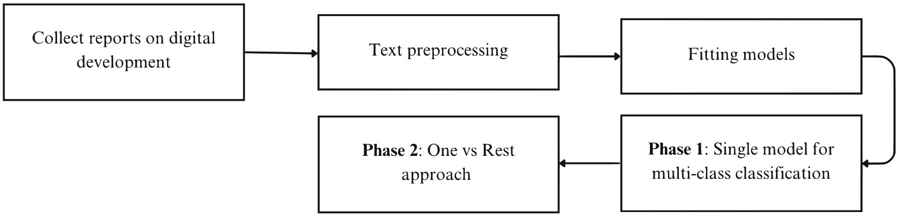

Collect reports on digital development
Text preprocessing
Fitting models
Phase 1: Single model for multi-class classification
Phase 2: One vs Rest approach

Fig. 1. Conceptual framework.

digital development interventions. As indicated in Fig. 2, the process is partially completed to date, and still, there are only a few reports from most parts of the world.

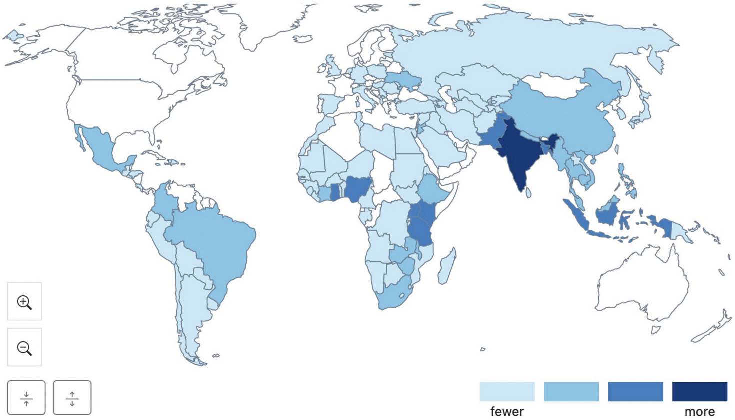


Fig. 2. Worldwide distribution of reports on digital development interventions.

The USAID DEEM dataset includes 851 documents[2] classified under twelve pre-defined intervention areas: Child Protection, Cybersecurity, Data Privacy, Data Systems & development, Digital Finance, Digital Inclusion, Digital Information Services, Digital Infrastructure Development, Digital literacy, Policy & regulation or Digital Services, E-government and Upskilling/Capacity Building. The document database includes a variety of reports, such as research papers, project reports and case studies. However, a heavy class imbalanced nature is observed among the twelve classes, as shown in Fig. 3.

---

[2] Since there is no intention to train a multi-label classification model for this study, duplications were eliminated, downsizing the dataset to 615 records.

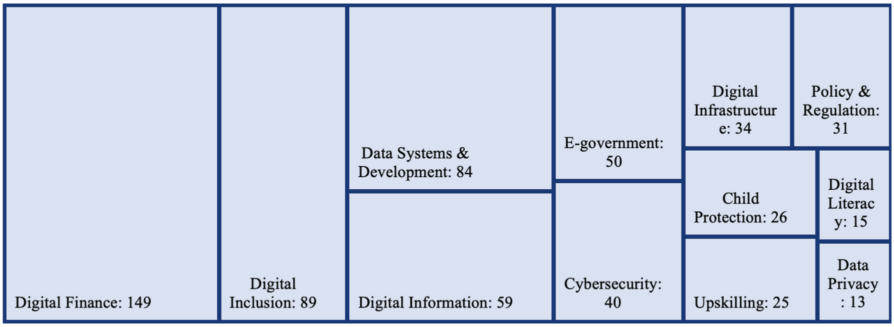


Fig. 3. Distribution of documents across intervention areas after removing duplicates.

Several factors were considered when choosing USAID DEEM as the custom dataset for this study. Since the study intends to evaluate classification models for reports on digital development, a dataset centring around the field was a must. Much research has been conducted on balanced datasets, making accuracy a more reliable estimator to measure the success of the classification. However, when gathering data from an area that is still evolving, it is impossible to expect a well-balanced dataset. So, to reflect practicality, it was necessary to have a dataset that is being used at an organisational level.

Furthermore, most studies in text classification have utilised datasets designed for research purposes, such as newsgroups and Reuters, which include only a few features to make the process easier and faster. USAID DEEM, where the document length ranged from a few words to 100,000+ words, enriched the research to handle datasets with many characteristics. Hence, this study employs the full content of a document rather than selecting a section of an article like the heading, abstract or introduction. The approach allows us to feed a 360-degree view of the documents, enhancing model performance, given that document classification is quite a new topic for digital development reports.

After defining the dataset, as for the conceptual framework in Fig. 1, next up was to preprocess the text data. Text preprocessing is the first step in text mining; that needs to be done following web scraping. Since all the documents from USAID DEEM were extracted in PDF format, converting them into text format was necessary before applying any preprocessing steps. As Diem et al. [9] suggested, the Optical Character Recognition (OCR) technology was used to read each document in the database to the text format. Finally, to transfer the text documents into a dataframe, the PySpark library was utilised [22].

Under text preprocessing, a series of steps such as lowercasing, tokenisation, stop words removal and lemmatisation was carried out [15] using the Python library NLTK. NLTK is an NLP toolkit that is used for most of the preprocessing steps, and Hardeniya [14] stated it as one of the most popular libraries for NLP as it was built on the learning curve of Python, making it fast and easy to work with. It is noteworthy to mention that a domain-specific stopword list was defined for successful execution. When defining the stop word list, words like "digital", "development", "project", etc., were included as they frequently occur in digital development reports, adding no value. Finally, to make the text corpus ready to be fed into ML algorithms, each text record was converted into a vector format using Term Frequency - Inverse Document Frequency (TF-IDF) vectorisation [2].

For the experiment, the final dataset of 615 records was utilised, with 70% of the data points considered as training data and the remainder as test data points. The 185 test data points were used to evaluate how well the models generalise on unseen data.

3.2 Defining Models

The classification algorithms were applied in two major phases, as mentioned in Fig. 1. However, before directly fitting a model, it was necessary to treat the class-imbalanced nature of the dataset as the unequal distribution of data points among the classes may result in poor performance in minority classes and model bias for the majority classes [1]. To further elaborate on minority and majority classes, in the USAID DEEM dataset, classes like Digital Finance and Digital Inclusion can be categorised as majority classes, while classes like Data Privacy and Digital Literacy come under minority classes.

Among the techniques to address the class-imbalanced nature of the dataset are oversampling, undersampling and adjusting class weights. Oversampling is where new data points are added to the minority classes either by adding new instances or by repeating existing instances [18]. Undersampling is where data points are removed from the majority classes [18] while the class weights method will assign higher weights for minority classes and vice versa to avoid the model bias resulting from imbalanced data [27]. For the underlying study, the oversampling technique was employed since the training dataset was limited to 430 data points.

The first phase of the analysis focused on training single models capable of multiclass classification on the USAID DEEM dataset. The ML algorithms trained were Decision trees, Naïve Bayes, k-NN, logistic regression, AdaBoost, SGD and SVM. These algorithms were picked in a way to represent different aspects of ML models, such as lazy learning (k-NN), eager learning (decision trees, SVM), discriminative models (logistic regression) and generative models (Naïve Bayes). Moreover, an ensemble learning approach was adapted to evaluate classification performance by combining the two best-performing ML algorithms based on the F1 score. The ensembling process used the majority voting mechanism to decide the class from the two base models [10].

Moving on to the study's second phase, the aim was to train the above-mentioned ML algorithms for binary classification on a class-imbalanced dataset. Given that the study deals with a multiclass dataset, to try out binary classification, a One vs Rest (OvR) approach was utilised. There were two assumptions behind adopting OvR to this study: deploying a single model to predict class labels might not perform better when the dataset is class imbalanced, and the uniqueness inherent to each class will make an algorithm record different performances among classes. Thus, instead of training one classification model for all the classes, comparing model performances and fitting the best algorithm for each class will generate better results in classification.

Though there is no motive to try multi-label classification at this stage, the OvR approach eases addressing multi-label classification as well (one document belonging to more than one class), which is the reality for this dataset. So, trying out OvR will allow the study to expand its scope to accommodate multi-label classification. At the moment, by applying an OvR approach, this study trains individual models that perform significantly better for at least a few of the classes and tries to understand what hinders the model performance for the rest of the classes.

3.3 Performance Measures

Several performance measurements, such as accuracy (correctly classified instances over total instances), precision (correctly classified positive instances over total positive instances predicted), recall (correctly classified positive instances over actual positive instances), and F1(harmonic mean of precision and recall), were used to measure the performance of each algorithm [11]. All these performance measures can be derived using the four features: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [4], as summarised in Table 1.

Table 1. Confusion matrix

	Predicted Label	
Actual Label	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Since the study deals with an imbalanced dataset, accuracy is unreliable. As an unbiased estimator, F1-score (1), a combination of precision and recall [25] can be used.

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{1}$$

The study's phase one performance was measured using the weighted average F1-score (2) to account for the class-imbalanced nature under multiclass classification [4].

$$\text{Weighted average F1} = \frac{\sum_{i=1}^{m} |y_i| \times \frac{2\times \text{tp}_i}{2\times \text{tp}_i + \text{fp}_i + \text{fn}_i}}{\sum_{i=1}^{m} |y_i|} \tag{2}$$

4 Results and Discussion

This section of the paper investigates the results of the models trained under phase one (training single models for multiclass classification) and phase two (OvR approach for binary classification). All these models have been executed using Google CoLab Jupyter Notebook with Python 3.10 version with a maximum RAM of 12 GB.

First, an Exploratory Data Analysis (EDA) was carried out to get insights into the dataset. As for Fig. 4, the twelve distinct classes of the dataset have similarities and dissimilarities, making it an interesting point to study how the models would perform classification.

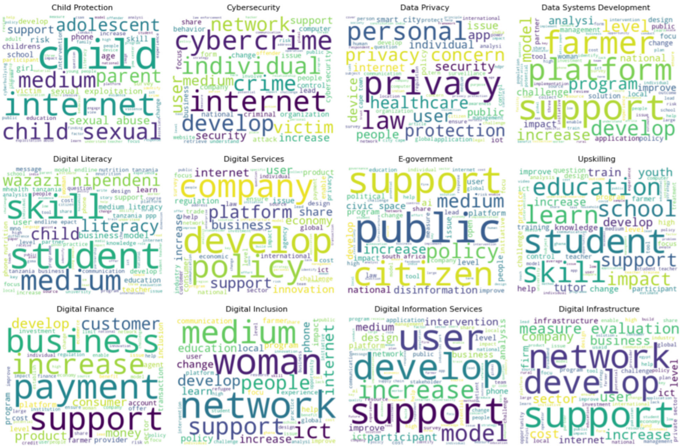


Fig. 4. Word clouds for intervention areas.

Before fitting the models under the first phase, the class labels, which took categorical values, were encoded numerically. Moreover, since the dataset was heavily class imbalanced, random oversampling was applied, which balanced the training dataset with each class having exactly 100 records. In line with the class-imbalanced nature of the dataset, the weighted average F-1 score was used over accuracy as an unbiased measure of model performances on the test dataset.

From the weighted average F-1 scores recorded in Table 2, it is evident that logistic regression outperformed the rest with an F1-score of 0.53, followed by SGD with a score of 0.51. It is a noticeable fact that AdaBoost has recorded the worst results for phase one. Even literature has strong evidence for AdaBoost reporting worst performances in text classification [3,12].

As per literature [17], F1- measures should be optimised to generalise a model for further classifications. The results indicated in Table 2 show that none of the models succeeds in going beyond 0.53, and most of them fall under a mid-range performance (except for Decision trees and AdaBoost, all the other classifiers have reported a performance score ranging from 0.42–0.53).

Table 2. Phase one classification report for ML algorithms.

ML Algorithm	Accuracy	Precision	Recall	F-1
Decision trees	0.34	0.38	0.34	0.34
Naïve Bayes	0.43	0.53	0.43	0.45
k-NN	0.39	0.52	0.39	0.42
Logistic regression	0.53	0.56	0.53	**0.53**
AdaBoost	0.07	0.09	0.07	0.05
SGD	0.51	0.59	0.51	0.51
SVM	0.47	0.45	0.47	0.42
Ensemble learning	0.48	0.43	0.48	0.42

The second phase of the study involved training ML models for each of the twelve classes separately. In each iteration, the dataset was binary encoded, where 1 represented the documents of the class in consideration, while the rest were labelled as 0. Table 3 records the performance measures for selected eight classes out of the twelve classes the models were trained. The eight classes were picked to represent the classes with the least number of documents, the classes with the most and those with a moderate number of documents. For the selected classes in Table 3, the document spread is: Child Protection-26, Cybersecurity-40, Data Privacy-13, Data Systems-84, Digital Finance-149, Digital Inclusion-89, Digital Information Services- 59, and Policy & Regulations-31.

Similar to phase 1, since the dataset is class imbalanced, the F1-score was utilised to evaluate the model performance on the test dataset. As per the values recorded in Table 3, the model performances were significantly better for most classes than fitting a single model for a multiclass problem. The first phase of the study achieved a maximum performance of 0.53 under logistic regression by fitting a single model for the test dataset. However, training separate models for each class using the OvR approach successfully resulted in better performances that reached up to 0.8 for some classes. Out of the best classifiers identified for the classes under the OvR approach, the least desirable performance of 0.4 was recorded for the Data Systems class under k-NN. Further supporting the conclusion of phase one, none of the classes recorded their best performance under decision trees and AdaBoost, while the best results were recorded for SGD and logistic regression.

The most noteworthy observation of this approach is the presence of clear evidence to prove that it is not always the same classification algorithm that works well for all the classes. For example, though the SGD model generated the highest performance for the Child Protection class with an F1-score of 0.86, the algorithm underperformed for the Digital Inclusion class, recording an F1-score of 0.25. Thus, the OvR approach allows us to train different classification algorithms for each of the twelve classes and optimise the classification results. This statement is further supported by the F1 scores recorded in Table 4 for each

Table 3. Classification report for ML algorithms under OvR.

ML Algorithm	Performance Measures	Child Protection	Cyber-security	Data Privacy	Data Systems	Digital Finance	Digital Inclusion	Digital Information	Policy & Regulations
Decision trees	Accuracy	0.93	0.94	0.96	0.82	0.81	0.74	0.79	0.79
	Precision	0.42	0.75	0.44	0.00	0.75	0.43	0.19	0.11
	Recall	0.45	0.40	0.57	0.00	0.54	0.47	0.23	0.18
	F-1	0.43	0.52	0.50	0.00	0.63	0.45	0.21	0.14
Naïve Bayes	Accuracy	0.81	0.77	0.92	0.82	0.70	0.58	0.85	0.66
	Precision	0.23	0.26	0.30	0.00	0.50	0.33	0.41	0.21
	Recall	1.00	1.00	0.86	0.00	1.00	0.79	0.64	0.94
	F-1	0.38	0.41	0.44	0.00	0.67	**0.47**	0.50	0.34
k-NN	Accuracy	0.90	0.95	0.94	0.68	0.82	0.49	0.69	0.72
	Precision	0.36	0.67	0.30	0.30	0.68	0.23	0.19	0.11
	Recall	0.91	0.80	0.43	0.61	0.77	0.53	0.50	0.29
	F-1	0.51	0.73	0.35	**0.40**	0.72	0.33	0.28	0.16
Log. Reg	Accuracy	0.95	0.95	0.98	0.78	0.86	0.73	0.86	0.89
	Precision	0.53	0.62	0.80	0.30	0.71	0.42	0.44	0.41
	Recall	0.82	0.87	0.57	0.18	0.91	0.44	0.50	0.41
	F-1	0.64	0.72	0.67	0.23	**0.80**	0.43	0.47	**0.41**
AdaBoost	Accuracy	0.95	0.96	0.96	0.78	0.81	0.75	0.85	0.86
	Precision	0.62	0.89	0.44	0.36	0.76	0.44	0.14	0.17
	Recall	0.45	0.53	0.57	0.27	0.55	0.37	0.05	0.12
	F-1	0.53	0.67	0.50	0.31	0.64	0.41	0.07	0.14
SGD	Accuracy	0.98	0.96	0.97	0.79	0.84	0.77	0.88	0.88
	Precision	0.9	0.77	0.62	0.39	0.71	0.54	0.47	0.14
	Recall	0.82	0.67	0.71	0.27	0.82	0.16	0.41	0.06
	F-1	**0.86**	0.71	**0.67**	0.32	0.76	0.25	0.44	0.08
SVC	Accuracy	0.97	0.96	0.96	0.83	0.84	0.75	0.89	0.90
	Precision	1.00	0.79	0.00	0.67	0.68	0.46	0.67	0.25
	Recall	0.55	0.73	0.00	0.06	0.89	0.40	0.18	0.06
	F-1	0.71	**0.76**	0.00	0.11	0.78	0.42	0.29	0.10
Ensemble learning	Accuracy	0.97	0.97	0.97	0.82	0.82	0.75	0.85	0.89
	Precision	1.00	1.00	0.75	0.43	0.92	0.46	0.41	0.33
	Recall	0.55	0.60	0.43	0.09	0.43	0.40	**0.64**	0.24
	F-1	0.71	0.75	0.55	0.15	0.59	0.42	0.50	0.28

Table 4. Phase one classification report for logistic regression.

Intervention area	Precision	Recall	F-1
Child Protection	1.00	0.64	0.78
Cybersecurity	0.60	0.75	0.67
Data Privacy	0.00	0.00	0.00
Data Systems	0.46	0.43	0.44
Digital Finance	0.71	0.91	0.80
Digital Inclusion	0.50	0.55	0.52
Digital Information	0.23	0.58	0.33
Policy & Regulations	0.30	0.32	0.33
Accuracy			0.53
Weighted Avg F-1	0.56	0.53	0.53

class under the logistic regression model trained during the first phase (multi-class classification). Though fitting the logistic regression model on test data generated a weighted average F1-score of 0.53, it performed poorly for some classes. The logistic regression model has performed well in predicting Child

Protection and Digital Finance documents, while the performance for the Data Privacy class is nil. Nevertheless, adopting the OvR approach in the Data Privacy class made it possible to report an F1 score of 0.67 under the SGD model. The eight classes in Table 3 have documents ranging from 13–149. Data Privacy is the class with the lowest number of documents, which is 13. However, it still managed to report a moderate performance of 0.67 under both logistic regression and SGD. Saying so, the Data Systems class did not meet an acceptable performance (F1-score of 0.4 under k-NN), given that it has 84 documents. The Digital Finance class, with the highest number of documents, met a significant performance of 0.8 for logistic regression. In contrast, with just 26 documents, Child Protection recorded the highest performance of 0.86 under the SGD model. This brings us to the point that it is not always the number of training samples available for each class that determines the model's performance.

Facts like dissimilarity among classes and similarity within classes also affect the performance of a model. In agreement, Child Protection and Digital Finance classes have several terms unique to the class. For example, words like "child", "sexual", and "adolescent" are unique for the Child Protection class (Fig. 4). But for a class like Digital Inclusion, the frequent words indicated in Fig. 4 are network, people, ict and phone. Given that the entire dataset is centred around digital development efforts, those words can be common to all the classes, resulting in weak performances for the models trained for such classes. With the recorded results, it can be stated that adapting an OvR approach generates more significant results, at least for some classes, than fitting a single model for all.

5 Conclusion

Automated text document classification is commonly used in various industries like customer service, news media, legal, healthcare, e-commerce, etc., to facilitate efficient information retrieval, aid decision making and enhance overall end-user satisfaction. With the rising number of digital development initiatives, a vast data pool with reports on such initiatives is evolving. This study has utilised several ML algorithms to automate the classification process of digital development reports, which will help world organisations such as the World Bank to effectively track and monitor development initiatives worldwide.

In the study's first phase, a single model was fitted for a multiclass dataset. However, the weighted average F1-score did not report beyond 0.53 due to insufficient documents for some intervention areas and similarities among classes. This is a drawback of the USAID DEEM dataset being used for the study. As nations have started to embrace digital development lately, USAID continues to add new reports to the database. Therefore, future performances can be optimised by augmenting the dataset with reports from underrepresented intervention areas.

To overcome the drawbacks of the first phase of the study, an OvR approach was adopted, and comparatively, it yielded better performance for the custom dataset. The improved performance is due to models being fitted for a binary classification task, i.e., the models were trained to distinguish between two classes as a document belonging to the class or not. On the other hand, in the multiclass classification approach, a model was trained to classify a document into one of the twelve classes, making the process complex as the models must learn variation among twelve classes.

Accordingly, for a multiclass classification task where training data points are limited, following an OvR approach will result in better performances. So, the recommended method for this study is to train separate models for each of the twelve classes and finally call them into a single function that can output all the classes a document belongs to, along with a probability score. Under this approach, unlike training a single model for multiclass classification, which maps one class for a document, the model will map more than one class for a document due to training errors. The above-stated issue is no harm to the USAID DEEM dataset, as it is more of a multi-label dataset. However, if one is dealing with a situation where a document can belong to only one of the classes, the class that most fits the document can be chosen based on the highest probability score.

For simplicity, if one is interested in training a single ML model for multiclass classification, the recommended algorithm is SGD. The conclusion is drawn based on the average of F1 scores reported across the twelve intervention areas in phase 2. In contrast, other algorithms performed exceptionally well for certain classes and reported very weak performances for the rest of the classes. Here, one might argue that the results reported under phase one indicated logistic regression as the best-performing model for multiclass classification. Though the particular model reported a higher average performance, it reported a null F1-score for the Data Privacy class. This is in contrast to SGD, which performed moderately for all the classes with a weighted average F1-score of 0.51.

However, it should be noted that no significant performance was yielded by either the OvR or the full model for some of the classes. Few classes remained underperforming due to the lack of data points and the limited differences between classes. To elevate the overall outcome of the research, future work should address training models on a larger dataset while considering the possibility of overfitting. As digital development keeps rising, so will the reports on such initiatives. Therefore, there is space to make the dataset sounder and extend the study to include development reports pooled by other organisations. This is a positive sign to integrate DL models in future work, which were found to be outperforming ML algorithms for large datasets [4].

References

1. Adil, M., Ansari, M.F., Alahmadi, A., Wu, J.Z., Chakrabortty, R.K.: Solving the problem of class imbalance in the prediction of hotel cancelations: a hybridized machine learning approach. Processes **9**(10), 1713 (2021). https://doi.org/10.3390/pr9101713
2. Aizawa, A.: An information-theoretic perspective of tf-idf measures. Inf. Process. Manag. **39**(1), 45–65 (2003). https://doi.org/10.1016/s0306-4573(02)00021-3

3. Al Qadi, L., El Rifai, H., Obaid, S., Elnagar, A.: Arabic text classification of news articles using classical supervised classifiers. In: 2019 2nd International Conference on New Trends in Computing Sciences (ICTCS), pp. 1–6. IEEE (2019). https://doi.org/10.1109/ictcs.2019.8923073
4. Behera, B., Kumaravelan, G., Kumar, P.: Performance evaluation of deep learning algorithms in biomedical document classification. In: 2019 11th International Conference on Advanced Computing (ICoAC), pp. 220–224. IEEE (2019)
5. Borko, H., Bernick, M.: Automatic document classification. J. ACM (JACM) **10**(2), 151–162 (1963). https://doi.org/10.1145/321160.321165
6. Caldas, C.H., Soibelman, L., Han, J.: Automated classification of construction project documents. J. Comput. Civ. Eng. **16**(4), 234–243 (2002). https://doi.org/10.1061/(asce)0887-3801(2002)16:4(234)
7. Church, K.W., Rau, L.F.: Commercial applications of natural language processing. Commun. ACM **38**(11), 71–79 (1995). https://doi.org/10.1145/219717.219778
8. Cohen, A.M.: An effective general purpose approach for automated biomedical document classification. In: AMIA Annual Symposium Proceedings, vol. 2006, p. 161. American Medical Informatics Association (2006)
9. Diem, M., Kleber, F., Sablatnig, R.: Text classification and document layout analysis of paper fragments. In: 2011 International Conference on Document Analysis and Recognition, pp. 854–858. IEEE (2011). https://doi.org/10.1109/ICDAR.2011.175
10. Dong, X., Yu, Z., Cao, W., Shi, Y., Ma, Q.: A survey on ensemble learning. Front. Comp. Sci. **14**(2), 241–258 (2019). https://doi.org/10.1007/s11704-019-8208-z
11. Forman, G., et al.: An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. **3**, 1289–1305 (2003)
12. Gutman, J., Nam, R.: Text classification of reddit posts. Technical report. New York University (2015)
13. Hakim, A.A., Erwin, A., Eng, K.I., Galinium, M., Muliady, W.: Automated document classification for news article in Bahasa Indonesia based on term frequency inverse document frequency (TF-IDF) approach. In: 2014 6th International Conference on Information Technology and Electrical Engineering (ICITEE), pp. 1–4. IEEE (2014). https://doi.org/10.1109/ICITEED.2014.7007894
14. Hardeniya, N.: NLTK essentials: build cool NLP and machine learning applications using NLTK and other python libraries. Packt Open Source. Packt Publishing, Birmingham (2015)
15. Kadhim, A.I.: An evaluation of preprocessing techniques for text classification. Int. J. Comput. Sci. Inf. Secur. (IJCSIS) **16**(6), 22–32 (2018)
16. Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., Brown, D.: Text classification algorithms: a survey. Information **10**(4), 150 (2019)
17. Lipton, Z.C., Elkan, C., Naryanaswamy, B.: Optimal thresholding of classifiers to maximize F1 measure. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014. LNCS (LNAI), vol. 8725, pp. 225–239. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44851-9_15
18. Mohammed, R., Rawashdeh, J., Abdullah, M.: Machine learning with oversampling and undersampling techniques: overview study and experimental results. In: 2020 11th International Conference on Information and Communication Systems (ICICS), pp. 243–248. IEEE (2020). https://doi.org/10.1109/ICICS49469.2020.239556

19. Pandey, N., Sanyal, D.K., Hudait, A., Sen, A.: Automated classification of software issue reports using machine learning techniques: an empirical study. Innovations Syst. Softw. Eng. **13**(4), 279–297 (2017). https://doi.org/10.1007/s11334-017-0294-1
20. Rasjid, Z.E., Setiawan, R.: Performance comparison and optimization of text document classification using k-NN and Naïve Bayes classification techniques. Procedia Comput. Sci. **116**, 107–112 (2017)
21. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. (CSUR) **34**(1), 1–47 (2002). https://doi.org/10.1145/505282.505283
22. Singh, P., Singh, P.: Natural Language Processing. Machine Learning with PySpark: with Natural Language Processing and Recommender Systems, pp. 191–218 (2019)
23. Ting, S., Ip, W., Tsang, A.H., et al.: Is Naive Bayes a good classifier for document classification. Int. J. Softw. Eng. Appl. **5**(3), 37–46 (2011)
24. USAID: Deem digital ecosystem evidence map. https://deem.digitaldevelopment.org. Accessed 24 May 2023
25. Wardhani, N.W.S., Rochayani, M.Y., Iriany, A., Sulistyono, A.D., Lestantyo, P.: Cross-validation metrics for evaluating classification performance on imbalanced data. In: 2019 International Conference on Computer, Control, Informatics and Its Applications (IC3INA), pp. 14–18. IEEE (2019). https://doi.org/10.1109/IC3INA48034.2019.8949568
26. Wei, F., Qin, H., Ye, S., Zhao, H.: Empirical study of deep learning for text classification in legal document review. In: 2018 IEEE International Conference on Big Data (Big Data), pp. 3317–3320. IEEE (2018)
27. Yu, S., Guo, J., Zhang, R., Fan, Y., Wang, Z., Cheng, X.: A re-balancing strategy for class-imbalanced classification based on instance difficulty. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 70–79 (2022). https://doi.org/10.1109/CVPR52688.2022.00017
28. Zhang, F., Fleyeh, H., Wang, X., Lu, M.: Construction site accident analysis using text mining and natural language processing techniques. Autom. Constr. **99**, 238–248 (2019). https://doi.org/10.1016/j.autcon.2018.12.016

Exploration of the Feasibility and Applicability of Domain Adaptation in Machine Learning-Based Code Smell Detection

Peeradon Sukkasem and Chitsutha Soomlek(✉)

Department of Computer Science, College of Computing, Khon Kaen University,
Khon Kaen, Thailand
peeradon_s@kkumail.com, chitsutha@kku.ac.th

Abstract. Machine learning-based code smell detection was introduced to mitigate the limitations of the heuristic-based approach and the subjectivity issues. Due to limited choices of the publicly available datasets, most of the machine learning-based classifiers were trained by the earlier versions of open-source projects that no longer represent the characteristics and properties of modern programming languages. Our experiments exhibit the feasibility and applicability of using a machine learning classifier well-trained on the earlier versions of open-source projects to classify four types of code smells, i.e., *god class*, *data class*, *feature envy*, and *long method*, in modern Java open-source projects without extensive feature engineering. The performance produced by the supervised machine learning algorithms was evaluated and compared. Particle swarm optimization and Bayesian optimization were adopted to enhance the performance of the machine learning classifiers, i.e., decision tree and random forest. The experimental results indicated that the machine learning-based code smell classifiers adapt poorly to the different target domain. The hyper-parameter optimization slightly improves the performance of the machine learning classifiers when classifying *feature envy*, *god class*, and *long method* in a modern Java project.

Keywords: Code smells · Machine learning · Hyper-parameter optimization

1 Introduction

A code smell indicates poor design or implementation choices made during software development [13]. Software maintenance and software evolution are inevitable. Software maintenance and software evolution ensure customer satisfaction and support new features due to advancement of technology. The software evolution activities unwittingly introduce code smells into a system [34]. Code smells cause unfavorable issues in a software system, particularly software maintainability and software quality. Although code smells do not affect the

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 74–87, 2023.
https://doi.org/10.1007/978-981-99-7969-1_6

functionality and outputs of a program, code smells can lead to future software failure and increase technical debt and maintenance cost [14,18,29]. To identify and eliminate a code smell, code refactoring is a well established solution. Code refactoring improves the design and quality of the code [13]. To support code refactoring, an effective code smell detection technique is required.

For decades, automated code analysis tools and researches often utilize the metrics-based approaches [2]. The metric-based approach suffers from subjectivity issues and is error prone due to the lack of common definitions of code smells. Programming languages are improved over time, and some code smells are language-specific. These problems cause the metric-based code smell detection fails to find the most suitable set of metrics and threshold values [15,28]. Finding metrics and calibrating the thresholds are exhaustive.

Alternatively, a machine learning-based approach trains the classifiers using software properties or attributes. Machine learning mitigates the subjectivity issues and provides reliable code smell detection results when a machine learning classifier is trained on high-quality data.

In recent years, machine learning-based code smell detection has been progressively studied [2,16]. The performance produced by a machine learning algorithm heavily relies on the quality of the training dataset.

It takes considerable time and effort to create a high-quality code smell dataset that is labeled by experienced software developers and experts. The widely used code smell datasets are of the `Qualitas Corpus` benchmark [37]. Many well-studied code smell datasets were built upon the `Qualitas Corpus` benchmark [21,41]. Although Di Nucci et al. [10] commented that the threats to the validity of their work are inherited from the threats related to the creation of the `Qualitas Corpus`, other studies report that the machine learning classifiers trained by those datasets achieve high performance results [1,9,25]. Considering the age of the `Qualitas Corpus` project and the modern programming paradigm, it is questionable that a machine learning-based code smell detection well-trained on the code smell dataset built upon the `Qualitas Corpus` benchmark would produce a comparable performance when identifying a code smell in a modern software project. To explore the feasibility and applicability of domain adaptation in machine learning-based code smell detection, we present the experiments with two goals:

(1) To evaluate whether the machine learning-based code smell classifiers trained over a dataset built upon the `Qualitas Corpus` benchmark can effectively detect code smells in a modern software project.
(2) To investigate whether hyper-parameter optimization (HPO) can improve the performance of the machine learning classifiers in the same settings as (1).

To these aims, we adopt the open-source Java projects from the `Qualitas Corpus` dataset and the labels from Fontana et al. [11,12] in our training process. A wide range of code metrics are extracted from the Java projects. The machine learning algorithms are trained to classify four types of code smells, i.e., *god class*, *data class*, *feature envy*, and *long method*. `MLCQ` dataset [23], which is the most up-to-date industry-relevant code smell dataset at the time of writing, is used as the testing data. The performance results are evaluated and compared. Three HPO techniques of particle swarm optimization and Bayesian optimization are applied to improve the performance of the machine learning classifiers. The experimental results indicate that the classifiers produce poor performance results when the domain is shifted. HPO techniques hardly improve the recall scores of the classifiers. There is a slight difference between the baseline and optimized version in a large picture.

2 Related Work

Research on code smells is advancing and focusing on machine learning-based code smell detection [2] and using code smells as prediction factors [1,7,10,12,21,25]. Recent research shows that machine learning-based code smell detection mitigates the subjectivity issues and produces promising results [35]. Considering that the machine learning algorithms should be trained on a reliable dataset labeled by experienced software developers, up-to-date and reproducible code smell datasets are limited [21]. Collecting a large number of software systems with the information needed and labeling code smells in the target source code is a laborious task. Zakeri-Nasrabadi et al. [41] did a survey on 45 code smell datasets and revealed that most of the existing datasets cover limited types of code smells. The datasets in the survey are imbalanced, lack of supporting severity levels, and are restricted to Java programming language. We argue that code smell datasets of other programming languages exist, e.g., [38,40,42], but there are smaller choices compared to Java. The work of Zakeri-Nasrabadi et al. [41] also confirms that the datasets proposed by Palomba et al. [26] and Madeyski et al. [23] are the most comprehensive dataset. The datasets from Fontana et al. [11,12] are mostly cited in research work. Fontana's datasets are also built from the Qualitas Corpus benchmark. Besides its popularity, the Qualitas Corpus benchmark is outdated. The projects in the dataset are primarily developed by Java 5. Fontana's datasets have been commented on several aspects such as the size, types of code smells, and the ratio between smelly and non-smelly samples which is not realistic compared to the nature of the code smell problem [10].

In 2020, Mandelski et al. introduced the `MLCQ` dataset [23], which contains code smells from the newer versions of open-source Java projects. It is larger in size, is labeled by experienced software developers, and is more comprehensive. Recent research also adopted the `MLCQ` dataset and achieved promising results, e.g., [20,24,35,39]. Each of them spent time on extensive metrics extraction and feature engineering. Therefore, domain adaptation in machine learning-based code smell detection would mitigate the issues. The closest work to ours is [31] proposed by Sharma et al. Unlike our research, their work focuses on training deep learning models from C# code samples and evaluating the models over Java code samples and vice-versa. Our research aims to train the machine learning classifiers on one programming language and seeks the possibility for the model trained on the previous version of the programming language to work on the newer version.

Another possibility to enhance a machine learning classifier is to apply HPO. The closest work to this research and to our previous work [36] is presented by Shen et al. [32]. The researchers trained the machine learning-based code smell classifiers on Fontana et al.'s datasets [11,12] and applied HPO to improve the performance. Four optimizers were applied to six machine learning classifiers to identify two types of code smells, i.e., *data class* and *feature envy*. They did not focus on domain shifts in the data distribution in their research. This research is a continuation of our previous work [36], in which we already employed one particle swarm optimizer and two Bayesian optimizers to the machine learning-based code smell classifiers. The three optimizers were applied to the machine learning classifiers to enhance the performance. The rest of this research article is brand new.

3 Research Methodology

3.1 Data Collection

The main goals of this research are to explore whether the machine learning-based code smell classifiers trained on a dataset built upon the `Qualitas Corpus` benchmark can effectively detect code smells in a modern software project and to improve the performance of the classifiers through HPO. The experiments require three data sources: software projects in the `Qualitas Corpus` benchmark, smell and non-smelly samples in the `Qualitas Corpus` benchmark, and smell and non-smelly samples in the projects built on the newer versions of Java.

The Qualitas Corpus Benchmark. The dataset was contributed by Tempero et al. in 2010 [37] and the latest version (release 20130901r) contains 112 open-source Java software systems. Later in 2013, 74 systems were selected to create a code smell dataset by Fontana et al. [12]. The `Qualitas Corpus` benchmark contains a total of 6,785,568 lines of code, 3,429 packages, 51,826 classes, and 404,316 methods. Fontana et al. also provide smell and non-smelly samples of *god class*, *data class*, *feature envy*, and *long method* in the dataset.

In this research, a large set of code metrics were extracted by using code analysis tools, i.e., Design Features and Metrics for Java (DFMC4J) [1] and Understand tool by SciTools [30]. Code metrics exhibit software properties in various perspectives and can be used as independent variables to train machine learning models. The dependent variable is the label, which indicates whether a class or a method is a code smell or not. The label of a code sample is obtained from Fontana et al.'s code smell dataset. The major challenge is to match a code sample to its label and the corresponding code metrics due to the different file path formats used. There are 140 positive (smelly) instances and 280 negative (non-smelly) instances for each type of code smells in our training dataset.

The MLCQ Dataset. Madeyski and Lewowski [23] contribute an industry-relevant code smell dataset. The `MLCQ` dataset contains nearly 15,000 code samples together with severity levels (i.e., *critical*, *major*, *minor*, and *none*). These code samples were collected from Java open-source projects available on GitHub within the year 2019. 26 professional software developers reviewed and classified the code samples into four classes, i.e., *data class*, *blob* (a.k.a. *god class*), *feature envy*, and *long method*. A severity level is also given to a reviewed code sample. The *none* severity level indicates a non-smelly or negative sample. The `MLCQ` dataset contains 1057, 974, 454, and 806 positive instances for *data class*, *god class*, *feature envy*, and *long method*, respectively. For the negative instances, there are 2964, 3045, 2883, and 2556, respectively. This dataset is considered as the most comprehensive available code smell dataset in the field [41].

3.2 Data Preparation

When an instance in Fontana et al. contains a null code metric value, the instance is removed. In the case of the `MLCQ` dataset, several procedures were performed. Some Java projects are no longer available on the GitHub repository. In total, 518 projects from the `MLCQ` dataset were analyzed. Information relative to the reviewers, for instance, code sample id, reviewer id, and reviewing timestamp were also excluded from the dataset. Since there are multiple reviewers labeled on an instance, the majority vote was used to determine the positive and negative instances. Then, the Understand tool by SciTools [30] was employed to extract the code metric values. Due to the granularity difference between the code analysis tool and the dataset, there are instances with a null metric value. In this case, the instances were removed.

To test machine learning on unseen data, the machine learning model must be trained over a training set having identical properties as the testing set. We carefully studied the definitions of the code metrics provided by Fontana et al. and the Understand tool. Only identical code metrics were selected.

3.3 Experiments

Fontana et al.'s code smell dataset was used as a training set. The machine learning algorithms were tested over the `MLCQ` dataset. The experiments evaluated the performance metrics, e.g., precision, recall, and f-score. For each experiment, 10-fold cross validation was implemented with 10 repetitions. The experiments were repeated for 10 iterations to avoid randomness and then the results were averaged to conclude the experiments.

Selecting Machine Learning Algorithms. As mentioned earlier, this study is a continuation of our previous work [36]. Our previous work and other research in the same area confirm that decision tree [6] and random forest [5] produce the best performance in the context of code smell detection [35]. Both machine learning algorithms not only produce high performance results but also provide the benefits of interpretability. Although we can include other machine learning algorithms, we chose to study decision tree and random forest to seek the feasibility and applicability of domain adaptation in machine learning-based code smell detection. The experiments can be repeated on other machine learning algorithms to explore more. Note that both machine learning algorithms were implemented by using Scikit-learn [27].

Selecting Hyper-parameter Optimization Techniques. This research selected hyper-parameter optimization (HPO) techniques based on the characteristics of the hyper-parameters of the machine learning algorithms. Considering the decision tree and random forest algorithm, the majority of the hyper-parameters are discrete. One of them is categorical. Decision tree and random forest possess the characteristic of a large hyper-parameter search space with multiple hyper-parameter types. From this perspective, Bayesian optimization using random forest (SMAC), Bayesian optimization using tree-structured parzen estimator (BO-TPE), and particle swarm optimization (PSO) are well-matched to the machine learning algorithms. The selected hyper-parameter and their configuration space are given in Table 1, which is inherited from our previous work [36]. Note that all HPO techniques use the same hyper-parameter configuration space and consider recall score as an objective function with 10 maximum optimization iterations.

Table 1. List of selected hyper-parameters with their characteristics and configuration space.

Hyper-parameter	**Characteristic**	**Range**
criterion	Categorical	‘gini’, ‘entropy’
max_depth	Discrete	[5, 50]
min_samples_split	Discrete	[2, 11]
min_samples_leaf	Discrete	[1, 11]
max_features	Discrete	[1, 64]
n_estimators	Discrete	[100, 300]

Bayesian optimization (BO) [33] is an iterative algorithm learning from previously found information. The algorithm consists of two main components: a probabilistic surrogate model and an acquisition function. BO works by building a surrogate model of the objective function, detecting the optimal on the surrogate model, and applying the optimal configuration to the real objective function. The surrogate model is updated with the new results and these procedures are repeated until the optimal solution is found, or the maximum number of iterations is reached. BO can use tree-parzen estimator (TPE) [3] and random forest (SMAC) [17] to support multiple types of hyper-parameters. Therefore, SMAC and BO-TYPE are included in this research.

Particle swarm optimization (PSO) [19] is an optimization technique inspired by biological theories. The process of PSO starts by randomly initializing a group of samples called particles. These particles explore the search space to search for the optimal point. Then, the obtained information is shared within a group. PSO algorithm is one of the most popularly used algorithms in metaheuristic research due to its ease of implementation, flexibility, and high performance. PSO also supports multiple types of hyper-parameters.

Several well-recognized Python libraries were employed to implement HPO. Hyperopt [4] (v.0.2.7) was used to implement BO-TPE. SMAC3 library [22] (v.1.4.0), which is the only Python library providing BO with random forest surrogate model at the time of writing, was adopted to implement SMAC. For PSO, Optunity [8] (v.1.1.1) was used to implement the algorithm.

4 Results and Discussion

Table 2 shows the ratio of instances used for training and testing per code smell. These two datasets have two major differences. First, Fontana et al.'s code smell dataset contains 420 instances, which is four times smaller than the `MLCQ` dataset for all types of code smells. Second, the positive and negative ratio of the testing set is heavily imbalanced. Obviously, for *feature envy*, there are only 3% of smelly instances in the testing set, while the training set contains more than 30%. For the rest code smells, the testing set contains approximately 20% less number of smelly instances.

Table 2. No. of training and testing instances, No. of negative and positive instance and the ratio of negative and positive per code smell.

Code smell	Dataset type	Dataset	No. of instance	Negative ratio	Positive ratio
Data class	training set	Fontana et al	420	0.67	0.33
	testing set	MLCQ	2154	0.87	0.13
Feature envy	training set	Fontana et al	420	0.67	0.33
	testing set	MLCQ	2035	0.97	0.03
God class	training set	Fontana et al	420	0.67	0.33
	testing set	MLCQ	2122	0.89	0.11
Long method	training set	Fontana et al	420	0.67	0.33
	testing set	MLCQ	2080	0.88	0.12

In the experiments, the performance evaluation results obtained from the validation set and the testing set are compared in three aspects, i.e., precision, recall, and f-score. When a classifier achieves a relatively close or higher prediction score than a validation score, the machine learning classifier is general enough to identify code smells on a software project developed by a newer version of Java. In other words, it is feasible and applicable for the machine learning-based code smell detection to adapt when there is a change in the data distribution. Otherwise, a new dataset containing the characteristics of modern Java programming language is required. Other strategies used to gain a better model adaptation are also needed. Figures 1–3 illustrates the evaluation results.

In terms of recall, in most cases, the performance of the machine learning classifiers was extremely decreased, except for *feature envy* as shown in Fig. 1. The recall produced by decision tree and random forest were dropped by ≈ 30% and ≈ 60% when classifying *data class* and *god class*, respectively. In the case of *long method*, both decision tree and random forest hardly predicted correct results on the target domain. Unlike *feature envy*, the decision tree classifier achieved slightly lower recall (≈ 5%) on the testing set. The random forest classifier produced ≈ 16% lower recall over the testing set than the validation set.

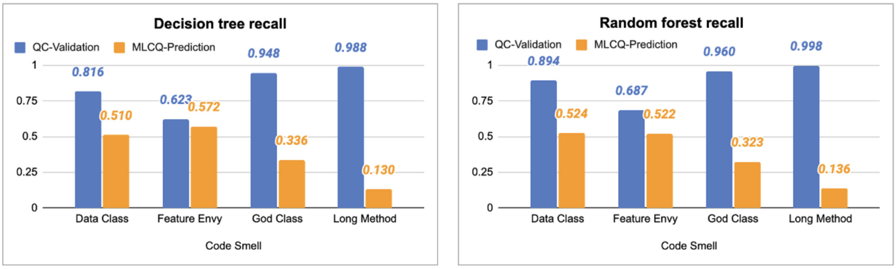


Fig. 1. Recall produced by decision tree (left) and random forest (right)

Figure 2 shows the precision results for code smell detection. It is clear that in most cases the classifiers achieved lower precision when facing the new target domain, except for *long method*. In this case, both algorithms achieved relatively close prediction and validation precision scores. Considering the machine learning algorithm, random forest classifiers outperformed decision tree in all cases. The random forest also achieved the highest precision at 100% when detecting *long method* in the `MLCQ` dataset.

In the case of f-score (see Fig. 3), domain shifting in code smell datasets makes a substantial difference in the validation and prediction scores. When facing a different target domain, the random forest slightly outperformed the decision tree and achieved the best f-score at 42.7% for *god class* classification.

Our experiments confirm that the machine learning-based code smell classifiers trained on a dataset built upon the `Qualitas Corpus` benchmark adapt poorly to domain shifting. The classifiers achieved lower performance results when they were trying to detect code smells in a modern software project. To improve the performance, HPO algorithms were applied to the machine learning models. The experimental results are summarized in Table 3–5.

In most of the cases, the optimized random forest classifiers produced higher recalls than their default settings. Table 3 indicates that random forest with PSO can achieve a better recall by ≈ 5% when classifying *feature envy*. For *long method*, PSO assists both algorithms to achieve slightly higher recalls. The random forest classifier applying SMAC can also achieve a slightly better recall when identifying *god class*.

Considering the precision (see Table 4), decision tree with SMAC produced higher precision results when classifying *god class* and *feature envy*. In case of the optimized random forest classifier, SMAC is the only algorithm that can assist the random forest classifier to achieve a slightly better result when classifying *feature envy*.

Table 5 compares the f-score obtained from the classifiers with the default configurations the proposed HPO techniques. SMAC can boost the performance by ≈ 1% and ≈ 2% for *god class* and *feature envy*, respectively. PSO can slightly improve both decision tree and random forest when classifying *long method*. However, none of the classifiers can exceed 45% of the f-score.

Table 3. Recall produced by the classifiers with default configuration and the proposed HPO techniques.

Code smell	Decision tree				Random forest			
	Default	BO-TPE	SMAC	PSO	Default	BO-TPE	SMAC	PSO
Data Class	0.51021	**0.51831**	0.46761	0.50106	**0.52359**	0.48486	0.50915	0.49225
Feature Envy	**0.57188**	0.55625	0.54375	0.53906	0.52188	0.56094	0.55313	**0.57188**
God Class	**0.33562**	0.33176	0.28884	0.30043	0.32275	0.33691	**0.33777**	0.33605
Long Method	0.13008	0.13008	0.13008	**0.13252**	0.13618	0.13049	0.13374	**0.13699**

We can conclude that the machine learning-based code smell classifiers trained over a dataset built upon the `Qualitas Corpus` benchmark can not produce desirable results when identifying code smells in a modern software project. The advancement of programming languages leads to changes in software properties and various aspects of software, which affect the performance of the classifiers. The experimental results confirm that the machine learning-based code smell classifiers adapt poorly to domain shifting. Although the hyper-parameter optimization techniques can slightly improve the prediction results in some cases, it is insufficient for the machine learning-based code smell classifiers

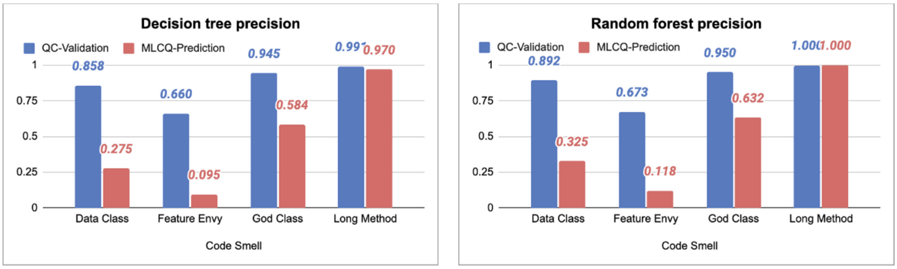


Fig. 2. Precision produced by decision tree (left) and random forest (right)

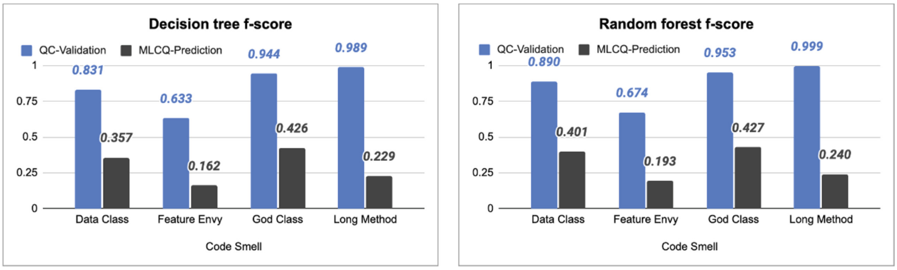


Fig. 3. F-score produced by decision tree (left) and random forest (right)

Table 4. Precision produced by the classifiers with default configuration and the proposed HPO techniques.

Code smell	Decision tree				Random forest			
	Default	BO-TPE	SMAC	PSO	Default	BO-TPE	SMAC	PSO
Data Class	**0.27473**	0.26738	0.25785	0.26127	**0.32547**	0.29867	0.30598	0.29844
Feature Envy	0.09473	0.10510	**0.10873**	0.10813	0.11839	0.11704	**0.11954**	0.11581
God Class	0.58377	0.60510	**0.62915**	0.62012	**0.63201**	0.61174	0.61159	0.61148
Long Method	**0.96970**	0.96970	0.96970	0.96541	**1.00000**	0.99108	1.00000	0.97702

Table 5. F-score produced by the classifiers with default configuration and the proposed HPO techniques.

Code smell	Decision tree				Random forest			
	Default	BO-TPE	SMAC	PSO	Default	BO-TPE	SMAC	PSO
Data Class	**0.35702**	0.35082	0.32947	0.34188	**0.40136**	0.36941	0.38173	0.37137
Feature Envy	0.16246	0.17634	**0.18072**	0.17950	0.19298	0.19349	**0.19647**	0.19239
God Class	**0.42616**	0.42559	0.39214	0.40391	0.42726	0.43438	**0.43516**	0.43364
Long Method	0.22939	0.22939	0.22939	**0.23293**	0.23969	0.23059	0.23586	**0.24014**

to work on a different target domain. There is no significant difference in the performance results produced by the default-configured and optimized classifiers. The concluded experimental results already grant the answers to both of the main goals of this research.

Threats to Validity. Note that there are subjectivity issues in both code metrics and code smells. Although we selected the same set of metrics from both the Fontana et al.'s code smell dataset and the `MLCQ` dataset, there are insignificant differences in the metric definitions due to different code analysis tools used. The selected code metrics might not represent the characteristics of the source code and the code smells. Moreover, the code smell instances of the two datasets were labeled by disparate groups of software developers. The Fontana et al.'s code smell dataset was labelled by master students whose background and experience are inconclusive. In case of the `MLCQ` dataset, the instances were labelled by a group of experienced professional software developers. The researchers also include the background experience and survey results of the developers.

5 Conclusions

This research explores the feasibility and applicability of domain adaptation in machine learning-based code smell detection. The experiments ran supervised machine learning classifiers well-trained on the earlier versions of open-source Java projects to identify code smells in the modern Java open-source projects without carrying out extensive feature engineering. Three hyper-parameter optimization techniques, i.e., Bayesian optimization using random forest (SMAC), Bayesian optimization using tree-structured parzen estimator (BO-TPE), and particle swarm optimization (PSO), were also applied to the machine learning-based code smell classifiers to improve their performance. The experiments measured precision, recall, and f-score.

Fontana et al.'s code smell dataset, which is a dataset built upon the `Qualitas Corpus` benchmark, was employed as a training set. The `MLCQ` dataset, which is a dataset built upon the newer-version Java projects, was used as a test set. Most of the open-source Java projects in the `Qualitas Corpus` benchmark are of Java 5, which is much older than Java versions used in the projects included in the `MLCQ` dataset.

The experimental results indicate that the machine learning classifiers trained over Fontana et al.'s code smell dataset incompetently detect four types of code smells in the `MLCQ` dataset. In other words, the classifiers adapt poorly to domain shifting in code smell detection problems. Although the hyper-parameter optimization techniques were adopted, the performance results of the enhanced machine learning classifiers are smell-specific. For example, random forest with PSO produced a higher recall and f-score when classifying *long method*. Decision tree with SMAC achieved higher precision results when classifying *god class* and *feature envy*.

While the interest in using machine learning-based techniques for code smell detection is inclined, creating more choices of high quality training datasets is a non-trivial and exhaustive task. Programming languages are also evolved overtime. It would be impractical to continue laboriously working on code smell dataset construction to keep up with the advancement of programming languages. This research opens a new opportunity for using a more sophisticated domain adaption technique to mitigate the limitations and supports the research community to mature more studies in the fields. To support the advancement in the research community, we made the dataset and source code publicly available at https://github.com/Peeradon06/Domain-Adaptation-ML-Based-Code-Smell-Detection.

Acknowledgements. This research was partially supported by the Department of Computer Science, College of Computing and the Graduate School, Khon Kaen University, Khon Kaen, Thailand.

References

1. Arcelli Fontana, F., Mäntylä, M.V., Zanoni, M., Marino, A.: Comparing and experimenting machine learning techniques for code smell detection. Empir. Softw. Eng. **21**(3), 1143–1191 (2015). https://doi.org/10.1007/s10664-015-9378-4
2. Azeem, M.I., Palomba, F., Shi, L., Wang, Q.: Machine learning techniques for code smell detection: a systematic literature review and meta-analysis. Inf. Softw. Technol. **108**, 115–138 (2019). https://doi.org/10.1016/j.infsof.2018.12.009
3. Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K. (eds.) Advances in Neural Information Processing Systems, vol. 24. Curran Associates, Inc. (2011)
4. Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., Cox, D.D.: Hyperopt: a python library for model selection and hyperparameter optimization. Comput. Sci. Discov. **8**(1), 014008 (2015). https://doi.org/10.1088/1749-4699/8/1/014008
5. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324
6. Breiman, L.: Classification and Regression Trees. Routledge, New York (2017). https://doi.org/10.1201/9781315139470
7. Caram, F.L., Rodrigues, B.R.D.O., Campanelli, A.S., Parreiras, F.S.: Machine learning techniques for code smells detection: a systematic mapping study. Int. J. Software Eng. Knowl. Eng. **29**, 285–316 (2019). https://doi.org/10.1142/S021819401950013X
8. Claesen, M., Simm, J., Popovic, D., Moreau, Y., De Moor, B.: Easy hyperparameter search using optunity. http://arxiv.org/abs/1412.1114. Accessed 15 Jan 2023
9. Dewangan, S., Rao, R.S., Mishra, A., Gupta, M.: A novel approach for code smell detection: an empirical study. IEEE Access **9**, 162869–162883 (2021)
10. Di Nucci, D., Palomba, F., Tamburri, D.A., Serebrenik, A., De Lucia, A.: Detecting code smells using machine learning techniques: are we there yet? In: 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER), pp. 612–621. IEEE (2018). https://doi.org/10.1109/SANER.2018.8330266
11. Fontana, F.A., Zanoni, M.: Code smell severity classification using machine learning techniques. Knowl.-Based Syst. **128**, 43–58 (2017)
12. Fontana, F.A., Zanoni, M., Marino, A., Mantyla, M.V.: Code smell detection: towards a machine learning-based approach. In: 2013 IEEE International Conference on Software Maintenance, pp. 396–399. IEEE (2013). https://doi.org/10.1109/ICSM.2013.56, http://ieeexplore.ieee.org/document/6676916/
13. Fowler, M.: Refactoring: Improving the Design of Existing Code. Addison-Wesley, Boston, MA, USA (1999)
14. Hall, T., Zhang, M., Bowes, D., Sun, Y.: Some code smells have a significant but small effect on faults. ACM Trans. Softw. Eng. Methodol. **23**(4), 1–39 (2014). https://doi.org/10.1145/2629648
15. Haque, M.S., Carver, J., Atkison, T.: Causes, impacts, and detection approaches of code smell: a survey. In: Proceedings of the ACMSE 2018 Conference, pp. 1–8. ACM, Richmond Kentucky (2018). https://doi.org/10.1145/3190645.3190697
16. Hasantha, C.: A systematic review of code smell detection approaches. J. Adv. Softw. Eng. Test. **4**(1), 1–9 (2021). https://doi.org/10.5281/zenodo.4738772
17. Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: Coello, C.A.C. (ed.) Learning and Intelligent

Optimization, pp. 507–523. Springer, Berlin Heidelberg, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25566-3_40
18. Kaur, A.: A systematic literature review on empirical analysis of the relationship between code smells and software quality attributes. Arch. Comput. Methods Eng. **27**(4), 1267–1296 (2020). https://doi.org/10.1007/s11831-019-09348-6
19. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN 1995 - International Conference on Neural Networks, vol. 4, pp. 1942–1948. IEEE, Perth, WA, Australia (1995). https://doi.org/10.1109/ICNN.1995.488968
20. Kovačević, A., et al.: Automatic detection of long method and god class code smells through neural source code embeddings. Expert Syst. Appl. **204**, 117607 (2022)
21. Lewowski, T., Madeyski, L.: How far are we from reproducible research on code smell detection? a systematic literature review. Inf. Softw. Technol. **144**, 106783 (2022). https://doi.org/10.1016/j.infsof.2021.106783, https://linkinghub.elsevier.com/retrieve/pii/S095058492100224X
22. Lindauer, M., et al.: SMAC3: A versatile bayesian optimization package for hyper-parameter optimization. J. Mach. Learn. Res. **23**, 1–9 (2021). https://doi.org/10.48550/ARXIV.2109.09831
23. Madeyski, L., Lewowski, T.: MLCQ: industry-relevant code smell data set. In: Proceedings of the Evaluation and Assessment in Software Engineering, pp. 342–347. ACM (2020). https://doi.org/10.1145/3383219.3383264
24. Madeyski, L., Lewowski, T.: Detecting code smells using industry-relevant data. Inf. Softw. Technol. **155**, 107112 (2023). https://doi.org/10.1016/j.infsof.2022.107112, https://linkinghub.elsevier.com/retrieve/pii/S095058492200221X
25. Mhawish, M.Y., Gupta, M.: Predicting code smells and analysis of predictions: using machine learning techniques and software metrics. J. Comput. Sci. Technol. **35**(6), 1428–1445 (2020). https://doi.org/10.1007/s11390-020-0323-7
26. Palomba, F., Bavota, G., Penta, M.D., Fasano, F., Oliveto, R., Lucia, A.D.: On the diffuseness and the impact on maintainability of code smells: a large scale empirical investigation. Empir. Softw. Eng. **23**(3), 1188–1221 (2017). https://doi.org/10.1007/s10664-017-9535-z
27. Pedregosa, F., Varoquaux, G., Gramfort, A., et al.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
28. Pereira dos Reis, J., Brito e Abreu, F., de Figueiredo Carneiro, G., Anslow, C.: Code smells detection and visualization: a systematic literature review. Arch. Comput. Methods Eng. **29**(1), 47–94 (2021). https://doi.org/10.1007/s11831-021-09566-x
29. Santos, J.A.M., Rocha-Junior, J.B., Prates, L.C.L., Nascimento, R.S.D., Freitas, M.F., Mendonça, M.G.D.: A systematic review on the code smell effect. J. Syst. Softw. **144**, 450–477 (2018). https://doi.org/10.1016/j.jss.2018.07.035
30. Understand by Scitools. https://www.scitools.com/. Accessed 31 May 2023)
31. Sharma, T., Efstathiou, V., Louridas, P., Spinellis, D.: Code smell detection by deep direct-learning and transfer-learning. J. Syst. Softw. **176**, 110936 (2021). https://doi.org/10.1016/j.jss.2021.110936, https://www.sciencedirect.com/science/article/pii/S0164121221000339
32. Shen, L., Liu, W., Chen, X., Gu, Q., Liu, X.: Improving machine learning-based code smell detection via hyper-parameter optimization. In: 2020 27th Asia-Pacific Software Engineering Conference, pp. 276–285. Singapore, Singapore (2020)
33. Snoek, J., Larochelle, H., Adams, R.P.: Practical bayesian optimization of machine learning algorithms. In: Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.) Advances in Neural Information Processing Systems, vol. 25. Curran Associates, Inc. (2012)

34. Sobrinho, E.V.D.P., De Lucia, A., Maia, M.D.A.: A systematic literature review on bad smells-5w's: which, when, what, who, where. IEEE Trans. Softw. Eng. **47**(1), 17–66 (2021). https://doi.org/10.1109/TSE.2018.2880977
35. Soomlek, C., van Rijn, J.N., Bonsangue, M.M.: Automatic human-like detection of code smells. In: Soares, C., Torgo, L. (eds.) DS 2021. LNCS (LNAI), vol. 12986, pp. 19–28. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88942-5_2
36. Sukkasem, P., Soomlek, C.: Enhance machine learning-based code smell detection through hyper-parameter optimization. In: 20th International Joint Conference on Computer Science and Software Engineering (JCSSE). IEEE (2023)
37. Tempero, E., et al.: The qualitas corpus: a curated collection of java code for empirical studies. In: 2010 Asia Pacific Software Engineering Conference, pp. 336–345. IEEE (2010). https://doi.org/10.1109/APSEC.2010.46, http://ieeexplore.ieee.org/document/5693210/
38. Vatanapakorn, N., Soomlek, C., Seresangtakul, P.: Python code smell detection using machine learning. In: 2022 26th International Computer Science and Engineering Conference (ICSEC), pp. 128–133. IEEE (2022)
39. Virmajoki, J., Knutas, A., Kasurinen, J.: Detecting code smells with AI: a prototype study. In: 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), pp. 1393–1398 (2022). https://doi.org/10.23919/MIPRO55190.2022.9803727
40. Wang, T., Golubev, Y., Smirnov, O., Li, J., Bryksin, T., Ahmed, I.: Pynose: a test smell detector for python. In: 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 593–605. IEEE (2021)
41. Zakeri-Nasrabadi, M., Parsa, S., Esmaili, E., Palomba, F.: A systematic literature review on the code smells datasets and validation mechanisms. ACM Comput. Surv. **55**, 1–48 (2023). https://doi.org/10.1145/3596908
42. Zhang, H., Cruz, L., Van Deursen, A.: Code smells for machine learning applications. In: Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI, pp. 217–228 (2022)

Web Usage Mining for Determining a Website's Usage Pattern: A Case Study of Government Website

Panunsiya Rawira and Vatcharaporn Esichaikul(✉)

Asian Institute of Technology, Pathum Thani 12120, Thailand
`{st122423,vatchara}@ait.asia`

Abstract. Web usage mining is a crucial research area that aims to uncover user behavior patterns from web log data because web usage mining can be used to analyze a website's usage. This study examined web usage mining to discover online users' usage patterns and used the results to redesign and improve the government website. This study aims to help online customers obtain a better experience. A dataset was collected from the Metropolitan Electricity Authority (MEA) website. Various algorithms, including association rule mining (Apriori and Frequent Pattern-Growth (FP-Growth)) and sequential pattern mining (Generalized Sequential Pattern (GSP) and Prefix-Span), were used to mine the user usage data. The first 30 frequently visited patterns of web usage mining results were selected for analysis and to develop a prototype. It revealed that most pages were accessed directly. Moreover, most users were interested in alternative energy information, the Fuel Adjustment Charge (at the given time) or the FT rate, the power outage announcement page, and reducing electric expenses information. Furthermore, the analysis indicated that customers were interested in making online processes, such as using the contact-us feature, downloading forms, and calculating their electric expenses. A survey was conducted with 30 participants and the results were compared with the usage data from the weblog. Welch's t-test was utilized to evaluate the prototype. The findings indicated that the newly implemented website, based on the user usage patterns, was more effective and reduced the user's usage time.

Keywords: Web usage mining · Machine learning · E-Government

1 Introduction

Web usage mining is a branch of web mining. It is the uncovering method to extract user entry patterns from a web server [3]. Web usage mining can discover data from web server logs files to investigate and explore users' insight information and patterns [8,18,19]. The process of web usage mining consists of data collection, data pre-processing, pattern discovery, and pattern analysis [8]. It is

---

Supported by AIT and MEA.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 88–100, 2023.
https://doi.org/10.1007/978-981-99-7969-1_7

a process to discover valuable and hidden data. Some examples of web usage mining are website traffic analysis, personalization, fraud detection, and website redesign [16].

Many organizations have digitized their services and knowledge by creating online channels. They deliver information and communication technology to give locals and businesses more access to their knowledge and services. They also designed and improved user interfaces (UI), search engines, navigation, and other features to increase the satisfaction of their visitors and customers [6]. Therefore, these transformations have generated vast usage data that contain valuable information [7]. This data can be analyzed to find helpful information that an organization can use to enhance its service quality and expand participation opportunities [10]. Furthermore, it can help businesses make decisions and maximize profitability [7].

The Metropolitan Electricity Authority (MEA) is a Thai state enterprise whose primary responsibility is to provide electric service to customers in Bangkok, Samut Prakan, and Nonthaburi. The organization has 18 office branches (excluding sub-branches). MEA provides its services through various channels to communicate and apprehend its customers. The official website is one of the applications among other MEA applications. It is located at "https://www.mea.or.th." The website primarily offers news and knowledge to its customers and stakeholders. It is displayed in both English and Thai languages. This site has eight main menus: Home, About Us, About electricity bill, Auction, Work, Services and Download, Electrical Safety, and Contact Us. Each main menu has several sub-menus. The landing page highlights news and activities. It also provides shortcut links to other MEA businesses. Thus, this site can be determined as a portal website since it spreads news and information and links to other MEA businesses and applications. Moreover, it also provides an electric calculation page which allows users to calculate the approximate electrical bills. However, this site contains a lot of information that can make it difficult for customers to find the needed information and cause customer dissatisfaction. Therefore, this research discusses web usage mining and machine learning techniques that can be used to discover users' usage patterns. This technique provides valuable information and reveals the user's insight and interest. Those companies can use it to redesign their websites, which helps online customers obtain a better experience.

2 Theoretical Background and Related Work

2.1 Web Usage Mining

Web usage mining is defined as a process to discover valuable and hidden data. It utilizes techniques from data mining. It explores the usage patterns using web log files to extract data. This approach aimed to extract and understand the customers' usage [4]. This approach involved extracting knowledge from server log files [15]. Web usage mining is a part of web mining that involves analyzing user behavior on websites.

2.2 Applications of Web Usage Mining

Web usage mining is widely used in different applications. First, web usage mining can personalize web content for user experience [5]. The process involves delivering web pages to individual users. This can be analyzed from user characteristics or preferences [17]. Second, web usage mining can prefetch and cache to improve web server performance. In addition, it helps system admins to plan and develop the proper prefetching and caching strategies. Third, the most widely used in this field are recommendation systems. It is a common application that recommends interesting links to the user. For example, an e-commerce website suggests products according to the forecast from user behavior [5]. Fourth, some studies used web usage mining in the security area. It is used to detect, prevent, and predict cyber-attacks. Fifth, web usage mining can enhance customer relationship management (CRM). For instance, the guide focuses on scenarios unique to businesses, like client recruitment, customer retention, cross-selling, and customer departure. Sixth, web usage mining is used to support design and modification. Usability is one of the most crucial factors to consider when designing and implementing websites. Web usage mining provides feedback on user behavior that is proper for website designers to improve web usability. It gives valuable information to support redesign decisions [17]. The results from web usage mining provide guidelines for improving the website's design according to the users' perspective [5]

2.3 Related Work

Web Usage Mining with Different Algorithms. Kumar & Rukmani [9] studied web usage mining specifically in the pattern discovery step. They analyzed the usage patterns using an Apriori and an FP growth by comparing the performance of memory usage and time usage in each algorithm. There are two techniques that can help mine frequent item sets for Boolean association rules. The Apriori algorithm is a popular choice for analyzing association rules. It was made to operate on databases with transactions. On the other hand, FP- growth consists of two phases. The first phase is constructing an FP tree, and the second is recursive Research of the FP tree and outputs all frequent patterns. They used a weblog file to analyze. As a result, the benefit of an Apriori is that it is comfortable to implement but costly to handle many candidate sets and scan the database frequently to match a large group of candidates by pattern. Thus, an FP-growth was present to overcome this drawback using an efficient FP-tree-based mining method. It is an order of volume quicker than other algorithms. However, one of the weaknesses of the FP-growth algorithm is the lack of suitable candidate-generating methods for massive datasets.

Charliepaul & Gnanadurai [2] compared the K-means and Apriori algorithms through an analysis of a web server log file. The Apriori algorithm is a traditional algorithm used to understand association rules. One of its benefits is that it is straightforward to implement. However, its drawback is that it requires multiple database scans and uses high memory. The algorithm is efficient in finding all frequent item sets. On the other hand, the K-means algorithm is a straightforward clustering technique that groups the same objects into existing categories. It is typically used in medical imaging and biometrics. Also, it is considered an unsupervised data mining algorithm. Its benefit is that it is faster than hierarchical clustering when k is small. However, its drawback is that predicting the K-value is challenging, and different initial partitions can result in different final clusters. In summary, the Apriori algorithm finds frequent item sets efficiently, while the K-means algorithm is faster when k is small. However, predicting the K-value is challenging; different initial sections can result in different final clusters. These two algorithms have unique benefits and drawbacks, and choosing the appropriate algorithm depends on the project's specific needs.

Han et al. [13] introduced a sequence pattern mining algorithm. They compared the performance of three algorithms, including Prefix-Span, Free-Span, and GSP algorithms. The methods are all put into practice with Microsoft Visual C++ 6.0. Various datasets are used to conduct experiments. This research generated the synthetic datasets using standard procedures. The result found that Prefix-Span and Free-Span are faster than GSP, and Prefix-Span is also quicker than Free-Span. It showed that Prefix-Span outperformed both the GSP and Free-Span algorithms.

Web Usage Mining to Improve the Website. Resul et al. [14] used web usage mining to help admins and designers improve the systems. They argued that the number of visitors is a measure of the effectiveness and quality of a university. This study identified system errors, system corrupts, and broken links. They used log data from the University of Fira, which contains 83 d (March 30th to June 20th, 2007). A weblog data type was a Common Log file format (CLF). A Nihuo Web Log Analyzer program was used to analyze it. They divided web usage mining into three steps: data preprocessing, pattern discovery, and pattern analysis. The result found that the most error in the data was an error status 404 not found.

Nina et al. [11] focused on the data preparation and pattern discovery process. This technique helps website owners understand how visitors use their sites effectively. This study used a web server logs file from the Sust Tube website. They proposed algorithms for the data preparation process. It consisted of four sub-steps. In data cleaning, the authors removed the records containing the image and sound files. In user identification, they converted the visitors' IP addresses by considering the extension of each visitor's domain. Then, they identify users using a random ID that the web server assigns to the web browser or cookies, which happens when the visitor's connected to a website for the first time. They also used cache busting to handle the cache. The last sub-step in data preparation is session identification. They used session timeout and time intervals between consecutive log entries to identify the session. After the data preparation step, this data goes to a pattern discovery step. Admin can find the most active countries visited by converting their IP address to its domain. They used path analysis to understand which pages visitors liked the most and how long they spent on a website. They also used an answer page link to find the most visited page. As a result, most visitors left after visiting four pages. This result suggested that vital information should be contained within four pages.

Patil & Khandagale [12] discussed the explosive growth of web traffic and users' navigational problems. It proposed a method for developers to realize authentic and anticipated usage behavior. The data was collected from http://www.southtexasshooting.org. This log file contained 98,938 log records from the year 2016. They modified the Generalized Sequential Pattern (GSP) and Prefix-Span. The result is used to update web links in an automated manner. They found it can improve the effectiveness and efficiency of user tasks and reduce developer time, making it useful for developers and users.

Website Usability. Ugras et al. [19] reviewed website usability studies conducted over ten years to identify research trends in website usability issues. They used the systematic review analysis based on their six research questions. The study included a total of 199 studies that met the inclusion criteria. The top three content of websites were education, business, and health. They found navigation is the most frequently addressed usability issue, excluding general usability issues followed by methodological issues, user interface (UI) design, and accessibility. Most research employed user-based Usability Evaluation Methods (UEMs). In addition, questionnaires and usability testing are the most commonly used methods for gathering user feedback. However, there was limited consideration of the user experience for special user groups (disabled users).

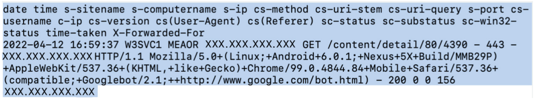
Belanche et al. [1] investigated the relationship between website usability and consumer behavior, specifically satisfaction and intention to use a website. It also examines how consumer perceptions of risk may influence the impact of website usability. They analyzed the literature reviews. Research has shown that website usability impacts user satisfaction, affecting their intention to use the website. Surprisingly, usability does not directly impact the intention to use but rather indirectly through user satisfaction. Additionally, the user's perceived risk influences the impact of usability on user satisfaction.

3 Methodology

This study follows four primary processes involved in web usage mining. According to the research objectives, the first objective is to analyze users' access patterns through web usage mining using a server log file. The second objective is to conduct pattern discovery using four mining algorithms: Apriori, FP-Growth, GSP, and Prefix-Span. The third objective is to design and implement the website involves four subtasks: designing the website, developing the prototype, conducting testing, and comparing the average usage time between the original website and the improved website using Welch's T-test.

3.1 Data Collection

The data was collected from an MEA website. The server log file collected data in the Internet Information Services (IIS) log file format. The total log durations were from February 27th to March 12th, 2023. The web server log files are divided into four durations: two weekdays and two weekends. The first weekdays are from February 27th to March 3rd, 2023. The second weekdays are from March 6th to 10th, 2023. Also, there are two weekend files: one from March 4th to 5th, 2023, and the other from March 11th to 12th, 2023. Figure 1 shows an example of a web server log file.



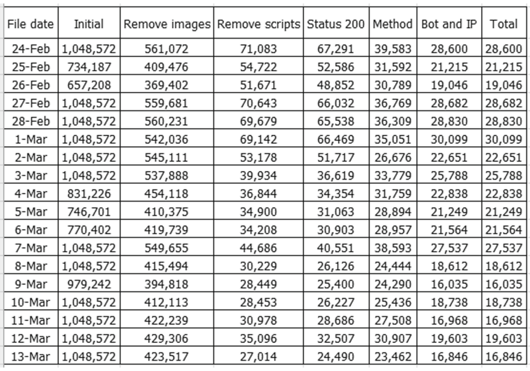
```
date time s-sitename s-computername s-ip cs-method cs-uri-stem cs-uri-query s-port cs-
username c-ip cs-version cs(User-Agent) cs(Referer) sc-status sc-substatus sc-win32-
status time-taken X-Forwarded-For
2022-04-12 16:59:37 W3SVC1 MEAOR XXX.XXX.XXX.XXX GET /content/detail/80/4390 - 443 -
XXX.XXX.XXX.XXX HTTP/1.1 Mozilla/5.0+(Linux;+Android+6.0.1;+Nexus+5X+Build/MMB29P)
+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/99.0.4844.84+Mobile+Safari/537.36+
(compatible;+Googlebot/2.1;++http://www.google.com/bot.html) - 200 0 0 156
XXX.XXX.XXX.XXX
```

Fig. 1. An example of a web server log file.

3.2 Data Pre-processing

Data Cleaning. A web log file records all activities that occur on a website. The data cleaning process removes any unnecessary items from the log data. There are various approaches to cleaning data. First, any executions or downloads not from the user requests must be removed. For example, images (JPG, GIF), scripts, and other videos executed on server log data need to be removed. All these types can be easily found and cleaned. Second, by examining the HTTP status code return, any unavailable or incorrect web pages on the server must be deleted. Similarly, the status code of a successful response ranges from 200–299, while other status codes are removed. Third, records of crawlers or bots must be excluded, as this study only focuses on human actions. Most crawlers declare themselves as agents, making it easy to identify them in the record. Figure 2 shows the result from the cleaning step.

User Identification. This step aims to determine the unique user who accesses the website, which can be done by identifying the IP address. There are several ways to consider a different user. First, it can be considered the client type. Any field with different parameters but the same IP address can be identified as another user. This field can assume it is another user [20]. Second, it can be considered based on site topology. If a person requests a page inaccessible from the website they just viewed, it has the same IP address. This request comes from a separate user. Third, it can consider using cookies. It is a small variable that keeps parameter values on the client side. The web server creates cookies from first-time users accessing the web and sends them to the client. It records any requests made through the same web server. These cookies help the web server identify the user and deliver the requested page without recreating it.



File date	Initial	Remove images	Remove scripts	Status 200	Method	Bot and IP	Total
24-Feb	1,048,572	561,072	71,083	67,291	39,583	28,600	28,600
25-Feb	734,187	409,476	54,722	52,586	31,592	21,215	21,215
26-Feb	657,208	369,402	51,671	48,852	30,789	19,046	19,046
27-Feb	1,048,572	559,681	70,643	66,032	36,769	28,682	28,682
28-Feb	1,048,572	560,231	69,679	65,538	36,309	28,830	28,830
1-Mar	1,048,572	542,036	69,142	66,469	35,051	30,099	30,099
2-Mar	1,048,572	545,111	53,178	51,717	26,676	22,651	22,651
3-Mar	1,048,572	537,888	39,934	36,619	33,779	25,788	25,788
4-Mar	831,226	454,118	36,844	34,354	31,759	22,838	22,838
5-Mar	746,701	410,375	34,900	31,063	28,894	21,249	21,249
6-Mar	770,402	419,739	34,208	30,903	28,957	21,564	21,564
7-Mar	1,048,572	549,655	44,686	40,551	38,593	27,537	27,537
8-Mar	1,048,572	415,494	30,229	26,126	24,444	18,612	18,612
9-Mar	979,242	394,818	28,449	25,400	24,290	16,035	16,035
10-Mar	1,048,572	412,113	28,453	26,227	25,436	18,738	18,738
11-Mar	1,048,572	422,239	30,978	28,686	27,508	16,968	16,968
12-Mar	1,048,572	429,306	35,096	32,507	30,907	19,603	19,603
13-Mar	1,048,572	423,517	27,014	24,490	23,462	16,846	16,846

Fig. 2. The result of data cleaning step

Session Identification. This method seeks to identify the user's access pattern and frequently used path [21]. It can be categorized into two classes. There are two types of heuristics: time-oriented and navigation-oriented heuristics. Time-oriented divided user activities into sessions using temporal information, whereas the navigation-oriented used the navigation behavior to identify the sessions.

Path Completion. This step aims to find and complete the page's missing entries. If a client requests a page unrelated to the previous page they requested, the request's source page can be determined using the referrer log. If the page appears in the user's recent request history, it indicates that the user navigated back using the "back" button [3].

3.3 Pattern Discovery

After pre-processing the data, this step discovers the user's pattern. Various techniques have been studied in previous works. According to the research objective, multiple algorithms have been selected to analyze results better. Four algorithms for mining frequent patterns are Apriori, FP-Growth, GSP, and Prefix-Span.

3.4 Pattern Analysis

There are several approaches to perform this step. This research focused on Online analytical processing (OLAP) analysis. The OLAP operations were performed in Python programming.

3.5 Website Design and Implementation

A website can be improved and redesigned using the knowledge obtained via web usage mining. The results reveal the frequently visited patterns which show the target page. This step consists of four sub-tasks: website design, developing a prototype, testing the prototype, and evaluating the results. In the first step, a use case and sequence diagram were created in this step. The prototype was created following the diagram. The third step is to conduct surveys. Finally, a Welch's t-test is used to perform a statistical analysis to analyze the result.

4 Analysis and Results

Web Usage Mining Results. The mining results indicated three pattern formats. First, the patterns started with a homepage and were followed by other pages. Second, the patterns contained a direct link to each page. Third, the patterns made by the system or the backend request pages. The pattern containing a direct link is the most common pattern from the mining result. In addition, this study does not concentrate on the homepage and collected data during the employment periods, which may affect the analysis results. The result revealed that the weekday and weekend files showed different results. On weekdays, the most common patterns were direct visits to the alternative energy page, FT page, residential electric rate, contact us, self-electric calculation, and power outage announcement. On weekends, the most common patterns were direct visits to power outage announcements, government electric assistance news, the FT page, alternative energy, residential electric rate, and self-electric calculation.

Furthermore, the study used four different algorithms (Apriori, FP-growth, GSP, and Prefix-Span) to analyze the patterns of website visits during weekdays and weekends. Each algorithm had specific minimum support values and thresholds that were defined. Both Apriori and FP-Growth defined the minimum support value as 0.003 and the minimum threshold as 0.05. The results indicated that 36 itemsets in the weekday and 27 in the weekend files met the rules. For the GSP and Prefix-Span algorithms, with a minimum support value of 30, the study found 162 sequences on weekdays and 61 on weekends that met the rules. The comparison between the four algorithms showed that they obtained similar patterns but different support numbers since they used different approaches to mine the usage patterns. However, the results of the patterns and support numbers are similar when compared to the same algorithm's type.

Redesigned Website Implementation. A new website was developed based on the results of web usage mining. For this study, four algorithms were considered. The preferred algorithm was the sequence pattern mining algorithm, which considers the order and timing of the items in the sequences. In addition, previous research has shown that Prefix-Span is more effective than the GSP algorithm, so the results were obtained using the Prefix-Span algorithm. This study combined the results from the weekday and weekend files to redesign the website and selected the top 30 frequent patterns, excluding the homepage and employment information. Similar content was categorized in the same menu. This research focused on residential customers. They can view the web, download files, submit forms, and make electrical calculations. This research considered placing

Table 1. A summary of the results of a weekday file.

Pattern	Apriori	FP-Growth	GSP	Prefix-Span
(‘Homepage’)	0.055877909	0.055877909	2552	2552
(‘Employment news’)	0.037288432	0.037288432	1703	1703
(‘Alternative energy’)	0.026296775	0.026296775	1201	1201
(‘Homepage’, ‘Employment’)	0.023713078	0.023713078	1083	1083
(‘Employment’)	0.021063695	0.021063695	962	962
(‘FT’)	0.019158766	0.019158766	875	875
(‘Residential electric rate’)	0.015764927	0.015764927	720	720
(‘Contact us’, ‘Captcha’)	0.01569924	0.01569924	717	717
(‘Self-electric calculation’, ‘Calculator.php’)	0.014363601	0.014363601	656	656
(‘Power outage announcement’)	0.011736113	0.011736113	536	536

Table 2. A summary of the results of a weekend file

Pattern	Apriori	FP-Growth	GSP	Prefix-Span
(‘Homepage’)	0.054334011	0.054334011	855	855
(‘Power outage announcement’)	0.041115913	0.041115913	647	647
(‘Government electric assistance news’)	0.040861718	0.040861718	643	643
(‘Homepage’, ‘Employment’)	0.034951703	0.034951703	550	550
(‘FT’)	0.026245552	0.026245552	413	413
(‘Employment’)	0.02586426	0.02586426	407	407
(‘Alternative energy’)	0.025165226	0.025165226	396	396
(‘Employment news’)	0.022750381	0.022750381	358	358
(‘Residential electric rate’)	0.017348754	0.017348754	273	273
(‘Self-electric calculation’, ‘calculator.php’)	0.017221657	0.017221657	271	271

Note. The number of each record in the Apriori and FP-Growth columns indicates the approximate percentage of records in the dataset in which the itemset appears. While the number in the GSP and Prefix-Span columns shows the number of instances, a sequence appears in the web log data.

the most frequently visited pages close to each other. We also placed other frequently visited pages together, such as electric rate combined with self-electric calculation, form, and electric knowledge, such as information and infographics on reducing electric costs. Furthermore, information with short periods but highly impact users, such as government assistance news, electric campaigns, and employment information, were positioned in the most accessible location. This study also considered the organization’s branding redesign.

5 Prototype Development and Evaluation

5.1 Prototype Development

This research focused on residential customers. They can view the web, download files, submit forms, and make electrical calculations. This research considered placing the most frequently visited pages close to each other. We also placed other frequently visited pages together, such as electric rate combined with self-electric calculation, form download, and electric knowledge, such as information and infographics on reducing electric costs. Furthermore, information with short periods but highly impact users, such as government assistance news, electric campaigns, and employment information, were positioned in the most accessible location.

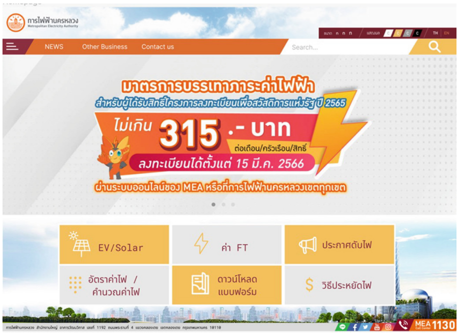

การไฟฟ้านครหลวง
Metropolitan Electricity Authority
TH
EN
NEWS
Other Business
Contact us
Search...
มาตรการบรรเทาภาระค่าไฟฟ้า
สำหรับผู้ได้รับสิทธิโครงการลงทะเบียนเพื่อสวัสดิการแห่งรัฐ ปี 2565
ไม่เกิน 315 .- บาท
ต่อเดือน/ครัวเรือน/สิทธิ์
ลงทะเบียนได้ตั้งแต่ 15 มี.ค. 2566
ผ่านระบบออนไลน์ของ MEA หรือที่การไฟฟ้านครหลวงเขตทุกเขต
EV/Solar
ค่า FT
ประกาศดับไฟ
อัตราค่าไฟ / คำนวณค่าไฟ
ดาวน์โหลด แบบฟอร์ม
วิธีประหยัดไฟ
MEA 1130

Fig. 3. A new design of a homepage.

5.2 Evaluation

After developing a website prototype, the surveys were used to gather the users' usage time finding pages. The sample size was collected from a web server log file on the original website and compared with 30 participants using the website prototype. Each participant was asked to find three pages resulting from web usage mining: the MEA EV or solar information page and the FT page. The hypothesis stated that the average usage time of users between the website with the original design and the website with the improved design has a significant difference. It found that the new website design created based on user usage patterns is more effective and can reduce the user's usage time.

6 Conclusion and Practical Implications

6.1 Conclusion

This study demonstrated the possible benefits of incorporating web usage mining into the design and development of government websites. First, we compared a weekday and a weekend file. The finding revealed different results between the files viewed on weekdays and weekends. The outcome indicates that people seek information that can improve their lives, with more serious information being accessed during weekdays and electric information affecting their weekend lives being viewed on weekends. People tend to be interested in alternative energy information and electric cost information. The result could indicate an increased awareness of new energy sources and the need to pay more attention to the electric cost information. Additionally, the analysis showed that customers desire to conduct online tasks. These findings indicated the potential for expanding online services in the future.

Second, this study used two association rule mining algorithms (Apriori and FP-growth) compared with two sequence pattern mining algorithms (GSP and Prefix-Span) to precisely the usage pattern result. The comparison between various algorithms revealed that they obtained similar patterns but different support numbers and total itemset. Therefore, these algorithms could be used to precisely the result. In addition, when choosing algorithms to mine the usage patterns, it is crucial to consider their type. Different algorithms use different approaches, which can impact the mining results.

Finally, we provided a new website prototype and compared the user's usage time between the original website and the improved version. A Welch's t-test was used to analyze the result since this approach does not assume equal sample sizes or variances between groups. The results indicated that the website, created based on user usage patterns, is more effective and can reduce the user's time spent on the website.

6.2 Practical Implications

According to the result, government websites can apply this approach to their organization. Various ways can be used in the future. It helps system administrators, website owners, website designers, and developers design the government website to comprehend how users interact with the services and information. The result of this research is based on users' insight patterns. They can use it to decide on delivery services and provide information to their customers by considering placing the most frequently visited pages close to each other. This study provides some suggestions that can be used to improve the government website to provide more helpful information that better meets customers' needs:

1) The organization can consider placing frequently visited pages close to each other for easy access.
2) The organization can consider grouping similar content under the same menu, which can simplify navigation.
3) The organization can provide more relevant information by considering the outcome and filtering out irrelevant information. For example, the organization can relocate the page within the frequently used patterns that obtained the low support numbers to an inner page instead of the homepage for better optimization.

4) The organization could create page caches for its most frequently visited pages to enhance its website performance.

References

1. Belanche, D., Casaló, L.V., Guinalíu, M.: Website usability, consumer satisfaction and the intention to use a website: the moderating effect of perceived risk. J. Retail. Consum. Serv. **19**(1), 124–132 (2012)
2. Charliepaul, C.K., Gnanadurai, G.I.: Comparison of K-mean algorithm and apriori algorithm-an analysis. Int. J. Eng. Technol. Sci. **1**(III) (2014)
3. Cooley, R., Mobasher, B., Srivastava, J.: Mining: information and pattern discovery on the world wide web. In: Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI) (1997)
4. Elhiber, M.H.A., Abraham, A.: Access patterns in web log data: a review. J. Netw. Innovative Comput. **1**(2013), 348–355 (2013)
5. Facca, F.M., Lanzi, P.L.: Recent developments in web usage mining research. In: Kambayashi, Y., Mohania, M., Wöß, W. (eds.) DaWaK 2003. LNCS, vol. 2737, pp. 140–150. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45228-7_15
6. Fourie, I., Bothma, T.: Information seeking: an overview of web tracking and the criteria for tracking software. In: Aslib Proceedings. Emerald Group Publishing Limited (2007)
7. Goebel, M., Gruenwald, L.: A survey of data mining and knowledge discovery software tools. ACM SIGKDD Explor. Newslett. **1**(1), 20–33 (1999)
8. Ibrahim, K.K., Obaid, A.J.: Web mining techniques and technologies: a landscape view. J. Phys: Conf. Ser. **1879**, 032125 (2021)
9. Kumar, B.S., Rukmani, K.V.: Implementation of web usage mining using APRIORI and FP growth algorithms. Int. J. Adv. Netw. Appl. **1**(06), 400–404 (2010)
10. Neelima, G., Rodda, S.: Predicting user behavior through sessions using the web log mining. In: 2016 International Conference on Advances in Human Machine Interaction (HMI), pp. 1–5. IEEE (2016)
11. Nina, S.P., Rahman, M., Bhuiyan, K.I., Ahmed, K.E.U.: Pattern discovery of web usage mining. In: 2009 International Conference on Computer Technology and Development, vol. 1, pp. 499–503. IEEE (2009)
12. Patil, S.S., Khandagale, H.P.: Enhancing web navigation usability using web usage mining techniques. Int. Res. J. Eng. Technol. **4**(6) (2016)
13. Pei, J., et al.: PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth. In: proceedings of the 17th international conference on data engineering, pp. 215–224. IEEE (2001)
14. Resul, D.A.S., Turkoglu, I., Poyraz, M.: Analyzing of system errors for increasing a web server performance by using web usage mining. IU J. Electr. Electron. Eng. **7**(2), 379–386 (2007)
15. Singh, A.P., Jain, D.R.: A survey on different phases of web usage mining for anomaly user behavior investigation. Int. J. Emerg. Trends Technol. Comput. Sci. **3**(3) (2014)
16. Sisodia, D. S., & Verma, S. (2012, May). Web usage pattern analysis through web logs: A review. In 2012 Ninth International Conference on Computer Science and Software Engineering (JCSSE) (pp. 49–53). IEEE

17. Suadaa, L.H.: A survey on web usage mining techniques and applications. In: 2014 International Conference on Information Technology Systems and Innovation (ICITSI), pp. 39–43. IEEE (2014)
18. Suharjito, S., Diana, D., Herianto, H.: Implementation of classification technique in web usage mining of banking company. In: 2016 International Seminar on Intelligent Technology and Its Applications (ISITIA), pp. 211–218 (2016)
19. Ugras, T., Gülseçen, S., Çubukçu, C., İli Erdoğmuş, İ, Gashi, V., Bedir, M.: Research trends in web site usability: a systematic review. In: Marcus, A. (ed.) DUXU 2016. LNCS, vol. 9746, pp. 517–528. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40409-7_49
20. Varnagar, C.R., Madhak, N.N., Kodinariya, T.M., Rathod, J.N.: Web usage mining: a review on process, methods and techniques. In: 2013 International Conference on Information Communication and Embedded Systems (ICICES), pp. 40–46. IEEE (2013)
21. Yuan, F., Wang, L.J., Yu, G.: Study on data preprocessing algorithm in web log mining. In: Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 03EX693), vol. 1, pp. 28–32. IEEE (2003)

Deep Learning

Deep-Learning-Based LSTM Model for Predicting a Tidal River's Water Levels: A Case Study of the Kapuas Kecil River, Indonesia

Kanada Kurniawan[1], Joko Sampurno[2](✉), Riza Adriat[2], Randy Ardianto[1], and Arie Antasari Kushadiwijayanto[3]

[1] Pontianak Marine Meteorological Station, Pontianak, Indonesia
[2] Department of Physics, Tanjungpura University, Pontianak, Indonesia
`jokosampurno@physics.untan.ac.id`
[3] Department of Marine Sciences, Tanjungpura University, Pontianak, Indonesia

Abstract. Accurate prediction of water levels in tidal rivers is crucial for effective disaster management in coastal areas. This study uses the LSTM deep learning method to forecast the water level dynamics of the Kapuas Kecil River and determine the optimal window size for precise predictions. Our results reveal an optimal window size of 336 h (equivalent to 14 days) for water level prediction using LSTM in this coastal region. Using this optimal window size, the LSTM model consistently outperforms GRU and RNN models in comparative assessments. These findings offer not only valuable insights into water level prediction in the study area but also the potential of deep learning to enhance flood and disaster management in similar river systems globally.

Keywords: LSTM · Deep learning · Water level · The Kapuas Kecil River

1 Introduction

Predicting water levels in tidal rivers like the Kapuas Kecil River delta is crucial for flood and disaster management, navigation and transportation, environmental stewardship, and infrastructure planning [1]. The navigation route from the river mouth to the Dwikora Port in Pontianak is narrow and shallow, with only a 20 km stretch, so large vessels urgently require tidal information to avoid grounding [2]. Other than that, accurate water level predictions can assist authorities in taking proactive measures to mitigate flood impacts and design resilient infrastructure capable of withstanding the expected water levels [3]. Fluctuations in water levels impact infrastructure, such as bridges, dams, and water intakes, so accurate water level predictions can inform the planning of infrastructure that is more resilient, safe, and functional in the long run. Additionally, since tidal rivers are prone to flooding, knowing when and how high the water level will rise can inform decision-making about the evacuation of affected areas, relocation of livestock to higher ground, and deployment of emergency services. For the industry players, accurate water

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 103–110, 2023.
https://doi.org/10.1007/978-981-99-7969-1_8

level predictions can assist in planning operations, such as determining the optimal times for vessel movement and avoiding hazards like shallow water and strong currents. Lastly, from the environmental perspective, accurate predictions can help environmental managers anticipate and mitigate the impacts of changes in water levels, which determine the health of ecosystems in tidal rivers.

A deep learning method called Long Short-Term Memory (LSTM) has been shown to be more reliable in predicting water levels in tidal rivers than other methods [4–6]. LSTM is a recurrent neural network commonly used for time-series prediction tasks [4, 7–9]. In LSTM-based time-series prediction, the model takes a sequence of previous observations (i.e., a data window) as input and predicts the next observation in the sequence. The window size used in LSTM-based time-series prediction typically consists of multiple observations. Using a window of past observations as input to the LSTM model captures temporal dependencies in the data and makes more accurate predictions.

The window size refers to the number of previous time steps of data that the LSTM model takes as input to predict the next time steps [10]. Considering its importance in deep-learning LSTM, the window size can determine the robustness of the water level prediction model. If the window size is too small, the model may fail to capture long-term patterns and dynamics in the data, resulting in inaccurate water level predictions. Likewise, if the window size is too large, the model may include too much irrelevant information, leading to overfitting and poor generalization of new data. Therefore, determining an optimal window size is crucial in building a robust and accurate LSTM model for water level prediction.

This study aims to determine the optimal window size of hourly water level data of the Kapuas Kecil River delta to predict future water levels using the deep learning LSTM method. We collected and analyzed the delta's water level data to determine the optimal window size to predict flood accurately and mitigate the disaster impacts. Here, we tested various window sizes and evaluated each size's impact on the performance of the LSTM model. Subsequently, we compare the performance of the optimal LSTM model with that of other models, namely Gated Recurrent Unit (GRU) and recurrent neural network (RNN), to assess its superiority.

2 Method

2.1 Study Area

The Kapuas Kecil River is situated in West Kalimantan, Indonesia. It traverses through the vast and diverse landscape of the region, flowing from its headwaters in the Kapuas River, branching down to its downstream, where it eventually meets the South China Sea (Fig. 1). The river follows a meandering path, winding its way through lush tropical rainforests and low-lying plains. It passes through numerous settlements and communities, serving as a lifeline for both rural and urban areas along its course. The river's length and width vary, with narrower stretches in some areas and wider expanses in others. As the river descends towards the coastal areas, the landscape transitions into swamps, mangroves, and tidal flats, forming the deltaic region.

The Kapuas Kecil River is influenced by the tidal movements of the South China Sea, giving it the unique characteristic of being a tidal river [2]. This tidal influence extends into the river's lower reaches, affecting water levels and the overall hydrological dynamics. The river's tidal nature has implications for navigation, water quality, and ecosystem dynamics, making it an interesting and complex system to study.

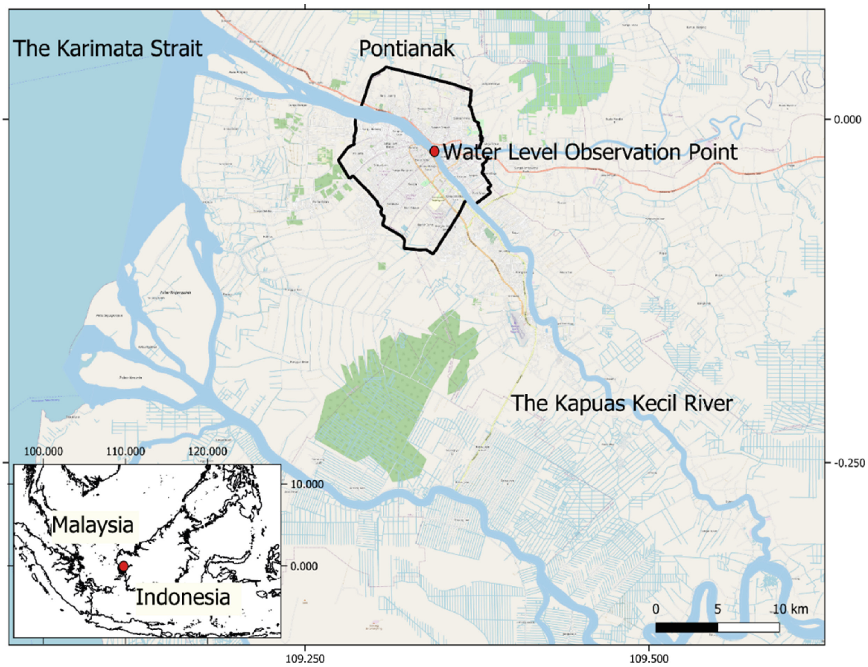


Fig. 1. The point of interest. The red dot indicates the observation point within the Kapuas Kecil River in the city of Pontianak. The background map was retrieved from OpenStreetMap [11].

2.2 Data

In this study, we conducted an in-depth analysis of a six-year dataset (2016–2021) comprising hourly water level measurements, as depicted in Fig. 2. These measurements exhibited notable variability, with the minimum water level recorded at 80 cm and the maximum reaching 299 cm. To better understand the central tendency of the dataset, we calculated the median water level, which was found to be 156 cm, as well as the mean, which averaged 159.5 cm. Importantly, all water level readings were referenced to the lowest astronomical tide (LAT), ensuring that our findings are grounded in the context of natural tidal fluctuations and the hydrodynamic conditions in the study area.

Using data from a six-year period also ensures that the data used in the study covers a range of weather conditions and hydrological events, including wet and dry seasons, storms, and periods of prolonged drought. This helps to ensure that the LSTM model trained on this data is robust and can make accurate predictions under a wide range of conditions. Overall, the hourly water level data over the six-year period is rich and

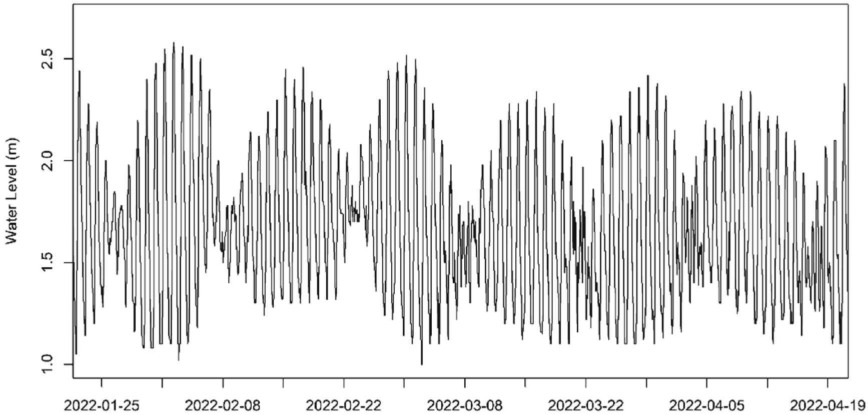


Fig. 2. The time series of the water level data observed at the point of interest within the Kapuas Kecil River, Pontianak

diverse, allowing for a detailed analysis of water level patterns and the development of a robust LSTM model for water level prediction in the Kapuas Kecil River delta.

2.3 LSTM Deep Learning Algorithm

With the data on water level dynamics, the LSTM deep learning algorithm predicts future water levels based on past observations [4]. First, we prepare the water level data for the LSTM model, including splitting the data into training and testing datasets, selecting the appropriate input and output variables, and scaling the data to ensure that all variables have similar ranges. Next, the LSTM model architecture is defined, consisting of a stack of LSTM layers, each containing a set of LSTM cells. Then, the model architecture is compiled before training, which involves specifying the loss function, optimizer, and evaluation metric for training.

The LSTM model is then trained on the training dataset, with the model weights updated iteratively using backpropagation [12]. The model is evaluated on the test dataset after each epoch to monitor performance. To make a prediction, the model takes the input sequence of past water level observations and generates the predicted value of the next time steps. The performance of the LSTM model is evaluated by comparing the predicted water levels with the actual values, using metrics mean absolute error (MAE), root mean squared error (RMSE), Kling-Gupta Efficiency (KGE), Nash-Sutcliffe Efficiency (NSE), and correlation coefficient (r). The model is then retrained and evaluated until satisfactory performance is achieved. Then, the optimum window size is determined based on the lowest error metric. Once the optimum window size is determined, a comparison of the LSTM model's performance with other models, namely GRU and RNN, is carried out to corroborate its superiority.

3 Result

In this study, the evaluation was conducted across a range of window sizes spanning from 1 to 8760 h, equivalent to one year. The results of our analysis are visually presented in Fig. 3, which illustrates the performance of the LSTM deep learning model in forecasting water levels in the study area.

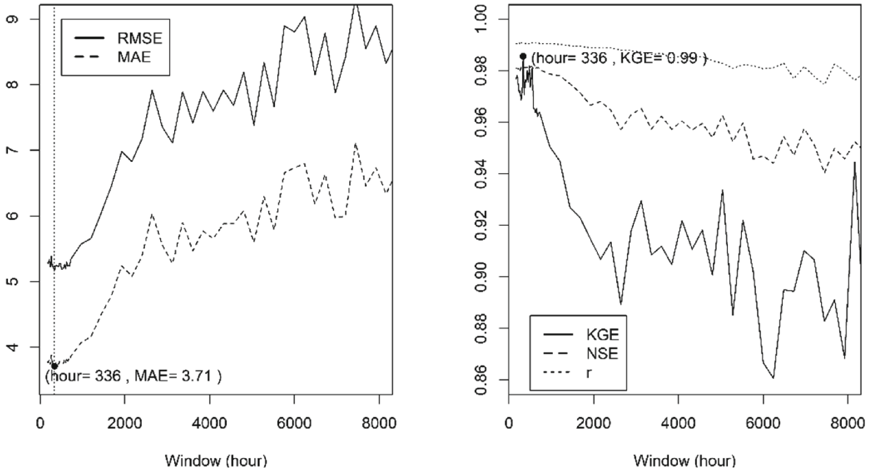


Fig. 3. LSTM performance based on the windows sizes

The findings revealed interesting trends in the performance of the LSTM model with different window sizes. The MAE values demonstrated a decreasing trend from a window size of 1 h to 335 h, reaching the lowest point at 336 h. Subsequently, from the window size of 336 h onward, the trend of MAE values increased. The RMSE values followed a similar pattern, decreasing from a window size of 1 h to 335 h, with the lowest point occurring at 336 h. Then, the RMSE values rose from 336 h onward.

The impact of window size on metrics such as KGE, NSE, and the correlation coefficient (r) was also assessed. Figure 3 demonstrates that their respective maximum values are achieved at the window size of 336 h. A remarkable peak in the KGE metric is observed at the 336-h window size, with an accuracy score of 0.99 achieved. This finding supports the previous assessment using the MAE and RMSE metrics.

Furthermore, the water level predictions from the top-performing LSTM model were tested against the observational data. This evaluation involved a comparative analysis with the predictions generated by GRU and RNN models, with the outcomes visually presented in Fig. 4. The result shows that the LSTM model performs slightly better than the other models. MAE, RMSE, and KGE support the superiority of the LSTM model, whereas NSE and r indicate consistency across all models (Table 1).

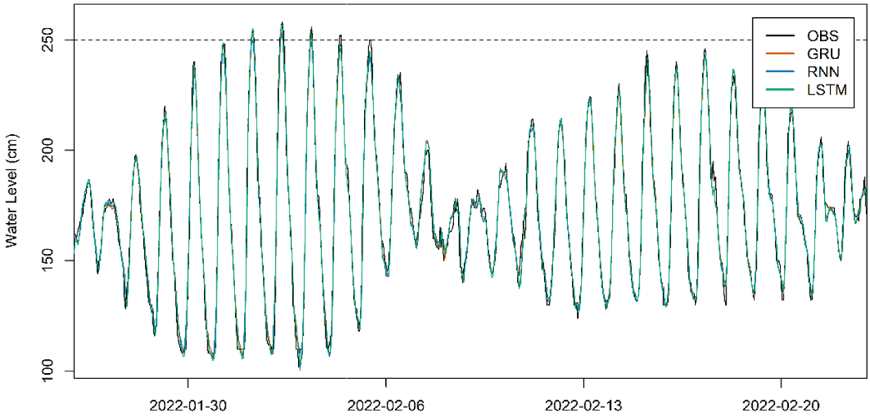


Fig. 4. Water level comparison between model outputs and observation, with the dashed line representing the threshold when the river water begins to overflow and inundate the city of Pontianak [13]

Table 1. The comparison between model performance

	MAE	RMSE	KGE	NSE	r
LSTM	3.71	5.17	0.99	0.98	0.99
GRU	3.83	5.28	0.97	0.98	0.99
RNN	3.96	5.44	0.96	0.98	0.99

4 Discussion

The primary objective of this study is to determine the optimal window size for the LSTM model in predicting tidal water levels. The investigation has led to a noteworthy finding, with the 336-h window size being the most effective setting. What makes this conclusion robust is the comprehensive evaluation carried out using a range of error metrics, including MAE, RMSE, KGE, NSE, and r [14, 15].

MAE quantifies the average absolute differences between predicted and observed values, with a lower MAE value indicating more accurate predictions [16]. Meanwhile, RMSE considers the squared differences and places greater emphasis on more significant errors [16], and KGE combines correlation, bias, and variability to assess model performance on a scale from $-\infty$ to 1, with 1 indicating a perfect agreement [17]. Finally, NSE compares error variance to observational variance, and r measures linear relationships between variables [14]. Each metric has its strengths and limitations, making their suitability context-dependent. For instance, applying a logarithmic transformation to values may enhance MAE or RMSE's performance for cases where low flows are more crucial. KGE may be preferred if the goal is to capture both timing and magnitude errors. By considering these diverse metrics, the current study ensures a thorough assessment of the model performance, taking into account various aspects of accuracy and precision.

Furthermore, the study goes beyond identifying the best window size for the LSTM model by comparing the performance with two other models, namely GRU and RNN. This comparative evaluation adds another layer of credibility to the findings. Notably, three of the five metrics (MAE, RMSE, and KGE) demonstrate the LSTM model's superiority over the alternative models. This consistency in results across multiple evaluation criteria strengthens the conclusion that the LSTM model with a 336-h window size is the most suitable choice for predicting water levels in the study area.

This study's comprehensive approach, combining various error metrics and model comparisons, leads to two conclusions. The first is the importance of considering multiple dimensions of model performance. The second is the LSTM model's effectiveness in addressing the specific predictive task of tidal water level forecasting in the area under study.

However, it is important to note a few limitations in this study. The identified optimal window size may be specific to the study area and dataset, and generalizing it to other scenarios should be done cautiously. Next, the study focuses only on accuracy, overlooking potential impacts on computational requirements and model complexity. Lastly, all error metrics involved (MAE, RMSE, KGE, NSE, and r) have limitations and may not provide a complete assessment of the model performance. Therefore, further research is needed to explore alternative window sizes, incorporate different evaluation metrics, and corroborate the findings in diverse regions and datasets.

5 Conclusion

This study employs LSTM deep learning to predict water levels in the study area and examines the influence of window sizes on prediction accuracy. A thorough evaluation revealed that the optimal window size for tidal water level prediction is the 336-h one. This conclusion was reached through a comprehensive analysis incorporating various error metrics (MAE, RMSE, KGE, NSE, and r) and confirmed by a test against GRU and RNN models. Three out of five metrics (MAE, RMSE, and KGE) support the LSTM model's superior performance. Therefore, configuring the LSTM model with a 336-h window is recommended for its effectiveness, reliability, and practicality in forecasting water levels accurately in the study area. This finding offers valuable insights to inform decision-making in water resource management and flood mitigation strategies in the study area and other similar river systems.

References

1. Guo, W.D., Chen, W.B., Yeh, S.H., Chang, C.H., Chen, H.: Prediction of river stage using multistep-ahead machine learning techniques for a tidal river of Taiwan. Water **13**(7), 920 (2021)
2. Sampurno, J., Vallaeys, V., Ardianto, R., Hanert, E.: Modeling interactions between tides, storm surges, and river discharges in the Kapuas River delta. Biogeosciences **19**(10), 2741–2757 (2022)
3. Xu, G., Cheng, Y., Liu, F., Ping, P., & Sun, J. A.: Water level prediction model based on ARIMA-RNN. In IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService), pp. 221–226 (2019)

4. Yang, C.-H., Wu, C.-H., Hsieh, C.-M.: Long short-term memory recurrent neural network for tidal level forecasting. IEEE Access **8**, 159389–159401 (2020)
5. Jung, S., Cho, H., Kim, J., Lee, G.: Prediction of water level in a tidal river using a deep-learning based LSTM model. J. Korea Water Resour. Assoc. **51**(12), 1207–1216 (2018)
6. Tran, Q.K., Song, S.K.: Water level forecasting based on deep learning: a use case of trinity River-Texas-the united states. J. KIISE **44**(6), 607–612 (2017)
7. Kratzert, F., Klotz, D., Brenner, C., Schulz, K., Herrnegger, M.: Rainfall–runoff modelling using long short-term memory (LSTM) networks. Hydrol. Earth Syst. Sci. **22**(11), 6005–6022 (2018)
8. Xiang, Z., Yan, J., Demir, I.: A rainfall-runoff model with LSTM-based sequence-to-sequence learning. Water Resources Res. **56**(1), e2019WR025326 (2020)
9. Han, H., Choi, C., Jung, J., Kim, H.S.: Deep learning with long short term memory based sequence-to-sequence model for rainfall-runoff simulation. Water **13**(4), 437 (2021)
10. Rizal, A.A., Soraya, S., Tajuddin, M.: Sequence to sequence analysis with long short-term memory for tourist arrivals prediction. J. Phys. Conf. Ser. **1211**(1), 12024 (2019)
11. OpenStreetMap contributors: Panet dump. https://planet.osm.org Accessed 16 Sept 2023
12. Zhang, J., et al.: Application of cost-sensitive LSTM in water level prediction for nuclear reactor pressurizer. Nucl. Eng. Technol. **52**(7), 1429–1435 (2020)
13. Sampurno, J., Ardianto, R., Hanert, E.: Integrated machine learning and GIS-based bathtub models to assess the future flood risk in the Kapuas river delta. Indonesia. J. Hydroinformatics **25**(1), 113–125 (2023)
14. Ruma, J.F., Adnan, M.S.G., Dewan, A., Rahman, R.M.: Particle swarm optimization based LSTM networks for water level forecasting: a case study on Bangladesh river network. Res. Eng. **17**, 100951 (2023)
15. Hashemi, R., Brigode, P., Garambois, P.A., Javelle, P.: How can we benefit from regime information to make more effective use of long short-term memory (LSTM) runoff models? Hydrol. Earth Syst. Sci. **26**(22), 5793–5816 (2022)
16. Kumar, V., et al.: Using machine learning to predict clinical outcomes after shoulder arthroplasty with a minimal feature set. J. Shoulder Elbow Surg. **30**(5), e225–e236 (2021)
17. Knoben, W.J., Freer, J.E., Woods, R.A.: Inherent benchmark or not? Comparing Nash-Sutcliffe and Kling-Gupta efficiency scores. Hydrol. Earth Syst. Sci. **23**(10), 4323–4331 (2019)

Data Augmentation for EEG Motor Imagery Classification Using Diffusion Model

Nutapol Soingern[1(✉)], Akraradet Sinsamersuk[1], Itthi Chatnuntawech[2], and Chaklam Silpasuwanchai[1]

[1] School of Engineering and Technology, Data Science and Artificial Intelligence, Asian Institute of Technology, Pathum Thani, Thailand
`nutapol1997@gmail.com`, `{st121413,chaklam}@ait.asia`
[2] National Science and Technology Development Agency, National Nanotechnology Center, Pathum Thani, Thailand
`itthi.cha@nanotec.or.th`

Abstract. Motor imagery classification using electroencephalogram (EEG) signals is an important research topic that has been extensively studied in the field of brain-computer interfaces (BCIs). However, due to the limited amount of available data, overfitting is a common problem, especially when using a deep-learning classifier. One way to address this is by performing data augmentation. In this paper, we investigate the efficacy of the diffusion model as a data augmentation method for motor imagery classification. We evaluated the diffusion method by comparing it with commonly-used EEG data augmentation techniques namely such as Noise Addition, Fourier Transform Surrogates, Frequency Shift, and SmoothTimeMask. The result shows that the diffusion method outperformed other methods in terms of classification accuracy by 17.49%. The Kullback-Leibler (KL) divergence is used for assessing the similarity between the training set (with and without augmentation) and validation set, thus showing the effectiveness of the diffusion approach compared to other techniques.

Keywords: Motor Imagery · EEG · Brain Computer Interface · Deep learning · Data Augmentation · Diffusion · KL divergence

1 Introduction

Brain-computer interfaces (BCI) establish a direct pathway between the human brain and a computer via signal processing and decoding techniques. One classic paradigm of EEG is motor imagery (MI), in which its physiological basis is based on body movements or imagined movements that can produce α (8–13 Hz) and β (13–30 Hz) event-related synchronization (ERS) and event-related desynchronization (ERD) rhythms in the motor-sensory areas of the brain [1]. Recently, deep learning (DL) model has been used for motor imagery classification. For

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 111–126, 2023.
https://doi.org/10.1007/978-981-99-7969-1_9

example, EEGNet [2] is a compact convolutional neural network designed for EEG-based brain-computer interfaces that effectively extracts spatial-temporal features from EEG signals. In any case, the paucity of data is a prevalent issue in the field of EEG classification, as it hinders the development and performance of DL models. Consequently, a common symptom is overfitting, which reduces the model's accuracy and robustness on test set [3].

Data augmentation (DA) has been widely used to improve the robustness and accuracy of DL by artificially increasing the number of training data. Traditional EEG data augmentation methods include `Noise Addition` [4–6], fourier transform surrogates [7], `Frequency Shifting` [8,9] and `SmoothTimeMask` [10]. `Noise Addition` [5,6] adds random white noise to all channels. Fourier transform surrogates [7] randomizes the Fourier-transform (FT) phases of temporal-spatial data generates surrogates that approximate examples from the data-generating distribution. `Frequency Shifting` [8,9] randomly shifts the frequency spectrum on all channels. Last, `SmoothTimeMask` [10] randomly masks consecutive time steps of the EEG signal and replaces them with zeros. The length of the masked segment was also selected randomly. The motivation is to force the model to disregard minor irrelevant events.

Recently, diffusion model [11] was proposed which generates synthetic data based on Langevin dynamics. These models naturally admitted a progressive lossy decompression scheme that can be interpreted as an extension of autoregressive decoding. Diffusion mode has been used as a DA method to generate synthetic training data for skin disease classification [12], etc. Moreover, `WaveGrad` [13] model is a research-based approach that has been developed to generate audio waveforms of superior quality. The proposed approach involves utilizing score matching [14] and diffusion probabilistic models to estimate gradients of the data density within a conditional model. The research methodology involves initializing the model with a Gaussian white noise signal and subsequently improving the signal quality through an iterative process that utilizes a gradient-based sampler. The sampler is conditioned on the mel-spectrogram. The `WaveGrad` model presents a method to balance inference speed and sample quality through the manipulation of refinement steps. Additionally, it serves as a connection between non-autoregressive and autoregressive models in relation to audio quality. According to the research findings, `WaveGrad` has the ability to produce high-quality audio samples with only six iterations.

In this work, we demonstrated the use of the diffusion model based on `WaveGrad` [13] as a DA method for motor imagery classification. We evaluated the effect of the proposed method by performing DA on BCI Competition IV 2a [15] with various size of synthetic data with five standards EEG MI models (EEGNet [2], ATCNet [16], EEG-ITNet [17], Deep ConvNet [18] and ShallowFBCSPNet [18]). The proposed method improve the performance outperform other traditional EEG data augmentation methods.

2 Related Work

We reviewed commonly-used data augmentation for EEG MI such as `Noise Addition`, `Fourier Transform Surrogates`, `Frequency Shift` and `Smooth TimeMask`.

2.1 Noise Addition

`Noise Addition` has two main categories for adding noise to the EEG signals in purpose of DA [19]. A common technique in EEG research involved introducing Gaussian noise with zero mean and a standard deviation of 0.1 to the recorded data [5]. The simulation of EEG data variability was commonly utilized to replicate the effects of electrode noise or subject movement during experimental procedures. The introduction of noise to the training data could enhance the robustness of the model by compelling it to learn features that were less susceptible to minor fluctuations in the data. Previous research has demonstrated that the inclusion of Gaussian noise in EEG signals enhances the efficacy of the MI classification model when applied to BCI competition IV dataset 2b [20], resulting in a 10% improvement in performance.

2.2 Fourier Transform Surrogates

The Fourier transform surrogates (`FTSurrogate`) method utilized the phase data of frequency elements, which were subsequently rearranged in a random manner while maintaining their original magnitude spectrum [7]. The generation of synthetic data samples has been utilized as a means to address the underrepresentation of certain classes. This approach has been shown to improve the balance of class distribution and enhance the accuracy of classification. The method proposed in this study has the potential to enhance classification performance either as a standalone technique or in conjunction with other data augmentation methods [7]. The extent of enhancement varies based on the particular dataset and classification issue. The extent of enhancement differs based on the particular dataset and classification issue. The study aimed to enhance the mean F1-score of a convolutional neural network utilized for sleep stage classification by 7% through the implementation of surrogate-based augmentation on the CAPSLPDB sleep database [21].

2.3 Frequency Shift

In the `Frequency Shift` method, the frequency spectrum of an EEG signal was randomly shifted to a different frequency range while maintaining the amplitude spectrum [9]. The proposed technique involved generating novel EEG signals that exhibit identical spectral characteristics as the initial signal, albeit with altered frequencies. The study's findings indicated that the `Frequency Shift` method was successful in enhancing the classification accuracy of certain EEG datasets.

In a study on motor imagery datasets, the implementation of the `Frequency Shift` method resulted in a 2.5% increased in classification accuracy when compared to the baseline method. Moreover, Jaderberg et al. (2021) proposed generating augmented EEG signals by applying various transformations, including the `Frequency Shift` method, to the original signals. The study assessed the efficacy of a novel method on various EEG classification tasks and demonstrated its superiority over conventional data augmentation techniques, including random cropping and flipping, as well as other learned data augmentation methods. The study's findings indicated that the suggested approach yields optimal performance and required less time for training compared to gradient-based methods in the class-agnostic context. Additionally, it surpassed gradient-free methods in the class-wise context. The research paper lacked a specific numerical value for the quantity of effects or enhancements. The effectiveness of this method in enhancing classification performance was observed in certain datasets, such as the BCI Competition IV 2a dataset [8].

2.4 SmoothTimeMask

`SmoothTimeMask` was a research methodology that utilized time-domain augmentation to introduce smoothness into a signal. This is achieved by masking contiguous time intervals [10]. The `SmoothTimeMask` algorithm was a technique that used to apply a smooth mask to a segment of a time series signal. This method involved generating a mask by randomly selecting a starting point and masking a fixed length of contiguous samples. A common technique that used to create a smooth transition between masked and unmasked regions is the application of a convolution with a Gaussian kernel to the mask. The introduction of smoothness in the augmented signal has the potential to prevent overfitting and enhance the generalization of the model [10].

2.5 WaveGrad

`WaveGrad` is a generative model for waveform generation that uses score matching and denoising to improve the quality of generated waveforms. The basic idea behind the method is to estimate the probability density function of a dataset using a generative model, and then use this estimate to generate new data points that are similar to the original data.

To achieve this, `WaveGrad` uses an autoregressive architecture that predicts each sample of the waveform conditioned on the previous samples. Specifically, `WaveGrad` uses a modified version of the WaveNet architecture that replaces the dilated convolutions with a set of learned gates and skips connections, which reduces the computational cost of the model.

`WaveGrad` trains the generative model using score matching, which is a technique that involves matching the gradient of the log-density function of the model to the gradient of the true log-density function of the data. The idea behind score matching is that the gradient of the log-density function is easier to estimate than the function itself, and that matching the gradients is sufficient to match the distributions.

Score matching is a technique used for estimating the probability density function (`PDF`) of a dataset by matching the score function of a model to the true score function of the `PDF`. The score function is the gradient of the log-density function, i.e., the vector of partial derivatives of the log-density function with respect to each input variable.

In score matching, the model is trained to minimize the difference between the score function of the model and the true score function of the `PDF`. This can be formulated as the following loss function:

$$L(\theta) = \sum_i |\nabla_x \log p(x_i;\theta) - \nabla_x \log \hat{p}(x_i)|^2 \tag{1}$$

where θ are the parameters of the model, x_i is a data point, $p(x_i;\theta)$ is the model's probability density function, and $\hat{p}(x_i)$ is the true probability density function of the dataset.

Denoising score matching is an extension of score matching that uses a denoising autoencoder to estimate the score function of the `PDF`. The denoising autoencoder is trained to remove noise from the input data, and the score function of the denoised data is used to estimate the true score function of the `PDF`. The loss function for denoising score matching is:

$$L(\theta) = \sum_i |\nabla_x \log \hat{p}(\tilde{x}_i) - \nabla_x \log \hat{p}(x_i)|^2 \tag{2}$$

where $\tilde{x}_i$ is the denoised version of x_i.

Weighted denoising score matching is a further extension of denoising score matching that accounts for noisy labels. The idea is to assign higher weights to samples with less noise and lower weights to samples with more noise. This can be achieved by introducing a weighting function $w(x_i)$ into the loss function:

$$L(\theta) = \sum_i w(x_i)|\nabla_x \log \hat{p}(\tilde{x}_i) - \nabla_x \log \hat{p}(x_i)|^2 \tag{3}$$

where θ are the parameters of the model, x_i is a data point, $\hat{p}(x_i)$ is the true probability density function of the dataset, $\tilde{x}_i$ is the denoised version of x_i, and $w(x_i)$ is a weighting function that assigns a weight to each sample based on the level of noise in its label.

`WaveGrad` further improves the optimization of the model by using a variant of stochastic gradient descent called Stochastic Gradient Hamiltonian Monte Carlo (SGHMC). SGHMC uses Hamiltonian dynamics to simulate the motion of particles in a potential energy landscape, which improves the exploration of the parameter space during optimization.

Overall, `WaveGrad` is able to generate high-quality waveforms that are comparable to or better than previous state-of-the-art methods. It achieves this by combining denoising and score matching with a modified version of the WaveNet architecture and SGHMC optimization.

3 Methodology

We compared the diffusion method against four DA methods and the baseline method without augmentation. Figure 1 shows how the training sets were obtained using different combinations of the DA method and sampling size. The sampling size was chosen at the ratio of 25, 50, 75, 100%. Five commonly-used models for MI classification was used. The models were trained using subject-dependent scheme and evaluated on their respective testing sets.

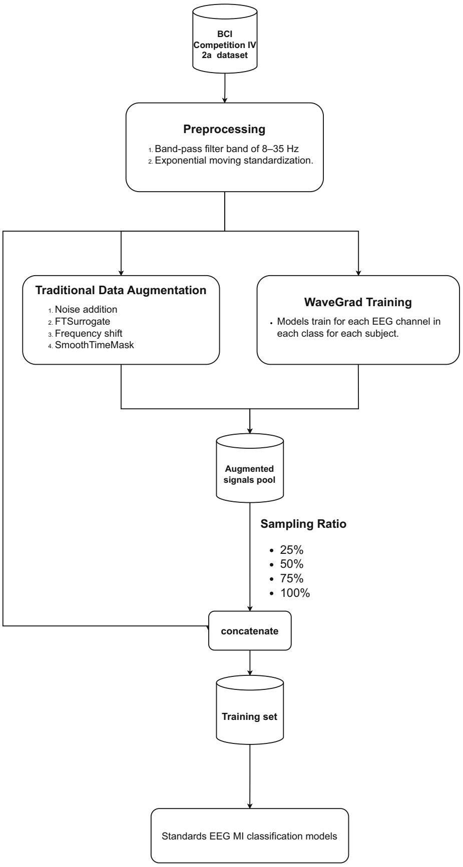

BCI Competition IV 2a dataset
Preprocessing
1. Band-pass filter band of 8–35 Hz
2. Exponential moving standardization.
Traditional Data Augmentation
1. Noise addition
2. FTSurrogate
3. Frequency shift
4. SmoothTimeMask
WaveGrad Training
Models train for each EEG channel in each class for each subject.
Augmented signals pool
Sampling Ratio
25%
50%
75%
100%
concatenate
Training set
Standards EEG MI classification models

Fig. 1. Showed how to obtain training sets when using traditional DA and `WaveGrad`. `WaveGrad` was trained for each EEG channel in each class for each subject. Thus, we trained a total of 792 models from four classes, 22 channels and 9 subjects.

3.1 Datasets

BCI Competition IV 2a [15] was a collection of EEG data from 9 subjects who participated in a cue-based BCI paradigm involving four distinct motor imagery tasks: imagining movement of the left hand (class 1), the right hand (class 2), both feet (class 3) and the tongue (class 4). Each subject completed the tasks in two distinct sessions on different days, with each session consisting of six runs separated by brief pauses, resulting in a total of 48 trials (12 for each of the four classes). The data were captured while the participants sat in a comfortable armchair in front of a computer screen, and a fixation cross appeared at the start of each trial. A cue consisting of an arrow pointing to the left, right, down, or up was used to prompt the subjects to perform the desired motor imagery task. The subjects were instructed to perform the motor imagery task until the fixation cross disappeared from the screen without receiving any feedback. Signals were sampled at 250 Hz and filtered between 0.5 Hz and 100 Hz using a 50 Hz notch filter in order to reduce line noise. We used one section for training sets and another for test sets.

3.2 Data Preprocessing

EEG signal was filtered with a band pass filter of 8–35 Hz followed by exponential moving standardization x_{t}. For exponential moving standardization, compute the exponential moving mean m_{t} at time t and as show in Eqs. (4). Then, compute the exponential moving variance v_{t} at time t as show in Eqs. (5). We set $factornew$ is 0.001 and esp is 0.0001. Finally, standardize the data point x_{t} at time t as show in Eqs. (6).

$$m_{\mathrm{t}} = factornew \cdot mean(x_{\mathrm{t}}) + (1 - factornew) \cdot m_{\mathrm{t}-1}, \tag{4}$$

$$v_{\mathrm{t}} = factornew \cdot (m_{\mathrm{t}} - x_{\mathrm{t}})^2 + (1 - factornew) \cdot v_{\mathrm{t}-1}, \tag{5}$$

$$x_{\mathrm{t}}^{'} = (x_{\mathrm{t}} - m_{\mathrm{t}})/max(\sqrt{- > v_{\mathrm{t}}}, esp), \tag{6}$$

3.3 Data Augmentation

Our objective was to assess the performance of each DA method with different ratios of augmented/synthetic data. Five DA methods (`Noise Addition`, `FTSurrogate`, `Frequency Shift`, `SmoothTimeMask`, `WaveGrad`) and four ratios (25%, 50%, 75%, 100%) = 20 combinations (5 methods × 4 ratios) of the training set are to be created. Here, ratios refer to the amount of augmented/synthetic data used. For example, given 25% ratio, a 25% augmented data is randomly sampled and add to the original training set. By comparing different ratios, we can better understand the impact of number of augmentation to accuracy improvement.

Similar to previous works [8,10,20,21], we used the subject-dependent scheme which augments data in subject level (i.e., each subject is treated separately) and channel level (i.e., each channel is treated independently).

Noise Addition: `Noise Addition` entails the inclusion of diverse forms of noise, such as Gaussian, Poisson, and others, that possess varying parameters to the original EEG signal. The raw EEG signal was subjected to additive noise by incorporating a Gaussian distribution with a standard deviation of 0.1.

Fourier Transform Surrogates: `Fourier transform surrogates` are a type of data generated by randomizing the phases of temporal-spatial data.

Frequency Shift: The technique of `Frequency Shift` is characterized by the alteration of the frequency of the EEG signal by a specific amount. We random shift the frequency by ±2 Hz.

SmoothTimeMask: `SmoothTimeMask` involves applying a smooth window function to mask a continuous segment of the signal, and optimizing it using gradient-based methods. The signal was randomly masked with a range of 100 sample points.

WaveGrad: It is first important that `WaveGrad` is originally a generative model, not a formal augmentation technique. Thus, in contrary to other DA methods, `WaveGrad` has to be trained before it can be used to generate a synthetic EEG signal. The dataset consists of 9 subjects and four MI classes. The EEG recording has 22 channels. Thus, the total number of `WaveGrad` models was (9 subjects $\times$ 22 channels $\times$ 4 classes) = 792 models. The training procedure and parameters were the same across all `WaveGrad` models. The learning rate was set to 0.0001 and the diffusion steps to 1000.

3.4 Evaluation

First, it is important to evaluate the quality of the augmented/synthetic data. A common way is through dimensionality reduction KL divergence. The success of the diffusion method in comparison to other methods is demonstrated by the Kullback-Leibler (KL) divergence, which is used to measure the similarity between the training set (with and without augmentation) and validation set. We expect that high-quality augmented or synthetic data should exhibit similarity between the training set (with and without augmentation) and the validation set.

Second, once the quality of the augmented/synthetic data are quantified, we are now ready to quantify the usefulness of data augmentation techniques on actual EEG tasks. We first selected five commonly-used motor imagery classification models (EEGNet, ATCNet, EEG-ITNet, Deep ConvNet and ShallowFBCSPNet) which would allow us to understand whether how complexity of the model relates with data augmentation. Here, note that we simply define the complexity based on the model's number of parameters. Accuracy was then measured across all 21 combinations (5 DA methods $\times$ 4 sampling ratios + 1 baseline method without augmentation). The details of each model were as follows:

EEGNet. EEGNet is a single CNN architecture that can accurately classify EEG signals from different BCI paradigms while being as compact as possible. The authors introduce the use of depthwise and separable convolutions to construct an EEG-specific model that encapsulates well-known EEG feature extraction concepts for BCI. They compare EEGNet to current state-of-the-art approaches across four BCI paradigms and show that EEGNet generalizes across paradigms better than, and achieves comparably high performance to, the reference algorithms when only limited training data is available across all tested paradigms.

ATCNet. ATCNet is predicting the onset of epileptic seizures using electroencephalogram (EEG) signals. The ATCNet consists of two blocks: an attention-based temporal convolutional (ATC) block and a transformer-based classification (TC) block. The ATC block is used to extract relevant features from the EEG signals, while the TC block is used to classify the extracted features into seizure and non-seizure classes. The proposed model is evaluated using the BCI Competition IV-2a (BCI-2a) dataset. The obtained accuracy ranges from 60.5% to 89.5%.

EEG-ITNet. EEG-ITNet uses inception modules and causal convolutions with dilation to extract rich spectral, spatial, and temporal information from multi-channel EEG signals with less complexity than other existing end-to-end architectures. The paper also provides a methodology for achieving intuitive visualisation structures such as topographic maps. The proposed EEG-ITNet model shows up to 5.9% improvement in classification accuracy compared to its competitors in different scenarios. The paper also explains and supports the validity of network illustration from a neuroscientific perspective.

Deep ConvNet. The deep ConvNet had four convolution-max-pooling blocks, with a special first block designed to handle EEG input, followed by three standard convolution-max-pooling blocks and a dense softmax classification layer. The authors found that recent advances in machine learning, including batch normalization and exponential linear units, together with a cropped training strategy, boosted the Deep ConvNets decoding performance, reaching at least as good performance as the widely used filter bank common spatial patterns (FBCSP) algorithm.

ShallowFBCSPNet. The shallow ConvNet are similar to the transformations of FBCSP. Concretely, the first two layers of the shallow ConvNet perform a temporal convolution and a spatial filter, as in the deep ConvNet. These steps are analogous to the bandpass and CSP spatial filter steps in FBCSP.

4 Results

4.1 Kullback-Leibler Divergence (KL Divergence)

To understand the quality of the generated data, we measured the similarity of signal by KL divergence process. We measure the KL divergence comparing of train set and test set, test set and train set with 25% of data augmentation, test set and train set with 50% of data augmentation, test set and train set with 75% of data augmentation, test set and train set with 100% of data augmentation, test set and data augmentation and train set and data augmentation. We randomize the augmentation data for this process 100 times and averaged the result. Overall, our method increases similarity as the ratios of augmented data are increased. When ratios are increased, the similarity for the SmoothTimeMask, Noise Addition, and FTSurrogate approaches that of the non-augmented data. On the other hand, as the augmented data from frequency shift increases, the similarity declines (Table 1).

Table 1. Average of KL divergence each data augmentation method. This result is based on 100 random iterations of each augmentation data process.

Data Augmentation	Test set vs Train set	Average of 25%	Average of 50%	Average of 75%	Average of 100%	Test set vs Augmentation	Train set vs Augmentation
`WaveGrad`	2421.27	2372.21	2336.35	2310.92	2294.04	2158.61	2090.67
`Noise Addition`	2316.79	2318.41	2324.93	2312.74	2316.05	2316.61	4.91
`FTSurrogate`	2316.79	2311.50	2305.57	2306.39	2306.27	2292.40	2226.75
`SmoothTimeMask`	2316.79	2320.10	2320.04	2322.51	2310.70	2316.79	0.00
`Frequency shift.`	2316.79	3513.38	4298.14	4888.52	5295.02	8287.84	8147.79

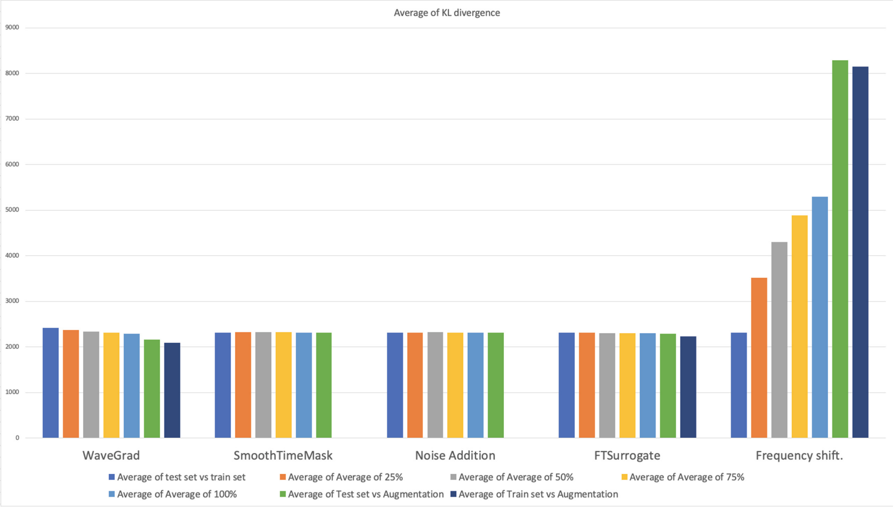


Fig. 2. Average of KL divergence each data augmentation method

4.2 Classification Performance

The baseline for the model was established by training it without any data augmentation, specifically without the use of synthetic data (i.e. 0% synthetic data). The average accuracy of standard EEG MI classification models were presented on the Table 2.

Table 2 showed the average accuracy of five standard EEG motor imagery (MI) classification models using different data augmentation techniques. The baseline accuracy of the models was 42.45%. The results showed that the highest accuracy was achieved using the `WaveGrad` technique, with an accuracy of 51.15%. This was an improvement of 8.7% over the baseline accuracy. As shown in Table 2, the highest accuracy values were obtained using `WaveGrad` (51.15%) and `Frequency Shift` (43.07%) techniques. The other two techniques, `Noise Addition` and Fourier transform surrogates, also improved the accuracy of the model to varying degrees. The `SmoothTimeMask` technique also resulted in higher accuracy values compared to the baseline, with the highest accuracy value of 38.84% obtained at a 25% size percentage.

Overall, our results demonstrated that DA techniques can be effective in improving the accuracy of EEG MI classification models. `WaveGrad` and `Frequency Shift` techniques were particularly promising, and were worth further investigation for future studies. Additionally, the size percentage of the DA techniques appeared to have an impact on model performance, suggesting that careful consideration of the amount of DA used was important for optimizing model accuracy.

Table 2. Average accuracy of five standards EEG MI classification models

Baseline Accuracy(%)	Size DA	`WaveGrad` (%)	σ	`Noise Addition`(%)	σ	`Frequency Shift`(%)	σ	Fourier transform surrogates(%)	σ	SmoothTimeMas(%)	σ
42.45	25%	**51.15**	17.61	35.83	13.95	**43.07**	18.28	41.25	14.31	**38.84**	12.38
	50%	**50.56**	17.38	33.46	11.10	44.49	17.82	37.03	10.85	36.27	10.84
	75%	51.64	17.24	33.77	11.23	43.58	18.25	37.01	10.14	34.95	10.73
	100%	50.78	17.72	31.39	8.84	42.61	18.80	36.20	11.03	34.31	9.51
Average Accuracy (%)		51.03	17.49	33.86	11.53	43.44	18.04	37.12	11.08	36.09	10.86

4.3 Size of Augmentation

We present the impact of data augmentation on the similarity by KL-divergence in Fig. 2. This result show that `WaveGrad` are improve similarity when incurred number of ratio of data augmentation to raw signal. We present the impact on accuracy of ratio of data augmentation to raw signal in Table 2. This result show that the `WaveGrad` are not improve the accuracy.

5 Discussion

In this study, we demonstrated the implementation of the `WaveGrad`-based Diffusion Model as a DA for EEG MI datasets. From the result, our method performed the accuracy based on the standard EEG MI classification models more than traditional DA. Thus, for the discussion section, we focused mainly on our method.

5.1 Key Improvement

The proposed methodology increased the variety of training sets, leading to an enhancement in the quality of the training set. As shown in Fig. 2, Our research indicated that our method enhanced the similarity of training sets with test set. The traditional DA (`Noise Addition`, `FTSurrogate`, and `SmoothTimeMask`) and training set without DA were the KL divergence are the most the same. Our method are improve the similarity of test set and train Fig. 2 our method more over another method are not improve the similarity. Spacial on `smooth time mask` and `noise add` the augmentation are not different the train set moreover the Frequency shift augment the seem make the train set and test set are more defence when decreased the similarity of train set and test set.

Our method are improve the similarity of test set and train set. Our method more over another method are not improve the similarity. Spacial on `SmoothTimeMask` and `Noise Addition` augmentation are not different the train set moreover the `Frequency shift` the seem make the train set and test set are more defence when decreased the similarity of train set and test set.

5.2 Subjects Level

In terms of research findings, it was observed that the data augmentation accuracy was highest for subjects who exhibited the greatest accuracy according to the standard EEG MI classification models prior to the implementation of data augmentation techniques. Subject 3's baseline accuracy was recorded to be 56.95% during the initial assessment. The research findings indicated that the accuracy for subject number 3 was improved by 15.46% using the implemented method. In contrast, the DA method employed in our study exhibited low efficacy when applied to the subject, as indicated by the subpar accuracy observed in relation to established EEG MI classification models. Subject 5's baseline accuracy was measured to be 28.08% in the study. The research findings indicate that a method has been implemented to enhance the accuracy of subject number 5, resulting in a 4.13% improvement. Table 3 displayed the mean increased in accuracy resulting from the implementation of our method.

5.3 Size of DA

The relationship between the size of DA and the performance of standard EEG classification was not a direct variation. Table 5 presented the average accuracy improvement achieved by our method on the DA size level. The limitations of the dataset that used to train the `WaveGrad` method have impacted its capacity to produce signals with greater valence. The observed limitation might account

Table 3. The Average accuracy improvement from our method on subject level

Subjects	Baseline Accuracy	Δ of `WaveGrad`
1	43.06	17.01
2	29.24	6.46
3	**56.95**	**15.36**
4	37.08	7.77
5	**27.08**	**4.13**
6	29.58	4.01
7	40.07	11.77
8	53.19	8.18
9	65.76	2.59

for the non-linear relationship between the size of the data augmentation and the corresponding increased in accuracy. Research suggested that increasing the number of samples may not necessarily lead to a significant enhancement in accuracy, particularly if the additional samples fail to encompass the complete spectrum of potential signal fluctuations.

However, despite this limitation, the `WaveGrad` method still outperformed the baseline accuracy on all levels of data augmentation, with the highest improvement achieved at a DA size of 75%. This finding suggested that even with a limited dataset, data augmentation techniques such as `WaveGrad` can still be effective in improving the accuracy of motor imagery classification using EEG signals.

It was worth noting that the results presented in the table were based on a single dataset, and the effectiveness of the `WaveGrad` method varies with other datasets or signal processing tasks. Future studies should investigate the generalizability of the `WaveGrad` method and explore its potential for improving accuracy in other EEG-based classification tasks.

The study concluded that although the `WaveGrad` method's capacity to produce signals with higher valence is affected by the restricted dataset used for its training, the outcomes presented in the table indicated that the method can enhance the precision of the motor imagery classification task using EEG signals. Additional investigation was required to establish the applicability of the `WaveGrad` approach to alternative datasets and signal processing assignments (Table 4).

Table 4. The Average accuracy improvement from our method on DA size level

Baseline Accuracy	DA size	`WaveGrad`	σ	Δ
42.45	25%	51.15	17.61	8.71
	50%	50.56	17.38	8.11
	75%	51.64	17.24	9.19
	100%	50.78	17.72	8.33

Table 5. The Average accuracy improvement from our method.

Model	Baseline Accuracy	`WaveGrad`	σ	Δ
ATCNet	47.68	53.63	18.45	5.94
Deep ConvNet	32.60	52.93	18.49	20.33
EEGITNet	46.84	54.96	16.76	8.12
EEGNet	34.18	37.04	8.00	2.85
ShallowFBCSPNet	50.93	56.62	15.85	5.69

5.4 Complexity of Model

In a low-complexity model, our method improves accuracy more than a high-complexity model because our method improves the similarity of the train set and the test set Table 5. As a result, our method has an impact on low-complexity models. However, high-complexity models are at risk of overfitting on a training set. Hence, our method does not improve accuracy on high-complexity models. However, our method does not guarantee accuracy.

5.5 Limitation

First, the proposed data augmentation method was developed based on a limited dataset, specifically the BCI competition IV dataset 2a. While the method showed promising results on this dataset, its effectiveness might be limited by the size and quality of the dataset used for its development. Future work could involve evaluating the generalizability of the method on other EEG MI datasets to improve its confidence in wider usage scenarios. Second, the proposed data augmentation method was developed specifically for the EEG MI task. It might not necessarily generalize well to other EEG-based tasks. Future work could involve evaluating the method on other EEG signal tasks, such as P300 and SSEVP, to assess its robustness and adaptability across different EEG-based applications.

6 Conclusion

In this study, we investigated the effectiveness of data augmentation techniques on improving the accuracy of standard EEG MI classification models. We used five standard EEG MI classification models to classify EEG signals into left and right hand movements. We evaluated the performance of data augmentation techniques by comparing them with the baseline model, which were trained without any data augmentation. The data augmentation techniques, we used `WaveGrad`, `Noise Addition`, `Frequency Shift`, Fourier transform surrogates, and SmoothTimeMas.

Our results showed that data augmentation techniques improved the performance of standard EEG MI classification models, with `WaveGrad` being the most effective technique. The accuracy of the baseline model was 42.45%, while the accuracy of the model trained with 75% synthetic data generated by `WaveGrad` was 51.64%. The other data augmentation techniques also improved the performance of the models, with `Noise Addition` and `Frequency Shift` being the least effective techniques.

References

1. Wolpaw, J.R.: Brain-computer interfaces. In: Handbook of Clinical Neurology, vol. 110, pp. 67–74. Elsevier (2013)
2. Lawhern, V.J., Solon, A.J., Waytowich, N.R., Gordon, S.M., Hung, C.P., Lance, B.J.: EEGNet: a compact convolutional neural network for EEG-based brain-computer interfaces. J. Neural Eng. **15**(5), 056013 (2018)
3. Bilbao, I., Bilbao, J.: Overfitting problem and the over-training in the era of data: particularly for artificial neural networks. In: Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), pp. 173–177. IEEE (2017)
4. Wang, F., Zhong, S., Peng, J., Jiang, J., Liu, Y.: Data augmentation for EEG-based emotion recognition with deep convolutional neural networks. In: Schoeffmann, K., et al. (eds.) MMM 2018. LNCS, vol. 10705, pp. 82–93. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73600-6_8
5. Parvan, M., Ghiasi, A.R., Rezaii, T.Y., Farzamnia, A.: Transfer learning based motor imagery classification using convolutional neural networks. In: 2019 27th Iranian Conference on Electrical Engineering (ICEE), pp. 1825–1828. IEEE (2019)
6. Li, Y., Zhang, X.R., Zhang, B., Lei, M.Y., Cui, W.G., Guo, Y.Z.: A channel-projection mixed-scale convolutional neural network for motor imagery EEG decoding. IEEE Trans. Neural Syst. Rehabil. Eng. **27**(6), 1170–1180 (2019)
7. Schwabedal, J.T., Snyder, J.C., Cakmak, A., Nemati, S., Clifford, G.D.: Addressing class imbalance in classification problems of noisy signals by using Fourier transform surrogates. arXiv preprint arXiv:1806.08675 (2018)
8. Rommel, C., Moreau, T., Paillard, J., Gramfort, A.: CADDA: class-wise automatic differentiable data augmentation for EEG signals. arXiv preprint arXiv:2106.13695 (2021)
9. Rommel, C., Paillard, J., Moreau, T., Gramfort, A.: Data augmentation for learning predictive models on EEG: a systematic comparison. J. Neural Eng. **19**(6), 066020 (2022)
10. Mohsenvand, M.N., Izadi, M.R., Maes, P.: Contrastive representation learning for electroencephalogram classification. In: Machine Learning for Health, pp. 238–253. PMLR (2020)
11. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems, vol. 33, pp. 6840–6851 (2020)
12. Akrout, M., et al.: Diffusion-based data augmentation for skin disease classification: impact across original medical datasets to fully synthetic images. arXiv preprint arXiv:2301.04802 (2023)
13. Chen, N., Zhang, Y., Zen, H., Weiss, R.J., Norouzi, M., Chan, W.: WaveGrad: estimating gradients for waveform generation. arXiv preprint arXiv:2009.00713 (2020)
14. Song, Y., Garg, S., Shi, J., Ermon, S.: Sliced score matching: a scalable approach to density and score estimation. In: Uncertainty in Artificial Intelligence, pp. 574–584. PMLR (2020)

15. Brunner, C., Leeb, R., Müller-Putz, G., Schlögl, A., Pfurtscheller, G.: BCI Competition 2008-Graz data set A. Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology, vol. 16, pp. 1–6 (2008)
16. Altaheri, H., Muhammad, G., Alsulaiman, M.: Physics-informed attention temporal convolutional network for EEG-based motor imagery classification. IEEE Trans. Ind. Inf. **19**(2), 2249–2258 (2022)
17. Salami, A., Andreu-Perez, J., Gillmeister, H.: EEG-ITNet: an explainable inception temporal convolutional network for motor imagery classification. IEEE Access. **10**, 36672–36685 (2022)
18. Schirrmeister, R.T., Springenberg, J.T., Fiederer, L.D.J., Glasstetter, M., Eggensperger, K., Tangermann, M., et al.: Deep learning with convolutional neural networks for EEG decoding and visualization. Hum. Brain Mapp. **38**(11), 5391–5420 (2017)
19. Lashgari, E., Liang, D., Maoz, U.: Data augmentation for deep-learning-based electroencephalography. J. Neurosci. Methods **346**, 108885 (2020)
20. Leeb, R., Brunner, C., Müller-Putz, G., Schlögl, A., Pfurtscheller, G.: BCI Competition 2008-Graz Data Set B, pp. 1–6. Graz University of Technology, Austria (2008)
21. Terzano, M.G., Parrino, L., Sherieri, A., Chervin, R., Chokroverty, S., Guilleminault, C., et al.: Atlas, rules, and recording techniques for the scoring of cyclic alternating pattern (CAP) in human sleep. Sleep Med. **2**(6), 537–554 (2001)

Thai Conversational Chatbot Classification Using BiLSTM and Data Augmentation

Nunthawat Lhasiw[1], Tanatorn Tanantong[1](✉), and Nuttapong Sanglerdsinlapachai[2]

[1] Thammasat Research Unit in Data Innovation and Artificial Intelligence, Department of Computer Science, Faculty of Science and Technology, Thammasat University, Pathum Thani, Thailand
nunthawat.lha@dome.tu.ac.th, tanatorn@sci.tu.ac.th

[2] Strategic Analytics Networks with Machine Learning and AI Research Team, National Electronics and Computer Technology Center, Pathum Thani, Thailand
nuttapong.sanglerdsinlapachai@nectec.or.th

Abstract. Chatbot platforms, e.g., Facebook and Line, have revolutionized human interaction in the digital age. In order to develop an automatic chatbot classification, there are several challenges especially for Thai chat messages. Conversational messages are usually short and ambiguous. Therefore, it is difficult to find a dataset for constructing an effective classification model. To address the limited size of the dataset, data augmentation techniques can be possibly applied. Data augmentation involves generating synthetic messages by applying various transformations to existing data samples while preserving their original meaning. In this study, the size and diversity of the dataset is increased by two methods, i.e., text augmentation using word2vec from Thai2Fit and English-Thai machine translation models proposed by VISTEC. Based on the augmented messages, a Deep Learning technique, BiLSTM, is used to construct a chatbot classification model. The experimental obtained results demonstrate that data augmentation can help to increase the classification performance.

Keywords: Chatbot Classification · Data Augmentation · Deep Learning · BiLSTM

1 Introduction

As the usage of social media platforms continues to rapidly increase [1], people can freely share their opinions and thoughts [2]. Online platforms e.g., Facebook and Line offer convenient and rapid channels for people to engage in conversations and share information. These platforms have the way people to interact, providing features such as automated question-answering systems and interactive exchanges. Users can communicate seamlessly, regardless of location or time constraints [3]. Conversational messages are typically brief and ambiguous, making it challenging to find a suitable dataset for constructing an effective classification model. One technique that can address the issue of insufficient data is data augmentation, which has been applied and presented in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 127–141, 2023.
https://doi.org/10.1007/978-981-99-7969-1_10

several studies [4–7]. Data augmentation serves to enhance both the size and quality of the training data, thereby improving the model's generalization capabilities. Some of the techniques involved in data augmentation include back translation and word embedding. More specifically, back translation relies on a transformer architecture, encompassing an encoder-decoder model. The paraphrasing approach utilizes machine translation models that translate text into an intermediate language and then back into the original language. In our approach, the original text is input into a Thai-English translation model, followed by an English-Thai mixture of experts translation model [2]. Another robust augmentation method is Word2vec, which employs a word embedding model trained on a public dataset to identify the most similar words for a given input word. This technique is referred to as Word2vec-based (learned semantic similarity) augmentation [10].

Moreover, deep learning techniques have shown great potential to understand the meaning of the sentence which is written in different forms [11]. One such technique is the Bidirectional Long Short-Term Memory (BiLSTM) model [8, 12], which extends the capabilities of the traditional Recurrent Neural Network (RNN) [9]. BiLSTM excels in capturing and understanding the meaning of both short and long text sequences, making it highly suitable for tasks such as text classification. By leveraging the power of BiLSTM, chatbots can better analyze and interpret user inputs, delivering more accurate and contextually relevant responses [10]. In this study, we have conducted a study on Thai Conversational Chatbot Classification but we have the small sample of chat message. To address this problem, we propose a method for Thai Conversational Chatbot Classification by employing the potential of data augmentation techniques to solve the small sample of chat message and deep learning models; namely, BiLSTM.

2 Related Works

2.1 Conversational Chatbot Classification

Many researchers have proposed deep learning models for classifying textual data. Among the related works, Maktapwong, P. and colleagues [11] introduced a chatbot application for breast cancer patients in Thailand. This application aims to increase communication channels and address issues related to a shortage of medical staff. In their study, they employed the deep learning model BiLSTM to construct chatbot conversation messages from breast cancer patients. They chose BiLSTM because it can effectively capture contextual information in chatbot conversations and demonstrated efficiency in text processing. The reported accuracy of their experiments was 86.90%.

Anki, P. and colleagues [9] proposed LSTM and BiLSTM models for classifying chatbot messages, implemented using the Python programming language. They subsequently evaluated the performance of both LSTM and BiLSTM models. Furthermore, they discussed the attributes of LSTM, which can learn data over long distances. They highlighted that combining LSTM models can lead to enhanced learning, as it enables data processing in two directions, encompassing both past and future information. Based on the experimental results, the deep learning BiLSTM model demonstrated outstanding performance in classifying chatbot messages, achieving an accuracy rate of 99.52%.

L. Shi and their team [8] introduced a BiLSTM model designed for understanding the context of dialogues in online chatbots. They collected the dataset for their study using the Scrapy tool, which extracted data from open-source projects like AngularJS, Bootstrap, and Chromium, yielding a total of 65,428 dialogues. The model evaluation demonstrated that the approach employed in the study successfully met its objectives, achieving an average precision, recall, and F1-score of 88.52%, 88.50%, and 88.51%, respectively.

R. Anhar and their colleagues [12] conducted a study using a BiLSTM model for question classification. Question classification plays a crucial role in question-answer systems as it directly impacts the accuracy of generated answers. Traditional approaches, such as Support Vector Machine (SVM), pattern matching, naive Bayes classification, and Latent Dirichlet Allocation (LDA), have been utilized for question classification. However, they are constrained by specific sentence patterns. To overcome these limitations, deep learning methods, including Bidirectional Long Short Term Memory (BiLSTM), have been proposed. In this study, the effectiveness of BiLSTM in question classification was evaluated, achieving an accuracy of 90.90% with a loss of 31.60%. Based on Literature reviews above, a Bi-LSTM model is potential effective model for classifying texts, particularly in the context of chat messages.

2.2 Text Classification Using Data Augmentation

There are various approaches to working with language in computers, such as employing back translation, substituting words with synonyms, or utilizing word embeddings. However, data augmentation in text-based tasks and natural language processing (NLP) remains a challenge. In related research, Beddiar, D.R and their team [2] proposed a data augmentation method to address issues related to insufficient datasets. Their method primarily utilized back translation, which was based on a transformer architecture incorporating an encoder-decoder model. The original text was processed through an English-French translation model, followed by a French–English mixture of experts' translation model. The study involved five original datasets, which included AskFM, Formspring, Olid, Warner and Waseem, and Wikipedia Toxic. The results were reported in terms of accuracy, F1 score, precision, and recall. Notably, they achieved a high recall score of 99.7% and a precision of 99.6% for the expanded version of the Warner and Waseem dataset. In summary, the best accuracy and F1 score, at 99.4%, were recorded for the expanded Wikipedia toxic comments dataset.

Phreeraphattanakarn, T. and their colleagues [4] encountered limitations in the available data, as well as unequal data group sizes within an automatic chatbot dataset. Through their study, the researchers discovered that data augmentation techniques can significantly enhance model performance. They employed this data augmentation technique to expand the training dataset, which primarily consisted of text data. Initially, the dataset comprised 339,985 pairs of sentences. The researchers applied data augmentation by leveraging word embeddings from the pre-trained Thai2fit model and cosine similarity to identify similar words within sentences. This approach resulted in a substantial increase in the number of sentence pairs, totaling 1,329,197. The results of the model's performance evaluation on the testing dataset and the augmented dataset yielded F1 scores of 0.088 and 0.071, respectively.

Ma, J. and their colleagues [5] employed data augmentation techniques to address limitations in their dataset by applying Back Translation and EDA (Easy Data Augmentation) to Chinese language data. The Chinese dataset comprised toxic comments obtained from the Kaggle platform. In their approach, the researchers translated the Chinese texts into English and then back-translated them to the original language. For this study, deep learning models, specifically LSTM and CNN, were utilized to classify toxic comments. The models' performance was assessed based on their accuracy, and the experimental results clearly demonstrated that the back translation technique significantly improved the effectiveness of the models.

Rizos, G., and their colleagues [6] aimed to tackle the challenge of augmenting text data for hate speech classification by introducing three text-based data augmentation techniques tailored specifically for text. These techniques encompass synonym replacement based on word embedding vector similarity. Deep learning approaches, particularly those employing word embedding techniques, have proven to be more effective in hate speech classification. The proposed framework exhibits a substantial improvement in hate speech detection, surpassing the baseline performance in the largest online hate speech database by an absolute 5.7% increase in Macro-F1 score and a 30% boost in hate speech class recall.

Fadaee et al. [7] have demonstrated that Back Translation (BT) can serve as a valuable method for expanding datasets without necessitating modifications to the training process of language translation models. The paper presents experimental results for the WMT news translation task, with a specific focus on German-English and English-German translation pairs. Furthermore, the study uncovered that the inclusion of synthetic data proves advantageous, particularly for words exhibiting high prediction loss during training.

3 Methodology

This section presents the process of data collection and preparation. Following that, we expanded the original dataset using two methods: text augmentation using word2vec from Thai2Fit (Data Augmentation by PyThaiNLP) and English-Thai machine translation models proposed by VISTEC (Data Augmentation by VISTEC). Next, we applied feature extraction. Lastly, we utilized the processed data to construct the classification model. Figure 1. Demonstrates a framework for Thai Conversational classification.

3.1 Data Collection and Preparation

This research collected chat text data from the office of the registrar, Thammasat University, with a specific focus on chat messages related to topics such as 'Document Request,' 'Registration,' 'Payment/Invoice,' 'Graduate Registration,' 'Grade,' and 'Student Profile.' The data collection period extended from March to April and from August to October 2021, resulting in a total of 3,609 chat messages. Each message was meticulously categorized by experts. Table 1 presents an example of messages and their corresponding classes.

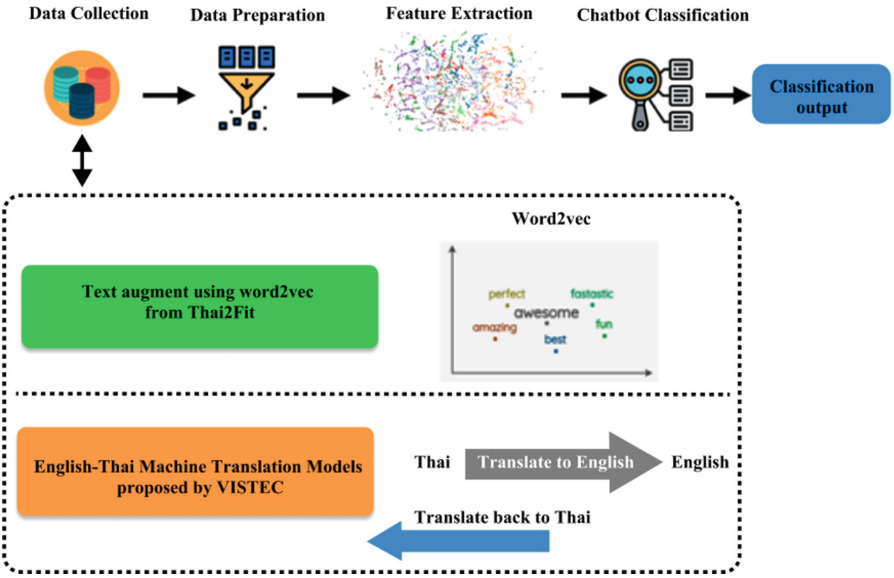


Fig. 1. Thai Conversational Chatbot Classification Framework.

Furthermore, data transformation is an important step in preparing the data for building a neural network for text classification in chatbot messages. In this study, data processing involves two main processes: 1) Data Transformation on a Message and 2) Data Transformation on a Class Label.

Regarding Data Transformation on a Message, encoding the data requires several preprocessing steps to clean the text and make it more suitable for subsequent analysis. These processes include:

1. Removal of special characters from a message: This step involves eliminating special characters such as ‘?’, ‘#’, and ‘,’ from the text.
2. Word Tokenization: The text is segmented into individual words to facilitate further processing.
3. Stop words removal: In this step, insignificant words, known as stop words, are removed from the text.
4. Indexing: The text is indexed, enabling efficient data retrieval and analysis.
5. Zero padding: All input sequences are adjusted to match the length of the longest text by replacing shorter sequences with zeros.

In terms of Data Transformation on a Class Label, Anhar [12] stated that the conversion of text categories into binary vectors using the one-hot encoding principle facilitates text categorization. This transformation represents each text category as a binary list, where the value 1 indicates membership in that specific category. The size of the text vector corresponds to the total number of categories. An example of data transformation on a class label is provided in Table 2.

Table 1. Chat messages annotated by experts.

Message	Class
อาจารย์คะพอจะมีไฟล์ใบขึ้นทะเบียนนักศึกษามีไหมคะ (Teacher, do you have the student registration form file?)	'Document Request'
ต้องลงทะเบียนวันทีลงล่าช้าหรอครับ (Do I need to register on the late date?)	'Registration'
แล้วใบเสร็จก็คือเอาไปเบิกของค่าราชการได้เลนมัยคะ (Can be used the receipt to withdraw government expenses?)	'Payment/Invoice '
จบแล้วค่ะ แต่ไม่มีปุ่มขึ้นบัณฑิตซักทีเลยค่า (There is no graduate button.)	'Graduate registration'
โดยปกติต้องมีเวลากำหนดการส่งเกรดมายังสำนักงานทะเบียนใช่ไหมครับ เพราะหากเด็กลงซัมเมอร์คะแนนจะต้องออกก่อนลงทะเบียนเรียน (Is there time to submit grades? Because, they must know the grade if a student wants registering on summer.)	'Grade'
นักศึกษาใหม่กรอกประวัติผิด (New students filled in wrong information)	'Student Profile'
นักศึกษารหัส 64 อยากทราบว่าใช้เวลานานมั้ยคะ กว่าสถานะรอขึ้นทะเบียน จะเปลียนเป็นปกติ พอดีมีความจำเป็นต้องยืมไอแพด และหนังสือจากห้องสมุดเร่งด่วนค่ะ (Student code 64, would like to know when the status is pending registration will change to normal. It is necessary to borrow an iPad and books from the library)	'Student Status'

Table 2. Example data transformation with annotated label.

Class	Label encoding						
'Document Request'	1	0	0	0	0	0	0
'Registration'	0	1	0	0	0	0	0
'Payment/Invoice'	0	0	1	0	0	0	0
'Graduate registration'	0	0	0	1	0	0	0
'Grade'	0	0	0	0	1	0	0
'Student Profile'	0	0	0	0	0	1	0
'Student Status'	0	0	0	0	0	0	1

3.2 Data Augmentation

We performed data augmentation on the original dataset using 2 methods: 1. text augmentation using word2vec from thai2fit (data augmentation by pythainlp) [13] and 2. english-thai machine translation models proposed by vistec (data augmentation by VISTEC) [14].

3.2.1 Text augmentation using word2vec from Thai2Fit (Data Augmentation by PyThaiNLP)

Text augmentation using word2vec from Thai2Fit (Data Augmentation by PyThaiNLP) [13] involved the application of the word2vec principle. Word2vec is a word embedding technique or vectorization method that assigns vectors to words, aiding in the search for words with similar meanings. An illustrative example of data augmentation using the Thai2Fit method is provided in Table 3.

Table 3. An example of increasing the amount of data using Data Augmentation by PyThaiNLP.

Class	Original Messages	Augmented Messages
‘Document Request’	อาจารย์คะพอจะมีไฟล์ใบขึ้นทะเบียนนักศึกษามีไหมคะ (Teacher, do you have the student registration form file?)	อาจารย์คะพอจะมีไฟล์ใบขึ้นทะเบียนนักศึกษามีไหมคะ (Dear teacher, may I inquire if you have the file for the student registration form?)
‘Registration’	ต้องลงทะเบียนวันทีลงล่าช้าหรอครับ (Do I need to register on the late date?)	ต้องลงทะเบียนวันทีลงล่าช้าหลิมครับ (You must register on the date of late entry.)
‘Payment/Invoice ‘	แล้วใบเสร็จก็คือเอาไปเบิกของค่าราชการได้เลยมั้ยคะ (Can be used the receipt to withdraw government expenses?)	แล้วrocksก็คือเอาไปเบิกของค่าราชการได้เลนเดี๋ยวคะ (Then the rocks are that you can take it to pay for government expenses. You can wait.)

3.2.2 English-Thai Machine Translation Models Proposed by VISTEC (Data Augmentation by VISTEC)

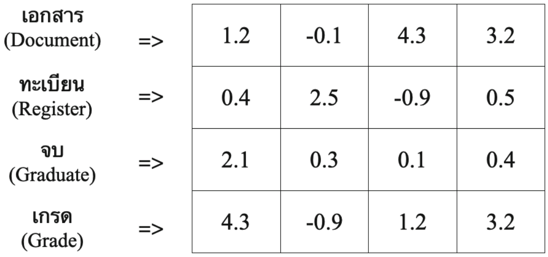
English-Thai machine translation models proposed by VISTEC (data augmentation by VISTEC) [14] is a backtranslation method. The backtranslation method is the process of translating the original language into English and then translating it back into the original language to generate new sentences. An example of original messages and augmented messages generated by data augmentation by VISTEC can be seen in Table 4.

Table 4. Original messages and augmented messages generated using data augmentation by VISTEC.

Class	Original messages	Augmented messages
‘Document Request’	อาจารย์คะพอจะมีไฟล์ใบขึ้นทะเบียนนักศึกษามีไหมคะ (Teacher, do you have the student registration form file?)	คุณมีใบอนุญาตสำหรับอาจารย์ของคุณหรือไม่? (Will you have to make your teacher?)
‘Registration’	ต้องลงทะเบียนวันที่ลงล่าช้าหรอครับ (Do I need to register on the late date?)	ฉันต้องลงทะเบียนสำหรับวันที่ล่าช้าหรือไม่? (Do I have to register for the late date?)
‘Payment/Invoice ‘	แล้วใบเสร็จก็คือเอาไปเบิกของค่าราชการได้เลนมั้ยคะ (Can the receipt be used to withdraw from the government?)	แล้วใบเสร็จจะถูกนำไปหักเป็นค่าบริการอย่างเป็นทางการมั้ยคะ (And will the receipt be deducted as an official service fee?)

3.3 Feature Extraction

Feature extraction holds a crucial role in text classification because it directly influences classification accuracy [15]. Word Embedding is a technique used to represent words in a distributed manner, improving the accuracy of natural language processing models. It involves extracting features from a specific type of word to enhance the efficiency of neural networks in classifying diverse data categories [16]. Figure 2 shows the example of feature extraction using word embedding principle.



เอกสาร (Document)	=>	1.2	-0.1	4.3	3.2
ทะเบียน (Register)	=>	0.4	2.5	-0.9	0.5
จบ (Graduate)	=>	2.1	0.3	0.1	0.4
เกรด (Grade)	=>	4.3	-0.9	1.2	3.2

Fig. 2. Features extracted by word embedding principle.

3.4 Chatbot Classification

Figure 3 illustrates the process of classifying Thai conversational chatbot messages. The initial step involves preparing the Thai Conversational Chatbot Messages. Data augmentation is performed using two distinct methods: one proposed by VISTEC and another by Py-ThaiNLP. The method by VISTEC utilizes the back-translation concept,

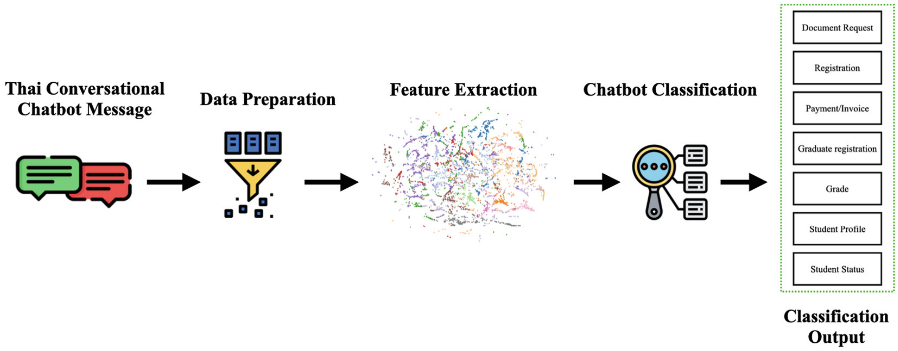


Fig. 3. Thai conversational chatbot messages classification processes.

employing the Transformer model for this purpose. The transformer functions as an encoder-decoder network. In this setup, the encoder captures pertinent information from an input sentence, while the decoder uses this information to generate an output sentence. This architecture is typically used for tasks like translation. On the other hand, the method by PyThaiNLP employs the word2vec concept, which assigns vector representations to words, facilitating the identification of words with similar meanings.

In the data preparation step, Thai conversational chatbot messages are converted into vectors. We then extract features from these vectors using the word embedding principle. The extracted features are utilized as input for our deep neural model, which aims to classify Thai conversational chatbot messages. In the chatbot classification phase, we employ BiLSTM. LSTM (Long Short-Term Memory) follows the basic architecture of RNN (Recurrent Neural Network) but effectively addresses one of the drawbacks of Simple RNN by learning long-term dependencies in text datasets [16]. BiLSTM shares the same underlying concept as LSTM [17] but adds bidirectional propagation. This means that the BiLSTM architecture learns from both past to future directions. The backward propagation layer essentially functions as the reverse of the forward LSTM, making the architecture more stable and capable as it operates from both ends. Finally, the following section will calculate and analyze the obtained results. You can observe the model architecture of BiLSTM for text classification in Fig. 4.

4 Experimental Settings and Results

4.1 Experimental Settings

In this study, the classification models are trained from 7 datasets, i.e., 1. Original dataset, 2. VISTEC dataset (Augmenting Original dataset with data augmentation by VISTEC), 3. VISTEC to VISTEC dataset (Augmenting VISTEC dataset with data augmentation by VISTEC), 4. VISTEC to Thai2fit dataset (Augmenting VISTEC dataset with data augmentation by PyThaiNLP), 5. Thai2Fit (Augmenting Original dataset with data augmentation by PyThaiNLP) and, 6. Thai2Fit to Thai2Fit dataset (Augmenting the Thai2Fit

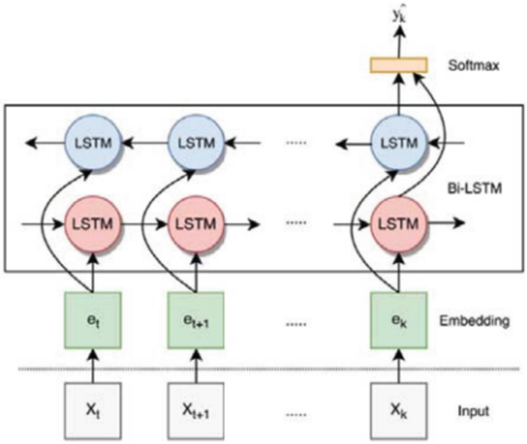


Fig. 4. The network structure of BiLSTM.

dataset with Data Augmentation by PyThaiNLP), and 7. Thai2Fit to VISTEC (Augmenting Thai2Fit dataset with data augmentation by VISTEC). The performance of the classification models is evaluated using well-established metrics, including precision, recall, F-value, and accuracy. Our results encompass outcomes from two key aspects: the 10-fold cross-validation and testing on an independent test dataset. The dataset used in this research comprises conversation texts exchanged between students and university staff. These conversations span a range of topics, including 'Document Request,' 'Registration,' 'Payment/Invoice,' 'Graduate registration,' 'Grade,' 'Student Profile,' and 'Student Status.' For more detailed information, please refer to Table 5, which also illustrates the extent of dataset expansion achieved through data augmentation by PyThaiNLP and VISTEC.

This research study has divided the data into 3 parts. The first two parts were used for model training, with 80% of the dataset. The remaining part was reserved for model testing, utilizing 20% of the dataset. In this study, we investigate the parameter settings of a classification model, specifically examining the values used for various parameters. These crucial parameters include the number of nodes in the BiLSTM layer, epoch, batch_size, dropout rate, loss function, and optimizer. The choice of these settings significantly influences the performance and effectiveness of the classification model. Our goal is to gain insights into how the values assigned to each parameter impact the overall performance of the model. You can find the specific values of each parameter in Table 6. Referring to Table 6, the number of nodes in the BiLSTM layer represents the number of output nodes within the layer. In our model, we selected 100 nodes for this purpose. It's worth noting that many studies utilize specific hyperparameter settings for building textual classification models. In our case, we adopted hyperparameter values from these studies as well. For instance, Maktapwong and colleagues [11] used 20 epochs to train their model. An epoch represents an iteration over the entire set of feature and label data.

Table 5. Size of datasets before and after expansion using data augmentation by PyThaiNLP, data augmentation by VISTEC, and a combination of original dataset with both techniques.

Class	Original dataset	VISTEC dataset	VISTEC to VISTEC dataset	VISTEC to Thai2fit dataset	Thai2Fit dataset	Thai2Fit to Thai2fit dataset	Thai2Fit to VISTEC dataset
Document Request	951	2222	8129	10181	34962	5330	17142
Registration	146	1904	5837	9814	32966	4569	16226
Payment/Invoice	713	1426	4424	9646	32426	3422	15661
Graduate registration	129	888	3164	8912	30536	2075	15157
Grade	1107	250	720	8509	29075	701	14328
Student Profile	444	292	1094	8473	28609	619	14117
Student Status	119	258	565	8511	28664	535	14179
Total	3609	7240	23933	64046	217238	17251	106810

Table 6. Hyperparameter setting value of the classification model.

Hyperparameter	Original dataset	VISTEC dataset	VISTEC to VISTEC dataset	VISTEC to Thai2fit dataset	Thai2Fit dataset	Thai2Fit to Thai2fit dataset	Thai2Fit to VISTEC dataset
number of nodes in BiLSTM layer	100						
epoch	20						
batch_size	128						
dropout rate	0.5						
loss function	categorical_crossentropy						
optimizer	Adam						

Anki and colleagues [9] employed a batch_size of 128 in their model. Batch size represents the number of training data instances used in a single iteration. Additionally, Ma and colleagues [5] utilized a dropout rate of 0.5. The dropout rate is a crucial parameter set to prevent overfitting in neural networks. They also employed the categorical_crossentropy loss function in combination with the Adam optimizer to construct their text classification model.

4.2 Experimental Results

In this section, we present the comprehensive results of our experiments, including two key components: the outcomes of 10-fold cross-validation and the results obtained from testing on the independent test dataset. The experiment results of the 10-fold cross-validation encompass the following 7 experiments, as follows: 1. Original dataset, 2. Original dataset + VISTEC dataset (augment1 dataset), 3. Original dataset + VISTEC to VISTEC dataset (augment2 dataset), 4. Original dataset + VISTEC to Thai2fit dataset (augment3 dataset), 5. Original dataset + Thai2Fit (augment4 dataset), 6. Original dataset + Thai2Fit to Thai2Fit dataset (augment5 dataset), and 7. Original dataset + Thai2Fit to VISTEC (augment6 dataset). The experiment results for the test dataset, which was evaluated using 7 test datasets, are presented in Table 8, as follow: 1. Original dataset, 2. VISTEC dataset (augment1 dataset), 3. VISTEC to VISTEC dataset (augment2 dataset), 4. VISTEC to Thai2fit dataset (augment3 dataset), 5. Thai2Fit dataset (augment4 dataset), 6. Thai2Fit to Thai2fit dataset (augment5 dataset) and 7. Thai2Fit to VISTEC dataset (augment6 dataset). Additionally, precision, recall, F-value, and accuracy metrics are utilized to evaluate the performance of chat message classification model.

From Table 7, a model trained with the original dataset exhibits excellent performance across multiple metrics, with precision at 97.01%, recall at 96.91%, F-score at 96.91%, and accuracy at 96.92%. To address the limitations of the original dataset's size, we increased its size by concatenating it with augmented datasets. When we combined the original data with the augment1 dataset, we observed a slight decrease in performance. However, merging the original dataset with the augment2 dataset resulted in a slight improvement, with precision at 94.51%, recall at 94.45%, F-score at 94.45%, and accuracy at 94.46%. Further enhancements in performance were achieved when joining the original dataset with the augment3 dataset. This combination yielded the highest precision (95.67%), recall (95.64%), F-score (95.64%), and accuracy (95.63%). Additionally, combining the original dataset with the augment5 dataset increased the volume and led to improved performance, with precision at 97.65%, recall at 97.63%, F-score at 97.63%, and accuracy at 98.38%. However, there was a significant drop in performance when integrating the original dataset with the augment6 dataset. Clearly, the highest performance was achieved when combining the original dataset with the augment4 dataset. This combination resulted in significantly improved precision (98.40%), recall (98.38%), F-score (98.38%), and accuracy (98.38%). For further performance evaluation, we also reported the results of using a test dataset to assess the model, as shown in Table 8.

According to Table 8, we evaluated the model using the test dataset for each variant. The original dataset exhibited moderate performance, achieving precision at 74.57%, recall at 72.57%, F-score at 73.00%, and accuracy at 80.00%. When we used the augment1 dataset for testing with the model trained from the original dataset and the augment1 dataset, there was an improvement in performance. Precision reached 79.00%, recall stood at 78.29%, F-score reached 78.71%, and accuracy reached 83.00%. Testing the model trained from the original dataset and the augment2 dataset with the augment2 dataset further enhanced the results. Precision increased to 82.14%, recall improved to 78.00%, F-score rose to 79.43%, and accuracy reached 84.00%. Evaluating the model trained with the original dataset and the augment3 dataset with the augment3 dataset yielded a precision of 74.00%, recall of 84.00%, F-score of 79.00%, and accuracy of

Table 7. Experiment results of 10 folds cross-validation.

Models trained with each dataset	Evaluation Matrices			
	Precision	Recall	F-score	Accuracy
original dataset	97.01%	96.91%	96.91%	96.92%
original dataset + augment1 dataset	94.50%	94.28%	94.28%	94.23%
original dataset + augment2 dataset	94.51%	94.45%	94.45%	94.46%
original dataset + augment3 dataset	95.67%	95.64%	95.64%	95.63%
original dataset + augment4 dataset	**98.40%**	**98.38%**	**98.38%**	**98.38%**
original dataset + augment5 dataset	97.65%	97.63%	97.63%	97.63%
original dataset + augment6 dataset	97.08%	97.06%	97.06%	97.06%

Table 8. Experiment results of the test dataset.

Models trained with each dataset	Evaluation Matrices			
	Precision	Recall	F-score	Accuracy
original dataset	74.57%	72.57%	73.00%	80.00%
augment1 dataset	79.00%	78.29%	78.71%	83.00%
augment2 dataset	82.14%	78.00%	79.43%	84.00%
augment3 dataset	74.00%	84.00%	79.00%	80.00%
augment4 dataset	**87.29%**	**86.71%**	**86.57%**	**91.00%**
augment5 dataset	86.14%	85.43%	85.57%	89.00%
augment6 dataset	76.86%	75.57%	76.00%	80.00%

80.00%. The highest performance was observed when testing the model from the original dataset and the augment4 dataset with the augment4 dataset. It achieved precision at 87.29%, recall at 86.71%, F-score at 86.57%, and accuracy at 91.00%. Similarly, the model trained from the original dataset and the augment5 dataset maintained consistently high performance, with precision at 86.14%, recall at 85.43%, F-score at 85.57%, and accuracy at 89.00%. Finally, testing the model from the original dataset and the augment6 dataset with the augment6 dataset resulted in precision at 76.86%, recall at 75.57%, F-score at 76.00%, and accuracy at 80.00%.

5 Conclusion

In this paper, we conducted a study on Thai conversational chatbot classification, despite having a small sample of chat messages. To address this limitation, we employed data augmentation methods, specifically Data Augmentation by PyThaiNLP and Data Augmentation by VISTEC, to expand the size of the datasets. We utilized the deep learning model, BiLSTM, for chat message classification. The experimental results, derived from the original dataset augmented with the Thai2Fit dataset using 10-fold cross-validation, showcased exceptional performance with a precision of 98.40%, recall of 98.38%, F-score of 98.38%, and accuracy of 98.38%. Moreover, the results from testing on the test dataset using 10-fold cross-validation yielded a precision of 87.29%, recall of 86.71%, F-score of 86.57%, and accuracy of 91.00%. Based on the experimental findings for Thai Conversational Chatbot Classification, it is evident that data augmentation by PyThaiNLP consistently outperforms data augmentation by VISTEC. For future work, we plan to explore different datasets and data augmentation methods to further enhance the performance of chatbot classification models.

References

Tanantong, T., Parnkow, M.: A survey of automatic text classification based on Thai social media data (2022). https://doi.org/10.4018/IJKSS.312578

Beddiar, D.R., Jahan, M.S., Oussalah, M.: Data expansion using back translation and paraphrasing for hate speech detection. 53 (2021). https://doi.org/10.1016/j.osnem.2021.1001

Lhasiw, N., Sanglerdsinlapachai, N., Tanantong, T.: A bidirectional LSTM model for classifying chatbot messages 173 (2021). https://doi.org/10.1109/iSAI-NLP54397.2021.9678

Phreeraphattanakarn, T., Kijsirikul, B.: Text data-augmentation using text similarity with manhattan siamese long short-term memory for Thai language (2021). https://doi.org/10.1088/1742-6596/1780/1/012018

Ma, J., Li, L.: Data augmentation for Chinese text classification using back-translation (2020). https://doi.org/10.1088/1742-6596/1651/1/012039

Rizos, G., Hemker, K., Schuller, B.: Augment to prevent (2019). https://doi.org/10.1145/3357384.3358040

Fadaee, M., Monz, C.: Back-translation sampling by targeting difficult words in neural machine translation (2018). https://doi.org/10.18653/v1/D18-1040

Shi, L., Xing, M., Li, M., Wang, Y., Li, S., Wang, Q.: Detection of hidden feature requests from massive chat messages via deep siamese network (2020). https://doi.org/10.1145/3377811.3380356

Anki, P., Bustamam, A.: Measuring the accuracy of LSTM and BiLSTM models in the application of artificial intelligence by applying chatbot programme (2021). https://doi.org/10.11591/ijeecs.v23.i1

Gong, X., Ying, W., Zhong, S., Gong, S.: Text sentiment analysis based on transformer and augmentation. https://doi.org/10.3389/fpsyg.2022.906061 (2022)

Maktapwong, P., Siriphornphokha, P., Tubglam, S., Imsombut, A.: message classification for breast cancer chatbot using bidirectional LSTM (2022). https://doi.org/10.1109/ITC-CSCC55581.2022.9895035

Anhar, R., Adji, T.B., Akhmad Setiawan, N.: question classification on question-answer system using bidirectional-LSTM (2019). https://doi.org/10.1109/ICST47872.2019.9166190

Source code for pythainlp.augment.word2vec.thai2fit. pythainlp.augment. word2vec.thai2fit - PyThaiNLP 4.0.2 documentation. https://pythainlp.github.io/dev-docs/_modules/pythainlp/augment/word2vec/thai2fit.html

Data Augmentation in NLP in a world that craves data, with Back Translation

Dzisevic, R., Sesok, D.: text classification using different feature extraction approaches (2019). https://doi.org/10.1109/eStream.2019.8732167

Pham, T.-H., Le-Hong, P.: End-to-end Recurrent Neural Network Models for Vietnamese Named Entity Recognition: Word-level vs. Character-level (2017)

Recurrent neural networks for prediction - lagout.org. https://doc.lagout.org/science/0_Computer%20Science/3_Theory/Neural%20Networks/Recurrent%20Neural%20Networks%20for%20Prediction.pdf

Wang, Y., Huang, M., zhu, xiaoyan, Zhao, L.: Attention-based LSTM for aspect-level sentiment classification (2016). https://doi.org/10.18653/v1/D16-1058

DeepFake Detection Using Deep Learning

Mariam Rehman[1](✉), Mehran Rasool[1], and Sadaf Safder[2]

[1] Government College University, Faisalabad, Pakistan
mariamrehman@gcuf.edu.pk
[2] National Center in Big Data and Cloud Computing, University of Agriculture, Faisalabad, Pakistan

Abstract. A deepfake is a computer-generated video or image where one person's face is replaced with another face of a person which uses a generative adversarial network (GAN) to create and alter images that are practically impossible for humans to distinguish from authentic ones. The development of GAN technology has led to significant improvements in image generation. This progress has made it difficult for humans to differentiate between generated images and raised concerns about their misuse and also creating security threats for the society. Deepfake image detection is an essential challenge because the current methods often face inaccuracy and time-consuming issues. The research work employed GAN discriminators to address these issues. The experiment used the CelebA dataset, which contained diverse celebrity images. Moreover, the proposed model consists of components i.e., generator and discriminator. The generator creates fake data from the training images, while the discriminator distinguishes real from the fake images. The result shows that the proposed approach is out-performing with an accuracy of 95.8% in identifying deep fake images, even in the presence of various manipulations. The proposed model is helpful for enhancing trust and security in the society.

Keywords: DeepFake · GAN · Deep Learning

1 Introduction

In today's digital era, the manipulation of visual content has become incredibly advanced. Deep fake technology uses special computer programs known as Generative Adversarial Networks (GANs). These programs can create computer-generated images and videos that resemble real ones. Earlier, the technology could be used to create art and entertainment that raises serious challenges [1]. Deepfake technology involves the use of convolutional neural networks, particularly Generative Adversarial Networks (GANs), to create fake images or videos by replacing one person's face with another's person. Recently, social media issues have highlighted instances in which celebrities' faces were illicitly swapped into explicit content, causing damage to their reputations and long-term harm to the identities of well-known persons and common people in society [2].

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 142–154, 2023.
https://doi.org/10.1007/978-981-99-7969-1_11

GAN model was first introduced by Ian J. Goodfellow in 2014 [2]. GAN was described as a connection between two components i.e., discriminator named as "D" and the generator named as "G." GAN method is an effective way to generate images. Over time, more advanced GANs such as BEGAN [3], PGGAN [4], StyleGAN [5], and StyleGAN2 [6]emerged. These advanced GANs have been successful in creating images for numerous purposes, such as image enhancement, translation, and image filling in missing parts.

The images generated by GANs are really hard to detect with the human eyes. This level of realism can provide an in-depth insight into particular issues especially when GAN-generated images are utilized for deception purposes. When these images are used to create fake news or to deceive ordinary people on social media by generating fake personal information, several serious challenges arise. Firstly, there are ethical problems, as the planned spread of incorrect information may affect public perception and damage trust in digital content. Secondly, there may be legal implications because these activities may violate laws governing misinformation, defamation, or identity theft. Finally, from a security aspect, GAN-generated images exploited for illicit purposes may lead to various problems, including financial scams and identity theft [7].

There are various detection algorithms have been developed to address the issues mentioned earlier which can be grouped into two categories i.e., Conventional and Deep learning methods. Conventional methods have been developed for extracting image properties such as texture and structural features which rely on manually created methodologies These methods offer a simple and relatable path for detecting fake images [8–11]. Although, the conventional methods have various drawbacks which require large processing resources to provide excellent performance across various scenarios. In comparison, deep learning methods have gained the interest of researchers due to their excellent outcome in image classification. Deep learning methods [12–16] have the advantage of being less computationally complex compared to conventional methods, but they lack interpretability. These methods require a large amount of data to train efficiently and are susceptible to overfitting, which can limit their practical application. Therefore, there is still a need for continual efforts to develop an efficient method for GAN-generated image detection.

GANs were originally designed for image creation, with a generator network producing synthetic images and a discriminator network discriminating real from fake images. Which involves training a GAN on a dataset that contains both real and GAN-generated fake images. The discriminator network is trained to distinguish between real and GAN-generated images, a process known as adversarial training. This rigorous training improves the network's capacity to identify DeepFakes precisely. Finally, the key contributions of this research are demonstrated by systematic experimentation, which evaluates the performance, accuracy, and reliability of the GAN-based detection model.

The experiment introduces a novel CNN-GAN based architecture designed to detect deep fake images. This innovative approach enhances the capacity to identify manipulated images by training on a preprocessed dataset to reduce the threats of deep fakes. Moreover, it provides an important tool for identifying the real images.

This research explores the use of celebrity image datasets in deepfake detection. The research improves the accuracy of identifying modified images by leveraging renowned celebrities. To employ these datasets provides a model for protecting celebrities from identity theft within deep fakes content.

This study contributes by thoroughly evaluating the proposed CNN-based GAN model's efficiency in detecting DeepFakes images.

The paper is organized in the following manner. Section 2 provides the detailed review of literature on GAN and its types. Section 3 presents the methodology and discusses the proposed detection model of the research work. Section 4 provides details about results and discussion about the experiments performed. Finally, the Section 5 describes conclusion and future work related to the research work.

2 Literature Review

GANs (Generative Adversarial Networks) are type of deep learning model that has revolutionized the field of generative modelling. GANs are used to create new data that is similar to a training dataset. Unlike other generative models that use explicit probability distributions, GANs learn to generate data by training two types of neural networks: a generator network and a discriminator network [17].

Mirsky et al. [18] examined DeepFakes by studying reenactment techniques such as manipulating facial expressions, lips, posture, or even an entire body, as well as conversion methods such as face swapping or transferring. Verdoliva et al. [19] provided an overview, distinguishing between conventional methods based on sensor-based, model-based, and supervised methods, as compared to methods based on deep learning techniques such as vCNN models. Shobhit et al. [20] investigated several images and video manipulation methods i.e., popular methods, and forgery detection methods. Syed Sadaf Ali et al. [21] designed advanced deep learning algorithms to detect double image compression forgeries, whereas Huang et al. [22] used data augmentation and single sample clustering to improve FakeLocator's detection of various DeepFake methods.

Table 1 illustrates s a comparison of several GAN types, considering their architectural structure, activation functions, number of layer, advantages, and limitations. The table demonstrates that by implementing a configuration consisting of 4 convolutional and de-convolutional layers, along with the activation functions Tanh, ReLU, and Leaky ReLU, the proposed model outperforms compared to others, achieving an accuracy of 95%.

Table 1. Comparison of GAN and its types

GAN Types	Year	Architecture	Activation function	Layers	Advantages	Limitations	Cite
GAN	2014	Multilayer perceptron	Mixture of ReLU and Sigmoid	1 intermediate layer	Handles a well-defined probability distribution, Rapidly generates samples	Vanishing gradient effect, the model performance	[1]
fGAN	2015	Multilayer perceptron	ReLU activation, sigmoid at final layer	5 layers	f-GAN framework based on the f-divergence approach	Stability of various f-divergence functions has not been specified	[2]
WGAN-GP	2017	Multilayer perceptron	ReLU	12 layers	Increase Image quality, improve gradient descent for the model	Excessive clipping cause gradient issue	[3]
Least Squares GAN	2017	Multilayer perceptron	ReLU and LeakyReLU	7 d-conv layers and 3-conv	Gradient stability, improved diversity mode	Use the decision boundary that effect image quality	[4]
LS-GAN	2020	Multilayer perceptron	Did not use sigmoid	maxpooling replaced strides	Address the model collapse issue	Lower image quality	[5]
Proposed Model	2023	Multilayer perceptron	Tanh, ReLU and LeakyReLU	4 conv and 4 d-conv layers	Address problem of gradients	May behave different with different dataset	

3 Methodology

3.1 Dataset

In this research, the CelebA attributes dataset has been used. CelebA is a huge dataset including over 200,000 celebrity images annotated with 40 attributes. This extensive dataset contains a broad variety of facial data, encompassing a wide range of changes in facial emotions, positions, and backgrounds because of its thorough annotations, which include parameters such as gender, age, the presence of spectacles, and facial expressions, the dataset is widely used in computer vision and machine learning domains. Therefore, the dataset has been selected to perform DeepFakes detection in this research work.

The dataset has been distributed into training, testing and validation sets. There are total of 202,599 images in the dataset. Out of total images, 162770 belongs to training set, 19961 belongs to testing data set while 19866 images belong to validation set. Among the dataset, 80% of images are used for training, 10% for tests, and 10% for validation. The details of distribution are given in Table 2.

Table 2. Distribution of training, testing, and validation dataset

Data Division set	Total images
Training Set	162770
Test Set	19961
Validation Set	19866

3.2 Preprocessing

The GAN preprocessing stage includes three critical operations i.e. image resizing, rescaling, and normalization. The goal of rescaling is to bring the dataset values into a consistent and standardized range, specifically between -0.9608 and 0.9608, which is accomplished by increasing the pixel values of each image by 2 and then subtracting 1. The normalization step is critical for accelerating convergence during training and minimizing gradient vanishing or exploding issues. Additionally, the images are resized to 32 x 32-pixel dimensions to maintain uniformity. The dataset resulting from the preprocessing steps, serves as an input for the GAN model while aligning with the model's requirements and supporting effective training and meaningful outcomes.

3.3 Convolutional Based GAN

A Convolutional Generative Adversarial Network is a cutting-edge deep learning model that has transformed generative modeling and image synthesis. Convolutional GANs is a version of the basic GAN architecture that are specifically designed for detecting GAN generated images. In a competitive training procedure, two neural networks i.e., the generator and the discriminator are integrated.

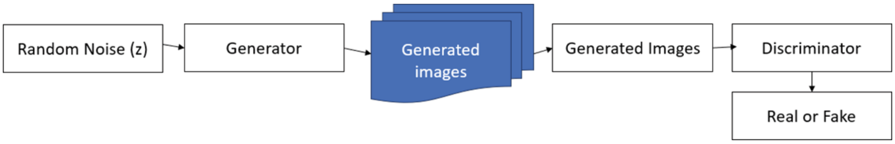


Fig. 1. Flow diagram of proposed Convolutional based GAN

Figure 1 demonstrates the process of generating random noise to feed into the proposed model. Later, the noise vector is fed into the generator, comprising of de-convolutional layers and activation functions. The noise is transformed within the generator, eventually resulting in the generation of an image. The generator takes a step-by-step method, improving the image with a series of convolutional layers. The layers are adept at identifying and capturing intricate features in a hierarchical manner, which enable the network to produce realistic images as training progresses. Meanwhile, the discriminator acts as an evaluator, rigorously scrutinizing the generated images in order to distinguish from real ones. The adversarial training dynamic pushes both the generator and discriminator to constantly improve the network.

3.4 Architecture of Convolutional GAN

The proposed model i.e., CNN-based GAN architecture, is a variant of Generative Adversarial Networks (GANs) optimized for image generation tasks. The model combines convolutional layers into both the generator and discriminator networks, significantly improving their ability to analyze image data. The model includes several crucial elements. Firstly, convolutional layers are added into both the generator and discriminator networks, allowing for the extraction of detailed visual features from edges to higher-level patterns. In the generator netwoek, strided convolutions are used to improve the generation of images while gradually increasing spatial resolution from lower to higher scales. In both networks, batch normalization is explicitly performed after convolutional layers, ensuring stable accelerated training via activation normalization. Secondly, there are de-convolutional layerswhich are also known as transposed convolution layers. These layers assist in the process of up-sampling. The layers play an important role in transforming low-dimensional noise vectors to higher-resolution images, gradually increasing their complexity and details. The weights of the generator are iteratively modified based on feedback from the discriminator throughout the training process. Notably, the discriminator's weights remain untrainable, ensuring that the focus is solely on refining the generator's performance. The interaction between the generator and discriminator is crucial to this process. The generator transforms random noise into realistic images by employing a series of convolutional layers, batch normalization, and activation functions such as leaky ReLU. In terms of loss computation, the generator's loss is evaluated using binary cross-entropy loss function. Simultaneously, the discriminator evaluates these generated images critically while distinguishing from real ones. The dynamic interplay is the core of adversarial training, in which the generator aims to outperform the discriminator by constantly improving its image generation capabilities. Figure 2 demonstrates the structural architecture of the proposed model.

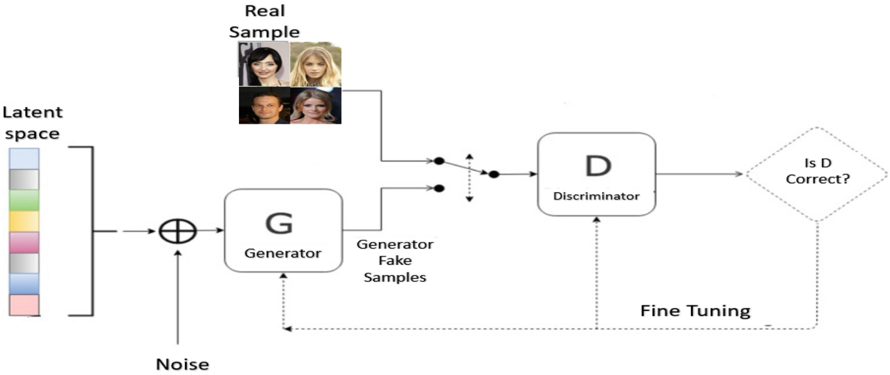


Fig. 2. Structural diagram of proposed Convolutional GAN

3.5 Parameters for Convolutional GAN

Table 3 provides a detailed summary of the parameters used in the DCC-GAN model. In Table 3, there are several parameters including the kernel size, which determines the filter's coverage in terms of pixels, and the stride, a value that affects the filter's movement over the input image.. Furthermore, the table describes the features associated with each convolutional layer, defining the intricate features learnt during the training process. Besides this, another key feature is batch normalization, which is defined as a technique employed across neural network layers to accelerate training, improve stability, and enable the use of higher learning rates. Moreover, the activation functions play an essential role in assisting the neural network to detect complicated patterns in the image. Notably, the model's generator component uses these parameters to convert a 100-dimensional random noise input into a $32 \times 32 \times 3$ image. Concurrently, the discriminator analyses images with convolutional layers and provides a probability score indicating the authenticity of the input image. These parameter values are required for the Convolutional GAN model's training.

Table 3. Model Parameters of Proposed model

Layers	Kernel matrix	Stride value	Features value	Batch Normalization	Activation function
Generalization input (100 × 1 × 1)					
D-Conv	4 × 4	1 × 1	256	Yes	ReLU function
D-Conv	4 × 4	2 × 2	128	Yes	ReLU function
D-Conv	4 × 4	2 × 2	64	Yes	ReLU function
D-Conv	4 × 4	2 × 2	3	No	Tanh function
Discrimination input (32 × 32 × 3)					
Conv	4 × 4	2 × 2	32	Yes	LeakyReLU function
Conv	4 × 4	2 × 2	64	Yes	LeakyReLU function
Conv	4 × 4	2 × 2	128	Yes	LeakyReLU function
Conv	4 × 4	1 × 1	1	No	–

3.6 Evaluation Metrics

The model's performance was evaluated using evaluation measures such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Precision, Recall, Specificity, and Accuracy were calculated using these measures to determine the model's efficacy. The following are detailed descriptions of these measures, as well as their associated mathematical formulas:

Precision

Precision is a measure of how effectively a model predicts a positive a positive outcome, representing the proportion of correct positive predictions produced by the model out of all positive predictions made by the model.

$$percision = True\ positive \div True\ positive + false\ positive$$

Specificity

Specificity is a measure of a model's ability to accurately detect negative cases in the context of binary classification evaluation metrics. It evaluates the proportion of true negative predictions among all actual negative cases.

$$specificity = True\ negative \div True\ negative + false\ positive$$

Recall/Sensitivity

In binary classification, recall, also known as sensitivity or true positive rate, is an important evaluation parameter which measures the model's ability to correctly identify all positive instances in a dataset out of all actual positive instances.

$$recall\ or\ sensitivity = True\ positive \div True\ positive + false\ negatives$$

F1 Score

The F1 Score is a binary classification metric that combines precision and recall to create a single score that describes a model's performance which is particularly useful when the class distribution is unequal or when false positives and false negatives have distinct costs or implications.

$$F1\ Score = 2 \times percision \times recall \div percision + recall$$

Accuracy

Accuracy is a fundamental evaluation metric used to evaluate a classification model's overall performance. Accuracy is a measure of how well a model performs and measures the ratio of correctly predicted images to the total number of images in the dataset.

$$\begin{aligned} Accuracy = {} & True\ positive + True\ negative \div True\ positive + false\ positive \\ & + false\ nagatives + true\ negativ \end{aligned}$$

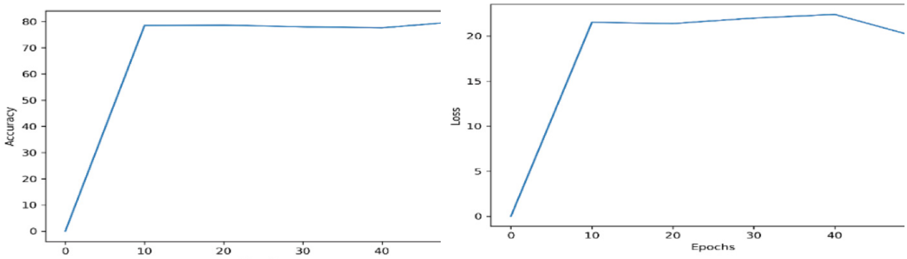
4 Result and Discussion

The study focuses on identifying CNN-GAN generated images, demonstrating greater performance when compared to other models. The GAN uses convolutional layers to improve its efficiency. In the context of GANs, the generator uses deconvolution to generate images from random variables, while the discriminator uses CNN to identify images as either real or fake. Convolutional layers extract features from images, while de-convolutional layers expand images based on the extracted characteristics.

The experiment was carried out on large datasets using a hybrid architecture that combines the Convolutional Neural Network (CNN) layer with Generative Adversarial Network (GAN) model. The experimentation environment was built on the Jupyter Notebook utilizing the computational capability of an available NVIDIA GPU. Moreover, the Keras 2.3.1 framework, Torch, and the Python-based open-source neural network deep learning libraries were used to implement CNN-GAN. The proposed model represents an enhancement towards exploiting deep learning technology in combination with GAN for the specific dataset. The model was trained across 100 epochs with 128 batch sizes each which allowed the model to learn intricate features and patterns from the dataset. Besides this, the model improved its effectiveness to detect DeepFakes images efficiently.

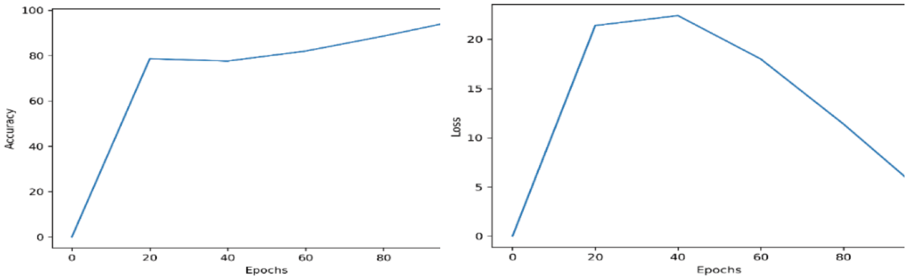
4.1 Analysis of CNN-GAN Model

The primary goal of the research was to determine the effect of the CNN-GAN model on system performance. The proposed models were evaluated using a large dataset to achieve good results. Moreover, the performance of the proposed model was evaluated based on the evaluation metrics i.e., accuracy and loss. The proposed model's accuracy was measured by its ability to accurately classifying images as real or fake, whereas the loss quantifies inaccuracies throughout the training process.



a. Accuracy over 50 epochs

b. Loss over 50 epochs



c. Accuracy over 100 epochs

d. Loss over 100 epochs

Fig. 3. Comparison of Loss and accuracy of model over 50 and 100 epochs

Figure 3 demonstrates that the discriminator's accuracy and loss curves are unstable at epoch 50. However, when the number of epochs is increased to 100, accuracy and loss demonstrates an upward trend before stabilizing. During the 100th epoch, the model reaches an accuracy of 95.8 percent. In Fig. 3, x-axis represents the number of training epochs, signifying the training cycle through the entire dataset. Meanwhile, the loss and accuracy are represented on the y-axis.

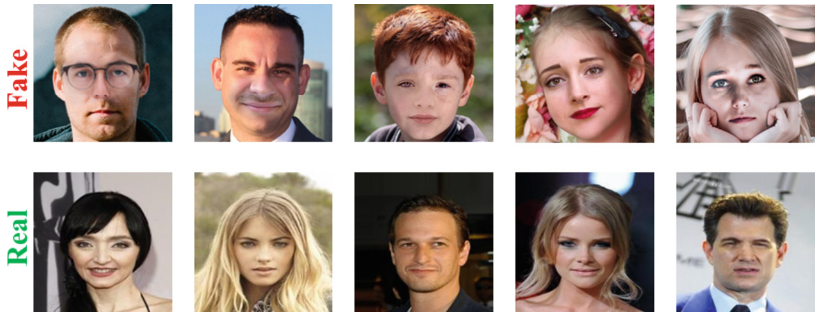
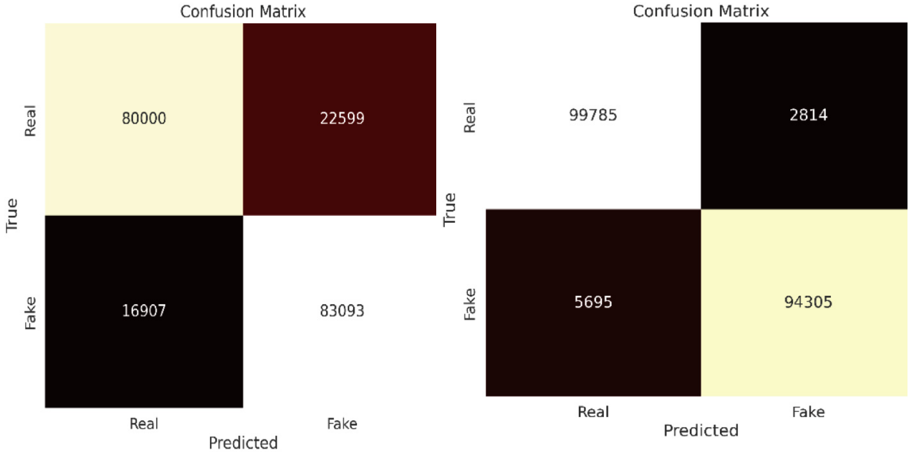


Fig. 4. Result of Classification Probability of Random Test Image

In the experiment, a random image has been chosen from the test dataset, and the model predicts whether the image is fake or real, as shown in Fig. 4. The method completely assesses the model's classification capabilities and capacity to generalize across multiple DeepFakes settings. Furthermore, the model aids in gaining insights about the model's proficiency in distinguishing various aspects and attributes associated with real and fake images, as observed through the projected probabilities.

Figure 5 compares confusion matrices produced after the model evaluations at 50th and 100th epochs. Each confusion matrix is a visual aid for evaluating the model's predictions and correspondence with actual images. The matrices provide information about the model's ability to classify images as real or fake correctly. The matrices show how many images are given to each class. Furthermore, the confusion matrices identify patterns and distribution entries that reflect the model's strengths and weaknesses, allowing for future upgrades and fine-tuning to improve overall classification performance.



a. Confusion Matrix 50 epochs

b. Confusion Matrix 100 epochs

Fig. 5. Comparison of Confusion Matrices Between 50 and 100 Epochs

Table 4 provides an exhaustive overview of major metrics of evaluation important for measuring the effectiveness of proposed Convolutional GAN model. Accuracy, which measures the overall correctness of predictions, is 0.958, which is the ratio of properly predicted instances to entire dataset size. The model's sensitivity (recall) of 0.972 represents the ability to identify true positive instances among all actual positives, which is critical for minimizing false negatives. Specificity, at 0.943, measures the model's ability to distinguish between true negatives among all actual negatives, with a focus on reducing false positives. Precision (0.946) computes the accuracy of positive predictions, emphasizing the model's ability to avoid misclassifying negatives as positives. Finally, the F1-Score (0.958) combines precision and recall into a balanced measure, indicating a model that excels in both high precision and recall.

Table 4. Evaluation matrices measurements for the proposed model

Dataset	Measures	Percentage
Large-scale CelebFaces Attributes (CelebA)	Accuracy	0.958
	Sensitivity	0.972
	Specificity	0.943
	Precision	0.946
	F1-Score	0.958

5 Conclusion

Generative Adversarial Network i.e., GAN generates synthetic data with a realistic appearance of images by developing an algorithm in a large number of iterations. DeepFakes technology is a hard challenge through which the differentiation can be made between real and fake. The said challenge allows researchers to provide optimal algorithms and solutions to detect DeepFakes. The proposed model is capable of working efficiently with the large dataset. The results depict that the accuracy of the proposed model is outperforming as compared to the other models. In the research work, the loss of discriminator is minimized in a number of iterations as compared to the generator's loss. The evaluation of the proposed model is performed through sensitivity and specificity. The generalization and convergence of GAN is a promising future research area for improvements.

5.1 Future Direction

There are several limitations of the proposed model i.e., difficulties with mode collapse, issues with gradient descent and usual convergence challenges in GANs. Moreover, there are many future challenges focusing on the improvement and generalization of GAN models.

References

1. Nguyen, T.T., et al.: Deep learning for deepfakes creation and detection: a survey. Comput. Vision Image Understand. **223**, 103525 (2022). https://doi.org/10.1016/j.cviu.2022.103525
2. Goodfellow, I., et al.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020). https://doi.org/10.1145/3422622
3. Li, Y., Xiao, N., Ouyang, W.: Improved boundary equilibrium generative adversarial networks. IEEE Access **6**, 11342–11348 (2018). https://doi.org/10.1109/ACCESS.2018.2804278
4. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings, pp. 1–26 (2018)
5. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. IEEE Trans. Pattern Anal. Mach. Intell. **43**(12), 4217–4228 (2021). https://doi.org/10.1109/TPAMI.2020.2970919

6. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: StyleGANv2, Cvpr, pp. 8107–8116 (2020)
7. Hany, H.: Creating, using, misusing, and detecting deep fakes. J. Online Trust Safety **1**(4) (2022). https://doi.org/10.54501/jots.v1i4.56
8. Deng, J., Zhang, X., Chen, H., Wu, L.: BGT: a blind image quality evaluator via gradient and texture statistical features. Sign. Process. Image Commun. **96**, 116315 (2021). https://doi.org/10.1016/j.image.2021.116315
9. Li, H., Li, B., Tan, S., Huang, J.: Identification of deep network generated images using disparities in color components. Sign. Process. **174**, 107616 (2020). https://doi.org/10.1016/j.sigpro.2020.107616
10. Tang, G., Sun, L., Mao, X., Guo, S., Zhang, H., Wang, X.: Detection of GAN-synthesized image based on discrete wavelet transform. Secur. Commun. Networks **2021**, 1 (2021). https://doi.org/10.1155/2021/5511435
11. Agarwal, B.: 1209: RECENT ADVANCES ON SOCIAL MEDIA ANALYTICS AND MULTIMEDIA SYSTEMS : ISSUES AND CHALLENGES Financial sentiment analysis model utilizing knowledge-base and domain-specific representation, pp. 8899–8920 (2023)
12. Mi, Z., Jiang, X., Sun, T., Xu, K.: GAN-generated image detection with self-attention mechanism against GAN generator defect. IEEE J. Sel. Top. Signal Process. **14**(5), 969–981 (2020). https://doi.org/10.1109/JSTSP.2020.2994523
13. Yang, J., Xiao, S., Li, A., Lan, G., Wang, H.: Detecting fake images by identifying potential texture difference. Futur. Gener. Comput. Syst. **125**, 127–135 (2021). https://doi.org/10.1016/j.future.2021.06.043
14. Zhao, S., Zhang, M., Ding, H., Cui, X.: MFF-Net: deepfake detection network based on multi-feature fusion. Entropy **23**(12), 10–13 (2021). https://doi.org/10.3390/e23121692
15. Chen, B., Ju, X., Xiao, B., Ding, W., Zheng, Y., de Albuquerque, V.H.C.: Locally GAN-generated face detection based on an improved Xception. Inf. Sci. (Ny) **572**, 16–28 (2021). https://doi.org/10.1016/j.ins.2021.05.006
16. Hsu, C.-C., Zhuang, Y.-X., Lee, C.-Y.: Deep fake image detection based on pairwise learning. Appl. Sci. **10**(1), 370 (2020). https://doi.org/10.3390/app10010370
17. Wang, X., Guo, H., Hu, S., Chang, M.-C., Lyu, S.: GAN-generated Faces Detection: A Survey and New Perspectives (2022). http://arxiv.org/abs/2202.07145
18. Mirsky, Y., Lee, W.: The creation and detection of deepfakes: a survey. ACM Comput. Surv. **54**(1), 1–41 (2021). https://doi.org/10.1145/3425780
19. Verdoliva, L.: Media forensics and DeepFakes: an overview. IEEE J. Sel. Top. Signal Process. **14**(5), 910–932 (2020). https://doi.org/10.1109/JSTSP.2020.3002101
20. Tyagi, S., Yadav, D.: A detailed analysis of image and video forgery detection techniques. Vis. Comput. **39**(3), 813–833 (2023). https://doi.org/10.1007/s00371-021-02347-4
21. Gupta, P., Rajpoot, C.S., Shanthi, T.S., Prasad, D., Kumar, A., Kumar, S.S.: Image forgery detection using deep learning model. In: 3rd International Conference on Smart Electronics and Communication ICOSEC 2022 - Proceedings, pp. 1256–1262 (2022). https://doi.org/10.1109/ICOSEC54921.2022.9951952
22. Huang, Y., Juefei-Xu, F., Guo, Q., Liu, Y., Pu, G.: FakeLocator: robust localization of GAN-based face manipulations. IEEE Trans. Inf. Forensics Secur. **17**, 2657–2672 (2022). https://doi.org/10.1109/TIFS.2022.3141262

Image Analysis and Object Recognition

Using the New YoLo Models in Detecting Small-Sized Objects in the Case of Rice Grains on Branche

Khang Nguyen Quoc, Anh Nguyen Quynh, Hoang Tran Ngoc,
and Luyl-Da Quach(✉)

FPT University, Can Tho City, Vietnam
{KhangNQCE160710,AnhNQCE160515}@fpt.edu.vn,
hoang2531992@gmail.com, luyldaquach@gmail.com

Abstract. Identifying a small-sized object is of interest to many studies, especially the rice grain on the branch. Due to its significance to the evaluation of the grain quality. In identifying rice seeds, there are also some difficulties in separating components such as branches and spikes. The study uses a dataset of images of rice branches with differences in shape, state, and size. After the pre-processing steps, the obtained data has images of relatively small rice grains. The research applies to object recognition advances such as new YOLO models (including YOLOv5, YOLOv6, and YOLOv7). On a dataset of 150 images and nearly 6000 instances, and the results are evaluated on many different epochs, the results show that the highest accuracy belongs to YOLOv7, which is 89.93% (Precision), 87.96% (Recall), and 91.33% (mAP). The study also opens up further studies in detecting diseases on rice, such as grain blight, cotton neck blast, etc.

Keywords: Yolo models · small-sized · count objects · count rice · rice on the branch

1 Introduction

Seed quality plays an essential role in assessing the quality of growth and development of seeds, which is a decisive factor for yield, nutritional composition, taste, and edible quality [1]. However, the evaluation of polymorphic seed quality is still carried out by manual method, which is costly in terms of labor as the assessment is based on chemical analysis in the extraction process. This method was performed through solid phase microextraction, spectroscopic techniques, infrared (IR) spectroscopy, and nuclear magnetic resonance (NMR) besides e-tongue and e-nose for the detection of flavor [2]. Realizing this disadvantage, several studies have been carried out to support the automation process in agriculture.

First, much research has been done in detecting, counting, and locating fruit trees in the garden. Research [3] uses the advancement of deep learning models applied to the data set with 1000 images and 41000 labeled instances of apple. The results obtained the highest accuracy of 92.68% with the CNN algorithm, a semi-supervised clustering

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 157–169, 2023.
https://doi.org/10.1007/978-981-99-7969-1_12

method based on Gaussian Mixture Models, with relatively satisfactory results. With similar work, research [4] uses fewer datasets with algorithms GMM and ResNet50 with the highest accuracy of 95.1%. Research [5, 6] builds MangoNet and MangoYolo networks to detect and count mangoes with the highest accuracy of 98.3%. Research [7] used Mask-RCNN to detect grapes with a camera mounted on an automatic grape harvester with green berries by imaging with an accuracy of 94% in vertical bud positioning and 85.6% in cutting minimal pruning on a dataset with 5700 patches from 38 images. Research [8] used InceptionV2, MobileNet, and Single Shot Multibox Detector (SDD) to detect and count avocados, apples, and lemons with the highest accuracy of 93%. Research [9] detects and counts pear fruit by YOLOv4 models with 98% accuracy. Another fruit of interest is the tomato, with much research focused on supporting the automation of detection, counting, and harvesting. Specifically, the research [10] used the MaskRCNN model on the dataset with different heights with the highest accuracy of 88%. Besides, research [11, 12] performed tomato flower counting by Faster R-CNN thresholding in computer vision with the highest accuracy of 96.02%. Through research, the problem in implementation is dominated by environmental factors such as trees covered by foliage, camera height, light, Etc., which have affected fruit detection and counting.

Similar to fruit detection, much research has focused on seed quality to determine product value. Research [13] counted plots as they affected the quality of tomato cultivars using LocAnalyzer. Research [14] created a TasselNetv2 model that counts wheat spikes based on context-augmented local regression networks with an accuracy of 91.01%. Research [15] used SeedGerm correlation and manual method to count the number of germinated grains of barley, cabbage, corn, pepper, and tomato with minimal error. Research [16] to detect and count the number of sorghum heads using CNN with the final efficiency R2 between humans and machines of 0.88. Research [17] determined the length, width, and height of panicle evaluated clustering with a high accuracy of 89.3% with a 10.7% omission and 14.3% commission rate. Research [18] counted rice and wheat grains using an Android device with an error of less than 2%.

The research shows that the implementation of seed counting has yet to be engaged in much research, but this is a decisive factor for the quality and yield of seed-harvesting crops. Besides, the research using the advancements of the YOLO model has been of great interest to object counting studies. Therefore, the study takes the first step in detecting rice seeds on rice panicles to serve as a basis for further studies. The research is divided into sections. Section 2 explores related studies in rice seed detection with its advancements and challenges to be made. Section 3 provides knowledge related to YOLO models; Section 4 proposes methods and assessments for data made through the actual collection of the research. Section 5 the results obtained from implementation. Section 6 discusses the results of the research, Sect. 7 conclusions related to the study, and suggestions for further research directions.

2 Related Works

Recent technological advances play an essential role in seed detection. Research [19] shows potential in using near-infrared spectroscopy, hyperspectral and multispectral imaging, Raman spectroscopy, infrared thermography, and soft X-ray imaging methods.

Research [20] classified 14 rice varieties of Oryza sativa with 3500 samples with nearly 50000 seeds. The machine learning methods used in this research include VGG16, VGG19, Xception, InceptionV3, InceptionResNetV2, LR, LDA, k-NN, and SVM. The highest accuracy is 95.15%. In addition, the research [21] used advances in technology using hyperspectral imaging and chemometrics on individually isolated particles with an overall accuracy of 88%. Similarly, the research [22] using hyperspectral imaging with deep learning algorithms gave an accuracy of 99.19% on the oat seeds dataset, and research [23] used transfer learning with the highest accuracy of 97.2% on soybean seed varieties. Another approach in particle detection is built on optical micrographs using image processing with relatively satisfactory results. In addition, the advancement of deep learning and transfer learning models has shown substantial development in agriculture, such as diagnosing diseases in chickens [24, 25], shrimp [26], foliar diseases [27, 28], fish [29], palm trees diseases [30].

The research in this section demonstrates a remarkable development in technological advances in the formulation and development of techniques for particle detection. However, the remaining problem of the studies is that they are being carried out with photos in the laboratory, using high equipment, so the cost problem is still limited. In addition, the tiny size of the seeds on the cotton when separated is analyzed in Section IV. Therefore, this study performing rice seed detection in paddy ears will create a challenge in formulating and developing follow-up studies that help support farmers through mobile devices (Fig. 1).

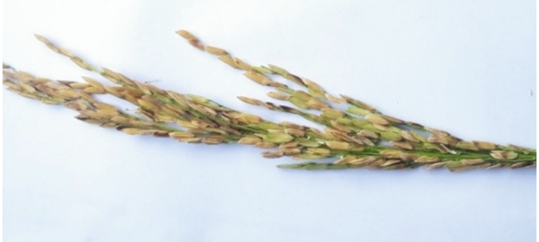


Fig. 1. The image illustrates the sample of the research.

3 YOLO Models

3.1 YOLO Network and Algorithms Brief

The YOLO algorithm divides the trained image into S × S grids of cells, where each cell has different detection tasks. This network structure is created from 24 convolutional layers and connected layers described (Fig. 2). In which the convolutional layers will extract the features of the image, and the full-connected layers will predict the probability and the coordinates of the object. After the fully connected layer, the tensor of $S \times S \times (B \times 5 + C)$ is output, where B represents the number of predicted targets in each mesh and C represents the number of object types. The final result is obtained by regressing the box object's position and evaluating the tensor data's type probability. Therefore, the YOLO algorithm can detect targets quickly but cannot detect small targets, or its detection efficiency is not good [31]. So many later versions of this algorithm since version 5 have tried to overcome this by changing the structure and adding image processing methods. In this study, we will evaluate the latest version of the YOLO models, including versions 5, 6, and 7.

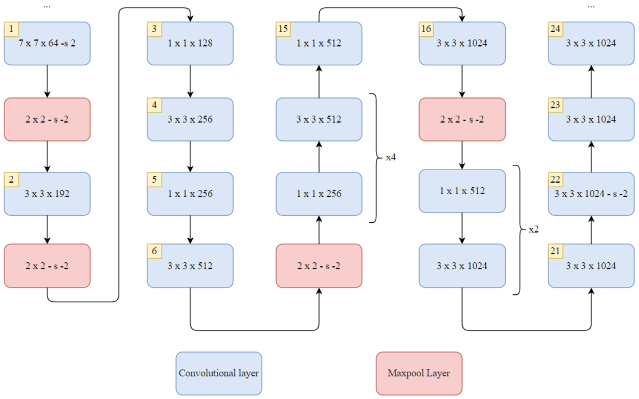


Fig. 2. Illustrating the Convolution layer network structure of the YOLOv1 models.

3.2 Overview of YOLO Models

YOLOv5 is an upgraded architecture from its predecessors to improve recognition efficiency. It has significantly improved the processing time of deeper networks [32]. This will become important with the project into larger datasets containing small objects and real-time detection. With the structure being changed with some properties:

- Backbone: from CSPResidualBlock (version 4) switch to using module C3.
- Neck: SPP + PAN -- >SPPF + PAN. Adopting a module similar to SPP, but twice as fast and calling it SPP - Fast (SPPF).
- Data Augmentation: Mosaic Augmentation, Copy-paste Augmentation, MixUp Augmentation.
- Anchor Box: Using the technique of applying genetic algorithm (GA) to the Anchor Box after the k-means step, the Anchor Box works better with the user's custom datasets but no longer works well with each Common Objects in Context (COCO).

YOLOv6 is a single-stage object detection model based on YOLO architecture [33]. This version is researched and developed into open source by the author Meituan. YOLOv6 achieves a more robust performance than Yolov5 when benchmarking against the MS Coco dataset. This model has evolved the backbone and neck layer with a new structure called EfficientRep Backbone and a Rep-PAN Neck. This change makes the number of parameters of v6 larger than v5, even two times. However, the training time between the two models is mainly about the same with the same amount of data, and v6 gives a higher probability of object type demonstrated on the COCO dataset.

YOLO-v7 is the latest model of the YOLO family published and studied by WongKinYiu [34]. The author confirms that this model is superior to the object development models. The change and many new structures applied make Yolov7 achieve high efficiency and faster time. Some new techniques to improve the efficiency of the above model such as the Label Assignment technique, Model Scaling, Backbone use Efficient Layer Aggregation Network (ELAN), Neck with SPPCSPC architecture developed from YOLOv4 model, Re-parameterization in YOLOv5 model.

4 Methodology

The research evaluated the three latest models of the YOLO family on a dataset containing small objects in the Object Detection domain, for YOLO models, SOTA one-stage object detection models, is popular set of object detection models used for real-time object detection and classification in computer vision. The proposed research method uses YOLO models to separate each rice grain on the rice smudging data set to evaluate the effectiveness of each model. The actual object separation results of each model compare the productivity of versions with each other.

4.1 Data Collection and Preprocessing

The dataset used in the rice grain analysis was provided by the Mekong Delta Rice Institute and captured through a Samsung Galaxy Note 10 phone. The rice image dataset used through selection includes 150 images. Image data will be processed through steps including data labeling and resizing (640 × 640 px) when included in the YOLO model. Using the above-sized image to make the objects reach a small size on each image. Nearly 6,000 objects were labeled using LabelImg and RoboFlow processing and labeling tools. The processing is illustrated below (Fig. 3).

The data is labeled in the YOLO format using a self-designed program that can provide a bounding box and x, y, height, and width label coordinates. On the other hand, for efficient training, the images are labeled with the open-source LabelImg tool and using the YOLO format. Afterward, the processed data will be pushed to RoboFlow to export two complete datasets with a train and valid ratio of 80:20 suitable for 3 YOLO models. When training and testing, the dataset will be exported by RoboFlow for each YOLO version.

Based on the results of the analysis, the research visualized the distribution of object box occurrences and sizes in the data set (Fig. 4). In Fig. 4a represents the position of the centroids of the object boxes in the image after scaling to (0,1) the image, and it can be observed that the objects are primarily concentrated in the center of the image; Fig. 4b represents the ratio of the size of the box object to the size of the image, where it can be observed that not only there are many objects of the same size but also some objects of different sizes.

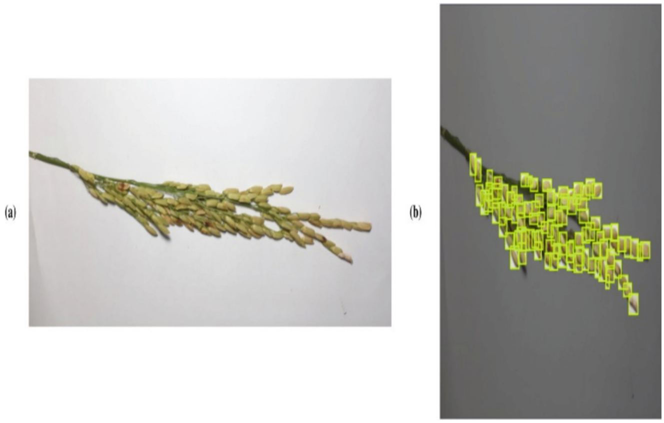


Fig. 3. Illustrate the rice image in the dataset: (a) Original image (b) Processed image.

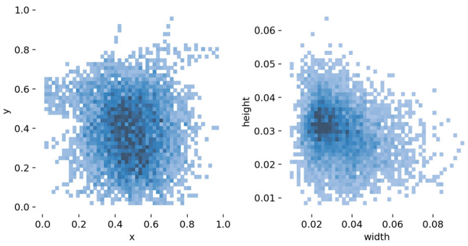


Fig. 4. Characteristics of the dataset used: (a) Location of objects in the box, (b) Size of the box.

4.2 Experimental Environment

The software and hardware parameters used for the model training in this document are shown under Table 1.

Table 1. Configuration parameters

Device	Configuration
System	Windows 10 Home Single Language
CPU	Intel(R) Core (TM) i7-1065G7
GPU	16 GB GPU Tesla T4
GPU accelerator	CUDA 11.2, Cudnn 11.0
Frames	Pytorch
Compilers	Google. LLC. Collab and Anaconda
Python version	3.8

4.3 Proposed Method

After the data collection and processing are completed, they will be fed into YOLO models for object detection training. YOLO models are initialized on the Google Colab Cloud platform through open source code developed on GitHub with some fixed parameters such as Table 2.

Table 2. Fixed training parameters

Hyperparameter	Value
Batch size	4
Image size	640 (px)
Epochs	200
Epochs evaluation interval	1
Learning rate	0.01

The version with little complexity will be used in which model parameters YOLOv5, YOLOv6, and YOLOv7. At the same time, these models also fit the dimensions of the trained images. The provided models use transfer learning techniques from pre-trained models on the COCO dataset of 80 classes to achieve results with small epochs, with the mean average in validation dataset by IoU (threshold equals 50%) and number of params are detailed in Table 3.

Table 3. Model properties

Model	Params (M)	$mAP^{val}{}_{50}$ (%)
Yolov5s	7.2	56.8
Yolov6s	17.2	60.4
Yolov7	36.9	69.7

The process of our method is illustrated simply through the steps that include data processing, model training, evaluation of the model's effectiveness on the test set, and comparison of each model's results based on metrics (Fig. 5).

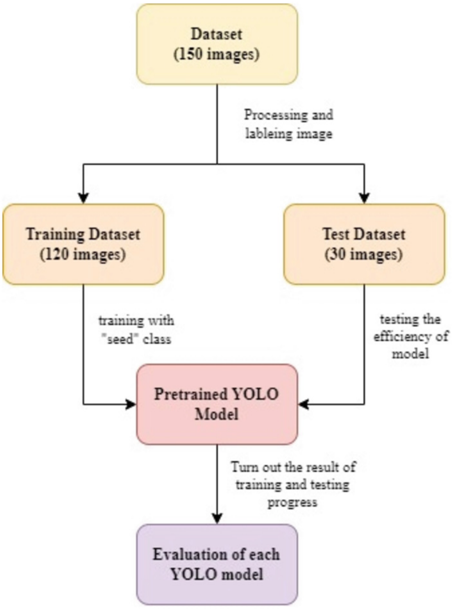


Fig. 5. Illustrate progress in research.

5 Result

5.1 Evaluation Metrics

To evaluate and compare the results of YOLO models, we use some standard metrics developers use in Deep Learning, such as Precision used to measure the quality of true prediction (TP) is performed by (1), Recall is the ratio of the number of true positives among those that are False Negative (TP + FN) (2). These are two measurement methods commonly used to assess understanding between models. The Precision Average (3) and Mean Precision Average (4) by N number of classes are the indicators used to evaluate the trained parameters of the models of the YOLO family.

$$P = \frac{TP}{TP + FP} \tag{1}$$

$$R = \frac{TP}{TP + FN} \tag{2}$$

$$AP = \frac{1}{11} \sum_{R_i} PR_i \tag{3}$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{4}$$

In particular, the research needs to determine whether the object prediction is true or false based on an intersection over Union (IoU) concept. IoU is the ratio between the intersection and union of the predicted bounding and the ground truth; the formula and illustration of IoU are shown in Fig. 6. The research uses the usual threshold to make predictions right or wrong is 50%.

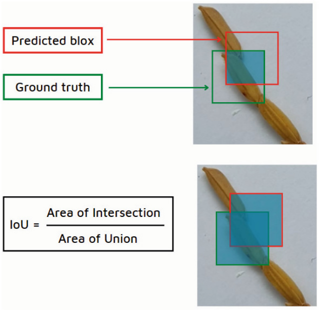


Fig. 6. First Section Illustrated IoU with rice samples on the dataset. Area of Intersection: the intersection of the predicted bounding box and ground truth. Area of Union: the combination of both the area of prediction bound box and ground truth.

5.2 Training Result

Through the steps from data processing, model structure, and configuration to training. The statistical research found that the results of the models based on the valid data set achieved the highest Mean Average Precision ($mAP^{0.5}$), with YOLOv7 reaching 94.22% at the 188th epoch. The results of the training process between 3 models are evident in Table 4 and Fig. 7. In which the highest result is YOLOv7 with evaluation parameters such as Precision (93.2%), Recall (84.27%), and F1 Score (88.51%) compiled in Table 4.

On the other hand, the study found that the YOLOv7 model significantly improved compared to the previous models (YOLOv5 and YOLOv6). As for the results obtained by YOLOv5 and YOLOv6 models during small object recognition training, Yolov7 only needed half the time to reach the same result, at epoch 60 reached 85.13% (compared to 82.68% in Yolov5) and 62 reached 85.81% (compared to 85.51% in Yolov6). However, the results also show that with the number of epochs less than 100, YOLOv6 gives stable results over time. However, this model still needs to drift to get high results and only gradually increase the effect more slowly. Between the YOLOv5 and YOLOv6 models not too much difference in results when training in the last epochs and almost saturating at nearly 85% mAP. In the research, YOLO models still make a difference and develop significantly between YOLO models.

Table 4. Results of the yolo models.

Model	Precision(%)	Recall(%)	F_1 Score	mAP(%)	Best Epoch
Yolov5s	81.53	80.89	81.2	82.68	165
Yolov6s	84.17	83.65	83.9	85.51	165
Yolov7	93.2	84.27	88.51	94.22	188

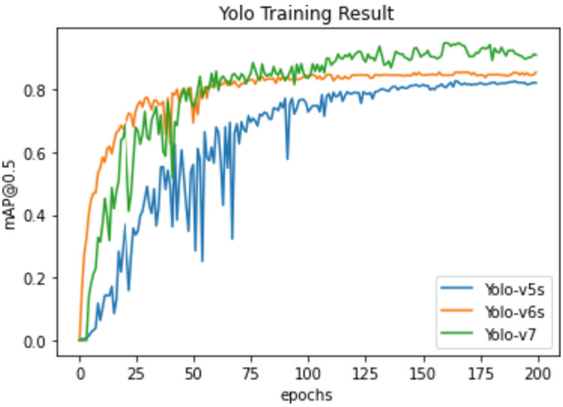


Fig. 7. The graph shows the mAP ratio of 3 YOLO models for each epoch.

6 Discussion

In object recognition, more specifically the YOLO family, the Yolov7 model is their authors claim a breakthrough with a performance and comprehension increase of more than 120% compared to the previous version. Moreover, the research also thought this model would achieve the best results compared to the generally tested models. The results are similar to the hypothesis that YOLOv7 achieved outstanding results compared to two models, YOLOv5s and YOLOv6s, respectively. This proves that YOLOv7 is a relatively new model and should be exploited, especially for small, hard-to-extract objects characterized by previous structures. However, this model still needs to perform better on the test dataset, as we can see the mAP object recognition is up to 94%. However, the correctness and accuracy depend on metrics such as Recall or Precision, and F1 Score is at most 90%. This can be explained by the fact that the size of our data set needs to be larger, specifically only 150 images, and only using the same structure with the image size of 640 × 640px. In addition, because the main purpose of the study is to compare the models, all the basic parameters mentioned above will be used, so there will be some limitations on the accuracy of the model. However, because the model is trained on a dataset that is close to reality, applying this model will also bring a system to help farmers identify rice seeds and then separate them from each other diseased and disease-free, serving the larger systems to bring up the disease status of rice quickly. For us, the research on comparing models and optimization algorithms still need to be explored. However, this topic contains potential positive solutions in this computer vision area.

7 Conclusion

In object detection, research shows that YOLO7v is the best of the last three versions of the YOLO family. The YOLOv7 model can identify small entities, even with less than 200 epochs. However, these models still cannot achieve too high accuracy because the data set needs to be larger, and the primary purpose is not. of the study is to compare the models with each other. Therefore, in the following studies, we may increase the dataset, apply many methods of image processing, evaluate other structures in the field of Object Detection, and propose a structure or method. The new algorithm for detecting small-sized objects. At the same time, drones will be applied to apply the Yolo model to identify objects on a large scale. The research also shows that the detection of diseases on rice, such as rice smudging and rice blast will be developed in the future.

References

1. Feng, L., Zhu, S., Liu, F., He, Y., Bao, Y., Zhang, C.: Hyperspectral imaging for seed quality and safety inspection: a review. Plant Methods **15**(1), 91 (2019). https://doi.org/10.1186/s13007-019-0476-y
2. Farghal, H.H., Mansour, S.T., Khattab, S., Zhao, C., Farag, M.A.: A comprehensive insight on modern green analyses for quality control determination and processing monitoring in coffee and cocoa seeds. Food Chem. **394**, 133529 (2022). https://doi.org/10.1016/j.foodchem.2022.133529
3. Hani, N., Roy, P., Isler, V.: MinneApple: a benchmark dataset for apple detection and segmentation. IEEE Robot. Autom. Lett. **5**(2), 852–858 (2020). https://doi.org/10.1109/LRA.2020.2965061
4. Häni, N., Roy, P., Isler, V.: A comparative study of fruit detection and counting methods for yield mapping in apple orchards. J. Field Robotics **37**(2), 263–282 (2020). https://doi.org/10.1002/rob.21902
5. Kestur, R., Meduri, A., Narasipura, O.: MangoNet: a deep semantic segmentation architecture for a method to detect and count mangoes in an open orchard. Eng. Appl. Artif. Intell. **77**, 59–69 (2019). https://doi.org/10.1016/j.engappai.2018.09.011
6. Koirala, A., Walsh, K.B., Wang, Z., McCarthy, C.: Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'MangoYOLO.' Precision Agric. **20**(6), 1107–1135 (2019). https://doi.org/10.1007/s11119-019-09642-0
7. Zabawa, L., Kicherer, A., Klingbeil, L., Töpfer, R., Kuhlmann, H., Roscher, R.: Counting of grapevine berries in images via semantic segmentation using convolutional neural networks. ISPRS J. Photogramm. Remote. Sens. **164**, 73–83 (2020). https://doi.org/10.1016/j.isprsjprs.2020.04.002
8. Vasconez, J.P., Delpiano, J., Vougioukas, S., Auat Cheein, F.: Comparison of convolutional neural networks in fruit detection and counting: a comprehensive evaluation. Comput. Electron. Agric. **173**, 105348 (2020). https://doi.org/10.1016/j.compag.2020.105348
9. Parico, A.I.B., Ahamed, T.: Real time pear fruit detection and counting using YOLOv4 models and deep SORT. Sensors **21**(14), 4803 (2021). https://doi.org/10.3390/s21144803
10. Afonso, M., et al.: Tomato fruit detection and counting in greenhouses using deep learning. Front. Plant Sci. **11**, 571299 (2020). https://doi.org/10.3389/fpls.2020.571299

11. Rahim, U.F., Mineno, H.: Tomato flower detection and counting in greenhouses using faster region-based convolutional neural network. J. Image Graph. **8**(4), 107–113 (2020). https://doi.org/10.18178/joig.8.4.107-113
12. Afonso, M., Mencarelli, A., Polder, G., Wehrens, R., Lensink, D., Faber, N.: Detection of tomato flowers from greenhouse images using colorspace transformations. In: Oliveira, P.M., Novais, P., Reis, L.P. (eds.) Progress in Artificial Intelligence: 19th EPIA Conference on Artificial Intelligence, EPIA 2019, Vila Real, Portugal, September 3–6, 2019, Proceedings, Part I, pp. 146–155. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-30241-2_13
13. Spetale, F.E., Murillo, J., Vazquez, D.V., Cacchiarelli, P., Rodríguez, G.R., Tapia, E.: LocAnalyzer: a computer vision method to count locules in tomato fruits. Comput. Electron. Agric. **173**, 105382 (2020). https://doi.org/10.1016/j.compag.2020.105382
14. Xiong, H., Cao, Z., Lu, H., Madec, S., Liu, L., Shen, C.: TasselNetv2: in-field counting of wheat spikes with context-augmented local regression networks. Plant Methods **15**(1), 150 (2019). https://doi.org/10.1186/s13007-019-0537-2
15. Colmer, J., et al.: SeedGerm: a cost-effective phenotyping platform for automated seed imaging and machine-learning based phenotypic analysis of crop seed germination. New Phytol. **228**(2), 778–793 (2020). https://doi.org/10.1111/nph.16736
16. Ghosal, S., et al.: A weakly supervised deep learning framework for sorghum head detection and counting. Plant Phen. **2019**, 2019/1525874 (2019). https://doi.org/10.34133/2019/1525874
17. Malambo, L., Popescu, S.C., Horne, D.W., Pugh, N.A., Rooney, W.L.: Automated detection and measurement of individual sorghum panicles using density-based clustering of terrestrial lidar data. ISPRS J. Photogramm. Remote. Sens. **149**, 1–13 (2019). https://doi.org/10.1016/j.isprsjprs.2018.12.015
18. Liu, T., et al.: Rice and wheat grain counting method and software development based on Android system. Comput. Electron. Agric. **141**, 302–309 (2017). https://doi.org/10.1016/j.compag.2017.08.011
19. Xia, Y., Xu, Y., Li, J., Zhang, C., Fan, S.: Recent advances in emerging techniques for non-destructive detection of seed viability: a review. Artific. Intell. Agric. **1**, 35–47 (2019). https://doi.org/10.1016/j.aiia.2019.05.001
20. Kiratiratanapruk, K., et al.: Development of paddy rice seed classification process using machine learning techniques for automatic grading machine. J. Sens. **2020**, 1–14 (2020). https://doi.org/10.1155/2020/7041310
21. Bao, Y., Mi, C., Wu, N., Liu, F., He, Y.: Rapid classification of wheat grain varieties using hyperspectral imaging and chemometrics. Appl. Sci. **9**(19), 4119 (2019). https://doi.org/10.3390/app9194119
22. Wu, N., et al.: Variety identification of oat seeds using hyperspectral imaging: investigating the representation ability of deep convolutional neural network. RSC Adv. **9**(22), 12635–12644 (2019). https://doi.org/10.1039/C8RA10335F
23. Zhu, S., et al.: A rapid and highly efficient method for the identification of soybean seed varieties: hyperspectral images combined with transfer learning. Molecules **25**(1), 152 (2019). https://doi.org/10.3390/molecules25010152
24. Quach, L.-D., Pham-Quoc, N., Tran, D.C., Hassan, M.F.: Identification of chicken diseases using VGGNet and ResNet models. In: Vo, N.-S., Hoang, V.-P. (eds.) Industrial Networks and Intelligent Systems: 6th EAI International Conference, INISCOM 2020, Hanoi, Vietnam, August 27–28, 2020, Proceedings, pp. 259–269. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-63083-6_20
25. Quach, L.-D., et al.: An AI-based chicken disease management system. In: Rosdiazli Ibrahim, K., Porkumaran, R.K., Nor, N.M., Prabakar, S. (eds.) International Conference on Artificial Intelligence for Smart Community: AISC 2020, 17–18 December, Universiti Teknologi

Petronas, Malaysia, pp. 707–716. Springer Nature Singapore, Singapore (2022). https://doi.org/10.1007/978-981-16-2183-3_68

26. Duong-Trung, N., Quach, L.-D., Nguyen, C.-N.: Towards classification of shrimp diseases using transferred convolutional neural networks. Adv. Sci. Technol. Eng. Syst. J. **5**(4), 724–732 (2020). https://doi.org/10.25046/aj050486
27. Quach, L.-D., Quoc, K.N., Quynh, A.N., Ngoc, H.T.: Evaluation of the efficiency of the optimization algorithms for transfer learning on the rice leaf disease dataset. Int. J. Adv. Comput. Sci. Appl. **13**(10) (2022). https://doi.org/10.14569/IJACSA.2022.0131011
28. Taujuddin, N.S.A.M., et al.: Detection of plant disease on leaves using blobs detection and statistical analysis. IJACSA **11**(8) (2020). https://doi.org/10.14569/IJACSA.2020.0110852
29. Sikder, J., Sarek, K.I., Das, U.K.: Fish disease detection system: a case study of freshwater fishes of Bangladesh. Int. J. Adv. Comput. Sci. Appl. **12**(6) (2021). https://doi.org/10.14569/IJACSA.2021.01206100
30. Abu-zanona, M., Elaiwat, S., Younis, S., Innab, N., Kamruzzaman, M.M.: Classification of palm trees diseases using convolution neural network. Int. J. Adv. Comput. Sci. Appl. **13**(6) (2022). https://doi.org/10.14569/IJACSA.2022.01306111
31. Wu, W., et al.: Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image. PLoS ONE **16**(10), e0259283 (2021). https://doi.org/10.1371/journal.pone.0259283
32. Jocher, G., et al.: ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration. Zenodo (2021). https://doi.org/10.5281/ZENODO.4418161
33. Li, C., et al.: YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications (2022). https://doi.org/10.48550/ARXIV.2209.02976
34. Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y. M.: YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors (2022). https://doi.org/10.48550/ARXIV.2207.02696

Segmentation of NKX2.5 Signal in Human Pluripotent Stem Cell-Derived Cardiomyocytes

Siem Jongsma[1], Verena Schwach[2], Simone A. Ten Den[2], Robert Passier[2,3], Fons J. Verbeek[1], and Lu Cao[1](✉)

[1] Leiden Institute of Advanced Computer Science, Leiden University, Niels Bohrweg 1, Leiden, The Netherlands
`l.cao@liacs.leidenuniv.nl`
[2] Applied Stem Cell Technologies, TechMed Centre, University of Twente, Enschede, The Netherlands
[3] Department of Anatomy and Embryology, Leiden University Medical Centre, Leiden, The Netherlands

Abstract. Human pluripotent stem cell-derived Cardiomyocytes (hPSC-CMs) become increasingly popular in recent years for disease modeling and drug screening. NKX2.5 gene is a key transcription factor that regulates cardiomyocyte differentiation. A human embryonic stem cell (hESC) reporter line with NKX2.5 in GFP signal allows us to monitor the specificity and efficiency of human cardiac differentiation. We intend to develop an automatic analysis pipeline for the NKX2.5 signal. However, the NKX2.5 signal captured from fluorescence microscopy is highly heterogeneous. It is not possible to be properly segmented using traditional thresholding methods. Therefore, in this paper, one machine learning method: enhanced Fuzzy C-Means clustering (EnFCM) and two deep learning models: U-Net and DeepLabV3+, are evaluated on the segmentation performance. Parameters have been tuned for each method so as to reach to the optimal segmentation performance. The results show that EnFCM reaches the performance of 0.85. U-Net and DeepLabV3+ have a superior performance. Their performances are 0.86 and 0.89 respectively.

Keywords: pluripotent stem cell derived cardiomyocyte · segmentation · machine learning · deep learning

1 Introduction

Stem cell technology is a rapidly developing field that potentially offers effective treatment for various diseases [11]. It provides a more faithful representation of the actual human diseases so that underlying mechanisms can be better understood. Stem cells can be used for the effective validation of safe medicines which could facilitate more predictive drug discovery and toxicity studies [28]. In addition, stem cells have the potential to replace animal experimentation in predictive toxicology [16].

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 170–184, 2023.
https://doi.org/10.1007/978-981-99-7969-1_13

Human pluripotent stem cell-derived Cardiomyocytes (hPSC-CMs) are valuable tools for disease modeling, assessing cardiotoxicity of drugs as well as identifying novel therapeutic compounds [19]. In recent years, tremendous efforts have been taken to integrate hPSC-CMs into high-throughput screening systems. Phenotypic analysis has been carried out and varied phenotypic readouts have been quantified including morphological changes [6,7], contractile properties [18,25],calcium transients [22,23] and electrophysiological parameters [20,32].

NKX2.5 is an essential transcription factor for activation and maintenance of the cardiac regulatory network. This transcription factor can be detected in cardiac progenitor cells, their progeny, and mature cardiomyocytes [17]. Therefore, genetically engineered NKX2.5 reporter cell line was developed to monitor cardiac cell populations during differentiation [10]. Recently, the hPSC NKX2.5 reporter line has been used for developmental toxicity testing [17].

Measuring NKX2.5 fluorescent signal from hPSC-CMs in a high-throughput manner is, however, challenging. We observed big variation of the green fluorescent protein (GFP) signal of NKX2.5 between batches and treatments as shown in Fig. 1 and 6. It can be caused by batch variation, differentiation efficiency as well as varied effect of treatments. The performance of traditional thresholding methods are not satisfactory, since a large amount of the methods can only segment part of the signal and cannot capture weak GFP signal in the fluorescent images. In this study, we are going to explore machine learning and deep learning methods for segmentation of NKX2.5 signal in hPSC-CMs.

2 Related Work

Image segmentation is a task in computer science that involves the delineation of regions of interest in an image. Segmentation is often used in medical or biomolecular imaging to automate or facilitate the division or recognition of specific structures in the images that are generated in the research [21].

There are many machine learning-based and deep learning-based methods which give superior performance for segmentation. For example, Fuzzy C-Means clustering is an unsupervised machine learning method which has been used for segmentation. It firstly assigns a degree of "belonging to foreground" for each pixel and sets the cut-off between foreground and background based on the minimization of intra-cluster variance [6].

In addition, a large number of deep learning models are successfully used in the research field of semantic segmentation. Semantic segmentation labels pixels in the image to the corresponding regions. One of the most widely used models in the field is U-Net. U-Net is a fully convolutional neural network that was first proposed in 2015 [27]. It is featured by its light weighted structure which makes it possible to train a deep learning model with a small training dataset such as thousands or even hundreds of images. It is extremely useful in the segmentation of a biological image dataset in which the number of training data is always limited.

There are several deep learning models which are commonly used as alternatives for U-Net [13] such as DeepLabV3+ [8] and Tiramisu [2]. From two studies in which DeepLabV3+ is used for similar segmentation tasks and directly compared to U-Net, it is concluded that DeepLabV3+ achieves slightly better performance than U-Net [9,14]. It is also found that U-Net does have a slightly better performance than DeepLabV3, which is the previous version of DeepLabV3+ [1]. In addition, it has been shown that DeepLabV3+ has a better performance than DeepLabV3 [31]. From two studies in which the performance of U-Net is compared to Tiramisu, it is concluded that Tiramisu has similar performance [12,15].

3 Methods

3.1 Preparation of the Cells

Double Reporter mRubyII-ACTN2 and GFP-NKX2.5 (DRRAGN) hPSCs were differentiated to hPSC-CMs as described in [26]. Around day 14 of differentiation, cells were dissociated and were FACS sorted for α-ActininmRybyII/w-Nkx2.5eGFP/w. Double positive CMs were seeded into 96 well special optics plates (PerkinElmer) at a density of 50,000 cells per well. The hPSC-CMs were maintained in a humidified incubator and were refreshed with CM-TDI medium twice a week [3]. 10–12 days after seeding, the hPSC-CMs were treated with dimethylsulfoxide (DMSO 4.23 mM) as control or with 1 μM of the anticancer drug Doxorubicin for 5 days.

3.2 Imaging

Images of NKX2.5 signal of hPSC-CMs were acquired using the high-throughput automated EVOS FL Auto 2 (Thermo Fisher) microscope equipped with a 40x Super-apochromat Olympus objective (NA 0.95) (Thermo Fisher, AMEP4754). The whole monolayer cell culture was scanned by automatically acquiring 55 images per well every 24 h for 5 days. During the 5 days, cells were maintained on the EVOS Onstage incubator.

3.3 Preparation of Ground Truth Data

The dataset, that is used in this study, consists of 1450 images from the hPSC-CMs research. The 1450 images are from 18 different batches. The size of the images is 1328×1048 and they are provided in a 8-bit gray scale format. The signal that is present in the images originates from the GFP signal representing the expression of NKX2.5 protein.

In order to train the deep learning models using these images, ground truths have to be created first. The ground truths, which is approved by the domain specialist, have to be manually created from the original images by converting it to a binary mask using the correct gray value threshold. This is done in the Fiji application [29] using a macro. By running the macro, the images in the selected directory are loaded one by one. A Gaussian blur is applied to suppress the noise in the image. Then, a threshold can be selected manually from which the binary mask is generated and saved. In Fig. 1, two examples are given for the original image and corresponding ground truth. In order to have a consistent set of ground truths, the annotator discussed with the biologists beforehand and did several trials together with the domain expert for quality control. Subsequently, all images were processed and were ready to feed into deep learning models for training.

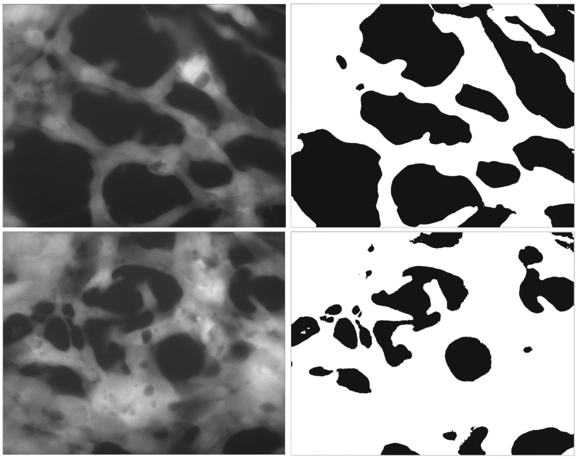


Fig. 1. Two examples of images from the dataset including the manually created ground truth binary mask.

3.4 Evaluation

An evaluation measure is used to assess the performance of the deep learning models. The metric that is used is Intersection over Union (IoU). For the calculation of the IoU metric, the intersection of the foregrounds of a ground truth image and a segmentation result is divided by the union of the same foregrounds, which can be seen in Fig. 2.

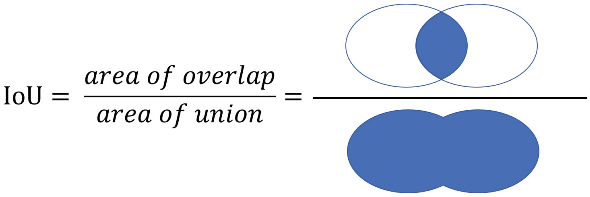


Fig. 2. Visual representation of Intersection over Union performance metric.

The implementation of the IoU metric is done using the MeanIoU function in the Python Keras library.

3.5 Fuzzy C-Means Clustering

The Fuzzy C-Means Clustering (FCM) is an unsupervised clustering method setting the threshold based on the minimization of intra-cluster variance. This method can successfully capture both strong and weak signals which is ideal to segment NKX2.5 signal. However, due to the high resolution of our images (1328 × 1048 pixels per image), it takes minutes (in a system with 2.80GHz processing speed and 8 GB RAM) to extract the signal from a single image. It is not optimal for a high throughput setup. In order to improve the processing speed, several improved versions based on FCM have been explored [5]. An accelerated version of the FCM Algorithm called EnFCM [30] was chosen to solve the speed problem. EnFCM treats each gray value from the histogram as a clustering candidate instead of each pixel from the image. Its energy function for minimization is expressed in Eq. 1:

$$J = \sum_{i=1}^{c} \sum_{l=1}^{q} n_l u_{il}^m \left\| l - v_i \right\|^2 \quad m > 1. \tag{1}$$

where c stands for the number of clusters, q represents the number of gray levels in the histogram, n_l is the number of pixels whose gray value equals to l. m is the fuzzyfication parameter. u_{il}^m is the degree of membership of gray level l. v_i is the center of the cluster. The iterative minimization of the objective function is realized by updating the membership u_{il} and the cluster centers v_i:

$$u_{il} = \frac{(v_i - l)^{-2/(m-1)}}{\sum_{j=1}^{c} (v_j - l)^{-2/(m-1)}} \quad \forall i = 1...c, \quad \forall l = 1....q,$$

$$v_i = \frac{\sum_{l=1}^{q} n_l u_{il}^m l}{\sum_{l=1}^{q} n_l u_{il}^m} \quad \forall i = 1...c.$$

In this way, the processing time is drastically reduced to 1–2 s per image using the same PC.

3.6 U-Net

The U-Net is a model that is built upon the architecture of the fully convolutional neural network [27]. The network consists of two parts: the contracting path and the expansive path. In the contracting path, the usual structure of a convolutional network is followed. This means that unpadded convolutions are applied, followed by a generic ReLU activation operator and a max pooling operation. Max pooling operations are applied for the downsampling of the image. In the expansive path, upsampling is performed first for every layer. Afterwards, up-convolution is done, followed by concatenation with the corresponding feature map that was acquired in the contracting path. As the last step, for every layer in the expansive path, ReLU activation is performed. The output is created by applying a sigmoid activation to acquire the correct range of values in the prediction.

3.7 DeepLabV3+

DeepLabV3+ (DLV3+) is a model that was proposed as an extension of the DeepLabV3 model. The structure of DLV3+ is similar to that of the U-Net. However, there are some important differences between the structure of U-Net and DLV3+. In DLV3+ a combination of atrous convolution and depthwise convolution is used, called atrous separable convolution. In this type of convolution the computational complexity is reduced, while still capturing multi-scale information [8]. ResNet is used as the backbone of the model for feature extraction in the encoder part [24]. Between the encoder and the decoder part, atrous spatial pyramid pooling is performed. This attempts to handle different object scales of a class in the image for better accuracy. The decoder part is comparable to the expansive path of the U-Net where the image is restored to its original size. The number of parameters of DLV3+ is considerably larger than for U-Net, so it is computationally more expensive to run.

3.8 Implementation and Experiments

The EnFCM is implemented as a Java plugin in ImageJ software [6]. We set the fuzzyfication parameter to 2. The parameter that we tuned is the prior probability of assigning pixels to foreground. We observed that the set [0.6, 0.7, 0.8] fits best to the actual signal coverage the best. A prior probability of 0.6 means that the chance of assigning the pixels to foreground is 60%.

The U-Net model is implemented in Python 3 using the Tensorflow Keras library. The implementation is based on the U-Net coding tutorial on GitHub (https://github.com/decouples/Unet/blob/master/unet.py). The platform Google Colaboratory (Colab) [4] is used to run the model and experiments. Colab has computational resources which can be used for running the code in a Python notebook on an online server. There are several hyperparameters which can be tuned in the U-Net model. An overview of the hyperparameters are given in Table 1. In addition, Xavier uniform initializer is used to initialize the weights in the layers. Adam optimization is used with a learning rate of 0.001.

Table 1. The parameters that are studied for the U-Net.

Parameter	Range that is studied
Number of images	[100, 200, 400, 500, 1000]
Number of epochs	[5, 10, 20, 25, 40]
Number of filter layers	[5, 8, 12, 16]
Batch size	[10, 20]

The set of images used is split into a training set and validation set by a ratio of 80/20 respectively. Several combinations of parameter values are tested to see which parameters have an effect on the performance of the model. Structured experiments are done for 100, 200 and 400 input images and an increasing number of training epochs. These small subsets were created from the total 1450 images by combining sets of images from all the different batches. Subsequently, the model is tested for 1000 input images.

The DLV3+ model is implemented in Python 3 using the Tensorflow Keras library, based on the implementation by Soumik Rakshit [24]. The model is adjusted to work for the binary segmentation task of this project. A pre-trained ResNet50 model (pre-trained on ImageNet) is used as backbone for the DLV3+ model for low-level features. Because the ResNet50 model is trained for use with images of size 512 × 512, the input images are resized to be this size. The output image has a size of 512 × 512 as well. This has to be taken into account when comparing the performance results of U-Net and DLV3+. He normal initializer is used to initialize the weights of layers. Again, Colab is used to run the code and carry out the experiments.

The testing procedure is carried out similarly to U-Net. Several combinations of values of parameters, as shown in Table 2, are tested. Evaluation is done using the same set of evaluation images as is used for U-Net for fair comparison between the two models.

Table 2. The parameters that are studied for the DLV3+.

Parameter	Range that is studied
Number of images	[100, 200, 300, 400, 1000]
Number of epochs	[10, 20, 30, 45]
Learning rate	[0.001, 0.01]

4 Results

4.1 EnFCM

In order to make EnFCM comparable to U-Net and DeepLabV3+, we evaluate the performance using the same testing dataset containing 50 images representing the total dataset distribution with ground truth. The three best results is shown in Table 3. As we can observe, when the prior probability is set to 0.70, the mean IoU performance reaches the highest score of 0.85. Reducing or increasing the prior probability does not help improve the segmentation performance. Therefore, 0.70 is the optimal prior probability for our image dataset with NKX2.5 signal.

Table 3. The top 3 prior probabilities that are studied for EnFCM.

Prior Probability	mIoU
0.60	0.822
0.70	0.850
0.80	0.773

4.2 U-Net

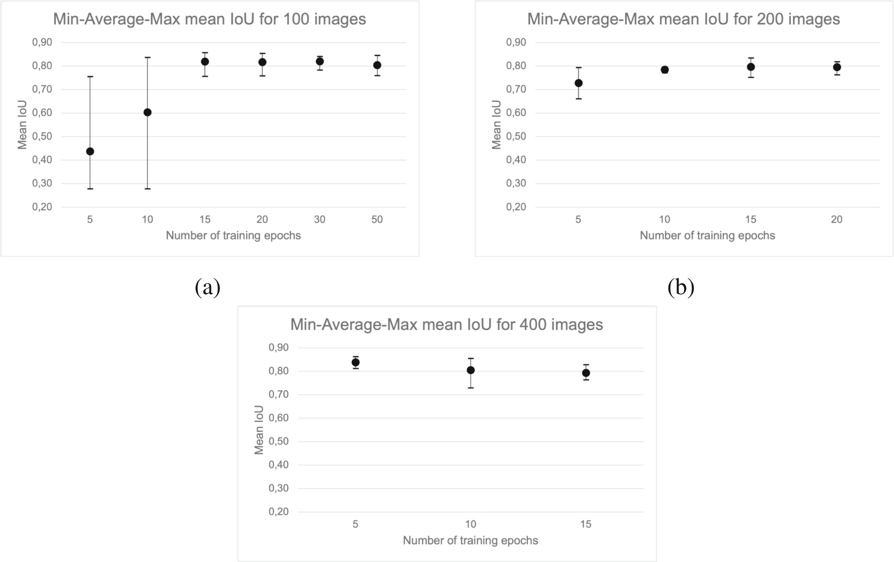
At first, various configurations of the parameter settings for the U-Net were tested. We observed that a larger training set does not necessarily result in a higher performance. An increased value for the number of filter layers resulted in a higher performance, but values for 8, 12 or 16 layers are comparable. A larger number of training epochs results in a higher performance. However, there is a plateau in performance improvement.

Based on the primary results stated above, a more structured experiment was carried out for the U-Net model using image sets of 100, 200 and 400 images to study the effect of the number of images and the number of training epochs. The parameters were kept constant at the following values: filter layers = 8, batch size = 10. The configurations are run three times to obtain an average of the performance.

The mean IoU results for 100 images are shown in Fig. 3(a). The average, lowest and highest values of the three runs are shown in the plot. It can be seen that for 5 and 10 training epochs the performance is very variable. For 15 training epochs and more, the performance is more stable and average mean IoUs above 0.80 are achieved. An increased number of training epochs above 15 does not considerably increase the performance further.

The results for 200 images are shown in Fig. 3(b). For 5 training epochs there is a larger variability in mean IoU values compare to the other numbers of training epochs. The average is also lower for 5 training epochs than for the other numbers of training epochs. The averages for 10, 15 and 20 training epochs differ slightly and the variability between the runs is comparable as well. Overall, an average mean IoU score of around 0.80 is achieved.

The mean IoU results for 400 images are shown in Fig. 3(c). The results of 5 training epochs give the highest performance and smallest variability between the runs. The average performance decreases for increasing numbers of training epochs. For 10 training epochs there is the largest variability between the runs.



(a) (b)

(c)

Fig. 3. Average mean IoU results for U-Net trained with a data set of 100, 200 and 400 images for increasing numbers of training epochs. The average is taken over three runs. The range of mean IoU score for the three runs is given using the bars and connected lines.

When the results for the three different numbers of training images are compared, the performance of the model does not increase when more training images are used. The average performance remains slightly above 0.80. A training set of 1000 images was tested as well to see if using almost all of the available images would improve the performance. However, this gave an average result of 0.794, which is lower than the results gotten from training using a lower number of images.

Final Model. A data set of 400 training images is created from the total 1450 images by combining sets of images from all the different batches. As a result, the created data set is the most representative of all the images that were gathered for this research. This general data set is used to train a final model using the parameter settings that is the most optimal from the previous experiments. The used parameters are: number of training epochs = 10, filter layers = 8, batch size = 10. ReLu is used as activation function. The mean IoU score of this model is 0.860. This model will be used to do a more detailed comparison between the U-Net and DLV3+ performance. The learning curve of the training of this model is shown in Fig. 4. In the curve it can be seen that the model has converged for both the training accuracy and the validation accuracy at around the third training epoch.

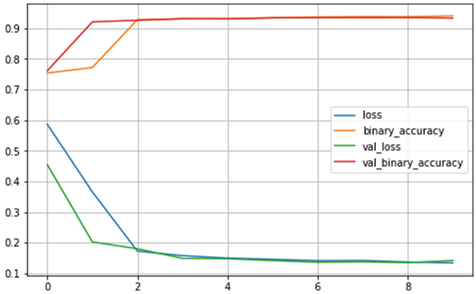


Fig. 4. Learning curve for the U-Net model. The model is trained for 10 epochs using a training set of 400 images.

4.3 DeepLabV3+

The DeepLabV3+ model is tested on performance for several configurations. First small sets of 100 and 200 images were used for the training with a small number of training epochs. This resulted in low performance scores. Then, the model was trained using 1000 images to determine if this improved the performance. It achieved a performance of above 0.75 when trained using 1000 images for 30 total training epochs. This value increased to 0.873 when the model was trained for 45 epochs. When the model was trained again for 45 epochs, but with a smaller number of training images (300 and 400), similar performance is achieved. The results indicate that if the model is trained for a larger number of epochs, a large set of training images is not essential for a high performance. Increasing the learning rate from 0.001 to 0.01 caused the model to converge faster, but the performance is less stable, and therefore lower, than for a learning rate of 0.001.

Final Model. The same general data set of 400 images is used to train a final model for DLV3+ as for U-Net. The model was created using the following parameter settings: number of training epochs = 45, batch size = 10, filter layers = 8 and learning rate = 0.001. The mean IoU score of this model is 0.890. This model will be used to do a more detailed comparison between the U-Net and DLV3+ performance. The learning curve of the training of this model is shown in Fig. 5. In the curve it can be seen that the validation loss is relatively high at the start of the training. It takes more than 30 epochs for the validation loss to decrease to a low value. The validation accuracy remains relatively low compared to the training accuracy for the first 33 epochs. This could indicate that the model is over-trained on the training data. However, after 33 epochs the validation accuracy increases to the same level as the training accuracy, which indicates that the model is generalized and is able to do proper predictions.

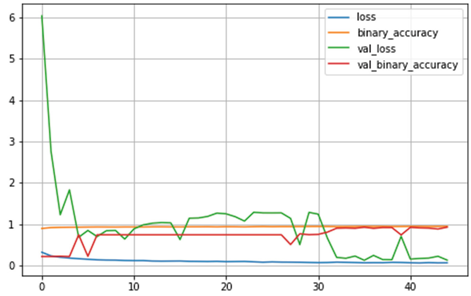


Fig. 5. Learning curve for the DLV3+ model. The model is trained for 45 epochs using a training set of 400 images.

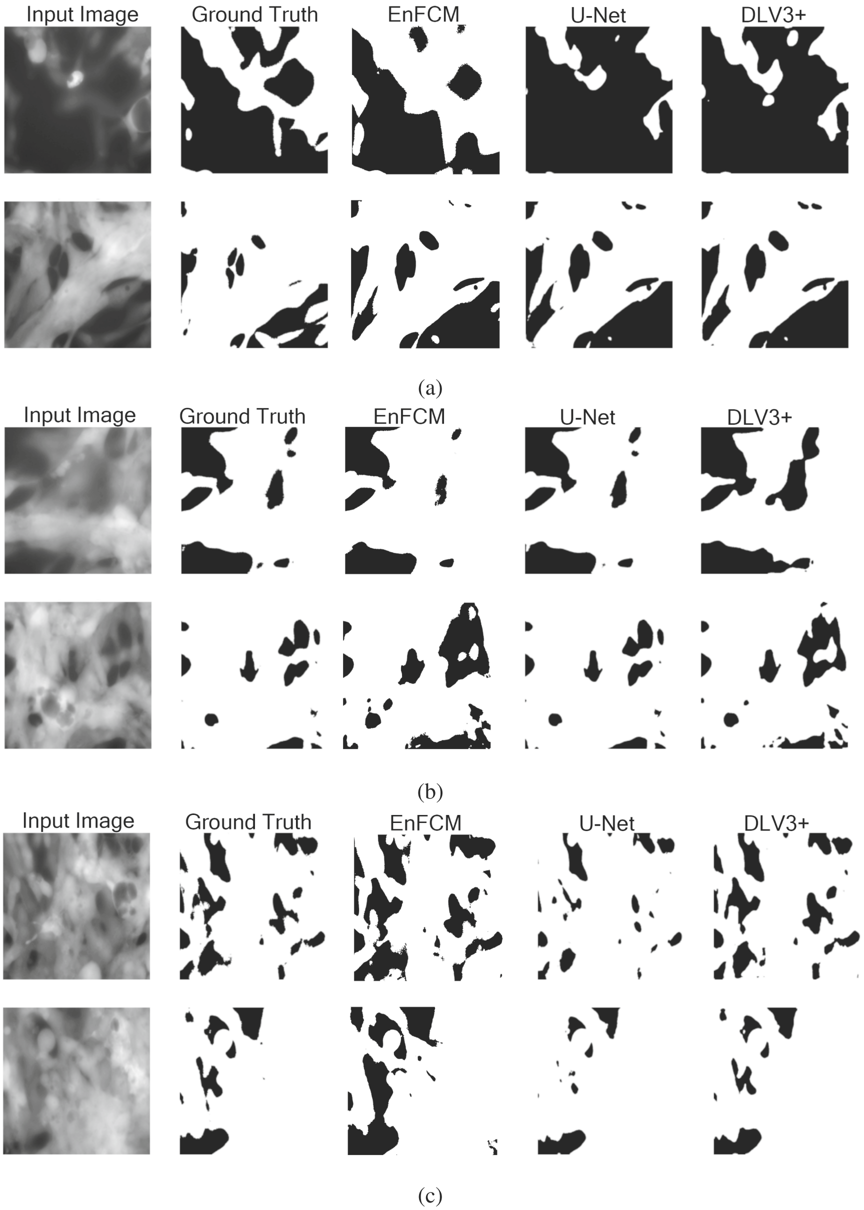
4.4 Comparison

The prediction results on the evaluation set for three methods are visualized (mIoU EnFCM = 0.850, U-Net = 0.860, mIoU DLV3+ = 0.890). Two examples are shown in Fig. 6(a). In these examples it can be seen that all three models did not give an accurate prediction when compared to the ground truth. EnFCM predicted too much foreground signal and both deep learning models predicted too much background instead of foreground signal. In Fig. 6(b), two examples are shown in which U-Net predicted more accurately when compared to the ground truth. In both examples the prediction by U-Net is very similar to the ground truth, but the prediction of EnFCM and DLV3+ contain more background region. In Fig. 6(c), two examples are shown in which DLV3+ predicted more accurately. In both examples DLV3+ showed a more correct amount of background signal compared to the EnFCM and U-Net predictions.

5 Discussion and Conclusion

In order to find a best method for segmentation of NKX2.5 signal for hPSC-CMs in a high throughput setup, three methods, including one machine learning method and two deep learning models, have been implemented, trained and evaluated. The first method that was evaluated, is EnFCM. It segments the NKX2.5 signal with a reasonable mean IoU of 0.85 when the prior probability parameter is set to 0.70. The speed of processing, which is within 1–2 s, is preferred for a high throughput setup. However, the segmentation performance is relatively lower than the other two deep learning models.

The second model is U-net. Several parameters were tuned for the U-Net model. The results showed that the U-Net model does not need a high number of training images and does not need to be trained for a high number of epochs to achieve a good performance, which was also stated in previous research [27]. A higher number of filter layers seems to improve the performance and the speed of convergence of the model slightly based on the number of training images. However, the model rapidly increases in size when the filter layers are increased, which causes the model to be very computationally expensive to run.



(a)

(b)

(c)

Fig. 6. Example prediction results.

Based on the results for the DeeplabV3+ model (Sect. 4.3), it can be concluded that the model is less efficient than the U-Net model. From the learning curves (Fig. 4 & 5) it can be seen that DLV3+ needs more training epochs to converge. The size of the DLV3+ model is also considerably larger than the size of the U-Net model (approx. 10 million parameters versus approx. 0.5 million parameters), which causes the model to be much less computationally efficient to train. Because of the ResNet50 backbone that is used for the implementation of the model in the encoder part, the sizes of the input and output images are set to 512 × 512. This limits the resolution of the output binary masks. This limitation could be solved by using a backbone that is trained using images of a larger size, but these were not available in the Keras library used at the time of the experiment. Increasing the learning rate in the DLV3+ model results in faster convergence of the model. The resulting model, however, is less stable and has, on average, a lower performance.

In Sect. 4.4, the prediction results of the three methods are visualized and compared. Based on visual inspection of the results, we observed that the predictions by U-Net are more accurate compare to the predictions by DLV3+ with respect to the ground truths. This is contradictory to the mean IoU performance scores of the models. This could be caused by that the images for the DLV3+ model are a smaller size and therefore the mean IoU score could give a slightly inaccurate indication when used to compare the models. Overall, we concluded that EnFCM predicts sometimes too much foreground signal and sometimes too much background region. It is due to the fact that there is a constant prior probability setting. In this study, we used 0.70. If the NKX2.5 signal in the image is less than 0.70, more foreground signal would be captured. If the signal in the image is more than 0.70, less foreground signal would be detected. In addition, we observed that DLV3+ predicts too much background signal. For the examples in which DLV3+ seemed to be more accurate, U-Net and EnFCM predicted too less background signal. The two examples, for which all three methods did not have an accurate prediction (Fig. 6(a)), have low NKX2.5 signal in the original image. This signal was not picked up, which could be explained by the large variance in the intensity of the signal in the image. However, for most images in the evaluation set the methods are able to distinguish most of the signal, so these are exceptions to the overall performance.

In conclusion, EnFCM provides reasonable predictions using the original size with a fast processing speed. the U-Net model is able to do the segmentation of images with a size of 1024 × 1024. The mean IoU performance of the U-Net is around 0.860. The model can converge to this score by training for at least 10 epochs using a train set of 100–400 images (batch size = 10 and filter layers = 8). The DeepLabV3+ model is able to do the segmentation of images with a size of 512×512. The mean IoU performance of this model is around 0.890. Convergence of this score can be achieved using a train set of 400 images for at least 35 epochs. The DeepLabV3+ model is computationally more expensive, needs more training epochs to converge and operates on images with a lower resolution. Adaptive learning rate could be tested in the future for faster convergence. Based on visual inspection, the prediction seems to be less accurate than U-Net.

This study has resulted in finding and validating a plausible segmentation method that can be integrated in the high throughput image analysis pipeline; i.e. U-Net. It enables automated monitoring of differentiation efficiency of hPSC-

CMs and facilitates screening of drugs for the toxicity and safety study. In the future, this work will be included in a high throughput analysis of phenotypical readouts for hPSC-CMs. The image-based phenotypical readouts can be combined with other high-throughput assays using functional and biochemical parameters to form a unique fingerprint for each drug under testing using hPSC-CMs as a cell model. It will further facilitate toxicity/safety screening and disease modeling, as well as drug discovery.

Acknowledgements. This work was supported by ERA-CVD 2016T092 and the Dutch Heart Foundation.

References

1. Abdollahi, A., Pradhan, B.: Integrating semantic edges and segmentation information for building extraction from aerial images using UNet. Mach. Learn. Appl. **6**, 100194 (2021)
2. Baghdadi, R., et al.: Tiramisu: a polyhedral compiler for dense and sparse deep learning (2020)
3. Birket, M.J., et al.: Expansion and patterning of cardiovascular progenitors derived from human pluripotent stem cells. Nat. Biotechnol. **33**(9), 970–979 (2015)
4. Bisong, E.: Google colaboratory. In: Building Machine Learning and Deep Learning Models on Google Cloud Platform, pp. 59–64. Apress (2019)
5. Cai, W., Chen, S., Zhang, D.: Fast and robust Fuzzy C-Means clustering algorithms incorporating local information for image segmentation. Pattern Recogn. **40**(3), 825–838 (2007)
6. Cao, L., van der Meer, A.D., Verbeek, F.J., Passier, R.: Automated image analysis system for studying cardiotoxicity in human pluripotent stem cell-derived cardiomyocytes. BMC Bioinform. **21**(1) (2020)
7. Cao, L., Schoenmaker, L., Ten Den, S.A., Passier, R., Schwach, V., Verbeek, F.J.: Automated sarcomere structure analysis for studying cardiotoxicity in human pluripotent stem cell-derived cardiomyocytes. Microscopy Microanal. **29**(1), 254–264 (2022)
8. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation (2018)
9. Dhingra, N., Chogovadze, G., Kunz, A.: Border-segGCN: improving semantic segmentation by refining the border outline using graph convolutional network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 865–875 (2021)
10. Elliott, D.A., et al.: NKX2-5egfp/w hESCs for isolation of human cardiac progenitors and cardiomyocytes. Nat. Methods **8**(12), 1037–1040 (2011)
11. Fontes, P.A., Thomson, A.W.: Stem cell technology. BMJ **319**(7220), 1308 (1999)
12. Guzder-Williams, B.P.: Fully convolutional networks for landcover classification and landcover change. In: AGU Fall Meeting Abstracts, vol. 2018, pp. H34B–01 (2018)
13. Innolitics, Reinhold, J., Shrestha, Y.: How to Choose a Neural Net Architecture for Medical Image Segmentation (2020)
14. Khan, Z., Yahya, N., Alsaih, K., Ali, S.S.A., Meriaudeau, F.: Evaluation of deep neural networks for semantic segmentation of prostate in T2W MRI. Sensors **20**(11), 3183 (2020)

15. Kholiavchenko, M., et al.: Contour-aware multi-label chest X-ray organ segmentation. Int. J. Comput. Assist. Radiol. Surg. **15**(3), 425–436 (2020)
16. Kim, T.W., Che, J.H., Yun, J.W.: Use of stem cells as alternative methods to animal experimentation in predictive toxicology. Regul. Toxicol. Pharmacol. **105**, 15–29 (2019)
17. Lauschke, K., et al.: Creating a human-induced pluripotent stem cell-based NKX2.5 reporter gene assay for developmental toxicity testing. Arch. Toxicol. **95**(5), 1659–1670 (2021)
18. Miklas, J.W., Salick, M.R., Kim, D.H.: High-throughput contractility assay for human stem cell-derived cardiomyocytes. Circ. Res. **124**(8), 1146–1148 (2019)
19. Oikonomopoulos, A., Kitani, T., Wu, J.C.: Pluripotent stem cell-derived cardiomyocytes as a platform for cell therapy applications: progress and hurdles for clinical translation. Mol. Ther. **26**(7), 1624–1634 (2018)
20. Paci, M., et al.: All-optical electrophysiology refines populations of in silico human iPSC-CMs for drug evaluation. Biophys. J. **118**(10), 2596–2611 (2020)
21. Pham, D.L., Xu, C., Prince, J.L.: Current methods in medical image segmentation. Annu. Rev. Biomed. Eng. **2**(1), 315–337 (2000)
22. Prajapati, C., Pölönen, R.P., Aalto-Setälä, K.: Simultaneous recordings of action potentials and calcium transients from human induced pluripotent stem cell derived cardiomyocytes. Biology Open (2018)
23. Psaras, Y., et al.: CalTrack: high-throughput automated calcium transient analysis in cardiomyocytes. Circ. Res. **129**(2), 326–341 (2021)
24. Rakshit, K.: Keras documentation: multiclass semantic segmentation using DeepLabV3+ (2021)
25. Ribeiro, A.J.S., et al.: Contractility of single cardiomyocytes differentiated from pluripotent stem cells depends on physiological shape and substrate stiffness. Proc. Natl. Acad. Sci. **112**(41), 12705–12710 (2015)
26. Ribeiro, M.C., et al.: A cardiomyocyte show of force: a fluorescent alpha-actinin reporter line sheds light on human cardiomyocyte contractility versus substrate stiffness. J. Mol. Cell. Cardiol. **141**, 54–64 (2020)
27. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
28. Rubin, L.: Stem cells and drug discovery: the beginning of a new era? Cell **132**(4) (2008)
29. Schindelin, J., et al.: Fiji: an open-source platform for biological-image analysis. Nat. Methods **9**(7), 676–682 (2012)
30. Szilagyi, L., Benyo, Z., Szilagyi, S.M., Adam, H.S.: MR brain image segmentation using an enhanced Fuzzy C-Means algorithm. In: Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No. 03CH37439), vol. 1, pp. 724–726 (2003)
31. Tsang, S.H.: Review: DeepLabv3+ - Atrous Separable Convolution (Semantic Segmentation) (2021)
32. Yamamoto, W., et al.: Electrophysiological characteristics of human iPSC-derived cardiomyocytes for the assessment of drug-induced proarrhythmic potential. PLoS ONE **11**(12), e0167348 (2016)

Deep Learning Implementation for Pineapple Sweetness Classification

Sayed Muchallil[1], Silvia Roza[1], Muhammad Hafez Al-Assad[1],
Yudi Candra[1], Maya Fitria[1,2], and Rahmad Dawood[1,2](✉)

[1] Department of Electrical and Computer Engineering, Universitas Syiah Kuala,
Banda Aceh, Aceh 23111, Indonesia
{sayed.muchallil,mayafitria,rahmad.dawood}@usk.ac.id,
{silviaroza,m.hafez,yudi.c}@mhs.usk.ac.id
[2] Telematics Research Center, Universitas Syiah Kuala, Banda Aceh, Aceh 23111,
Indonesia

Abstract. In 2022, Indonesia produced roughly 3.203.775 metric tons of pineapples. The sweetness of pineapples is an important factor in determining their selling price. Typically, a pineapple's sweetness is determined manually, which is time-consuming and prone to human error, resulting in a lower selling price and pineapple waste due to unsold fruit. To address this issue, we developed a deep learning-based algorithm for identifying the sweetness of pineapples based on the hypothesis that a pineapple's sweetness is closely related to its color and maturity. To train the algorithm, we took 872 images of freshly harvested Smooth Cayenne pineapples and measured their sweetness with a Brix measuring device. We use two algorithms, EffisientNetB4 and RestNet-50, both with a learning rate of 0.000001, to differentiate between sweet and not-so-sweet pineapple. Both algorithms detected pineapple sweetness with the same accuracy of 84.09%. However, RestNet50 had a greater loss than EffisintNetB4.

Keywords: Pineapple · EffisientNetB4 · ResNet-50 · Sweetness

1 Introduction

Indonesia produced more than 3 million pineapples in 2022 alone [1]. Aceh province, one of 38 provinces in Indonesia, contributed to the pineapple harvest. Based on the contribution, we chose the province as our research area. However, the sorting process relies intensely on humans. There are two drawbacks to using the human senses to classify pineapples. First, it is a time-consuming process. It is also vulnerable to human error.

Typically, people determine the maturity or ripeness of pineapple based on four factors: color, leaves, smell, and touch. These factors are judged using human abilities, whose perception differs from person to person. Two people may have different perspectives on the same issue because one is overworked. This is known

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 185–197, 2023.
https://doi.org/10.1007/978-981-99-7969-1_14

Table 1. Manual Method to Determine the Pineapple Ripeness.

Factors	Senses	Decision
Color	golden yellow, no green patches	ripe
Leaves	soft and flexible	ripe
Smell	fragrant	ripe
Touch	soft	ripe

as human error. Table 1 shows how to decide the maturity of pineapples using the human senses.

We proposed a deep learning algorithm to classify pineapple sweetness based on the combination of the Brix degree value, the color, and the scale size of pineapples. Brix values are measured using a refractometer, as shown in Fig. 1. An image of sweet and unsweet pineapples can be seen in Fig. 2 for color and scale size. The composite we use to build the dataset will be described in detail in the next section.

We chose EffisientNetB4 because it is a part of the EfficientNet family known for its efficiency and accuracy in image classification. This algorithm can be one of the best fits for differentiating between object color, texture, and size. All of these elements contribute to our study determining the ripeness and sweetness of pineapples. This algorithm is also well-known for its transfer learning capability. Since we also plan to do our next experiment with other fruit, we hope to implement this benefit in future research.

The other algorithm we use in our research is RestNet50. This algorithm is well-known for determining fruit maturity, ranging from unripe, half-ripe, ripe, and over-ripe. With this capability in determining the ripeness stages, the classification of sweet or unsweet pineapple will be highly accurate.

In our study, there are two significant elements in determining the sweetness. First are the physical factors, such as scales and color. Sweet pineapples have large scales of a uniformly distributed golden color. In contrast, unsweet pineapples' skin is mostly a greenish color and has smaller scales. We used 12 Brix values as the threshold to separate the sweet from unsweet pineapples. If the value is 12 or higher, the pineapples are sweet; otherwise, they are unsweet.

The paper is structured as follows: The following section is a literature review of the deep learning algorithms used in agriculture. Next, we provide our research methodology, including how we collected our dataset. Then, the results and discussion are presented. Finally, we conclude our research findings.

2 Related Works

2.1 Related Works

The Philippines' pineapple volume in 2021 was about 2.8 tonnes [2]. Based on the data, [3] studied how to automate the process of determining the ripeness of pineapples based on the Philippine Standard employing Support Vector Machine

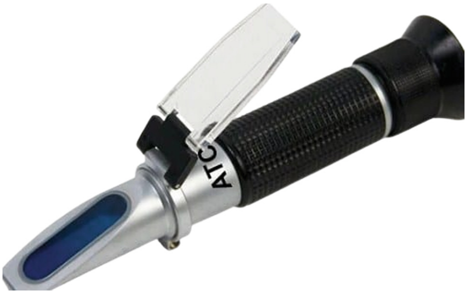
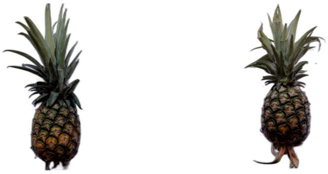


Fig. 1. A Refractometer.



(a) Sweet (b) Unsweet

Fig. 2. Sweet and Unsweet Pineapple Comparison.

(SVM) and Hue, Saturation, Value (HSV) Color Space. The results suggested that the system wrongly classified two ripe pineapples as not ripe and overripe, so the accuracy for the ripeness is only 87.5%.

In 2019, [4] proposed a MobilNet architecture for a pineapple sorting machine based on skin color only to classify the level of pineapple ripeness. They divided the ripeness into three categories: unripe, partially ripe, and ripe pineapples. MobileNet was used as one of the Convolutional Neural Network (CNN) architectures to help classify without many human workers. The results from confusion metrics in training were 79% accuracy for unripe pineapples, 82% for partially ripe pineapples, and 100% for fully ripe pineapples after training. The test accuracy was 98.38% and 90.77% when evaluating with the test set.

Cocoa harvesting still used the conventional method to determine the ripeness by perceiving the hollow sound from tapping the pod. [5] studied the method using acoustic sensing devices to record noiseless acoustic signals generated from tapping cocoa pods while on the tree. The experiment only resulted in two classes: ripe and unripe. Frequency-domain analysis was used to extract the spectral characteristics, including the first three dominant resonant frequencies, their corresponding amplitudes, and their power spectral densities. Time-domain features were also used, such as Short-time Energy and Zero-Crossing Rate. The eleven acoustic features of unripe and ripe samples were examined using Scatter plots. Two datasets are used: 272 for training and 120 for testing. The training dataset has eleven feature vectors whose class labels are already known as either ripe or unripe. The experimental results showed that combining the first two dominant resonant frequencies into a feature vector using an SVM classifier tool gave the maximum classification accuracy. The classification model output was tested and found to correctly classify cocoa ripeness with 95.8% overall accuracy.

[6] proposed pineapple quality evaluation system is a double inputs single output (DISO) system whose input variables consist of feature extraction of pineapple size and weight. The output is the pineapple quality classification determined by fuzzy data decision processing under the Thai Agricultural Commodity and Food Standard (TACFS 4-2003). The input variables are fuzzified by designating them a singleton fuzzy set, the fuzzy set of the output variable is deduced by the max-min composition, and the fuzzy relation describes the TACFS 4-2003. The experiment is divided into three parts: Part A is the process of image acquisition, image processing, and feature extraction. Next, part B determines the weight inspection. Finally, Part C assigns fuzzy decision-making to evaluate the pineapple quality grading standard and report the outcome. The experiment aims to determine the effectiveness of the proposed system by comparing the experiment results to the actual measurements. The experiment objects were 24 fresh pineapples devoid of crowns, each of which was introduced into a light control box and automatically graded for pineapple quality. The size of each pineapple was measured using a vernier caliper and a weighing instrument. The experiment results revealed that the proposed grading system for pineapples without a crown is 87.5% accurate.

Previous studies presented an approach to predicting and detecting pineapple ripening periods using overlapping anchor boxes, Intersection Over Union (IOU), and cross-stage partial backbones [7]. The dataset includes 36,000 photos from five growth stages of pineapples taken at a local garden. The model is trained with 30,000 independently labeled images and evaluated on a test set of 6,000 images. The implementation process involves using the YOLOv5 configuration, editing the parameters, and creating the data.yml file with parameters. The model is tested and evaluated using Google Colab, a high GPU-based environment. The training model achieved good results in detecting and identifying pineapples and their ripening periods, with an accuracy of over 95%. The results showed that YOLOv5 outperformed YOLOv3. The YOLOv4 also cannot compete with YOLOv5 in terms of speed.

A group of researchers from Indonesia believe that Artificial Intelligence (AI) can provide time efficiency and minimize human error compared to traditional methods [8]. This research focuses on a web-based AI application that can count pineapple objects in a wide area using hundreds of aerial photographs as a training dataset. The accuracy of pineapple object counting with AI model utilization will optimize the use of resources such as water, fertilizer, insecticides, and packaging materials. The utilization of UAVs in the agroindustry is growing, with UAV-captured image data being utilized as training data to recognize and count specific objects. To help the experiment, the author developed an interactive web-based application for counting pineapple objects using aerial photographs of a pineapple farm. The final result is an interactive web-based application with AI-supported functions that report the total number of pineapple objects in a photographed and stitched area. The research involved prototyping a Crowdsourcing Annotation System and a concept of Information System Design for Deep Learning Based Plant Counting Automation.

[9] tried to identify passion fruit ripeness using K-Means Clustering and Artificial Neural Networks. The fruit is classified into three phases: ripe, near-ripe, and unripe. The passion fruit-sorting system with artificial intelligence is an innovation for industrial markets, as it is cost-efficient and effective for large production processes. The data used in this research were 95 passion fruit videos divided into 80% for training and 20% for testing. The training data comprised 75 passion fruit videos, with 40 ripe, 10 near-ripe, and 25 unripe passion fruit.

[10] also conducted a study on pineapple maturity in Thailand. The proposed model is divided into two modules: Spiral Pattern Modeling, which operates on the whole pineapple image, and Pineapple Scale Modeling, executed on each scale, which starts with image acquisition by obtaining a pineapple panorama image by registering a series of images taken from different views of the fruit on a turntable. The process then finds scale locations in a panorama image by generating a binary image of candidate areas of scales and identifying them by thresholding the blue channel of the image. Standard morphological closing and distance transformation are applied to the result, resulting in an image with groups of pixels at the center of each scale. The component labeling algorithm is applied, and the centroid of each component is calculated and used to represent the scale location. The proposed method focuses on identifying the boundary of each pineapple scale using techniques such as Geodesic Active Contours (GAC) and Scale Location Detection.

3 Research Material and Methodology

The proposed workflow can be seen in Fig. 3. We start our research by collecting data to build our dataset. Since the data are digital images, we have to preprocess the images. The preprocessing step includes but is not limited to resizing the pineapple images. Next, we augment our data to increase the number of samples in our dataset. Finally, using the data, we run the model training and testing.

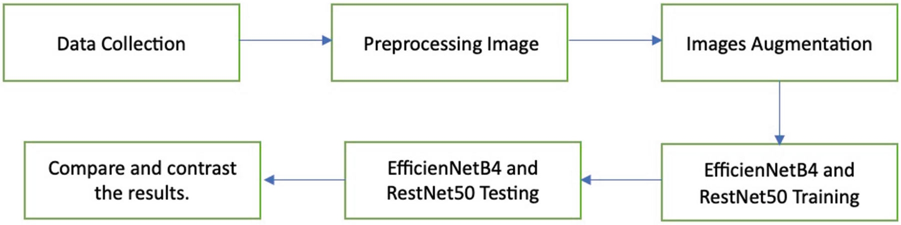

Data Collection
Preprocessing Image
Images Augmentation
EfficienNetB4 and RestNet50 Training
EfficienNetB4 and RestNet50 Testing
Compare and contrast the results.

Fig. 3. Research Workflow.

The first step of our data collection phase is to take four pictures of each pineapple. The camera setup can be seen in Fig. 4. The images are saved to a different directory based on the camera that shoots the pineapple. Since we have four cameras, we have four directories named “1”, “2”, “3”, and “4”. These directories are inside the sweet (Manis) and unsweet (Tidak Manis) directories. The directories’ structure is shown in Fig. 5.

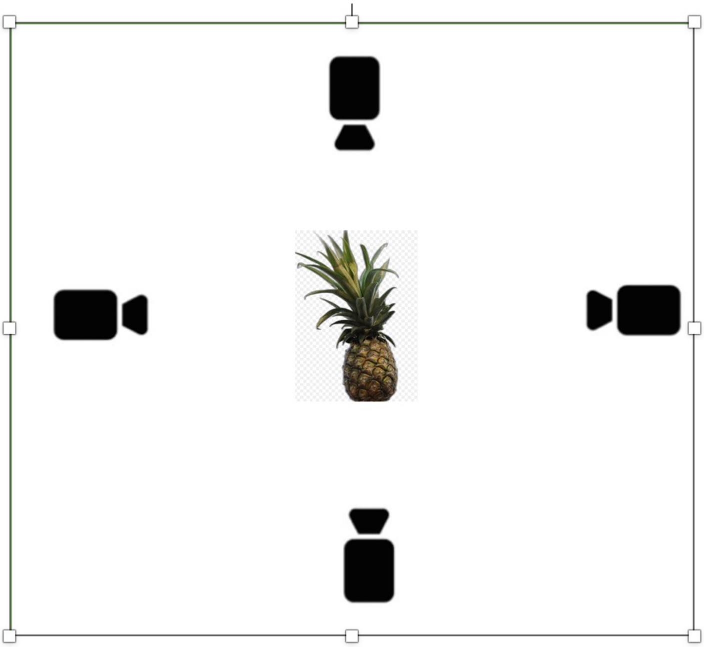
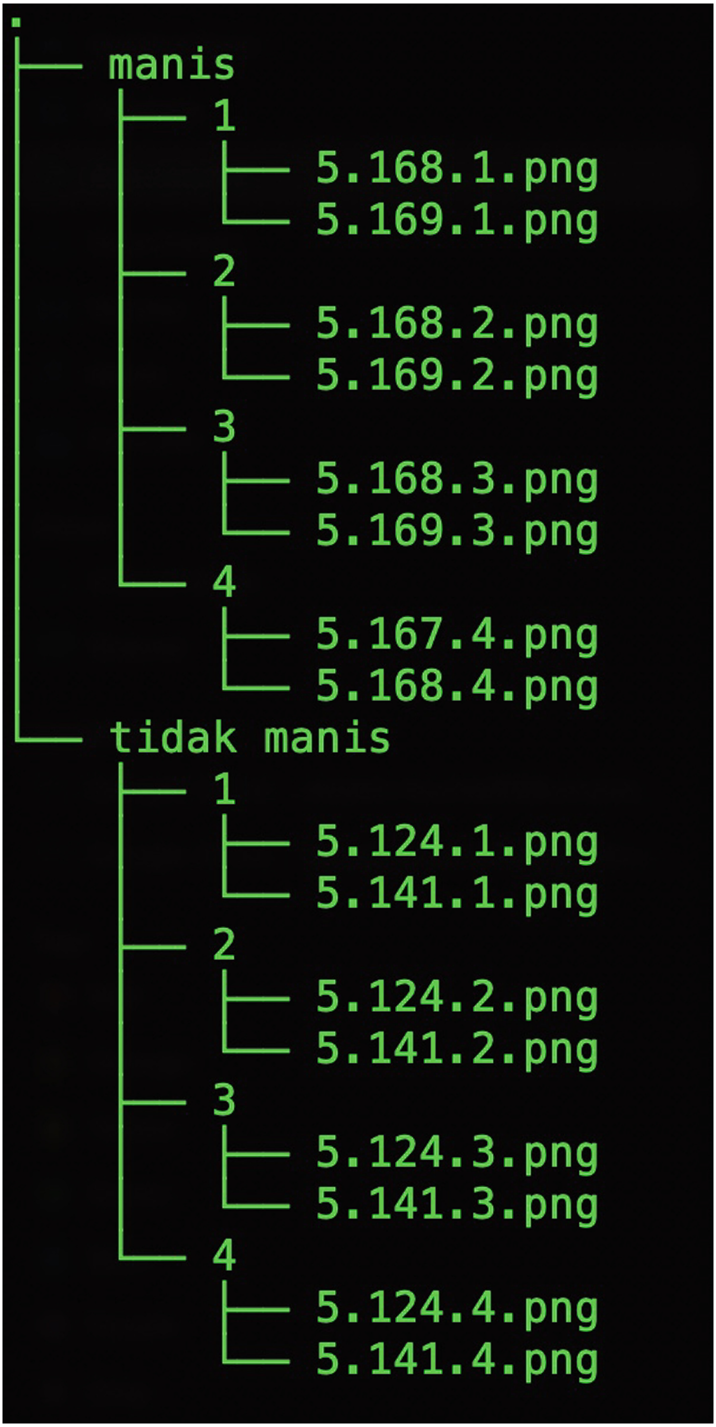


Fig. 4. Camera Setup.

Before saving the pineapple images to the computer, we implement segmentation to remove the background of every image. Then, we resize the image to 224 × 224 pixels. Finally, the augmentation process took place. The data augmentation became an essential part of our research to increase the diversity of the training data. We hope that this process will improve the performance. Flipping, rotation, and Gaussian blurring are the augmentation processes used in our experiment. Some of the images in the dataset are flipped horizontally or vertically. The other images are rotated, ranging from 15 to 360°. Also, we chose Gaussian blurring from 0.1 to 0.5. Finally, we augment.

Building the dataset, we also measure the Brix value of every pineapple. If the value is 12 or higher, we consider the pineapple sweet. A small amount of pineapple is extracted, and we put the juice into the refractometer to get the Brix values. The process can be seen in Fig. 6. The detail of data used in



```
.
├── manis
│   ├── 1
│   │   ├── 5.168.1.png
│   │   └── 5.169.1.png
│   ├── 2
│   │   ├── 5.168.2.png
│   │   └── 5.169.2.png
│   ├── 3
│   │   ├── 5.168.3.png
│   │   └── 5.169.3.png
│   └── 4
│       ├── 5.167.4.png
│       └── 5.168.4.png
└── tidak manis
    ├── 1
    │   ├── 5.124.1.png
    │   └── 5.141.1.png
    ├── 2
    │   ├── 5.124.2.png
    │   └── 5.141.2.png
    ├── 3
    │   ├── 5.124.3.png
    │   └── 5.141.3.png
    └── 4
        ├── 5.124.4.png
        └── 5.141.4.png
```

Fig. 5. Directory Structure.

our experiment can be seen in Table 2. We divide our total data for training, validation, and testing the architectures, as shown in Table 3.

Table 2. Total Data in our Dataset.

	Original Data	Augmented Data	Total Data
Sweet	460	140	600
Unsweet	216	384	600
Total	676	524	1200

Table 3. Data Segmentation for Training, Validation, and Testing.

	Sweet	Unsweet	Total
Training	340	340	680
Validation	150	150	300
Testing	110	110	220
Total	600	600	1200

4 Result and Discussion

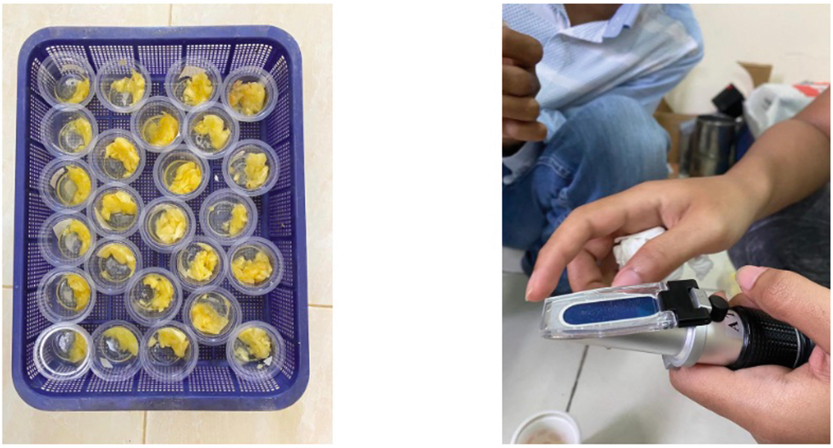
The result is presented in two parts. First, we will discuss the EfficientNetB4 architecture. The second is RestNet50. We show the confusion matrix at the beginning in both parts to compare the accuracy. Next, we compared the training and validation process for accuracy and loss to indicate whether the model is considering overfitting. The overall result is shown in Table 4.

Table 4. Result Comparison for Both Architectures.

Architectures	Accuracy	Loss	Precision	Recall	F-Score
EfficientNetB4	0.8409	0.3143	0.8409	0.8409	0.8409
ResNet-50	0.8409	0.4185	0.8000	0.9090	0.8510

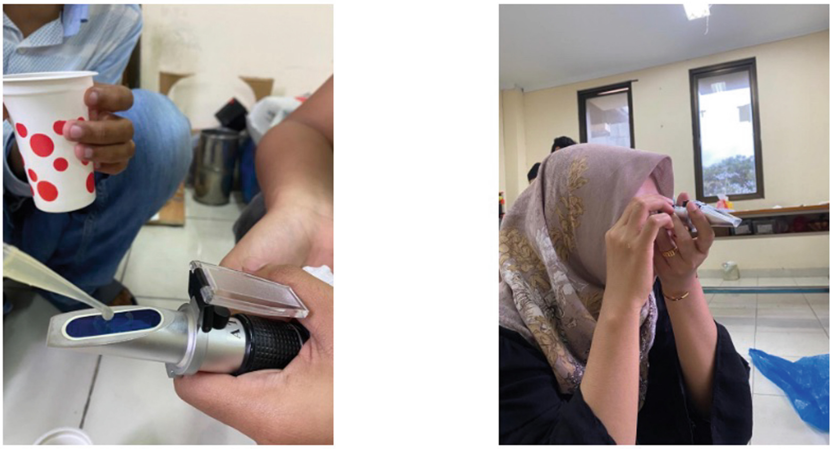
EfficientNetB4

The confusion matrix in Fig. 7 suggests that some sweet pineapples are recognized as unsweet, and the same number of unsweet pineapples are detected as sweet pineapples. From this matrix, this model's accuracy is 84.09%.



Pineapple extract

Refractometer



Juice sampling

Brix Value

Fig. 6. Brix Measurement Process.

Figure 8 shows the validation and training learning curve accuracy for the EfficientNetB4 architecture. The validation and training do not have a wide gap or a different trend. From the result, it can be inferred that our model does not promote overfitting. The loss graph in Fig. 9 also suggests the same result. In addition, the graph also displays that the EfficientNetB4 performance increases as the epoch number rises. Meanwhile, the loss decreases when the number of epochs increases, which is also a good sign. In addition, the graph also displays that the EfficientNetB4 performance increases as the epoch number rises. The loss decreases when the number of epochs increases, which is also a good sign.

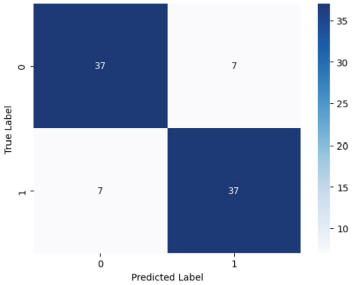


Fig. 7. EfficientNetB4 Confusion Matrix.

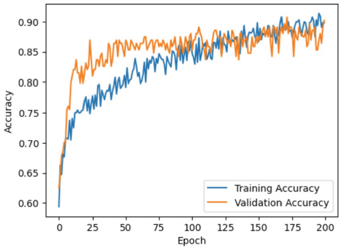


Fig. 8. EfficientNetB4 Accuracy Training and Validation Learning Curve.

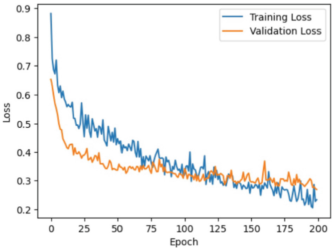


Fig. 9. EfficientNetB4 Loss Training and Validation.

RestNet50

When using the ResNet-50 model, the data that were accurately predicted as belonging to the sweet and not-sweet classes were 34 and 40, respectively. Ten unsweet data were incorrectly detected as sweet pineapples, and four sweet pineapples were indicated as being unsweet. Unlike the first architecture, this architecture is more open to overfitting/underfitting based on the accuracy learning curve as shown in Fig. 11, where the validation graph can be higher than the training result. Data loss in training is stable after epoch 25. However, the validation loss can be lower than the training result and decreases after epoch 75. The loss learning curve comparison can be seen in Fig. 12.

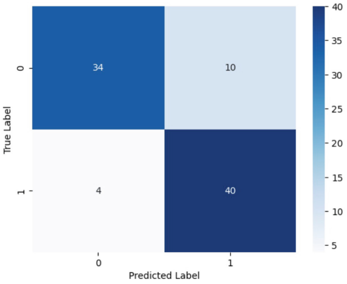


Fig. 10. RestNet-50 Confusion Matrix.

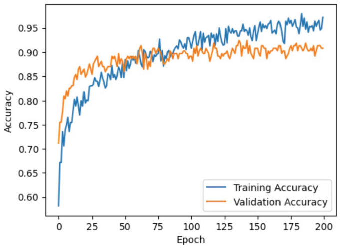


Fig. 11. RestNet-50 Accuracy Training and Validation Learning Curve.

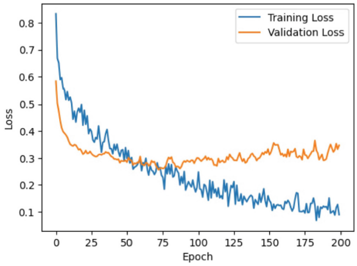


Fig. 12. RestNet-50 Loss Training and Validation.

5 Conclusion

Our study has resulted in two important results. First, the pineapple dataset can be used for other CNN architectures. The other result is the comparison of these two architectures regarding the accuracy, loss, precision, recall, and F1-score. Regarding accuracy, both architectures, EfficientNetB4 and RestNet-50, reach 84.09% in our study by not considering the training time. However, RestNet-50 produces more loss than the other architecture. Nevertheless, RestNet-50 has a higher recall and F1-score than EfficientNetB4.

6 Future Research

This study is our initial research in determining the maturity of pineapples. We plan to conduct further research on pineapple ripeness and sweetness detection. The future study will investigate the relationship between sweetness and the pineapple species. Further, our next experiment can be conducted with a non-destructive method to determine pineapple sweetness.

Acknowledgment. Keio University and APNIC Foundation supported this research under the CBR grant scheme 2022.

References

1. Statistics Indonesia. http://bps.go.id/indicator/55/62/1/produksi-tanaman-buah-buahan.hrml. Accessed 30 June 2023
2. Statistica. https://www.statista.com/statistics/751582/philippines-pineapple-production/. Accessed 1 July 2023
3. Aguilar, E.J.L., Borromeo, G.K.P., Villaverde, J.F.: Determination of pineapple ripeness using support vector machine for Philippine standards. In: 2021 IEEE 7th International Conference on Control Science and Systems Engineering (ICCSSE) on Proceedings, China, pp. 283–287. IEEE (2021). https://doi.org/10.1109/ICCSSE52761.2021.9545163
4. Chaikaew, A., Thanavanich, T., Duangtang, P., Sriwanna, K., Jaikhang, W.: Convolutional neural network for pineapple ripeness classification machine. In: 2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) on Proceedings, Pattaya, Thailand, pp. 373–376, IEEE (2019). https://doi.org/10.1109/ECTI-CON47248.2019.8955408
5. Arenga, D.Z.H., Cruz, J.C.D.: Ripeness classification of cocoa through acoustic sensing and machine learning. In: 2017 IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM) on Proceedings, Manila, Philippines, pp. 1–6. IEEE (2017). https://doi.org/10.1109/HNICEM.2017.8269438
6. Suksawat, B., Komkum, P.: Pineapple quality grading using image processing and fuzzy logic based on Thai Agriculture Standards. In: 2015 International Conference on Control, Automation and Robotics in Proceedings, Singapore, pp. 218–222. IEEE (2015). https://doi.org/10.1109/ICCAR.2015.7166035
7. Sharma, A.K., Nguyen, H.H.C., Bui, T.X., Bhardwa, S., Van Thang, D.: An approach to ripening of pineapple fruit with model Yolo v5. In: 2022 IEEE 7th International Conference for Convergence in Technology (I2CT) in Proceedings, Mumbai, India, pp. 1–5. IEEE (2022). https://doi.org/10.1109/I2CT54291.2022.9824067
8. Rahutomo, R., Perbangsa, A.S., Lie, Y., Cenggoro, T.W., Pardamean, B.: Artificial intelligence model implementation in web-based application for pineapple object counting. In: 2019 International Conference on Information Management and Technology (ICIMTech) in Proceedings, Jakarta/Bali, Indonesia, pp. 525–530. IEEE (2019). https://doi.org/10.1109/ICIMTech.2019.8843741

9. Sidehabi, S.W., Suyuti, A., Areni, I.S., Nurtanio, I.: Classification on passion fruit's ripeness using K-means clustering and artificial neural network. In: 2018 International Conference on Information and Communications Technology (ICOIACT) in Proceedings, Yogyakarta, Indonesia, pp. 304–309. IEEE (2018). https://doi.org/10.1109/ICOIACT.2018.8350728
10. Kaewapichai, W., Kaewtrakulpong, P., Prateepasen, A., Khongkraphan, K.: Fitting a pineapple model for automatic maturity grading. In: 2007 IEEE International Conference on Image Processing in proceedings, San Antonio, TX, USA, pp. I-257–I-260. IEEE (2007). https://doi.org/10.1109/ICIP.2007.4378940

Improving Low Light Object Detection Using Image Enhancement Models

Muhammad Omer Farooq Bhatti(✉) and Matthew N. Dailey

Asian Institute of Technology, Khlong Nueng, Pathum Thani, Thailand
`omerfbhatti@gmail.com`, `dailey.matthew@gmail.com`

Abstract. This paper explores the particular issue of object detection under low-light situations, in which, due to low exposure, object detectors may perform poorly. We present a comparative study of various state-of-the-art image enhancement models for the purpose of facilitating object detection under low light. We also propose a new method for robust low-light object detection that shows substantial improvements over state-of-the-art baselines. The proposed approach increases detection robustness to different lighting conditions and establishes a state-of-the-art mAP_{50} of 79.5% on the ExDark dataset.

Keywords: Deep learning · Low-light perception · Object detection

1 Introduction

Despite a great deal of progress made to date on autonomous driving systems (ADS), fully automated driving without the need for a human fail-safe still eludes the industry. A significant portion of the problem stems from issues in perception modules, such as those arising from low and varying illumination across the field of view. Some modern ADSs utilize multiple sensor modules to overcome this problem, but since sophisticated sensors result in increased manufacturing costs, major ADS vendors such as Tesla have recently announced their intention to switch to camera-only perception [9].

Although ADS research has come very far since its inception, there are still unsolved problems left to be tackled. Real-time object detection [13] and segmentation [4] are now practical, with very good accuracy under standardized conditions. But the benchmark datasets [8,16] on which these models are typically evaluated only contain good quality images of well-lit scenes. When the ADS encounters non-standard conditions, there is still significant potential for system failure as the model encounters out-of-domain data. Such conditions include disparity between training data and encountered conditions, low-light scenarios, illumination variance, glare/occlusion, and adverse weather conditions such as rain and fog.

Within the area of low-light perception, object detection under low illumination conditions is a critical problem that needs to be addressed. Low light situations lead to a lack of information in the input image, which makes it harder

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 198–212, 2023.
https://doi.org/10.1007/978-981-99-7969-1_15

for downstream models to learn patterns that will generalize well to unseen data [17]. Previous approaches to this problem have utilized simple pre-processing methods such as histogram equalization, other image enhancement techniques, or raw sensor data to account for the low illumination conditions. There have also been approaches that explore end-to-end training [11] and single-step approaches [7]. The best current benchmark for object detection under low-light conditions is the ExDark [17] dataset.

Beyond the simple pre-processing methods mentioned above, another approach for low-light object detection is to use a more sophisticated image enhancement model on the low-light image prior to carrying out the high-level vision task. This has been demonstrated by Guo et al. [10] for the face detection task. The reasoning behind the aim of enhancement as a pre-processing step is that low-light images may contain features that are not easily learned by a detection model such as YOLOv5, suggesting a pre-processing step that makes the features/information more discernible to the downstream model. However, these types of image enhancement methods are limited by the fact that they cannot generate information that is not already present in the input image; they can only enhance certain features to make them easier to learn by a model with specific capacity and a particular optimization objective. A single model, using domain adaption or a multi-tasking approach, could in principle enable greater knowledge sharing, which would result in better latent features, reducing the required number of overall model parameters and enabling faster processing [22].

Related to this issue, the evaluation metrics currently used to benchmark image enhancement models, namely Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM), are not reflective of the performance of the image enhancement models specifically for high-level vision tasks such as object detection or image segmentation. Training a separate model using task-agnostic criteria is not necessarily optimal for producing a processing stream that is proficient at high-level vision tasks. Evaluation protocols that consider both image enhancement and object detection together are needed.

In this paper, we compare several state-of-the-art low-light enhancement models in terms of their utility for the low-light object detection task. We further propose a new model for robust low-light object detection that improves on the state-of-the-art, and we show that the features learned by this model generalize across various lighting conditions much better than the alternatives.

2 Related Work

Research to overcome the above-mentioned limitations is ongoing. Several learning methods that address the low-light image problem have been proposed, including approaches based on an auto-encoder architectures, CNNs, retinex theory, and generative models, among others.

2.1 Auto-Encoder and CNN Based Methods

LLNet [18] was one of the first deep learning approaches to the problem of enhancing image illumination. It utilizes a stacked sparse deep auto-encoder (SSDA) with three encoder layers and three decoder layers for image enhancement and de-noising. This approach is, however, quite computationally heavy, as the encoder and decoder layers require many mathematical operations; furthermore, LLNet scores relatively lowly on the SSIM metric. MBLLEN [19] is a CNN-based model optimized on three loss functions: structure loss, context loss, and region loss. MBLLEN gives state-of-the-art results for image enhancement; however, the model's complexity is relatively high, disqualifying it for real-time applications such as ADSs.

2.2 Retinex Theory Based Methods

There have been several attempts, such as LightenNet [15], RetinexNet [5], and KinD [24], that combine retinex theory and deep learning to approach the low-light image enhancement problem. The retinex theory approach seeks to decompose the image into reflectance and illumination maps, after which the reflectance map is passed through a de-noising mechanism and the illumination map is passed through an illumination-enhancing mechanism, followed by a step in which the results are recombined to obtain an ideal result. The theory operates on the principle that the reflectance map of an image remains constant under any illumination condition. Among these models, KinD is the best performing model both in terms of evaluation metrics and computational efficiency.

2.3 Other Approaches

Other approaches include the use of adversarial training, deep curve estimation, and transformers to generate an enhanced image from a low-light image. Among these, EnlightenGAN [12] and Zero-DCE [10] are self-supervised methods. Zero-DCE learns a mapping from low-light to corresponding enhanced images by estimating a pixel-wise intensity mapping curve. The parameters defining the mapping curve for a specific image are only dependent on the image itself and are optimized using four non-reference functions: spatial consistency loss, exposure control loss, color constancy loss, and illumination smoothness loss. This approach does not require a paired image training dataset, so it is much easier to adapt to different scenarios in which paired data are not available or are insufficiently realistic.

EnlightenGAN utilizes adversarial training with a self-regularized perceptual loss function and a global-local discriminator to generate enhanced images. IAT [6] uses attention-based methods to convert an input RGB image into raw data and tweak the parameters of an ISP (Image Signal Processing) pipeline to reconstruct the enhanced image. This method only requires 90k parameters and rivals Zero-DCE in terms of least computation time. It also provides state-of-the-art results on the LOL dataset [5]. However, this approach does require a paired image dataset for training.

MAXIM [23] uses a MLP-based architecture to perform low-level vision tasks. It has produced state of the art results for various vision tasks such as de-blurring, de-hazing, de-raining and image enhancement. However, MAXIM, being based on a MLP architecture, requires a large amount of computation and thus may not be suitable for real-time use. Ma et al. [20] present SCI, a model which uses a self-calibrated module and unsupervised training loss in order to constrain the illumination learning and aid convergence between results of each stage in a multi-stage training operation. This model is much less complex in comparison to MAXIM and easily able to run in real time.

Most of these approaches require a dataset consisting of paired low-light and normal-light images, which can be hard to acquire. Datasets such as LOL [5], MIT-Adobe FiveK [2], and SID [3] are usually used for training this type of model. These datasets contain a large number of images of the same scene, captured using high and low exposure times. The images may be taken both indoors as well as outdoors and contain common objects and scenes, but lack diversity in terms of scenes.

Table 1. Comparison of low-light image enhancement results on the LOL dataset.

Model	PSNR	SSIM
LLNet	**17.959**	0.713
LightenNet	10.301	0.402
RetinexNet	16.774	0.462
EnlightenGAN	17.483	0.677
MBLLEN	17.902	0.715
KinD	17.648	**0.779**
Zero-DCE	14.861	0.589

Note. Reprinted from Li et al. [14].

2.4 Low-Light Image Enhancement for High-Level Vision Tasks

There are at least two ways to address the problem that low-light and low-exposure images cause for high-level vision tasks. The first approach seeks to first improve the exposure of an input image so that it can then be fed through a deep neural network for another vision task on high quality images. The second approach is to utilize a unified neural network to perform the high-level vision task (such as object detection) directly on low-light images.

Most current work takes the first approach. Improving the exposure of low-light images is potentially beneficial in many applications, not only for ADSs. Many researchers thus approach the problem as a general low-level vision task.

However, these efforts have the downside of being designed primarily to perform well on image enhancement metrics and have not been integrated into high-level vision systems. Al Sobbahi and Tekli [1] present a comparative study of some image enhancement and object detection models for low-light object detection, with YOLOv3 providing the best results.

Sasagawa et al. [22] explore the possibility of knowledge distillation and domain adaptation for training a unified YOLO model to "see in the dark." They utilize a SID model [3] fused with a YOLO backbone using glue layers in order to enable a YOLO model pre-trained on the MS-COCO dataset to detect objects in dark images. Since the SID model uses RAW image data as input, the composed model likewise only utilizes RAW image data for detection. Guo et al. [11] combine multiple image enhancement networks with Faster-RCNN [21] in order to improve low-light object detection.

Another approach has been taken by MAET [7], a multi-task model providing object detection in low-light images. To the best of our knowledge, this model provides the previous state-of-the-art results (mAP_{50} 74%) on the ExDark [17] dataset.

We perform a comparative evaluation of some recent image enhancement models for low-light object detection using YOLOv5 detectors, and we present a new model obtained through joint training that achieves state-of-the-art results.

3 Methodology

In order to evaluate the performance of low-light enhancement models in terms of their performance on a high-level vision task, we need to integrate the model into a pipeline for the relevant high-level vision task. We first enhance the input image by propagating it through a low-light enhancement model, and then we feed the resulting enhancement image to the relevant task model. In this work, we evaluate the effect of image enhancement models on object detection with various YOLOv5 models. An alternative approach is to fuse the enhancement and detection models, utilizing features extracted by a low-light enhancement module as input to the backbone of the task model, as explored successfully by Sasagawa et al. [22].

To evaluate the models, we use a labelled dataset (ExDark [17]), which consists of low-light images, to serve as a benchmark for the relevant tasks. For purposes of comparison, we first establish a baseline of how a standard object detection model, pre-trained on MS-COCO, performs on the EXDark dataset. Then we examine the efficacy of fine-tuning the YOLOv5 model on the EXDark training set. Alternatively, we utilize low-light enhancement models to pre-process the images used for fine-tuning the YOLOv5 model. Finally, we compare the various methods' test set performance.

The low-light enhancement models to be used lie in two categories: supervised and unsupervised. As discussed earlier, supervised models require labelled training data. Most low-light image enhancement models use the LOL dataset for training and testing. LOL is mostly composed of paired indoor images acquired under low and normal light conditions. On the other hand, the images in the ExDARK dataset are a mix of unpaired indoor and outdoor images, so they are out-of-domain relative to the LOL dataset.

Utilizing a supervised model trained on a dataset such as LOL presents us with the possibility that the model will not generalize to our target dataset. On the other hand, models trained using the unsupervised approach generally do not perform as well on commonly used evaluation metrics (PSNR, SSIM), as shown in Table 1, but the same cannot be said for downstream tasks such as detection without empirical evidence. Given this, we perform a comparison of various supervised and unsupervised models in terms of their efficacy for the object detection task on the low-light images in the ExDARK dataset. We use a common evaluation metric for object detection (mAP_{50}) in order to gauge each model's performance.

Besides the modular approaches just described, we also introduce a new method for building robust low-light detection models called EnlightenGAN-JT. In this approach, the enhancement module is trained jointly with a YOLOv5 model on ExDark training images of size 416 × 416. The self-preservation loss implemented using a VGG16 model in the original EnlightenGAN (used to encourage fidelity of the reconstructed image to the original) is removed, and the YOLOv5 loss is additionally backpropagated through its input pixels then through the EnlightenGAN model. The EnlightenGAN-JT model that achieves the highest accuracy on the validation set is saved and used to pre-process the entire ExDark dataset. This pre-processed dataset is then used to train the YOLOv5 model from scratch.

The EnlightenGAN-JT model is produced in several steps by first pre-training YOLOv5 on MS-COCO then fine-tuning it on the EnlightenGAN-enhanced dataset. Next, EnlightenGAN is trained jointly with this YOLOv5 model by backpropagating the YOLO loss through the input pixels then through the EnlightenGAN model. For joint training, the enhancement and detection modules are joined using three convolutional layers with a skip connection. The EnlightenGAN model is then frozen during re-training of YOLOv5 from scratch on the enhanced images.

4 Experimental Setup

Below we provide a list of commonly used terms and define their use in the rest of this paper.

Low Light Refers to the original ExDark dataset and any YOLO model trained on this dataset.

Zero-DCE Deep learning model that learns a pixel-wise mapping to brighten the input image. May refer to the image enhancement model itself, the ExDark dataset enhanced by the Zero-DCE model, or any YOLO model trained on this enhanced dataset.

EnlightenGAN Generative model based on a UNet-like architecture for low-light image enhancement. May refer to the image enhancement model itself, the ExDark dataset enhanced by the EnlightenGAN model, or any YOLO model trained on this enhanced dataset.

EnlightenGAN-JT (ours) Our method for robust low-light object detection. May refer to the EnlightenGAN model jointly trained with the YOLOv5 model, the ExDark dataset enhanced by this model, or any YOLO model trained on this enhanced dataset.

For evaluation of low-light image enhancement using YOLOv5s models, the small version, we utilize the default hyperparameters and fine-tune the model for 150 epochs. The training is performed using a single NVIDIA RTX3060 GPU and takes approximately three hours. For the training of YOLOv5x, the largest version, we begin with MS-COCO pre-trained parameters, then we perform a hyperparameter search and fine-tune for 5 epochs. For purposes of training, the ExDark dataset is split as per Table 2. All models are trained on images of size 416 × 416.

5 Results

We present the results of four experiments in this section. The first experiment explores the use of pre-trained YOLOv5 models. The second and third experiments explore the use of fine-tuned YOLOv5s and YOLOv5x models. The fourth experiment explores the robustness of the features learned by the fine-tuned YOLOv5x model. Detailed results with discussion for each experiment are given in the following sections.

5.1 Experiment 1: Pre-trained YOLOv5 Models on ExDark

This section presents an evaluation of various pre-trained YOLOv5 models on the ExDark test set. The YOLOv5 models are pre-trained on the MS-COCO dataset then evaluated on two versions of the ExDark dataset. The first is the original dataset (Experiment 1.1), and the second dataset is obtained by preprocessing using the EnlightenGAN model (Experiment 1.2).

Table 2. Training, validation and test split of the images in the ExDark dataset.

Dataset	Training	Validation	Test	Total
ExDark	3000	1800	2563	7363

The ExDark dataset consists of 12 object classes. The train-val-test split of the dataset is given in Table 2. The images in the dataset are classified into ten categories based on their illumination profile (Fig. 1).

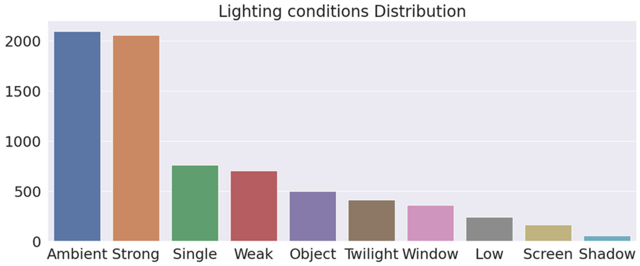


Fig. 1. Distribution of illumination profiles in the ExDark dataset.

The MS-COCO dataset comprises an 80 object category superset of the ExDark object categories. In order to evaluate a pre-trained YOLOv5 model on the ExDark dataset, we first need to re-label the ExDark test dataset using the class IDs from the MS-COCO dataset. In order to do that, we take the object class IDs used to label the objects in the MS-COCO dataset and annotate the corresponding objects from the ExDark test set using these IDs. This allows us to evaluate a model trained on the MS-COCO on the ExDark dataset.

Table 3 shows the results of Experiment 1.1, the evaluation of pre-trained YOLOv5 models on the original ExDark test set.

Table 3. Experiment 1.1 results. Evaluation of pre-trained YOLOv5 models on the original ExDark test set.

Model	YOLOv5s	YOLOv5l	YOLOv5x
mAP_{50}	58.1	68.6	**69.6**

Table 4 shows the results of Experiment 1.2, the evaluation of pre-trained YOLOv5 models on the enhanced ExDark test set. The ExDark dataset, in this case, has been enhanced using the pre-trained EnlightenGAN model prior to its use for the evaluation of the YOLOv5 models.

Table 4. Experiment 1.2 results. Evaluation of pre-trained YOLOv5 models on ExDark dataset enhanced using EnlightenGAN.

Model	YOLOv5s	YOLOv5l	YOLOv5x
mAP_{50}	57.2	67.6	**68.7**

The results show that, contrary to intuition, the pre-trained models perform slightly worse on the dataset pre-processed using EnlightenGAN than on the original images. For all three models, we can see a similar drop in accuracy for detection after pre-processing of the images. Observing this pattern, we can assume that despite enhancing the illumination in the images, the propagation of images through EnlightenGAN introduces a slight domain shift as it changes the distribution of pixel intensities in the images. Consequently, we see that the pre-trained YOLOv5 models have a drop of about 0.9% on the pre-processed dataset compared to the original. This result counters the notion that image enhancement is necessarily beneficial; a domain shift can disrupt downstream tasks.

5.2 Experiment 2: Evaluation of Fine-Tuned YOLOv5s

From Experiment 1, the baseline mAP_{50} for low light object detection using the pretrained YOLOv5 models on the ExDark dataset is 58.1 and 69.6 for YOLOv5s and YOLOv5x, respectively. Naively, we might assume that enhancing the input images using a low-light enhancement module, such as EnlightenGAN, will result in an improved mean average precision score. But, as observed in the results of Experiment 1.2, this seems to introduce a domain shift that reduces the resulting accuracy.

As followup, in Experiment 2, we fine-tune the YOLOv5s model on the ExDark dataset beginning with the default hyperparameters for the MS-COCO dataset. The dataset is pre-processed using various image enhancement models, and the performance on the original dataset is used as a baseline for comparison.

Results are shown in Table 6. We see that processing of images using an image enhancement model results in improved mean average precision, so long as the detector is fine-tuned on the enhanced image domain. The best image enhancement model in these settings turns out to be Zero-DCE, which is trained in a self-supervised manner. We obtain a substantial improvement of 1.6% over the baseline (i.e., without any image enhancement). The second-best image enhancement model, in terms of accuracy, is MBLLEN, which is trained using a paired image dataset (LOL) in a supervised manner. We can conclude that there is no significant advantage in choosing a model trained under a self-supervised regime over one which is not. However, in terms of inference speed, the MBLLEN model is not as fast as Zero-DCE or EnlightenGAN. Additionally, the self-supervised training of EnlightenGAN and Zero-DCE frees us from the need for a domain-specific paired image dataset; this enables deployment in more use cases.

5.3 Experiment 3: Evaluation of Fine-Tuned YOLOv5x

In addition to the small YOLOv5s model, we also train the more complex YOLOv5x model on the enhanced datasets in order to observe the effect of increased capacity on accuracy. Figure 2 shows examples of images processed by the three different image enhancement models along with the corresponding original image from the ExDark dataset. From the images, it may be observed that the Zero-DCE and EnlightenGAN-JT images are more color-consistent with the original image than the EnlightenGAN image. Furthermore, jointly training EnlightenGAN with a YOLOv5 model (EnlightenGAN-JT) results in a model

Table 5. Experiment 3 results. Evaluation results for YOLOv5x models fine-tuned on the ExDark dataset.

Model	mAP_{50}	$mAP_{50:95}$
Low Light	78.1	45.1
EGAN [12]	78.1	48.1
SCI [20]	78.9	45.8
Zero-DCE [10]	79.4	**49.7**
EGAN-JT (ours)	**79.5**	47.3
EGAN-JT (YOLOv5s)	67.8	38.9

that outputs images with illumination levels considerably lower than those of the original EnlightenGAN output. We infer that during joint training, the EnlightenGAN-JT model only adjusts illumination levels to the point required to minimize YOLOv5 losses and no further.

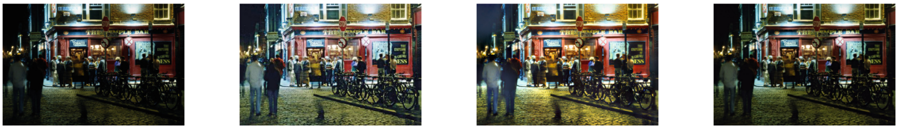

(a) (b) (c) (d)

Fig. 2. Comparison of selected Image enhancement models. a) Original image b) Zero-DCE c) EnlightenGAN and d) EnlightenGAN-JT (ours)

Though the YOLOv5x models were selected on the criteria of best mAP_{50}, results including mAP_{50} and $mAP_{50:95}$ are presented in Table 5. YOLOv5x trained on the EnlightenGAN-JT dataset has the highest mAP_{50} value, with the Zero-DCE dataset being the second best. However, YOLOv5x trained on Zero-DCE enhanced data gives the best $mAP_{50:95}$ accuracy (an increase of 4.6% over baseline). In the case of mAP_{50}, we obtain an increase of 1.4% over the use of the original (low light) dataset by utilizing EnlightenGAN-JT enhanced data.

5.4 Experiment 4: Evaluation of Fine-Tuned YOLOv5x Models Under Simulated Lighting Conditions

Loh et al. [17] conclude that the objects in images under low-light conditions present significantly different features to a deep learning model than those captured in well-lit images. The authors show a clear demarcation in feature space between learned representations from low-light and normal light images. Given this, the question arises as to how robust are the features learnt by YOLOv5 models trained on these processed datasets and how well they generalize to different lighting conditions.

Table 6. Experiment 2 results. Evaluation of fine-tuned YOLOv5s model on enhanced ExDark dataset. Reported numbers are mAP_{50}.

Class	Original	EnlightenGAN	Zero-DCE	MBLLEN	KinD	IAT
Person	66.4	65.8	67.1	**67.2**	66.0	67.0
Bicycle	73.3	73.9	**76.1**	74.9	73.9	73.8
Car	67.9	69.5	**70.6**	69.9	69.4	69.3
Motorcycle	50.7	**53.8**	49.6	51.7	49.4	51.7
Bus	80.3	81.5	82.7	**85.1**	82.2	79.5
Boat	63.6	64.4	**65.1**	63.6	64.9	**65.1**
Cat	56.7	55.6	**57.4**	54.9	55.8	57.1
Dog	67.8	**70.1**	68.9	67.2	65.0	69.3
Bottle	59.2	**63.7**	62.4	62.7	57.1	58.8
Cup	53.6	53.8	56.0	**56.8**	49.9	54.4
Chair	54.6	53.9	54.8	**56.8**	52.1	54.1
Table	34.0	35.9	**37.4**	36.3	34.2	35.6
All	60.7	61.8	**62.3**	62.2	60.0	61.3

In order to answer this question, we test all the YOLOv5x models on the original ExDark dataset without applying any image enhancement (Experiment 4.1). For further analysis, the evaluation results of all the models are broken down with respect to the illumination profiles of the images. The results are presented in Table 7. The EnlightenGAN-JT model is clearly able to generalize best in this scenario. We can deduce that the features learned by the YOLOv5 model from the EnlightenGAN-JT dataset are generalizable even if we do not apply the low-light image enhancement pre-processing step.

In Experiment 4.2 (Table 8) we further examine the results of applying the EnlightenGAN-JT model to the original (low light) dataset and the model enhanced dataset. The evaluation on the enhanced dataset shows that the proposed model performs better, except for the "Screen" and "Shadow" lighting conditions.

We further test the robustness of the models by simulating different illumination conditions using gamma correction on the test dataset prior to detection (Experiment 4.3). The results are featured in Fig. 3. The results show that for a range of gamma values, the EnlightenGAN-JT model tends to generalize better to both darker and brighter images. The YOLOv5 model trained on the original dataset (low light) is more accurate only on the very dark images.

Table 7. Experiment 4.1 results. YOLOv5x models trained on un-enhanced (low-light)/enhanced (others) ExDark train set and evaluated on un-enhanced ExDark test set.

Lighting	Low-light	EGAN-JT	ZDCE	SCI	EGAN
Strong	73.2	**74.1**	71.0	70.6	70.8
Ambient	**81.3**	81.0	78.8	80.1	79.8
Single	**84.8**	82.3	79.7	79.6	81.4
Weak	70.9	**73.2**	69.5	68.3	67.4
Object	72.8	75.7	76.2	**78.2**	74.4
Twilight	**85.8**	81.2	80.7	84.5	79.8
Window	**79.8**	79.3	77.9	68.3	**79.8**
Low	72.3	**75.3**	71.1	73.3	67.5
Screen	87.7	88.8	**89.4**	88.2	89.0
Shadow	75.9	**78.0**	68.0	77.0	75.4
All	78.1	**78.5**	75.8	76.3	76.0

Table 8. Experiment 4.2 results. Evaluation of YOLOv5x model trained on ExDark train set enhanced by EGAN-JT and tested on ExDark test set with and without EGAN-JT enhancement.

Lighting	w/o enhancement	with enhancement
Strong	74.1	**75.4**
Ambient	81.0	**81.5**
Single	82.3	**84.0**
Weak	73.2	**75.3**
Object	75.7	**77.7**
Twilight	81.2	**81.4**
Window	79.3	**80.1**
Low	75.3	**76.9**
Screen	**88.8**	87.6
Shadow	**78.0**	70.0
All	78.5	**79.5**

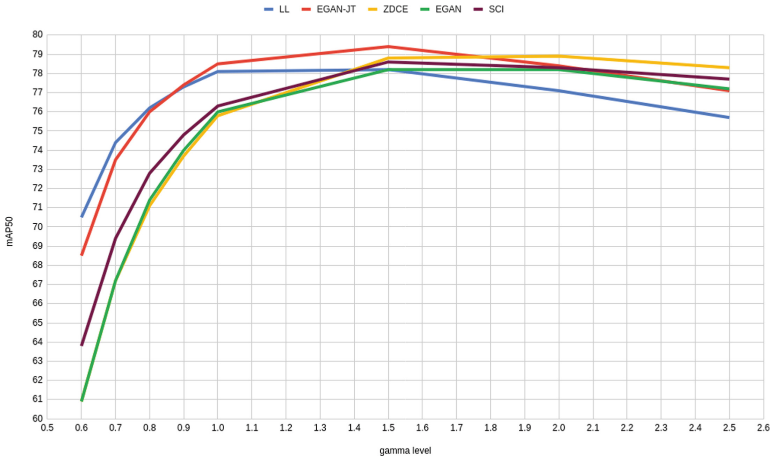


Fig. 3. Experiment 4.3 results. Evaluation after gamma correction of images

6 Conclusion

We show that appropriate use of image enhancement models to pre-process images can improve the accuracy of an object detection model substantially. We compare different low-light enhancement models to evaluate their efficacy. Our experiments show that Zero-DCE and EnlightenGAN-JT (ours) produce the best results, with a 1.4% improvement in mAP_{50} and a 4.6% improvement in $\mathrm{mAP}_{50:95}$.

We further conduct experiments to test the applicability of our trained models over a range of input pixel intensity distributions. We simulate various lighting conditions by applying gamma correction to the images in the ExDark test set prior to detection. We evaluate our trained models on the simulated test set. We conclude that the EnlightenGAN-JT (our) model produces better results over a range of lighting conditions (gamma values) than the other models.

We observe that the commonly used metrics for image enhancement models do not necessarily predict their utility for high-level vision tasks. Without the results presented here, we might assume that PSNR would be a useful metric for image enhancement, as it provides a measure of the signal clarity in the output image. Similarly, SSIM provides a seemingly useful measure of the preservation of structural information in the output image. However, when measured in terms of PSNR and SSIM, the Zero-DCE model performs worse than MBLLEN, KinD and EnlightenGAN. But when Zero-DCE is utilized in conjunction with a high-level vision task (object detection), it produces the best results of all four models.

When we trained the EnlightenGAN model jointly with a YOLOv5 model, we obtained an improvement over the performance of the pre-trained model. To build the hybrid model, we omitted the self-preservation loss [12] from the EnlightenGAN and instead backpropagated the YOLO loss through the EnlightenGAN model during training. Using the resulting EnlightenGAN model produces images that, as we have demonstrated, are more color-consistent with the original images, have comparatively lower illumination levels (than original EnlightenGAN), and have less noise, especially in the dark regions of the image.

References

1. Al Sobbahi, R., Tekli, J.: Comparing deep learning models for low-light natural scene image enhancement and their impact on object detection and classification: overview, empirical evaluation, and challenges. Sig. Process. Image Commun. **109**, 116848 (2022). https://doi.org/10.1016/j.image.2022.116848, https://www.sciencedirect.com/science/article/pii/S092359652200131X
2. Bychkovsky, V., Paris, S., Chan, E., Durand, F.: Learning photographic global tonal adjustment with a database of input/output image pairs. In: The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition (2011)
3. Chen, C., Chen, Q., Xu, J., Koltun, V.: Learning to see in the dark. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3291–3300 (2018). https://doi.org/10.1109/CVPR.2018.00347
4. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. ArXiv abs/1706.05587 (2017)
5. Wei, C., Wang, W., Yang, W., Liu, J.: Deep retinex decomposition for low-light enhancement. In: British Machine Vision Conference (2018)
6. Cui, Z., et al.: You only need 90k parameters to adapt light: a light weight transformer for image enhancement and exposure correction (2022). https://doi.org/10.48550/ARXIV.2205.14871, https://arxiv.org/abs/2205.14871
7. Cui, Z., Qi, G.J., Gu, L., You, S., Zhang, Z., Harada, T.: Multitask AET with orthogonal tangent regularity for dark object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 2553–2562, October 2021
8. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes (VOC) challenge, vol. 88, pp. 303–338 (2010)
9. Forbes: Can tesla really do without radar for full self-driving? In: Forbes, July 2021. https://www.forbes.com/sites/jamesmorris/2021/07/03/can-tesla-really-do-without-radar-for-full-self-driving/?sh=38b2e9d77007
10. Guo, C.G., et al.: Zero-reference deep curve estimation for low-light image enhancement. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1780–1789, June 2020
11. Guo, H., Lu, T., Wu, Y.: Dynamic low-light image enhancement for object detection via end-to-end training. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 5611–5618 (2021). https://doi.org/10.1109/ICPR48806.2021.9412802
12. Jiang, Y., et al.: Enlightengan: deep light enhancement without paired supervision. IEEE Trans. Image Process. **30**, 2340–2349 (2021)
13. Jocher, G., et al.: ultralytics/yolov5: v7.0 - YOLOv5 SOTA realtime instance segmentation, November 2022. https://doi.org/10.5281/zenodo.7347926
14. Li, C., et al.: Low-light image and video enhancement using deep learning: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **44**(12), 9396–9416 (2021)
15. Li, C., Guo, J., Porikli, F., Pang, Y.: Lightennet: a convolutional neural network for weakly illuminated image enhancement. Pattern Recogn. Lett. **104**, 15–22 (2018). https://doi.org/10.1016/j.patrec.2018.01.010, https://www.sciencedirect.com/science/article/pii/S0167865518300163

16. Lin, T.Y., et al.: Microsoft coco: common objects in context. In: European Conference on Computer Vision (2014)
17. Loh, Y.P., Chan, C.S.: Getting to know low-light images with the exclusively dark dataset. Comput. Vis. Image Underst. **178**, 30–42 (2019). https://doi.org/10.1016/j.cviu.2018.10.010
18. Lore, K.G., Akintayo, A., Sarkar, S.: LLNet: a deep autoencoder approach to natural low-light image enhancement. Pattern Recogn. **61**, 650–662 (2017)
19. Lv, F., Lu, F., Wu, J., Lim, C.S.: MBLLEN: low-light image/video enhancement using CNNs. In: BMVC (2018)
20. Ma, L., Ma, T., Liu, R., Fan, X., Luo, Z.: Toward fast, flexible, and robust low-light image enhancement. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5627–5636 (2022)
21. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**, 1137–1149 (2015)
22. Sasagawa, Y., Nagahara, H.: Yolo in the dark - domain adaptation method for merging multiple models. In: ECCV (2020)
23. Tu, Z., et al.: Maxim: Multi-axis MLP for image processing. In: CVPR (2022)
24. Zhang, Y., Zhang, J., Guo, X.: Kindling the darkness: a practical low-light image enhancer. In: Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, pp. 1632–1640. ACM, New York, NY, USA (2019). https://doi.org/10.1145/3343031.3350926

Unsupervised Segmentation of High-Throughput Zebrafish Images Using Deep Neural Networks and Transformers

Shima Javanmardi[1](✉), Xiaoqin Tang[2], Mehrdad Jahanbanifard[1], and Fons. J. Verbeek[1]

[1] Leiden Institute of Advanced Computer Science (LIACS), Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
s.javanmardi@liacs.leidenuniv.nl
[2] School of Computer and Information Science, Southwest University, Chongqing 400715, China

Abstract. Zebrafish is a popular model system for biomedical analysis, especially for compound screening in drug research. In this paper, we present a comprehensive investigation aimed at enhancing the processing pipeline for segmenting zebrafish larvae images. The emphasis is on the application of an unsupervised segmentation method for segmenting zebrafish in Optical Projection Tomography (OPT) images. We propose a novel pipeline that integrates the Transformer and U-Net, a convolutional neural network for biomedical image segmentation, to achieve accurate segmentation of zebrafish larvae images. This accuracy is critical for precise 3D reconstruction. Leveraging transfer learning, we broaden the capabilities of our trained model to segment OPT images.

This approach is intended to enhance the robustness and versatility of our pipeline, allowing it to cater to a broad range of imaging modalities beyond traditional microscopic images. The developed processing pipeline is then used for 3D reconstruction of the segmented areas, demonstrating its potential for advanced biomedical analysis. Our findings confirm the efficiency and accuracy of the proposed pipeline providing robust tools for future Zebrafish-based research, particularly in the domains of drug screening and cancer treatment.

Keywords: Machine Learning · Deep Learning · Image Segmentation · Transformer · Transfer Learning · 3D Reconstruction

1 Introduction

Zebrafish has proven to be an essential model system in biomedical research, due to its genomic similarity to humans and amenability for studying various biological phenomena, including developmental biology, toxicology, cancer, infectious diseases, and drug targeting [1, 2]. Historically, the majority of related studies have relied on 2D imaging and analysis. However, the shift towards comprehensive 3D image analysis is paramount for garnering profound insights and precise shape measurements, particularly in assessing overall size and form. Confocal Laser Scanning Microscopy (CLSM) is now a widely used imaging modality, providing superior spatial resolution.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 213–227, 2023.
https://doi.org/10.1007/978-981-99-7969-1_16

Our research responds to the need for meticulous 3D reconstruction from zebrafish images. Our approach involves creating a robust pipeline using Transformer and U-Net models for the segmentation of zebrafish in Optical Projection Tomography (OPT) images. The application of an unsupervised segmentation method is pivotal for acquiring precise 2D masks that form the foundation of subsequent 3D reconstructions.

The integrity of these 3D reconstructions is dictated by the quality of segmentation, highlighting the importance of our proposed pipeline. The segmentation process yields a sequence of shapes used in the 3D reconstruction process. Accurate segmentation is imperative, as the derived masks must represent the subject thoroughly for a successful reconstruction. The capabilities of this pipeline extend beyond the initial segmentation process. Using transfer learning, we employ the trained model to segment OPT images, further enhancing the versatility of our pipeline. This leads to 3D reconstructions from the segmented OPT images, which provide powerful tools for examining various biological phenomena.

In conclusion, our advanced processing pipeline offers a valuable tool for 3D image analysis in zebrafish research. This capability to precisely segment, reconstruct, and analyze 3D images has significant implications in numerous biological research areas, from toxicology to drug discovery. Our integrated approach not only improves our understanding of zebrafish biology but also lays a solid foundation for future advances in this field.

2 Related Work

Zebrafish, recognized by their distinctive horizontal stripes along their body sides [3], have gained significant importance as a model organism in various biological research fields. This species offers valuable insights into vertebrate development, genomics, physiology, behavior, toxicology, and disease studies [4]. Zebrafish are commonly bred in captivity due to their manageable care requirements and rapid breeding cycles, enabling researchers to acquire large quantities of zebrafish within a short timeframe. Moreover, the genomic similarities between zebrafish and humans [5], allow us to leverage zebrafish research findings to gain a better understanding of biological phenomena in humans. Zebrafish possesses a well-characterized genome assembly and gene annotation, and advancements in genetic engineering have facilitated the creation of various transgenic zebrafish lines [6]. For instance, fluorescent reporter genes can be introduced into genes of interest, enabling visualization of gene expression patterns within the organism. The transparent nature of zebrafish during early life stages makes it suitable for non-invasive monitoring of expression patterns resulting from the incorporation of these fluorescent reporter genes.

The utilization of zebrafish as disease models is crucial since these models faithfully mimic human diseases, encompassing the origin, progression, and resolution processes [7, 8]. By precisely editing genes in zebrafish, it is possible to create models harboring disease-causing alleles found in humans, thus capturing some of the disease's causes observed in patients [9, 10]. Additionally, genetic and transgenic models often exhibit the same molecular pathway alterations that underlie human diseases, such as cancer cell states and transcriptional signatures [7, 8]. Notably, zebrafish, being vertebrates, possess similar tissues and developmental biology as humans. Consequently, zebrafish models have been established for a broad range of conditions, including various cancers [11], liver disease [12], blood disorders [13], heart disease [14], and behavioral disorders [15, 16].

The next section of this paper delves into the material and methods employed to conduct the study. It begins by introducing the dataset utilized for analysis, followed by a comprehensive explanation of the various models employed in the research.

3 Material and Methods

3.1 The VAST BioImager

In order to extract the necessary information for our research, we employed specific techniques to analyze zebrafish. The Vertebrate Automated Screening Technology (VAST) BioImager system was utilized as a solution for efficiently screening large quantities of zebrafish larvae in an automated manner [6]. This imaging system incorporates a capillary-based setup that enables the loading and secure positioning of the larvae for imaging purposes. By utilizing a stepper motor, the capillary holding the specimen can be rotated, allowing for the acquisition of a series of axial-view images of the zebrafish larva at user-defined step sizes.

The VAST BioImager is compatible with both bright-field microscopes and fluorescence-capable microscope models. It is equipped with its own integrated camera, which serves two crucial functions: controlling the position of the larva within the capillary to ensure visibility within the microscope camera's field of view and capturing overall views of the vertebrate specimen. However, it is important to note that the overall views obtained from the VAST camera are at a lower resolution compared to the microscope camera. Therefore, the microscope camera is exclusively responsible for capturing organ-level views of the zebrafish larvae.

3.2 Dataset

Our dataset has been generated in 2018 using the MM-HTAI architecture and internal VAST camera. This dataset consists of a total of 123 zebrafish larvae, comprising 39 specimens at 3 days post-fertilization (dpf), 36 specimens at 4 dpf, and 48 specimens at 5 dpf. During the imaging process, most of the zebrafish larvae were captured 100 times from various angles to cover a full rotation. However, there were a few instances where three specimens were positioned incorrectly by the VAST BioImager, resulting in imaging failures for certain angles.

All images in this dataset were saved as TIFF files with a resolution of 1024 pixels by 250 pixels. To provide a visual representation, Fig. 1 showcases four bright field images of a typical 4 dpf zebrafish larva from the dataset. To effectively utilize the dataset, we utilized a total of 18,375 samples derived from the 123 zebrafish specimens. To ensure a robust training and evaluation process, we divided the dataset into two subsets: an 80% training set and a 20% validation set.

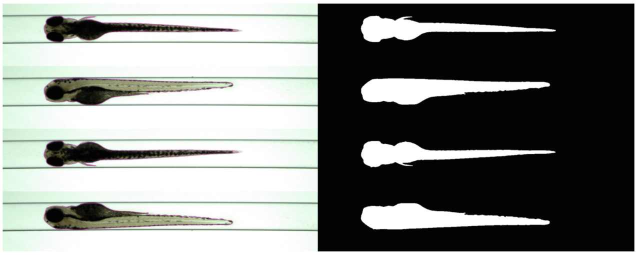


Fig. 1. These images depict different views, including ventral, left lateral, dorsal, and right lateral orientations. Additionally, the corresponding low-throughput segmentations are presented, where the purple line represents the boundary of the segmentation.

3.3 Preprocessing

In the preprocessing phase, we implemented a sequence of steps to enhance the quality of the images. These steps involved utilizing various image processing techniques as described below:

Initially, we applied a median filter with a filter size of 3 to each image, aiming to reduce noise and improve the overall image quality. Subsequently, a Gaussian filter with a sigma value of 2 was applied to further smooth the denoised image.

For segmentation purposes, we employed the Otsu thresholding method to create a binary image. This algorithm automatically determined an optimal threshold value to distinguish the foreground from the background. Pixels with intensities higher than the threshold were assigned a value of 1 (white), while those below were assigned a value of 0 (black), resulting in a binary representation of the image.

To refine the segmented structures and eliminate small artifacts, we performed morphological opening using a disk-shaped structuring element with a size of 3. This process helped enhance the definition of relevant structures, while reducing noise and unwanted details.

Finally, we combined the denoised image with the binary mask obtained from morphological opening. This was achieved by element-wise multiplication of the denoised image with the processed binary mask. The objective of this step was to preserve the relevant structures while suppressing noise and irrelevant elements, thereby improving the overall quality of the preprocessed images. Through the application of these preprocessing steps, our aim was to enhance the images within the dataset and optimize subsequent training and validation processes. These procedures effectively reduced noise, improved segmentation accuracy, and ensured the reliability and effectiveness of the data used in our experiments.

3.4 Segmentation Models

U-Net Models

The U-Net model used in our study for image segmentation consists of five "floors" and a total of 7,774,049 parameters. The model architecture, represented by the provided code snippet, follows a series of convolutional layers, batch normalization, and activation functions [17].

This U-Net model architecture provides a powerful framework for segmenting zebrafish larvae images. Figure 2 shows an illustration of this network.

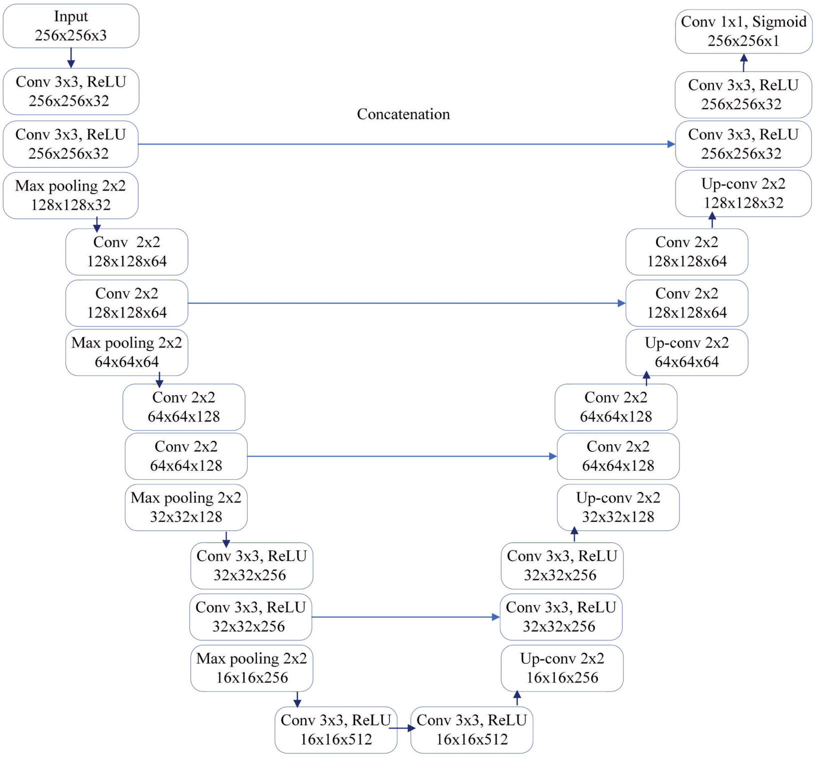


Fig. 2. U-Net architecture used for segmentation.

The U-Net architecture begins with the input layer, which is followed by a series of blocks that progressively downsample the input image while increasing the number of filters. Each block consists of two convolutional layers with 3 × 3 filter sizes, followed by batch normalization and ReLU activation functions.

To perform upsampling and refine the segmentation, a series of transpose convolutional layers, also known as deconvolutional layers, are used. These layers gradually upsample the feature maps and concatenate them with the corresponding block outputs from the downsampling path. Each upsampling block consists of a transpose convolutional layer followed by batch normalization and ReLU activation. This process is repeated four times, progressively restoring the resolution of the feature maps.

Finally, a 3×3 convolutional layer with a sigmoid activation function is applied to generate the segmentation output. The output represents a binary mask, with intensities indicating the likelihood of each pixel belonging to the segmented object.

The model is compiled using the Adam optimizer with a learning rate of 0.005 and a binary cross-entropy loss function. Additionally, metrics such as mean squared error (MSE) and accuracy are computed during the training process to monitor model performance.

The second model used in our study for image segmentation is a 4-layer U-Net architecture. This model incorporates a deeper structure, particularly in the lower levels, featuring a triple stack of Conv2D layers. This design allows for more elaborate processing at a higher semantic level, enabling better representation of complex patterns and structures.

In terms of model size, the 4-layer U-Net consists of a total of 10,655,809 trainable parameters. Due to its larger capacity, a higher number of epochs were implemented during training to effectively capture the intricate details of the data. Although overfitting may occur with such a model, precautions will be taken in future iterations by implementing appropriate callbacks. Interestingly, even after 100 training epochs, the phenomenon of overfitting did not significantly impact the model's performance.

During training, the learning process reached a plateau after approximately 35 epochs on the validation set. This suggests that the model achieved a stable level of performance and further training did not yield substantial improvements. The validation set serves as an essential benchmark for evaluating the model's generalization capabilities.

The incorporation of this 4-layer U-Net model enhances our segmentation pipeline by allowing more advanced and nuanced processing of zebrafish larvae images. The increased depth of the architecture and the extensive training contribute to improved segmentation accuracy and the ability to capture fine-grained details necessary for robust image analysis.

TransUnet Model

The third model in our study was inspired by the work of J. Cheng et al., [18]. This model architecture incorporates a 4-layer U-Net with Transformer blocks added at the bottom, which apply attention mechanisms to the feature maps generated by the Convolutional Layers. The concept of combining the spatial information of Convolutional Neural Networks (CNN) with the attention and global processing capabilities of Transformer architectures was initially proposed in the original Vision Transformer paper. Figure 3 shows the architecture of this model. We have selected this design because it offers two advantages: firstly, it enables us to leverage intermediate high-resolution CNN feature maps during the decoding phase, and secondly, our findings demonstrate that the hybrid CNN-Transformer encoder delivers superior performance compared to using a pure Transformer as the encoder [18].

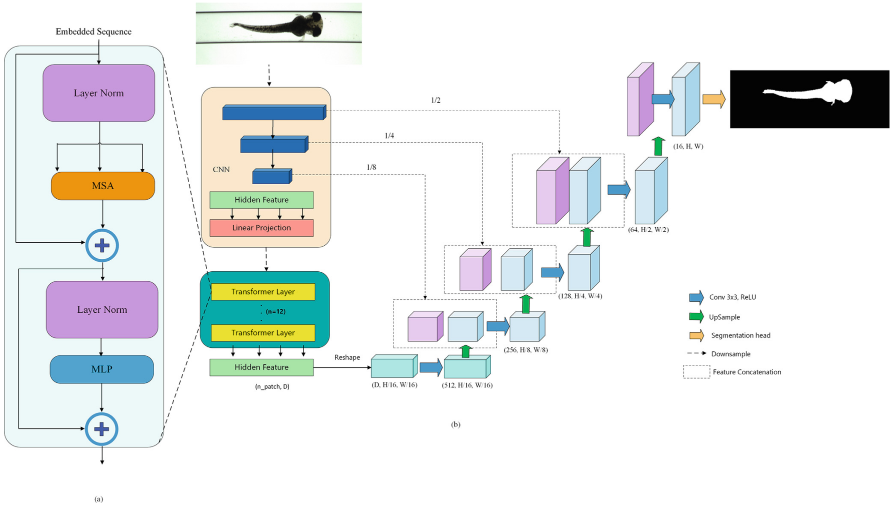


Fig. 3. Illustrates an overview of the framework (TransUnet) used in this study. In (a), a sche-matic of the Transformer layer is presented, showcasing the key components and their interac-tions. (b) showcases the architecture of the proposed TransUNet model, highlighting the major components and their connections.

With an input image $X \epsilon \mathbb{R}^{H \times W \times C}$, where H represents the height, W represents the width, and C represents the number of channels, our objective is to generate a pixel-wise label map of the same spatial resolution $H \times W$. To avoid unnecessary complexity and ensure a clear presentation, we have chosen to omit detailed explanations of all the layers and intricate network details. Instead, we provide a high-level overview of the framework to facilitate understanding and focus on the key aspects that are relevant to our research objectives.

It is important to note that this model is significantly different in terms of the number of parameters compared to the previous U-Net models. In fact, it is approximately 10 times larger than the largest U-Net model and it has 98,477,889 parameters. The authors of the paper did not train the network from scratch; instead, they pre-trained it on the ImageNet dataset. While I may explore pre-training the model on ImageNet in the future, at this stage, my focus is on building and evaluating the network to assess its performance.

The integration of Transformer blocks within the U-Net architecture offers a promising approach for medical image segmentation. By combining the strengths of CNNs and Transformers, this model has the potential to capture both local spatial features and long-range dependencies, leading to improved segmentation accuracy.

Although the larger number of parameters poses a computational challenge, evaluating the performance and exploring the capabilities of this model will provide valuable insights for further research and potential improvements in the future. In this architecture, the decoder performs upsampling on the encoded features, and subsequently, these upsampled features are fused with the high-resolution CNN feature maps to facilitate accurate localization.

4 Results

In this section, we present the results of our segmentation models for zebrafish larvae images. We evaluate the performance of various segmentation methods, including U-Net models with different architectures, as well as the TransUNet model. The results provide valuable insights into the segmentation quality, achieved by our models, further validating the efficacy of our approach.

4.1 Model 1

In the results section, the performance of the first model, a 5-layer U-Net, was evaluated using various metrics, including Mean Squared Error (MSE), Accuracy, Precision, Recall, F1 Score, and Training Time. The results are summarized in Table 1 and visualized in Fig. 4(a)- 4(f), where each figure corresponds to one metric.

Mean Squared Error (MSE) measures the average squared difference between the predicted and actual segmentation masks. For this model, the MSE value achieved was 3.7254e-04, indicating a low overall error rate in the segmentation predictions.

Accuracy represents the proportion of correctly classified pixels in the segmentation masks. The model achieved an accuracy of 0.9995, indicating a high level of accuracy in the segmentation results.

Precision measures the proportion of true positive predictions among all positive predictions. In this case, the precision obtained was 0.9983, suggesting that a high percentage of the predicted positive pixels were indeed correct.

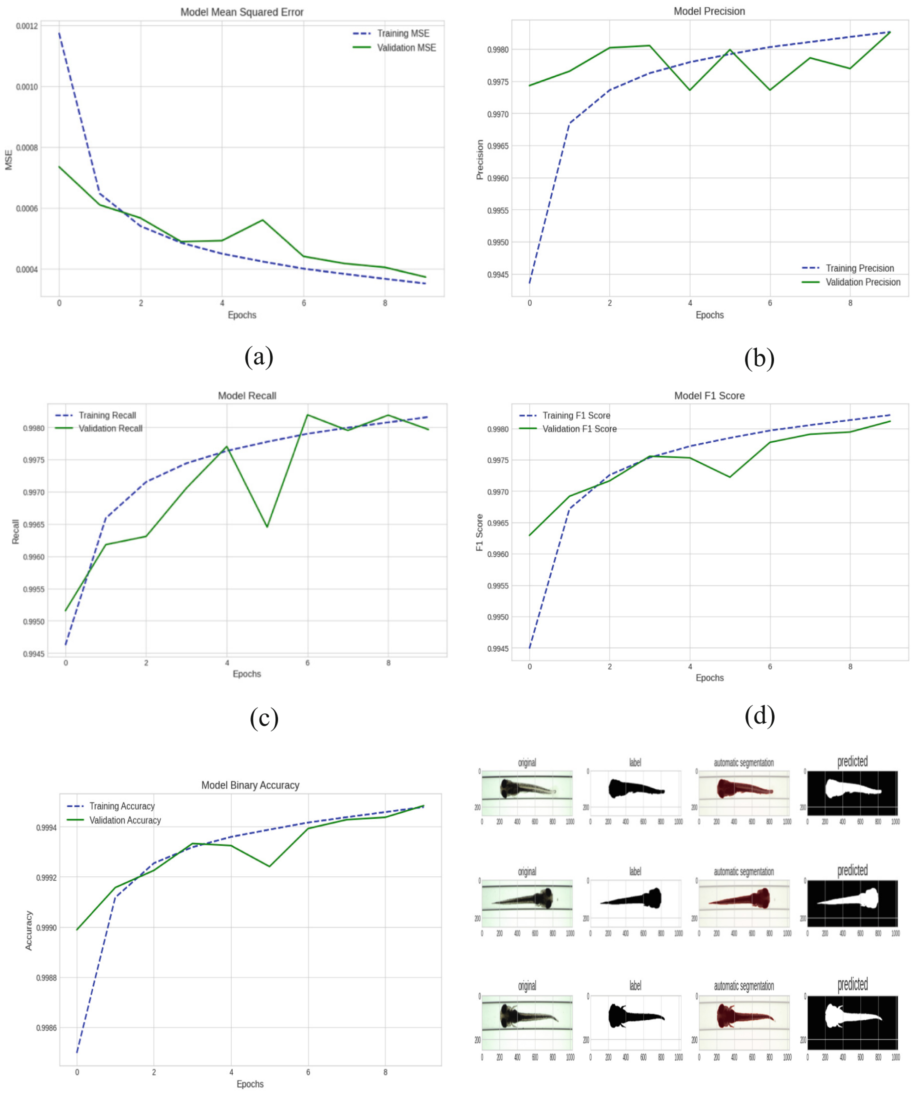
Recall, also known as sensitivity or true positive rate, quantifies the proportion of actual positive pixels correctly identified by the model. The 5-layer U-Net achieved a recall score of 0.9980, indicating a high ability to capture the true positive pixels in the segmentation process.

The F1 Score is the harmonic mean of precision and recall, providing an overall measure of the model's performance. The F1 Score achieved by the model was 0.9981, reflecting a well-balanced performance in terms of precision and recall.

Training Time indicates the time taken by the model to complete the training process. In this case, the model required approximately 29.21 min to train on the given dataset. Overall, the results of the 5-layer U-Net model demonstrate its effectiveness in accurately segmenting zebrafish larvae images. The low MSE value, high accuracy, precision, recall, and F1 Score validate the model's ability to accurately capture and classify the structures of interest. The relatively short training time further highlights the efficiency of the model in achieving high-quality segmentation results.

4.2 Model 2

The second model, a 4-layer U-Net, achieved promising results in the segmentation of zebrafish larvae images. The results are summarized in Table 1 and visualized in Fig. 5(a)- 5(f), where each figure corresponds to one metric.



(a) (b)

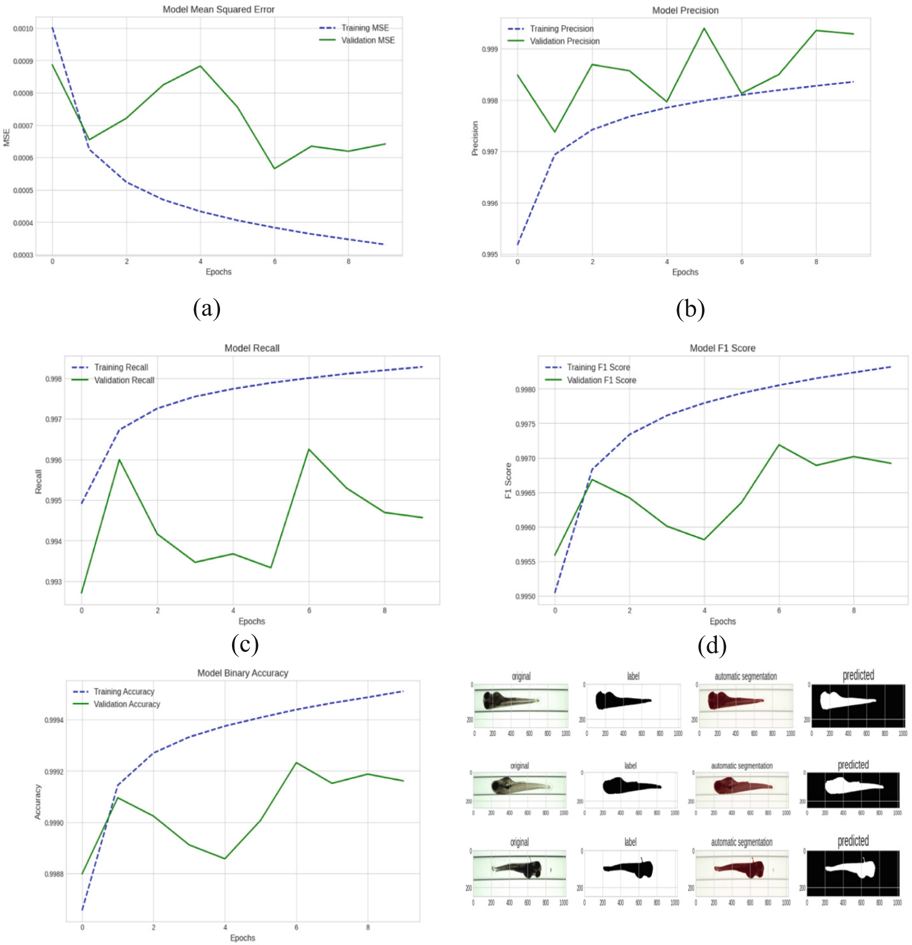
(c) (d)

(e) (f)

Fig. 4. Evaluation results from segmentation using model 1. Plots (a)-(e) show MSE, Precision, Recall, F1 Score and Accuracy of model1. Plot (f) shows some example from output of model1.

It obtained a Mean Squared Error (MSE) of 6.4200e-04, an Accuracy of 0.9992, Precision of 0.9993, Recall of 0.9946, and F1 Score of 0.9969. Although the MSE was slightly higher compared to the first U-Net model, this model demonstrated comparable or higher performance in terms of accuracy, precision, recall, and F1 Score.

The training time for the second model was approximately 53.39 min, which was longer than the first model. Overall, the results confirm the effectiveness of the 4-layer



(a) (b)

(c) (d)

(e) (f)

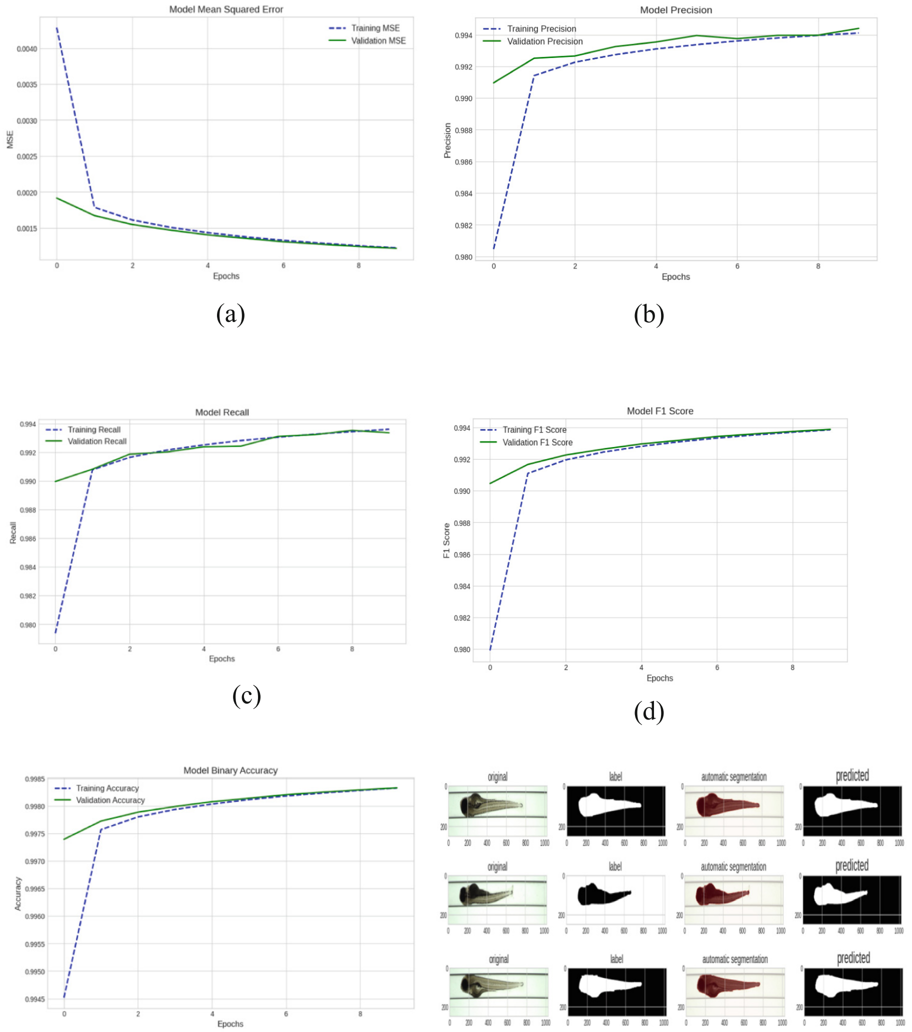
Fig. 5. Evaluation results from segmentation using model 2. Plots (a)-(e) show MSE, Precision, Recall, F1 Score and Accuracy of model 2. Plot (f) shows some example from output of model 2.

U-Net model in accurately segmenting zebrafish larvae images, with slightly different trade-offs compared to the first model.

4.3 TransUnet

The third model, TransUNet, combines a 4-layer U-Net architecture with Transformer blocks that incorporate attention mechanisms into the feature maps generated by the Convolutional Layers.

The evaluation of this model yielded the following results: Mean Squared Error (MSE) of 0.0012, Accuracy of 0.9983, Precision of 0.9944, Recall of 0.9934, and F1 Score of 0.9939. The training time for this model was approximately 59.43 min.



(a) (b)

(c) (d)

(e) (f)

Fig. 6. Evaluation results from segmentation using TransUnet. Plots (a)-(e) show MSE, Precision, Recall, F1 Score and Accuracy of model. Plot (f) shows some example from output of model.

Comparing these results to the previous two models, the TransUNet model exhibits slightly higher MSE and lower accuracy, precision, recall, and F1 Score values. However, it is important to note that these differences are relatively small and still indicate strong performance in segmenting zebrafish larvae images.

Furthermore, the visualizations in Fig. 6(a)- 6(f) demonstrate that the plots for MSE, Precision, Recall, F1 Score, and accuracy are smoother with less fluctuation compared to the previous models. This smoother behavior indicates improved convergence and stability in the training process of the TransUNet model.

Overall, the results highlight the efficacy of the TransUNet model in accurately segmenting zebrafish larvae images. Despite some minor differences in performance compared to the previous models, the TransUNet model exhibits smoother convergence, suggesting improved stability during training.

The results of different segmentation models for zebrafish are presented in Table 1. The findings suggest that Model1 excels in recognizing the zebrafish area and precisely segmenting it from the background, although with only a slight advantage over the second model. Nevertheless, the other two models also produce noteworthy results.

Table 1. Results of different segmentation models

Segmentation models	MSE	Accuracy	Precision	Recall	F1_Score	Time (Min)
Model 1_5B	**0.00037**	**0.9995**	**0.9983**	**0.9980**	**0.9981**	**29.21**
Model 2_4B	0.00064	0.9992	0.9993	0.9946	0.9969	53.39
TransUNet	0.0012	0.9983	0.9944	0.9934	0.9939	59.43

5 Unsupervised Segmentation

In our study, we designed and trained three different segmentation models, taking advantage of the extensive VAST-Bioimaging dataset of zebrafish images. This rich dataset, comprised of various imaging modalities and scenarios, provided a robust platform for developing comprehensive and versatile segmentation models. Each model was fine-tuned using the training set, optimizing its ability to segment zebrafish accurately from diverse backgrounds and lighting conditions.

Significantly, we leveraged transfer learning to further enhance the capabilities of our models. By using the knowledge gained from training on the VAST-Bioimaging dataset, we could effectively apply these pre-trained models to the task of segmenting zebrafish in Optical Projection Tomography (OPT) images, a substantially different imaging modality. This approach of utilizing unsupervised learning for segmentation dramatically reduced the need for manually labeled OPT images, which are often time-consuming and expensive to produce. Thus, our pipeline not only increased the efficiency of the segmentation process but also extended the application of our trained models to new, unlabelled datasets, illustrating the potential of unsupervised segmentation in biomedical imaging. Figure 7 displays a sample of an OPT image along with its mask after preprocessing and segmentation. However, due to the high transparency of the tail, it has become indiscernible and remains unsegmented.

6 Reconstruction

The reconstruction of 3D images from 2D zebrafish images plays a pivotal role in enabling comprehensive analysis and understanding of complex biological phenomena [19]. By accurately segmenting zebrafish larvae and obtaining precise 2D masks, we

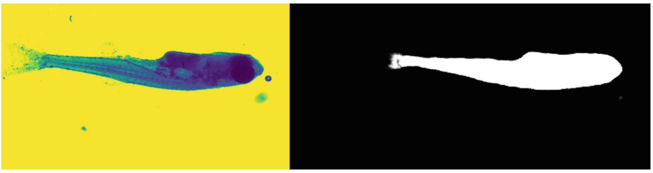


Fig. 7. OPT image segmentation using transfer learning

can reconstruct the 3D samples and unlock valuable information about their morphology and structure. This process facilitates in-depth measurements, such as volumetric analysis, shape preservation, and spatial relationships between different structures within the zebrafish larvae. The silhouette-based 3D reconstruction method, in particular, is more suitable for this type of data, ensuring accurate shape preservation and reconstruction fidelity. Figure 8 shows the process. ψ, is a vector includes series of correlated camera configurations such as angle of φ and the rotation angle of γ around Z axis. The reconstructed 3D images provide a more comprehensive representation of the specimens, allowing for a deeper exploration of developmental processes, disease progression, and treatment responses over time. Furthermore, by incorporating time-series analysis, researchers can monitor dynamic changes and gain insights into the temporal patterns and mechanisms underlying various biological phenomena. The utilization of 3D reconstruction techniques in zebrafish research, especially with the silhouette-based approach, holds immense potential for advancing measurement capabilities in high-throughput imaging. These measurements contribute significantly to our understanding of biology in general and have practical applications in drug discovery. As a result, they empower researchers to make informed decisions for future experiments and interventions.

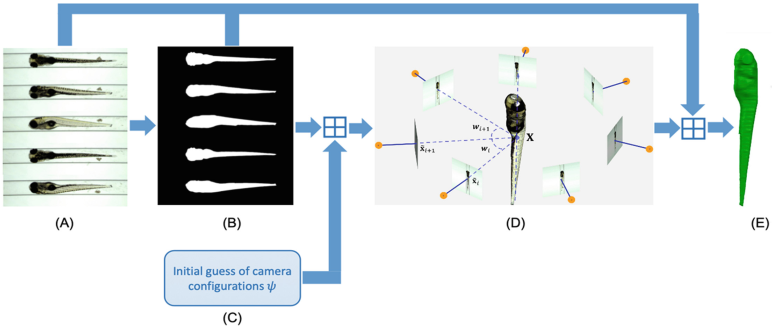


Fig. 8. Flowchart for 3D reconstruction and measurements of zebrafish larvae using high-throughput axial-view imaging. (A) Acquiring axial-view images with the VAST camera. (B) Segmentation for obtaining 2D shapes. (C) Initializing camera configurations based on VAST BioImager estimates. (D) Visualizing parameterization and calibration of the axial-view camera system. (E) Reconstructed 3D zebrafish models with volumetric representation [6].

In summary, the 3D reconstructed images obtained through our pipeline open new avenues for time-series analysis in zebrafish research, particularly in the domains of drug screening and cancer treatment [20]. The ability to monitor and quantify the growth and response of cancer cells over time provides valuable information for developing targeted therapies and improving treatment strategies. This integrated approach holds promise for advancing our understanding of disease mechanisms and ultimately improving patient care.

7 Conclusion and Future Work

In our research, we have established a robust pipeline for the segmentation of zebrafish larvae images, using transformer networks and transfer learning methodologies with U-Net and TransUNet models. We applied the trained models to the segmentation of Optical Projection Tomography (OPT) images, effectively creating an unsupervised segmentation approach. The resulting 3D reconstructions from these segmented OPT images offer a unique insight into various biological phenomena.

Future work will focus on addressing overfitting issues and expanding our models' performance through diverse strategies and advanced machine learning techniques. The objective is to refine these methodologies continuously, paving the way for more detailed insights into complex biological systems and advancing biomedical research.

References

1. Mishra, Y.K., et al.: Zebrafish (danio rerio) as an ecotoxicological model for nanomaterial induced toxicity profiling. Precision Nanomed. **4**(1), 750–782 (2021)
2. Wei, S., et al.: Perturbing tumor cell metabolism with a Ru (II) photo-redox catalyst to reverse the multidrug resistance of lung cancer. Sci. China Chem. **66**(5), 1482–1488 (2023)
3. Nord, H., Dennhag, N., Muck, J., Hofsten, J.: Pax7 is required for establishment of the xanthophore lineage in zebrafish embryos. Mol. Biol. Cell **27**(11), 1853–1862 (2016)
4. McCluskey, B.M., Postlethwait, J.H.: Phylogeny of zebrafish, a 'model species', within danio, a 'model genus.' Mol. Biol. Evol. **32**(3), 635–652 (2015)
5. Barbazuk, W.B., et al.: The syntenic relationship of the zebrafish and human genomes. Genome Res. **10**(9), 1351–1358 (2000)
6. Guo, Y., Veneman, W.J., Spaink, H.P., Verbeek, F.J.: Three-dimensional reconstruction and measurements of zebrafish larvae from high-throughput axial-view in vivo imaging. Biomed. Opt. Express **8**(5), 2611–2634 (2017)
7. Phillips, J.B., Westerfield, M.: Zebrafish models in translational research: tipping the scales toward advancements in human health. Dis. Model. Mech. **7**(7), 739–743 (2014)
8. Bradford, Y.M., et al.: Zebrafish models of human disease: gaining insight into human disease at ZFIN. ILAR J. **58**(1), 4–16 (2017)
9. Liu, K., Petree, C., Requena, T., Varshney, P., Varshney, G.K.: Expanding the CRISPR toolbox in zebrafish for studying development and disease. Frontiers Cell Dev Biol. **7**, 13 (2019)
10. Prykhozhij, S.V., Berman, J.N.: Zebrafish knock-ins swim into the mainstream, Disease Models and Mechanisms, vol. 11, no. 10, p. dmm037515 (2018)
11. Cagan, R.L., Zon, L.I., White, R.M.: Modeling cancer with flies and fish. Dev. Cell **49**(3), 317–324 (2019)

12. Goessling, W., Sadler, K.C.: Zebrafish: an important tool for liver disease research. Gastroenterology **149**(6), 1361–1377 (2015)
13. Rissone, A., Burgess, S.M.: Rare genetic blood disease modeling in zebrafish. Front. Genet. **9**, 348 (2018)
14. Zhao, Y., Zhang, K., Sips, P., MacRae, C.A.: Screening drugs for myocardial disease in vivo with zebrafish: an expert update. Expert Opin. Drug Discov. **14**(4), 343–353 (2019)
15. Griffin, A., Hamling, K.R., Hong, S., Anvar, M., Lee, L.P., Baraban, S.C.: Preclinical animal models for dravet syndrome: seizure phenotypes, comorbidities and drug screening. Front. Pharmacol. **9**, 573 (2018)
16. Kalueff, A.V., et al.: Towards a comprehensive catalog of zebrafish behavior 1.0 and beyond. Zebrafish **10**(1), 70–86 (2013)
17. Buda, M., Saha, A., Mazurowski, M.A.: Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm. Comput. Biol. Med. **109**, 218–225 (2019)
18. Chen, J., et al.: Transunet: transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306 (2021)
19. Guo, Y., Xiong, Z., Verbeek, F.J.: An efficient and robust hybrid method for segmentation of zebrafish objects from bright-field microscope images. Mach. Vis. Appl. **29**, 1211–1225 (2018)
20. Fukushima, H.C.S., et al.: Zebrafish toxicological screening could aid leishmaniosis drug discovery. Lab. Anim. Res. **37**(1), 1–11 (2021)

Exploring Deep Learning Techniques for Vision-Based Vehicle Re-Identification: A Traffic Intersection Case Study

Biplav S. Regmi(✉), Matthew N. Dailey, and Mongkol Ekpanyapong

Asian Institute of Technology, Khlong Nueng, Thailand
`{st123439,mdailey,mongkol}@ait.asia`

Abstract. We present a comprehensive review of recent deep learning techniques for vehicle re-identification (Re-ID) in Intelligent Transportation Systems (ITS). Vehicle Re-ID involves recognizing and matching specific vehicles across different cameras, against challenges such as pose variation, illumination, and weather. The study summarizes vision-based approaches, discusses challenges involved in real-time applications, and provides a brief comparison of methodologies and datasets available for the task. Additionally, a case study showcases an application of vehicle Re-ID at traffic junctions for the purpose of enforcing traffic rules, including identifying traffic light violations and capturing clear views of license plates for violator identification. By providing a summary of existing work and suggesting future research directions, we aim to assist researchers in advancing intelligent transportation systems and contributing to the development of smarter cities.

Keywords: Vehicle Re-ID · ITS · Smart City

1 Introduction

Progress in computer vision has led to numerous smart city initiatives aimed at enhancing the liveability and vibrancy of urban areas. Within this context, intelligent transportation systems (ITSs) represent a crucial component of smart city strategies. The major research in the area include vehicle detection, vehicle tracking, vehicle type recognition, and vehicle re-identification. Vehicle Re-ID has long been one of the important research problems in ITS [1] and it has broad applicability in urban monitoring. Ensuring the efficient functioning of a transportation system requires the ability to accurately track and identify vehicles captured in one video camera feed across multiple other camera feeds at the same or multiple locations for various traffic-related applications.

Supported by AIT AI center.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 228–242, 2023.
https://doi.org/10.1007/978-981-99-7969-1_17

Vehicle Re-ID is a type of image retrieval that recognizes and matches specific vehicles in images or videos captured by different cameras. This may involve fine-grained classification challenges such as identifying the specific make and model of the vehicle in question. The goal is to find the image in a gallery that is most likely to the same object as the query image. To enable a reliable tracking of instances across cameras, it is essential to extract unique identity features from each image or sequence of image patches and use those features to conduct feature matching across multiple cameras. Similarly, a great deal of research has focused on person re-identification [2]. However, vehicle re-identification is considered more difficult than person re-identification because of the overall similarity between vehicles and the significant variations among different views of the same vehicle. Distinguishing between individual people is possible in principle due to their unique facial features, while in contrast, vehicles possess similar shapes and colors, resulting in relatively small inter-class differences. The challenge in vehicle re-identification lies in the substantial variations within the same class, as images of the same vehicle can appear extremely dissimilar depending on the resolution and viewpoint of the camera capturing the image and the conditions under which the images are captured.

Vehicle Re-ID models suffer from two major environmental variations: low level perceptual variations such as illumination and resolution, as well as high level semantic variation such as viewpoint and occlusion. Deep learning models perform well at most tasks when there is sufficient training data to form their representations, but learning a robust and distinctive identity representation for vehicle in an unconstrained environment is still a very challenging task. It is impractical and expensive to collect data representing every possible variation in appearance, even if it is technically possible. If data preparation is inaccurate, noise will be introduced, degrading the model. Additionally, there is the danger of domain bias between training and test sets, which will negatively affect the accuracy of the resulting model if a new unforeseen condition is introduced in a target test image. Many state of the art techniques rely on adjusting or augmenting the samples in the original dataset using transformations or synthesis. Given this challenging problem and rich literature on its solution, this paper makes the following contributions:

- Identify challenges and produce an overview of recent approaches to the task of vehicle Re-ID that use deep learning.
- Compare publicly available benchmark datasets and evaluation metrics for the vehicle Re-ID task.
- Implement a use case scenario involving video feeds from traffic light junctions in which vehicles must be re-identified based on images from one or more low resolution overview cameras in sequences of images of the same set of intersections captured by high resolution zoomed in cameras.

2 Literature Review

Among the current approaches to vehicle Re-ID, there are two primary approaches. First we treat the problem as a classification problem, where we assume a large dataset of vehicle images is provided with labels. This type of model can in principle perform fine-grained classification of vehicles. However, this approach is susceptible to overfitting as the number and variety of vehicle models increase. Consequently, accuracy limitations can arise when dealing with a large number of samples. The second approach revolves around vehicle verification. In vehicle verification, a pair of vehicle images accompanied by their respective vehicle IDs is provided as input, and the task at hand is to ascertain whether the two vehicles represent the same ID or different IDs. A loss function appropriate for verification is employed to facilitate learning such that gradually reducing the loss facilitates distinguishing between different vehicles and accurately predicting the sameness of pairs with the same ID.

Cross-camera Re-ID challenges tend to be limited in terms of the amount of data available for learning, prompting the adoption of data augmentation approaches that combine multiple datasets or synthesize additional vehicle image pair to help train a robust network. New data can be synthesized using either 3D-based models or adversarial learning techniques. Controlled generation by 3D rendering engines is used to simulate different lighting conditions in a study by Bak et al. [3]. While artificial images may not be appropriate for training, due to their significant differences in style and appearance, they can still be useful in quantitatively analyzing the impact of visual factors in Re-ID. After Ian Goodfellow introduced GANs [4] in 2014, they have been also used in Vehicle Re-ID for generating vehicle image to add diversity in training data [5–7]. Zheng et al. [5] were pioneers in employing GANs for Re-ID tasks with unlabeled samples and incorporated regularization using label smoothing for leveraging unsupervised data. However, synthesized images can suffer from limited diversity, because they rely on a small set of original samples as prior information. Additionally, generating unrealistic samples in the synthesis process will hinder their usefullness in regularizing feature representations and ensuring generalization across different cameras.

On the basis of the specific model's feature learning approaches, we can further divide this topic into methods based on local features, metric learning, generative learning, attention mechanisms and other or mixed approaches. We consider each of these approaches in detail in the following subsections.

2.1 Methods Based on Local Features

Early research on vehicle Re-ID mainly dealt with finding global features that extract feature vectors from entire images [8,9]. However, this approach has limited accuracy. Therefore, researchers started to explore complementary local features to address this issue [10,11]. In this approach, various regional features that are discriminative in nature are employed to supplement the more global backbone features. These mechanisms explicitly identify and concentrate on informative sections of the vehicle to extract discriminative features, in addition to capturing robust features from the overall image. To localize regional features, researchers typically utilize additional annotations, such as keypoint localization, vehicle orientation, key-parts, bounding boxes, and part segmentation. He et al. [10] propose an end-to-end framework that uses part-regularized local features as a supplement to global representations using a detection branch. Their approach enhances the development of discriminative features that distinguish subtle differences and aid in aligning vehicle parts across various views to establish identity accurately. In a similar vein, Ma et al. [11] introduce a model with enhanced part features. They aim to learning an efficient feature representation for vehicle re-identification. Their approach incorporates a Grid Spatial Transformer Network (GSTN) that automatically finds vehicles and divides them into local regions for feature extraction. Additionally, residual attention mechanisms are employed to further refine the fine-grained identification process. By fusing the refined part features, an effective feature embedding is generated, leading to improved accuracy. However, its reliance on accurate annotations or key-part localization is time-consuming and resource intensive. As a result, it is preferable to develop more approaches that can focus accurately on informative parts without relying on corresponding ground truth annotations for complementary features.

2.2 Methods Based on Metric Learning

Within the domain of vehicle Re-ID, metric learning targets to minimize the distance between pairs of vehicle images with the same ID while maximizing the distance between pairs of vehicle images with different IDs. The commonly used loss functions for metric learning in this context include contrastive loss, triplet loss, and quaternion loss. Given two images $X1$ and $X2$, the corresponding feature vectors f_{x_1} and f_{x_2} are computed using forward propagation in a neural network. Similarity is measured by computing the square root of the sum of squared differences between the elements of the feature vectors, which is defined as

$$d_{x_1,x_2} = \|\mathbf{f_{x_1}} - \mathbf{f_{x_2}}\|. \tag{1}$$

Contrastive Loss: This loss is typically used with a siamese network consisting of two identical sub-networks sharing weights that calculates a similarity or distance between two inputs, which is updated using contrastive loss, which is given as

$$L_c = \frac{1}{2N} \sum_{i,j \in \text{pair}} d^2_{x_i,x_j} + \sum_{i,j \notin \text{pair}} \max(0, \delta - d_{x_i,x_j})^2. \tag{2}$$

where δ controls the threshold at which samples are considered sufficiently dissimilar.

In the case of vehicle Re-ID, such a Siamese network takes in pairs of vehicle images, maps them separately to the same vector space, and outputs a distances between the feature vectors. The network is trained with L_c, with pairs of images labeled as either same ID (positive samples) or different IDs (negative samples). The goal of the loss function is to reduce the predicted distance between “same” sample pairs while maximizing the distance between “different” sample pairs.

When the distance between similar samples is large or the distance between dissimilar samples is small it results in higher loss value. By minimizing this loss function during training, the network learns to effectively distinguish between vehicles with distinct IDs. Liu et al. [9] introduce a deep convolutional network with their structure.

Triplet Loss: Triplet loss is a modification of contrastive loss that utilizes three inputs rather than two. The network takes in an anchor image (denoted a), a positive image that belongs to the same category as the anchor (denoted p), and a negative image that belongs to a different category than the anchor (denoted n). The objective is to learn a feature representation under which the distance between the anchor image and the positive image is minimized, while the distance between the anchor image and the negative image is maximized:

$$L_t = \frac{1}{N}\sum_{i=1}^{N} \max\left(0, d_{ap} - d_{an} + \delta\right). \qquad (3)$$

where δ controls the threshold at which anchors are sufficiently similar to its positives while sufficiently dissimilar to the negatives.

Ghosh et al. [12] propose a scheme called Relation Preserving Triplet Mining (RPTM) that utilizes matching of features to ensure that triplets maintain natural sub-groupings for an object. By incorporating view consistency, RPTM establishes a pose-aware and well-conditioned triplet loss that attains good performance over benchmark data.

Both contrastive loss and triplet loss can suffer from slow convergence and be sensitive to the specific selection of positive and negative samples, which can negatively affect the learning process in vehicle Re-ID. Choosing paired images can be time-consuming due to the large number of possible pairs. A delicate balance needs to be maintained while choosing pairs to avoid those that are too challenging to optimize the network and those that are too easy (i.e., the positive and negative samples are too similar or different). Another potential drawback of applying metric learning in vehicle Re-ID is that the feature space can end up being overcrowded or overly sparse, making it impossible to distinguish between different classes of vehicles.

2.3 Methods Based on Generative Learning

Generative adversarial networks (GANs) comprise a generator and discriminator for unsupervised learning [4]. The generator samples from a predefined distribution and uses those samples to generate synthetic data, while the discriminator is trained to differentiate between synthetic and real data. The goal is to strike a balance in which the generator produces synthetic samples that closely resemble real data, while the discriminator becomes proficient in identifying the synthetic data generated by the generator. Lou et al. [6] utilize an adversarial learning network called EALN to generate multi-view samples and hard negatives in the embedding space. Their approach enhances the network's discriminatory ability and robustness by generating hard negative samples within the designated embedding space. In order to address the issue of Re-ID under multiple viewing angles in vehicle identification, GANs can be trained to generate various perspective images from a single perspective image. Similarly, Peng et al. [7] propose a cross-camera adaptation framework for vehicle Re-ID in large multi-camera datasets. The framework transfers images from multiple cameras into a common space to reduce bias and creates samples with similar distributions. To further enhance performance, the authors introduce an attention alignment network that focuses on discriminative regions while suppressing background interference. However, using GANs for image generation can be challenging, as convergence can be difficult to achieve and training must ensure a balance between the generator and discriminator models to avoid unstable situations.

2.4 Methods Based on Attention Mechanisms

One recent development in Re-ID is the use of visual attention masks in combination with deep learning. Masks help emphasize important features in an image by suppresing irrelevant information. Deep neural networks can be trained to learn to focus on relevant areas of an image, leading to the development of two different types of attention mechanisms: soft and hard attention. Soft attention is deterministic, emphasizes certain areas or channels, and is differentiable, which is important for calculating gradients and learning attention weights. In contrast, hard attention centers around specific points and involves a prediction process that is random in nature and gives more emphasis on dynamic changes. The key differences between the two approach are the level of focus and predictability. An example of soft attention can be found in the work of Teng et al. [13], who present a Spatial and Channel Attention Network (SCAN) that contains two branches: a spatial attention branch and a channel attention branch. The two branches adjust the weights of the outputs in distinct regions and channels to emphasize discriminative information. The resulting attention model refines feature maps and automatically extracts more discriminative features. Khorramshahi et al. [14] propose a model called Adaptive Attention for Vehicle Re-identification (AAVER), which addresses the variability of keypoint importance based on pose of the vehicle. The authors observe that the conventional approach of solely focusing attention on keypoint locations is insufficient. The AAVER model consists of two pathways: one for capturing global appearance that focuses on extracting overall vehicle features, while the other extracts orientation-based part appearance that gives attention to keypoints for extracting local discriminative features. In a separate study by Khorramshahi et al. [15], attention mechanisms were introduced to condition feature maps on visible key points. The authors employ diverse datasets to train their networks and utilize triplet embedding for feature vector dimensionality reduction. The methods used by Khorramshahi et al. are based on hard attention. He et al. [16] develop a novel approach in which images are encoded as a sequence of patches and input to a transformer. The method achieves impressive results on multiple Re-ID benchmarks when compared to traditional CNN-based methods. Notably, this work was the first to employ a purely transformer-based approach to Re-ID. Attention mechanisms enhance accuracy by automatically identifying distinctive regions, but tend to concentrate on larger regions and overlook subtle differences at the pixel level. This could lead to less effectiveness in scenarios where the datasets are not adequately labeled or the backgrounds are complex.

2.5 Other Approaches

Liu et al. [17] introduce a vehicle re-identification framework named PROVID that utilizes deep neural networks and multimodal data to match vehicles in large-scale video surveillance. The framework follows two progressive steps: progressive search in the feature domain and a close-to-far search in the real-world space. For achieving precise vehicle search, the authors verified license plate number with a Siamese neural network. Jiao et al. [18] improved cross-resolution Re-ID performance by integrating image super-resolution with Re-ID, which effectively addresses resolution mismatch issues across multiple cameras. Zheng et al. [19] present a unified deep convolutional framework called DF-CVTC, which incorporates meaningful attributes such as camera views, vehicle types, and colors to guide the learning of deep feature representations. The collaboration among these components enhances the discriminative power of the learned representations. Khorramshahi et al. [20] presents a novel training framework that combines supervised and self-supervised objectives for vehicle Re-ID. This approach, named SSBVER, generalizes well across different backbone architectures, consistently boosting accuracy and accommodating resource constraints while achieving the best results to date on the VeRiWild dataset [21].

Despite the advancements in deep learning-based vehicle Re-ID over the years, challenges still exist due to the limited variation and complexity in available datasets. In a custom traffic surveillance scenarios, images from different cameras may have different resolutions, angles, lighting conditions, and color, making it difficult to identify the same vehicle across multiple cameras.

3 Datasets

In recent years, the accuracy of vehicle re-identification has improved due to the availability of large public datasets and advancements in deep learning technology that surpass the limitations of hardware performance. Datasets play a vital role in evaluating the performance of vehicle Re-ID systems. They ideally represent real-world surveillance camera data. Some of the well-known benchmark dataset are described in following subsections.

3.1 CompCars [8]

The CompCars dataset is a comprehensive collection of 136,726 complete car and 27,618 car part images, divided into network and monitoring subsets. The dataset includes annotations for 163 car makes with 1,716 models, along with labeled attributes such as maximum speed, displacement, doors, seats, and car type.

3.2 VeRi-776 [9]

VeRi-776 is a vehicle Re-ID dataset that comprises 50,000 real surveillance camera images depicting 776 distinct vehicles. Each vehicle is represented by 2 to 18 images captured from different viewpoints, exhibiting variations in resolution, occlusion, and illumination. The dataset includes annotations for spatial and temporal relations, license plate details, and additional vehicle information such as type, color, and model.

3.3 VehicleID [5]

The VehicleID dataset contains 221,763 daylight images of 26,267 vehicles from various surveillance cameras in a Chinese town, including 90,196 annotated images detailing model information for 10,319 vehicles.

3.4 VRID-1 [22]

The VRID dataset contains 10,000 images of 1,000 vehicles, each captured at least 10 times from 326 surveillance cameras in Guangdong city, China, between 7am and 5pm over a week. The images in this dataset vary in resolution and pose, ranging from 400 × 424 to 990 × 1134 pixels.

3.5 PKU-VD [23]

The PKU-VD dataset, created by Peking University's National Video Technology Engineering Laboratory, includes two substantial vehicle datasets, VD1 and VD2, based on real-world traffic in two cities. VD1 features high-resolution traffic camera images, while VD2 focuses on surveillance video capture. The datasets provide detailed annotations such as unique identity numbers, precise models, and colors. With nearly 1.1 million annotated images in VD1 and 0.8 million in VD2, the dataset is comprehensive in its coverage of popular vehicle models and colors, ensuring consistency and expansiveness for vehicle Re-ID and related research.

3.6 PKU-Vehicle [24]

Introduced by Peking University, the PKU-Vehicle dataset features 10 million vehicle images from surveillance cameras across multiple Chinese cities. The dataset is diverse in location, weather, lighting, shooting angles, and vehicle brands. It poses a challenge due to varying image resolutions from different cameras.

3.7 BoxCars [25, 26]

Developed by the Bruno University of Technology in the Czech Republic, the BoxCars dataset includes two versions: BoxCars21k and BoxCars116k. BoxCars21k features 21,250 cars, 63,750 images, 27 brands, and 148 models. Its larger counterpart, BoxCars116k, contains 27,496 cars, 116,826 images, 45 brands, and 693 models. Both datasets come with annotations such as 3D bounding boxes, type, make, and model information.

3.8 Vehicle-1M [27]

The Vehicle-1M dataset includes 936,051 frames of 55,527 vehicles with 400 different models, captured from various angles during day and night using surveillance cameras in multiple Chinese cities during both daytime and nighttime from different angles, including front and back views.

3.9 VRIC [28]

Distinguishing itself from other datasets, VRIC features diverse variations in resolution, motion blur, illumination, viewpoint, and occlusion. It comprises 60,430 images of 5,622 vehicles, captured by 60 unique surveillance cameras across various road scenes in Guangdong, China.

3.10 CityFlow [29]

CityFlow is a pioneering dataset for vehicle Re-ID and cross-camera tracking, featuring synchronized high-definition video from 40 cameras at 10 city intersections. It includes 229,680 labeled instances of 666 vehicles, making it both large and unique.

3.11 VERI-Wild [21]

VERI-Wild is a large-scale challenging dataset for vehicle Re-ID collected from 174 CCTV cameras across a 200 square km urban area. It contains 12 million raw images, later refined and annotated by 11 volunteers into 416,314 images representing 40,671 unique vehicles. Captured in a real-world, unconstrained setting, the dataset poses practical challenges such as viewpoint variation, varying illumination, complex backgrounds, and severe occlusion, making it ideal for testing algorithms and models.

4 Evaluation Strategy

To assess the efficacy of Re-ID models, various evaluation metrics have been proposed. This section outlines some of the commonly used strategies for evaluating the performance of vehicle re-identification algorithms.

4.1 Rank

The most commonly used rank evaluation metrics in vehicle reidentification are Rank-1 and Rank-5 accuracy. Rank-1 accuracy measures the percentage of query images for which the correct match appears at the top position in the gallery, while Rank-5 accuracy measures the percentage of query images for which the correct match appears within the top five positions in the gallery.

4.2 Mean Average Precision (mAP)

Mean Average Precision (mAP) is a more broad metric that takes into account the accuracy of all the ranked matches for all query images and computes an average precision score across all queries. This method takes both precision and recall into consideration. To compute the average precision (AP) for a query image, the number of gallery images with the same identity as the query image is denoted as k. Then, AP can be calculated by summing up the precision values at all the positions where a relevant image is retrieved and dividing the sum by the total number of relevant images in the gallery set:

$$AP = \frac{1}{k} \sum_{i=1}^{k} \frac{1}{p_i}, \tag{4}$$

where p_i represents the minimum number of images that must be examined to obtain the top i accurate results in the image sequence that has been sorted based on similarity.

mAP is then calculated by taking the mean of the AP measurements performed over q query images:

$$mAP = \frac{1}{q} \sum_{i=1}^{q} AP_i. \tag{5}$$

AP_i denotes the Average Precision for the i-th query image. The mean is calculated by summing up all the AP values and dividing by the total number of query images.

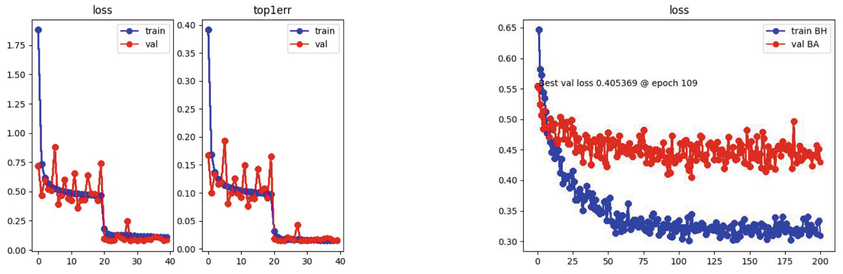
5 Experiments and Results

In this section, we present an example of applying vehicle Re-ID at traffic junctions. The objective is to match a vehicle violating traffic light rules across different viewpoints and resolutions within the junction. Utilizing an alternate viewpoint helps to obtain a clear view of the license plate of the violating vehicle. To accomplish this task, we collected data and annotated ground truth correspondences for five different traffic intersections in Thailand. The data were used to train a custom CNN model producing a embedding feature vector of size 128. Our training set consists of 225 paired tracks, while the validation set contains 32 paired tracks for hyperparameter tuning and early stopping. The model's performance is then evaluated on a test set of 32 paired tracks from the same intersections that are not in the training or validation datasets. This dataset contains vehicle tracks in light and heavy traffic conditions including occlusion conditions during various daytime and nighttime conditions. Initially, we trained the model on the MARS dataset [30], which is a benchmark for person re-identification commonly used to pre-train Re-ID models for other tasks. We then fine-tune this model checkpoint on our training set, leveraging transfer learning to adapt it to our specific application domain. We experiment with batch hard triplet loss function with and without margin(δ), to optimize the model's performance. For the model hyperparameter settings, we use Stochastic gradient descent optimizer with a momentum of 0.9, a learning rate of 0.0001, and a weight decay of 0.01 The learning rate is reduced by a factor of 10 every 20 epochs. The experiments were performed on a PC with Nvidia RTX3060 GPU.

To train our model, we utilized a batch size of 64 in each iteration consisting of 32 randomly selected image patches from a track of a vehicle of concern as anchor images thus ensuring image patches from two different vehicle tracks are included in each batch. In order to leverage use of triplet mining, we associate anchor images of a track with 32 random positive image patches for the same track but captured from different camera views, and 32 randomly negative image patches from different tracks. All image patches in our triplet batch dataset are standardized to a size of 128 × 128 pixels. In a preliminary experiment, the triplet pairs for each track were generated in the above mentioned fashion before model training and remains fixed throughout model training at all epochs which we refer as static batch assignment. This approach resulted in model overfitting due to insufficient diversity of image patches of anchors, positives and negatives in a batch. To address this issue, we conducted another experiment using dynamic batch assignment where we generated new triplet pairs for the anchors of each track after every epoch thus ensuring the positives and negatives in triplets are randomised over the period of training. This approach facilitated better generalization of the model since positive and negative image patches with respect to anchors within a track are better differentiated in the feature space. This adjustment effectively resolved the problem of model over-fitting, as illustrated in Fig. 1b and 1c. When the triplet loss was optimised with the margin factor(δ) equal to 0.1, the training was more stabilized, as illustrated in Fig. 1d. In this case, feature representations are learned in such a way that the distance between the anchor and positive examples is less than the distance between the anchor and negative examples by at least the specified margin.

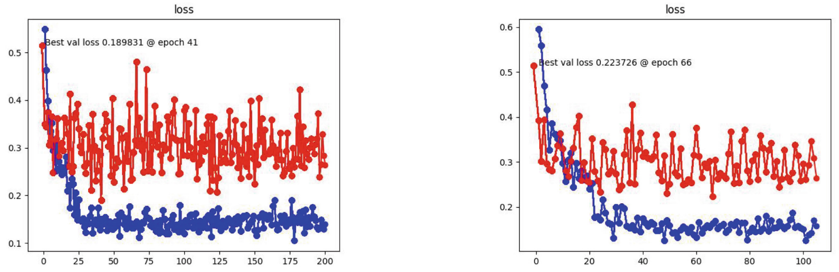
Next, we assessed the performance of our model on the test dataset using the track-to-track evaluation strategy. This approach involves computing the average embedding of the tracks based on the embeddings of individual images within each track, separately for each camera views. By calculating the cosine similarity difference between the weighted embedding of a track in a camera with all other tracks in another camera, we generate ranks for each track. To evaluate our model, we employ the mAP (mean Average Precision) and CMC

(Cumulative Match Curve) metrics. The results of this evaluation are presented in the Table 1.



(a) Loss and top-1 error curve of model when pre-training on MARS

(b) Training and validation loss when fine-tuning using static batch assignment and batch hard triplet loss



(c) Training and validation loss when fine-tuning with dynamic batch assignment and batch hard triplet loss

(d) Training and validation loss when fine-tuning with dynamic batch assignment and batch hard triplet loss, and a margin of 0.1

Fig. 1. Training curves for Re-ID models

Based on the aforementioned experiments, it is worth mentioning that dynamic batch hard triplet loss outperforms the static hard triplet loss in the Re-ID task. The utilization of the dynamic batch hard triplet loss with a margin of 0.1 as our loss function leads to the highest mAP of 0.862. Additionally, both dynamic batch models, with and without the margin, achieve a rank-1 accuracy of 78.125. However, the model with the margin performs better in terms of rank-2 and rank-3 accuracy. Conversely, the model without the margin demonstrates higher rank-4 and rank-5 accuracy in our test dataset.

Table 1. Experimental Results

Experiment	mAP	Rank-1	Rank-2	Rank-3	Rank-4	Rank-5
Static batch hard triplet loss	0.587	46.875	53.125	68.75	68.75	71.875
Dynamic batch hard triplet loss with no margin	0.859	**78.125**	87.5	90.625	**96.875**	**96.875**
Dynamic batch hard triplet loss with margin of 0.1	**0.862**	**78.125**	**90.625**	**93.75**	93.75	93.75

6 Conclusion

In this study, we analyzed the trends of deep learning based methods for vehicle Re-ID tasks. We critically analyzed the advantages associated with multiple approaches to this task. A list of public vehicle datasets is presented in brief with their notable unique features, which we hope will help guide future researchers. We also summarized the evaluation metrics used in this domain to measure model efficiency.

We further demonstrated the results of an experiment with a practical use case scenario in this field. The experiment suggests that metric learning based on dynamic batch hard triplet loss is most suitable to optimise the model for such task. In the future, we will seek to improve the accuracy of the Re-ID task by metric learning approach more efficiently in a single end to end framework. Also, to address the issue of high intra-class variability and enhance accuracy when performing inference across different perspectives, we might require large-scale vehicle dataset that capture multiple viewpoints of an identity. Methods such as GANs can also be used to learn a transformation model that can infer views from other perspectives. By synthesizing multi-view features from a single-view feature and embedding them into the distance space using metric learning, we aim to improve vehicle Re-ID in multiple viewing perspectives. Furthermore, we will also seek to use vision transformer models in the future as a possible way to further improve model accuracy.

References

1. Wang, H., Hou, J., Chen, N.: A survey of vehicle re-identification based on deep learning. IEEE Access **7**, 172443–172469 (2019)
2. Wu, D., et al.: Deep learning-based methods for person re-identification: a comprehensive review. Neurocomputing **337**, 354–371 (2019)
3. Bąk, S., Carr, P., Lalonde, J.-F.: Domain adaptation through synthesis for unsupervised person re-identification. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11217, pp. 193–209. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01261-8_12
4. Goodfellow, et al.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020)
5. Zheng, Z., Zheng, L., Yang, Y.: Unlabeled samples generated by GAN improve the person re-identification baseline in vitro. In: IEEE ICCV, pp. 3754–3762 (2017)
6. Lou, Y., Bai, Y., Liu, J., Wang, S., Duan, L.-Y.: Embedding adversarial learning for vehicle re-identification. IEEE Trans. Image Process. **28**(8), 3794–3807 (2019)

7. Peng, J., Jiang, G., Chen, D., Zhao, T., Wang, H., Xianping, F.: Eliminating cross-camera bias for vehicle re-identification. Multimedia Tools Appl. **81**(24), 34195–34211 (2022)
8. Yang, L., Luo, P., Change Loy, C., Tang, X.: A large-scale car dataset for fine-grained categorization and verification. In: IEEE CVPR, pp. 3973–3981 (2015)
9. Liu, H., Tian, Y., Yang, Y., Pang, L., Huang, T.: Deep relative distance learning: tell the difference between similar vehicles. In: IEEE CVPR, pp. 2167–2175 (2016)
10. He, B., Li, J., Zhao, Y., Tian, Y.: Part-regularized near-duplicate vehicle re-identification. In: IEEE CVPR, pp. 3997–4005 (2019)
11. Ma, X., Zhu, K., Guo, H., Wang, J., Huang, M., Miao, Q.: Vehicle re-identification with refined part model. In: International Conference on Multimedia & Expo Workshops, pp. 603–606. IEEE (2019)
12. Ghosh, A., Shanmugalingam, K., Lin, W.Y.: Relation preserving triplet mining for stabilising the triplet loss in re-identification systems. arXiv:2110.07933 (2021)
13. Teng, S., Liu, X., Zhang, S., Huang, Q.: SCAN: spatial and channel attention network for vehicle re-identification. In: Hong, R., Cheng, W.-H., Yamasaki, T., Wang, M., Ngo, C.-W. (eds.) PCM 2018. LNCS, vol. 11166, pp. 350–361. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00764-5_32
14. Khorramshahi, P., et al.: A dual-path model with adaptive attention for vehicle re-identification. In: IEEE ICCV, pp. 6132–6141 (2019)
15. Khorramshahi, P., Peri, N., Kumar, A., Shah, A., Chellappa, R.: Attention driven vehicle re-identification and unsupervised anomaly detection for traffic understanding. In: CVPR Workshops, pp. 239–246 (2019)
16. He, S., Luo, H., Wang, P., Wang, F., Li, H., Jiang, W.: Transreid: transformer-based object re-identification. In: IEEE ICCV, pp. 15013–15022 (2021)
17. Xinchen Liu, W., Liu, T.M., Ma, H.: Provid: progressive and multimodal vehicle reidentification for large-scale urban surveillance. IEEE Trans. Multimedia **20**(3), 645–658 (2017)
18. Jiao, J., Zheng, W.S., Wu, A., Zhu, X., Gong, S.: Deep low-resolution person re-identification. In: AAAI Conference on Artificial Intelligence, vol. 32 (2018)
19. Li, et al.: Attributes guided feature learning for vehicle re-identification. IEEE Trans. Emerg. Top. Comput. Intell. **6**(5), 1211–1221 (2021)
20. Khorramshahi, P., Shenoy, V., Chellappa, R.: Robust and scalable vehicle re-identification via self-supervision. In: IEEE CVPR, pp. 5294–5303 (2023)
21. Lou, Y., Bai, Y., Liu, J., Wang, S., Duan, L.: Veri-wild: a large dataset and a new method for vehicle re-identification in the wild. In: IEEE CVPR, pp. 3235–3243 (2019)
22. Li, X., Yuan, M., Jiang, Q. and Li, G.: VRID-1: a basic vehicle re-identification dataset for similar vehicles. In: IEEE ITSC, pp. 1–8 (2017)
23. Yan, K., Tian, Y., Wang, Y., Zeng, W., Huang, T.: Exploiting multi-grain ranking constraints for precisely searching visually-similar vehicles. In: IEEE ICCV, pp. 562–570 (2017)
24. Bai, Y., Lou, Y., Gao, F., Wang, S., Yuwei, W., Duan, L.-Y.: Group-sensitive triplet embedding for vehicle reidentification. IEEE Trans. Multimedia **20**(9), 2385–2399 (2018)
25. Sochor, J., Herout, A., Havel, J.: Boxcars: 3d boxes as CNN input for improved fine-grained vehicle recognition. In: IEEE CVPR, pp. 3006–3015 (2016)
26. Sochor, J., Špaňhel, J., Herout, A.: Boxcars: improving fine-grained recognition of vehicles using 3-d bounding boxes in traffic surveillance. IEEE ITSC **20**(1), 97–108 (2018)

27. Guo, H., Zhao, C., Liu, Z., Wang, J., Lu, H.: Learning coarse-to-fine structured feature embedding for vehicle re-identification. In: AAAI Conference on Artificial Intelligence, vol. 32 (2018)
28. Brox, T., Bruhn, A., Fritz, M. (eds.) Pattern Recognition: 40th German Conference, GCPR 2018, vol. 11269. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12939-2
29. Tang et al. Cityflow: a city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In: IEEE CVPR, pp. 8797–8806 (2019)
30. Zheng, L., et al.: MARS: a video benchmark for large-scale person re-identification. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 868–884. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_52

Large Language Models

ChatGPT Knows Your Attacks: Synthesizing Attack Trees Using LLMs

Olga Gadyatskaya(✉) and Dalia Papuc

LIACS, Leiden University, Leiden, The Netherlands
{o.gadyatskaya,d.papuc}@liacs.leidenuniv.nl

Abstract. Attack trees are a popular method to represent cyberattack scenarios. It is often challenging for organizations to design attack trees for relevant systems and scenarios, as this requires advanced security expertise and the engagement of many stakeholders. In recent years, many studies in academic literature have proposed methods for automating attack tree creation from system models or from libraries of attack patterns. However, these approaches are not yet mature enough to be of practical use in organizations.

The advent of large language models (LLMs) opens new opportunities for helping organizations in designing attack trees. We can envisage that organizations would be able to speed up attack tree design and benefit from LLMs like ChatGPT if they could rely on the quality of produced models. In this study, we investigate the feasibility of using ChatGPT to synthesize attack trees for specific scenarios. We propose a method to make ChatGPT to output attack tree-like models, we propose an approach to evaluate the quality of synthesized attack trees, and we evaluate these in two case studies. Our results show that LLMs like ChatGPT can indeed be valuable companions for designing attack trees. Yet, as expected, ChatGPT often fails to capture the meaning of the refinement operators, and the human analyst engaging with ChatGPT still needs to monitor the quality of the results.

Keywords: Attack trees · large language models · ChatGPT · threat model synthesis

1 Introduction

Cyber security is paramount today when all nation-states, organizations, and individuals regularly face different cyber threats. To prepare for imminent cyber attacks, organizations conduct cyber risk management, a recurrent process to identify relevant cyber risks and treat them to diminish their likelihood and impacts. Identification of future risks is a challenging task, as it requires being able to anticipate *what will come*. Cyber security professionals rely on their own extensive knowledge and experience, acquire data about cyber attacks

---

O. Gadyatskaya and D. Papuc—Equal contribution.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 245–260, 2023.
https://doi.org/10.1007/978-981-99-7969-1_18

incurred by other organizations (*aka* threat intelligence), and utilize systematic approaches to elicit cyber threats [15,35].

Among those systematic approaches, attack trees are a popular graphical security model that represents cyber threats from the attacker's perspective in a hierarchical manner. Introduced in the 90 s by security experts B. Schneier [31] and E. Amoroso [1], it has steadily gained popularity in both industry and academia as a versatile and easy-to-learn method for modeling threats, which also enjoys theoretical underpinnings that support advanced security analysis [14,25,34].

One of the challenges that security practitioners face when working with attack trees is the task of creating a tree for a specific scenario or system. Practitioners have to understand the scenario very well, and they also need to be well-versed in cyber attack techniques. Frequently, attack tree creation is done as teamwork, involving multiple practitioners possessing diverse expertise [14]. This is an expensive and time-consuming process. At the same time, practitioners often have doubts about the completeness of the produced tree, as they are afraid to miss important attack scenarios [11,14]. Thus, in recent years many academic researchers turned to the problem of automated attack tree generation, with proposals developed in studies, e.g., [12,13,18,20,27,33], among others. We refer the readers to the recent survey by Widel et al. [34] for an extensive overview of academic attack tree generation approaches. The challenge with the proposed approaches is that they require to have a formal model of an underlying system, from which the attack tree will be generated [13,27,33], or to have a library of relevant annotated attack scenarios [4,20]. Both of these requirements might not be feasible for smaller organizations with restricted security budgets and are challenging to fulfill even for large and mature organizations. Thus, the practical usefulness of existing attack tree generation approaches is currently limited [34].

In recent years a new powerful AI tool has emerged: large language models (LLMs) trained on huge datasets that are able to answer questions posed by human analysts in natural language in real-time. Perhaps, one of the most well-known LLM today is GPT [26] that powers the famous ChatGPT platform[1] by OpenAI, but there exist other advanced LLMs, such as Bloom [30] and Vicuna [8]. LLMs with a user interface like ChatGPT allow a human interlocutor to receive structured knowledge on any topic of their choice. ChatGPT has been shown to be a knowledgeable companion in, e.g., healthcare [29] and law fields [22]. Moreover, the value of LLMs has been demonstrated in the software engineering field, where AI assistants allow developers to write code faster [3]. Thus, in this work, we set out to explore the following research question: *are LLMs like ChatGPT able to help security analysts produce attack trees for relevant scenarios?*

To the best of our knowledge, this question has not yet been addressed in the literature. Recently, Charan et al. [7] have explored the usage of ChatGPT for generating malicious code payloads for the most prevalent MITRE ATT&CK techniques, and Gupta et al. [16] have demonstrated that ChatGPT can generate

---

[1] https://chat.openai.com/.

code of various ransomware samples and other types of malicious code. ChatGPT has also been assessed for security risks it poses for organizations and users [6,9,36]. Gupta et al. [16] have discussed various application scenarios for ChatGPT in the context of cyber defenses, yet they have not assessed the quality of defensive strategies that could potentially be developed. We were not able to find previous studies on applying ChatGPT for synthesizing attack tree models.

Our research question is far from trivial. First, we need to assess whether generalist LLMs available to practitioners possess enough specialist security knowledge. This aspect relates to *completeness* of produced attack trees, as we want to make sure that LLMs can output trees where all relevant attack scenarios are present. Second, it is known that sometimes LLMs like ChatGPT produce answers that might sound plausible but are in fact incorrect [16,32]. This might affect *soundness* of synthesized attack trees if these models include attacks that are actually not relevant to the scenario in question. Finally, LLMs need to be able to output *well-formed* attack trees, i.e. readable enough and structured according to the attack tree formalism.

2 Attack Trees Background

Since the paper is on the topic of ChatGPT, let us start by questioning ChatGPT what an attack tree is. As expected, ChatGPT is aware of attack trees and is capable of providing a meaningful explanation of them (see example in Fig. 1). This explanation is very close to the attack tree concepts introduced in seminal studies, e.g., [23,25,34].

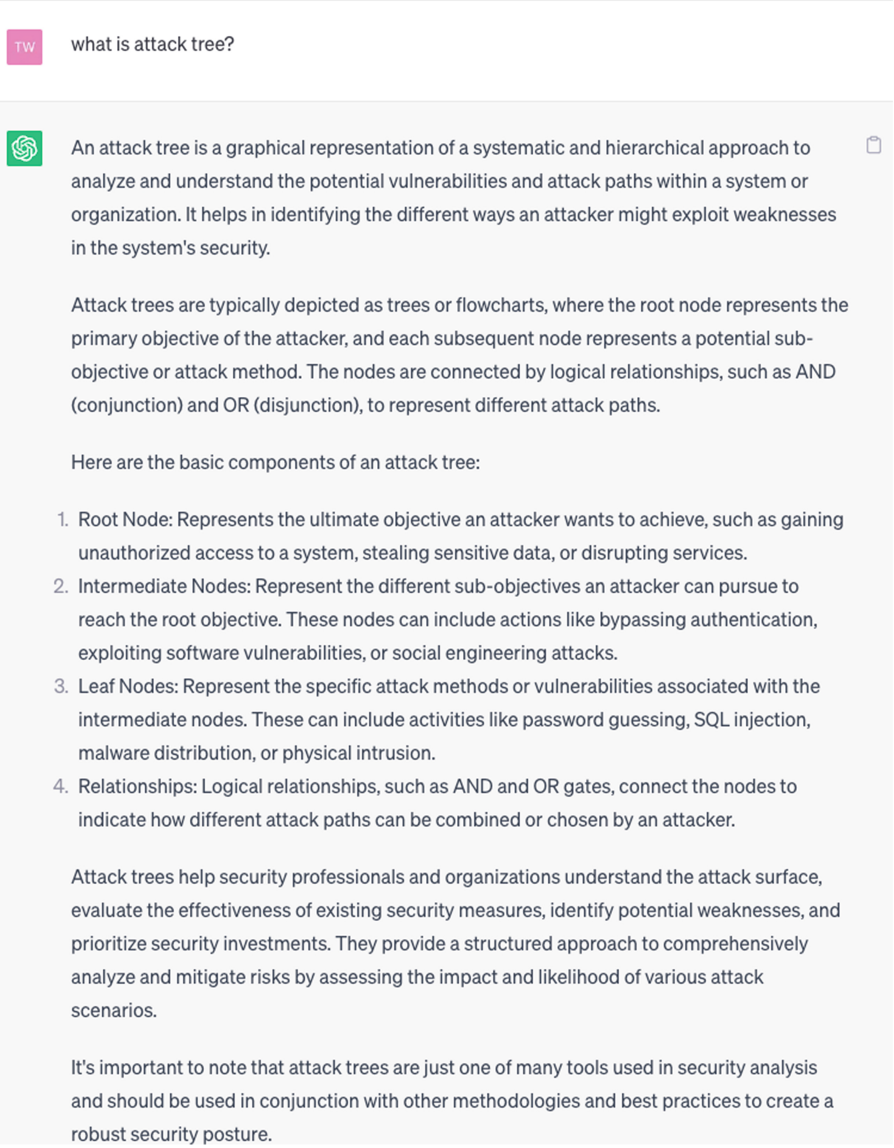
More formally, we define an attack tree following Buldas et al. [5], as we are interested in labels of the intermediate nodes as well.

Definition 1 (Attack tree). *Given a set of labels $\mathcal{L}$, an attack tree t is constructed following the grammar below (where $l \in \mathcal{L}$):*

$$t ::= l \,|\, \mathtt{OR}(t,\ldots,t)_l \,|\, \mathtt{AND}(t,\ldots,t)_l \,|\, \mathtt{SAND}(t,\ldots,t)_l$$

In this definition, we formalize attack trees with three types of *refinement operators*: OR, specifying that at least one of the children nodes needs to be fulfilled by the attacker in order to succeed at the parent node; AND that means that each of the children nodes needs to be fulfilled, but the order of execution is immaterial (can in principle be done in parallel); and finally SAND that denotes that all children nodes need to be fulfilled in the prescribed order in order for the parent attack to succeed. These are so-called SAND-trees introduced by Jhawar et al. [19].

Following Buldas et al. [5] we define the *labels* function to obtain a set of labels of an attack tree, and the $refs$ function to obtain a set of labels of nodes that are refined (i.e. all nodes except the leaf nodes, which are not further refined).



what is attack tree?

An attack tree is a graphical representation of a systematic and hierarchical approach to analyze and understand the potential vulnerabilities and attack paths within a system or organization. It helps in identifying the different ways an attacker might exploit weaknesses in the system's security.

Attack trees are typically depicted as trees or flowcharts, where the root node represents the primary objective of the attacker, and each subsequent node represents a potential sub-objective or attack method. The nodes are connected by logical relationships, such as AND (conjunction) and OR (disjunction), to represent different attack paths.

Here are the basic components of an attack tree:

1. Root Node: Represents the ultimate objective an attacker wants to achieve, such as gaining unauthorized access to a system, stealing sensitive data, or disrupting services.
2. Intermediate Nodes: Represent the different sub-objectives an attacker can pursue to reach the root objective. These nodes can include actions like bypassing authentication, exploiting software vulnerabilities, or social engineering attacks.
3. Leaf Nodes: Represent the specific attack methods or vulnerabilities associated with the intermediate nodes. These can include activities like password guessing, SQL injection, malware distribution, or physical intrusion.
4. Relationships: Logical relationships, such as AND and OR gates, connect the nodes to indicate how different attack paths can be combined or chosen by an attacker.

Attack trees help security professionals and organizations understand the attack surface, evaluate the effectiveness of existing security measures, identify potential weaknesses, and prioritize security investments. They provide a structured approach to comprehensively analyze and mitigate risks by assessing the impact and likelihood of various attack scenarios.

It's important to note that attack trees are just one of many tools used in security analysis and should be used in conjunction with other methodologies and best practices to create a robust security posture.

Fig. 1. Definition of attack trees according to ChatGPT

Definition 2 (Labels function). *Given $\mathcal{T}$ – the universe of attack trees, define labels* : $\mathcal{T} \to \mathcal{P}(\mathcal{L})$ *as follows:*
$labels(t) = \{l\}$ *when* $t \equiv l$;
$labels(t) = \{l\} \cup labels(t_1) \cup \cdots \cup labels(t_n)$ *when* $t \equiv \mathtt{OR}(t_1, \ldots, t_n)_l$, *or* $t \equiv \mathtt{AND}(t_1, \ldots, t_n)_l$, *or* $t \equiv \mathtt{SAND}(t_1, \ldots, t_n)_l$

Definition 3 (Refinements function). *Define* $refs : \mathcal{T} \to \mathcal{P}(\mathcal{L})$ *as follows:*
$refs(t) = \{\emptyset\}$ *when* $t \equiv l$;
$refs(t) = \{l\} \cup refs(t_1) \cup \cdots \cup refs(t_n)$ *when* $t \equiv \texttt{OR}(t_1, \ldots, t_n)_l$, *or* $t \equiv \texttt{AND}(t_1, \ldots, t_n)_l$, *or* $t \equiv \texttt{SAND}(t_1, \ldots, t_n)_l$

3 Properties of Synthesized Attack Trees

It is known that LLMs can produce unreliable answers [32]. It is thus important to verify produced models and assess the quality of attack trees synthesized by LLMs. So, before proceeding to synthesize attack trees, we propose and discuss several properties that will establish the quality of the produced models.

Arguably, there are two aspects that security experts care about regarding the quality of attack tree models:

- *Completeness* refers to the presence of all important subnodes and scenarios. This property is very important for the practical acceptance of the produced attack trees, as, as we mentioned, practitioners are seeking reassurance that the attack trees they use include all relevant attacks.
- *Soundness* refers to the absence of incorrect/misleading nodes and scenarios. This property is especially important given the ability of LLMs to hallucinate, i.e., confidently output completely wrong results. If practitioners know that the synthesized attack trees frequently include incorrect attacks, they might be wary of adopting the LLM-produced models.

These properties establish important practitioner requirements on attack tree models. Yet, it is practically impossible to obtain a fully sound and complete attack tree for any real-life system or scenario, as security is a property that cannot be easily formalized and quantified, and there are always *unknown unknowns*: threats that are not yet known (or of which we do not yet know that they are irrelevant). At the same time, the idea of which threats are relevant is based on the context in which the target system functions and also on the needs of the organization. Thus, attack steps that might seem wrong (unsound) for one organization, could be valid and relevant for another organization. To conclude, we expect that completeness and soundness properties cannot be fully formalized in the context of freely synthesized or manually designed attack trees, but they can be evaluated by human practitioners.

To make the first initial step towards evaluating the completeness and soundness of synthesized attack trees, we use the Jaccard similarity coefficient as a coarse approximation.

Definition 4 (Label similarity of attack trees). *Let* t_1 *be a tree depicting the desired outcome (reference attack tree), e.g., complete translation of a scenario into an attack tree, and* t_2 *be a target attack tree. We define their label similarity as*

$$J(t_1, t_2) = \frac{labels(t_1) \cap labels(t_2)}{labels(t_1) \cup labels(t_2)}$$

We note that it is possible to also attempt to define the completeness and soundness of synthesized attack trees using set difference operators, with the set of all labels included in the reference attack tree but not included in the target attack tree being the measure of completeness, and the set of all labels included in the target tree but not included in the reference tree being the measure of soundness. Yet, we believe these definitions would be meaningful only for substantially large attack trees. As the case studies we have explored in this paper are relatively small, the set difference metrics might not be very informative.

Another (syntactic) property important for the produced attack tree models:

- *Well-formedness*: The synthesized attack trees should be syntactically correct (respect Definition 1), and the refinements should make sense to an expert (e.g., some attack step that clearly defines alternatives should be refined using the OR refinement).

For an attack tree t we define its well-formedness measure W as a percentage of valid refinements: $W(t) = \frac{\text{number of valid refinements}}{|refs(t)|}$. We note here that the number of valid refinements in an attack tree is something that is not formalized in our paper, and it is to be calculated by a human.

We now propose the following validation approach for our method:

- We select two case study scenarios, which have already been analyzed by independent researchers and for which human-designed attack trees exist in the literature.
- We synthesize attack trees for these scenarios using ChatGPT[2].
- We compare human-designed and ChatGPT-synthesized attack trees, using human-designed trees as the ground truth, by applying the defined metrics of label-similarity with the ground truth and the well-formedness function.

4 Attack Tree Synthesis Methodology

We now present the method proposed for synthesizing an attack tree for a chosen scenario in an interaction with ChatGPT.

The first hurdle a security analyst has to overcome is to make ChatGPT to share information on cyber attack options. ChatGPT includes a policy layer that inhibits the generation of sensitive content, including information related to hacking. This policy layer has been shown to be vulnerable to different types of prompts that can *convince* ChatGPT that the intentions of the user are benign and thus enable production of sensitive content [17,21]. Still, we do not consider this aspect to be an issue for the proposed method, as our goal is to **demonstrate that a language model such as ChatGPT includes enough information about cyber attacks and their relationships that can be utilized**

---

[2] We use the free version available at https://chat.openai.com/.

to produce an attack tree. While retrieval of this information using the standard ChatGPT interface might require to use of some *prompt hacks* [16], security analysts can use other LLMs of choice, where they can interact with the model more freely.

We ask ChatGPT to generate an attack tree based on a description of a scenario from the literature. We also ask ChatGPT to take into account a few of the existing works in literature before synthesizing an attack tree given one of the attack goals. We proceed by first presenting the scenarios and then introducing the prompts we have used to synthesize the trees.

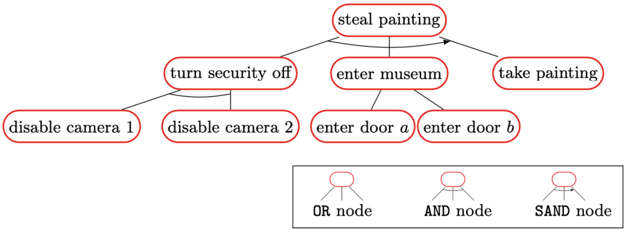


Fig. 2. An attack tree for stealing a painting in a museum, taken from [28]

4.1 Case Study 1

This case study concerns stealing a painting from a museum. The scenario description is taken from [28]: "A museum has two possible entries, both monitored by the same two cameras. The two cameras have a mutual protection system (distinct from the visual surveillance) so that they monitor each other: if a camera gets frozen while being monitored by the other, then an alarm is triggered. In order to neutralize a camera, the attacker can launch a virus on any camera: this virus immediately disables its ability to monitor the other camera, and then, possibly after some time, it freezes the camera. Additionally, the freezing is temporary so that a frozen camera may recover from freezing. Assume at least one camera has been infected by a virus". The reference tree from [28] is in Fig. 2.

4.2 Case Study 2

The second case study concerns tampering with a power meter with the purpose of reducing the electricity bill. We focus on one of the three approaches presented in [10], namely the *do it yourself* approach (we refer the reader to Figs. 1 and 2 in [10] for the reference attack tree): "Consider a fifth-year student of an engineering school, whom we will name Antoine, who is renting an apartment where he needs to pay for the electricity consumption. Antoine would like to lower his electricity bill and he decided to achieve this by reconfiguring the power meter in his apartment. The meter under study is equipped with an optical port that allows a user to connect to the meter using an optical probe. In order to reconfigure his power meter via optical port, Antoine has to have physical access to the power meter and reconfigure it using appropriate software tools. Since the power meter is located in the apartment where Antoine lives, we assume that accessing the power meter is a basic action, i.e., the corresponding node is not refined. In order to reconfigure the power meter with the help of software, we have identified the following scenario that Antoine can follow, taking into account his knowledge, capabilities, and financial profile: The do-it-yourself approach - Antoine reconfigures the meter himself by using unauthorized software tools".

4.3 Prompting

We have used the following prompts to generate an attack tree:

a. Consider scenario X. Could you please depict an attack tree for this approach? Here, X is the scenario description in Sect. 4.1 (a simplified version of this description, as explained in the next subsection) and the description in Sect. 4.2, respectively.
b. (1) Summarize three articles related to attack trees (we opted for a mixture of theoretical aspects, i.e., formal methods, [34] and examples [2,10]). (2) Give an example of an attack tree for stealing a museum painting given the information you learned about attack trees in the 3 articles.

The full conversations for generating an attack tree for the prompts of type *a* can be accessed here[3] and here[4], while for that of *b* can be accessed here[5].

4.4 Results

We present the synthesized attack trees obtained through interacting with ChatGPT. Figure 3 presents the attack tree obtained when given the description in Sect. 4.1. The outcome is clearly nowhere close to the original tree, Fig. 2. Hence, we simplify the descriptive text to the following before generating the tree: "A museum has two possible entries, entry A and entry B. There exist two security cameras, camera 1 and camera 2. Camera 1 supervises entry A and entry B. Camera 2 supervises entry A and entry B." While this description yields a better result, ChatGPT struggles with the concept of an entry being supervised by two cameras. Yet it can adjust if asked, e.g., *Given that entry A is supervised by both camera 1 and camera 2, how would you adjust the attack tree?*

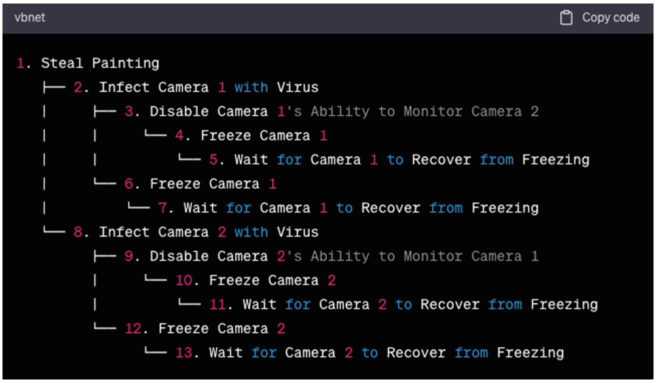
Then, we obtain the tree in Fig. 4, which shares common aspects with the tree in Fig. 2, such as the existence of choice among the entry options, the disabling of both cameras and achieving of all sub-goals of the root node (though not sequentially). Figure 5 presents another attack tree generated by ChatGPT

---

[3] https://chat.openai.com/share/d76dddc5-b560-40c0-8c2d-fcb82f23b60b.
[4] https://chat.openai.com/share/b1e33042-93f4-4df5-988c-453bd123c209.
[5] https://chat.openai.com/share/ce495a3c-1b66-4c11-b4b2-7afc3737ab67.

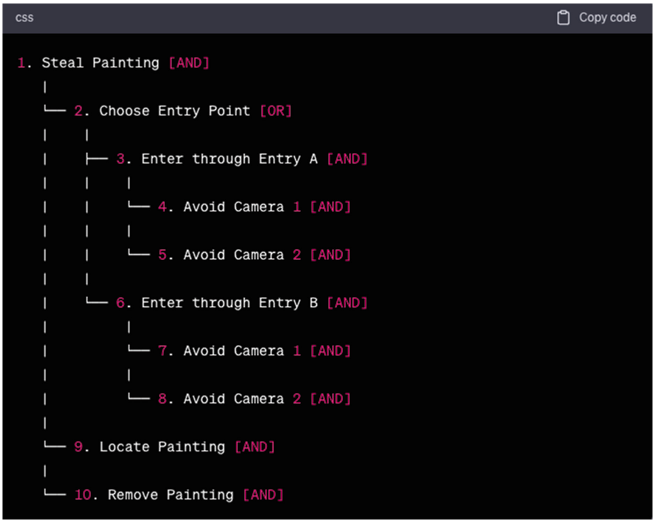
where only the root node (main attacker's goal) was provided. We observe a lot more creativity in providing the sub-goals, and this also indicates its ability to adjust given more details as in *a*.

Figure 6 presents the attack tree obtained when given the description in Sect. 4.2.



```
1. Steal Painting
   ├── 2. Infect Camera 1 with Virus
   |     ├── 3. Disable Camera 1's Ability to Monitor Camera 2
   |     |     └── 4. Freeze Camera 1
   |     |           └── 5. Wait for Camera 1 to Recover from Freezing
   |     └── 6. Freeze Camera 1
   |           └── 7. Wait for Camera 1 to Recover from Freezing
   └── 8. Infect Camera 2 with Virus
         ├── 9. Disable Camera 2's Ability to Monitor Camera 1
         |     └── 10. Freeze Camera 2
         |           └── 11. Wait for Camera 2 to Recover from Freezing
         └── 12. Freeze Camera 2
               └── 13. Wait for Camera 2 to Recover from Freezing
```

Fig. 3. Case study 1, prompt type *a*, full description. Generated AT for stealing a museum painting



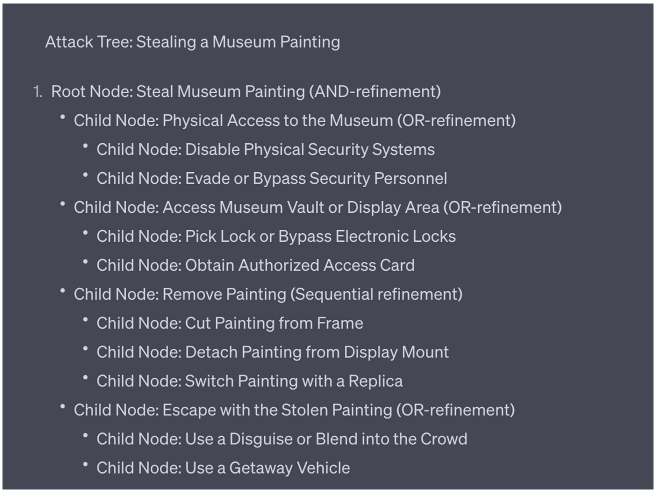
```
1. Steal Painting [AND]
   |
   └── 2. Choose Entry Point [OR]
   |    |
   |    ├── 3. Enter through Entry A [AND]
   |    |    |
   |    |    └── 4. Avoid Camera 1 [AND]
   |    |    |
   |    |    └── 5. Avoid Camera 2 [AND]
   |    |
   |    └── 6. Enter through Entry B [AND]
   |         |
   |         └── 7. Avoid Camera 1 [AND]
   |         |
   |         └── 8. Avoid Camera 2 [AND]
   |
   └── 9. Locate Painting [AND]
   |
   └── 10. Remove Painting [AND]
```

Fig. 4. Case study 1, prompt type *a*, simplified description. Generated AT for stealing a museum painting

4.5 Evaluation

Table 1 presents the summary of evaluation results based on the label similarity metrics and the well-formedness function. Evaluation of well-formedness (which are valid refinements) is based on the judgment of the authors.

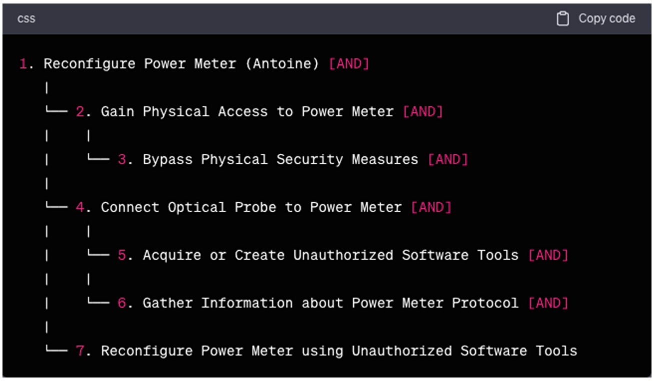
From the table, we can observe that the synthesized attack trees do contain similar attack scenarios to the reference trees, yet they also contain other attacks. We note that the target trees do not contain impossible or irrelevant (unsound) attacks. The well-formedness of the synthesized attack trees is mostly acceptable, but not perfect. The painting attack contains two AND nodes (3 and 6 in Fig. 4) that are refined wrongly to a human analyst (entering via some entry is not achieved by avoiding two cameras), yet this is a common pattern of thought in



Attack Tree: Stealing a Museum Painting

1. Root Node: Steal Museum Painting (AND-refinement)
    - Child Node: Physical Access to the Museum (OR-refinement)
        - Child Node: Disable Physical Security Systems
        - Child Node: Evade or Bypass Security Personnel
    - Child Node: Access Museum Vault or Display Area (OR-refinement)
        - Child Node: Pick Lock or Bypass Electronic Locks
        - Child Node: Obtain Authorized Access Card
    - Child Node: Remove Painting (Sequential refinement)
        - Child Node: Cut Painting from Frame
        - Child Node: Detach Painting from Display Mount
        - Child Node: Switch Painting with a Replica
    - Child Node: Escape with the Stolen Painting (OR-refinement)
        - Child Node: Use a Disguise or Blend into the Crowd
        - Child Node: Use a Getaway Vehicle

Fig. 5. Case study 1, prompt type *b*. Generated AT for stealing a museum painting



```
1. Reconfigure Power Meter (Antoine) [AND]
   |
   └── 2. Gain Physical Access to Power Meter [AND]
   |    |
   |    └── 3. Bypass Physical Security Measures [AND]
   |
   └── 4. Connect Optical Probe to Power Meter [AND]
   |    |
   |    └── 5. Acquire or Create Unauthorized Software Tools [AND]
   |    |
   |    └── 6. Gather Information about Power Meter Protocol [AND]
   |
   └── 7. Reconfigure Power Meter using Unauthorized Software Tools
```

Fig. 6. Case study 2, prompt type *a*. Generated AT for tampering the power meter

attack tree design by novices (as observed by one of the authors who teaches attack trees to students). Another common (for humans) mistake can be observed in Fig. 6, where node 2 is AND-refined, but only has one child (node 3).

5 Discussion

We have found that ChatGPT, an advanced LLM, has (limited) capacity to synthesize meaningful attack trees with correct refinement operators, both when presented with a scenario to be analyzed and when presented with just the main goal of the attack. From our experimentation, the main added benefit of ChatGPT is its capability to serve as an accessible knowledge base. It is easy for practitioners to interact with ChatGPT and to query it interactively regarding possible attack options.

Table 1. Analysis of the synthesized trees

Target tree (reference tree)	Label similarity	Well-formedness
Figure 4 (the reference tree is Fig. 2)	7/9	2/4
Figure 5 (no reference tree)	N/A	5/5
Figure 6 (the reference tree is in Figures 1&2 in [10]	5/12	2/3

We have observed that when high-level attack scenarios are requested from ChatGPT, they are all relatively plausible, and ChatGPT does an excellent job in positioning attack scenario components along the MITRE ATT&CK framework[6]. This framework is one of the standards in the security industry nowadays, meaning that there are a lot of reports for LLMs to learn from, but also that the produced security models, including attack trees, are immediately comprehensible and relevant to practitioners as they use the same language.

Limitations. Yet, we have observed several limitations of ChatGPT regarding synthesizing attack tree models. First, ChatGPT is not able to reason well with the refinement operators. While it is perfectly able to recite their definitions, the refinement operators it proposes are frequently wrong (using AND instead of OR and vice-versa; not using any refinement operators at all; frequently changing between refinement types in subsequent answers, especially if questioned about the reasoning behind the operator choice; resistance in assigning a refinement operator to the root node[7]). This is understandable as, while having reasonable logical reasoning capabilities [24], ChatGPT is a language model and not a logical reasoning tool. However, it is plausible to expect that in the near future, more

---

[6] https://attack.mitre.org/.

[7] In many synthesized attack trees the root node does not have a refinement assigned. However, we were able to interact with ChatGPT in such a way that the root refinement operator was ultimately provided, by asking, e.g. *Which of the children of the root node shall be achieved for a successful attack?*.

advanced AI systems will be developed that will combine large language models and logical reasoning capabilities. Such advanced AI systems will be quite capable of synthesizing well-formed attack trees.

While the observed issues with the refinement operators seem to be the most important challenge related to logical reasoning and the ability to synthesize useful attack trees, this lack of logical reasoning capabilities also translates into the limited ability of ChatGPT to reason about new complex scenarios (logical reasoning based on natural language comprehension), e.g., if asked to produce an attack tree for a given system description. Again, this is something that has already been observed by other researchers [24] in other application domains.

The second limitation is the hallucinations of LLMs when they produce a plausibly-sounding yet wrong output. As we mentioned earlier, here a lot depends on the context. For high-level attack scenarios, the produced attack trees are often plausible, because they integrate the typical security worries of any organization: authentication bypass, social engineering, web application hacking, and so on. Yet, the more detailed output we request from ChatGPT, the more mistakes can be uncovered, for example, related to its incorrect interpretation of the given scenario. We foresee that the more detailed and organization- and system-specific scenario is to be considered, the more expertise will be required from the ChatGPT interlocutor to interpret the output and correct it when necessary. Indeed, ChatGPT is capable of correcting itself with the right prompting, but the analyst needs to be confident in their own knowledge, as self-correction of ChatGPT can also be a hallucination (an incorrect modification stemming from the request of the interlocutor to reconsider).

The third limitation is the lack of ways to check how complete is the ChatGPT-produced attack tree. We note that this is a problem that is common for manually designed attack trees, where the creative process relies on the expertise of one or more security analysts. Attack trees generated from system models can be shown to be complete with respect to the underlying model [27,34]. It would be interesting to see if advances in AI and LLMs in particular would allow to achieve some guarantees on the completeness of the output with respect to the data consumed by the model in training.

There are also limitations of our study itself that we need to consider. We only considered two case studies, and both are relatively small. While we believe that our examples are illustrative and allow to draw conclusions, in the future more extensive experimentation is needed. Another limitation of our work is that the proposed quality metrics are not perfect: the label similarity metric is only an approximation of the soundness and completeness that we are after, and the well-formedness metric relies on the judgment of a human analyst. We believe that these issues reflect the nature of attack trees as produced by human analysts: another human analyst is required to assess their quality. As another limitation related to the metrics, we have not explored whether these metrics are relatively stable or not, and how will they be affected if ChatGPT regenerates an attack tree after a new query. It would be interesting to investigate this in subsequent, more extensive studies. Finally, we note that there are other ways to validate synthesized attack trees. For example, in the future one can organize focus groups or interviews with independent cybersecurity experts to manually review the synthesized attack trees.

Future Research Directions. We believe that our experiments open up new interesting research directions in the attack tree theory and application domain. For example, we have engaged with it to augment existing attack trees. This is one of the problems investigated by library-based attack tree synthesis approaches [20]: for an existing high-level scenario captured as a basic attack tree, the task is to find the relevant vulnerabilities that the attacker can exploit to achieve the basic goals. We note that this is something that ChatGPT is also familiar with (see the explanation of leaf nodes in Fig. 1). In our experiments (not reported in this paper for the lack of space) we saw that, while ChatGPT is able to augment a given tree with more nodes containing vulnerabilities to be exploited, this augmentation is far from ideal, as the proposed vulnerabilities target different systems (Windows, Linux, etc.), and not all of them seem to be quite relevant. At the same time, it is feasible to implement a natural-language processing tool that will verify the output from ChatGPT based on the information available in NVD[8], a database containing detailed information about known vulnerabilities, and other public resources, so, overall, this is a promising research direction.

There are also interesting research questions concerning applying the semantics of attack trees to measure how close reference and target attack trees are. We currently use the label similarity metric that does not consider the refinement structure similarity. More fine-grained similarity metrics based on attack tree semantics [19,25] will allow to estimate completeness and soundness of synthesized attack trees in a better way. Finally, as we mentioned in Sect. 4.4, synthesized attack trees sometimes include incorrect refinements, very similar to mistakes that novices make when applying attack trees. It would be interesting to investigate how these errors come to be in the trained LLMs.

Ethics. Our research has been approved by the relevant Ethics Review Committee of our institution (ref. 2023-013). We note that is a standard practice in the security community to work on identifying security threats and discussing them openly, including by publishing attack trees representing threat scenarios (as our case studies demonstrate). We thus consider that our research does not violate the principles of ethical research. As we noted earlier, ChatGPT includes a Moderation API (the policy layer) that inhibits the generation of sensitive content, including information related to hacking. Our experiments and previous research results [16,17,21] show that the policy layer is vulnerable to different prompt attacks, and thus attack-related information can still be produced. We must note that the OpenAI Usage Policy[9] does not prohibit using ChatGPT for producing cyber attack-related information. It only forbids the generation of malicious software (code), which our study does not do.

---

[8] https://nvd.nist.gov/.

[9] https://openai.com/policies/usage-policies Accessed on June 15, 2023.

6 Conclusions

In this study, we have investigated the capability of an advanced LLM such as ChatGPT to synthesize attack trees, a graphical security model widely used in practice to represent complex attack scenarios. We have also discussed the properties expected by practitioners for the adoption of ChatGPT-synthesized attack trees. We have proposed a way to assess the quality of the synthesized attack trees by comparing them with available manually designed attack trees and by assessing their well-formedness.

Our study demonstrates that ChatGPT is capable of producing relevant and meaningful attack tree models, yet it currently lacks logical reasoning to be able to judge the applicable refinement operators and to elicit attack trees closely corresponding to complex security scenarios. However, the outcomes are, overall, promising, and ChatGPT is clearly able to assist human analysts in brainstorming about different attack scenarios. With the advent of advanced AI systems, we can expect that AI-synthesized security models (such as attack trees) can become the accepted norm in organizations which will help them to prepare the best cyber defenses.

References

1. Amoroso, E.G.: Fundamentals of Computer Security Technology. Prentice-Hall, Inc., Hoboken (1994)
2. Bagnato, A., Kordy, B., Meland, P.H., Schweitzer, P.: Attribute decoration of attack-defense trees. Int. J. Secur. Softw. Eng. **3**(2), 1–35 (2012). https://doi.org/10.4018/jsse.2012040101
3. Barke, S., James, M.B., Polikarpova, N.: Grounded Copilot: how programmers interact with code-generating models. Proc. OOPSLA **7**(OOPSLA1), 85–111 (2023)
4. Bryans, J., Liew, L.S., Nguyen, H.N., Sabaliauskaite, G., Shaikh, S., Zhou, F.: A template-based method for the generation of attack trees. In: Laurent, M., Giannetsos, T. (eds.) WISTP 2019. LNCS, vol. 12024, pp. 155–165. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-41702-4_10
5. Buldas, A., Gadyatskaya, O., Lenin, A., Mauw, S., Trujillo-Rasua, R.: Attribute evaluation on attack trees with incomplete information. Comput. Secur. **88**, 101630 (2020)
6. Carlini, N., et al.: Extracting training data from large language models. In: USENIX Security Symposium, vol. 6 (2021)
7. Charan, P., Chunduri, H., Anand, P.M., Shukla, S.K.: From text to MITRE techniques: exploring the malicious use of large language models for generating cyber attack payloads. arXiv preprint arXiv:2305.15336 (2023)
8. Chiang, W.L., et al.: Vicuna: an open-source chatbot impressing GPT-4 with 90%* ChatGPT quality (2023). https://vicuna.lmsys.org. Accessed 14 Apr 2023
9. Derner, E., Batistič, K.: Beyond the safeguards: exploring the security risks of ChatGPT. arXiv preprint arXiv:2305.08005 (2023)
10. Fila, B., Wideł, W.: Attack–defense trees for abusing optical power meters: a case study and the OSEAD tool experience report. In: Albanese, M., Horne, R., Probst, C.W. (eds.) GraMSec 2019. LNCS, vol. 11720, pp. 95–125. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36537-0_6

11. Fraile, M., Ford, M., Gadyatskaya, O., Kumar, R., Stoelinga, M., Trujillo-Rasua, R.: Using attack-defense trees to analyze threats and countermeasures in an ATM: a case study. In: Horkoff, J., Jeusfeld, M.A., Persson, A. (eds.) PoEM 2016. LNBIP, vol. 267, pp. 326–334. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48393-1_24
12. Gadyatskaya, O.: How to generate security cameras: towards defence generation for socio-technical systems. In: Mauw, S., Kordy, B., Jajodia, S. (eds.) GraMSec 2015. LNCS, vol. 9390, pp. 50–65. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29968-6_4
13. Gadyatskaya, O., Jhawar, R., Mauw, S., Trujillo-Rasua, R., Willemse, T.A.C.: Refinement-aware generation of attack trees. In: Livraga, G., Mitchell, C. (eds.) STM 2017. LNCS, vol. 10547, pp. 164–179. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68063-7_11
14. Gadyatskaya, O., Trujillo-Rasua, R.: New directions in attack tree research: catching up with industrial needs. In: Liu, P., Mauw, S., Stølen, K. (eds.) GraMSec 2017. LNCS, vol. 10744, pp. 115–126. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74860-3_9
15. Gritzalis, D., Iseppi, G., Mylonas, A., Stavrou, V.: Exiting the risk assessment maze: a meta-survey. ACM Comput. Surv. (CSUR) **51**(1), 1–30 (2018)
16. Gupta, M., Akiri, C., Aryal, K., Parker, E., Praharaj, L.: From ChatGPT to ThreatGPT: Impact of generative AI in cybersecurity and privacy. arXiv preprint arXiv:2307.00691 (2023)
17. Huang, X., et al.: A survey of safety and trustworthiness of large language models through the lens of verification and validation. arXiv preprint arXiv:2305.11391 (2023)
18. Ivanova, M.G., Probst, C.W., Hansen, R.R., Kammüller, F.: Attack tree generation by policy invalidation. In: Akram, R.N., Jajodia, S. (eds.) WISTP 2015. LNCS, vol. 9311, pp. 249–259. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24018-3_16
19. Jhawar, R., Kordy, B., Mauw, S., Radomirović, S., Trujillo-Rasua, R.: Attack trees with sequential conjunction. In: Federrath, H., Gollmann, D. (eds.) SEC 2015. IAICT, vol. 455, pp. 339–353. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18467-8_23
20. Jhawar, R., Lounis, K., Mauw, S., Ramírez-Cruz, Y.: Semi-automatically augmenting attack trees using an annotated attack tree library. In: Katsikas, S.K., Alcaraz, C. (eds.) STM 2018. LNCS, vol. 11091, pp. 85–101. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01141-3_6
21. Kang, D., Li, X., Stoica, I., Guestrin, C., Zaharia, M., Hashimoto, T.: Exploiting programmatic behavior of LLMs: dual-use through standard security attacks. arXiv preprint arXiv:2302.05733 (2023)
22. Katz, D.M., Bommarito, M.J., Gao, S., Arredondo, P.: GPT-4 passes the bar exam. Available at SSRN 4389233 (2023)
23. Kordy, B., Mauw, S., Radomirovic, S., Schweitzer, P.: Attack-defense trees. J. Log. Comput. **24**(1), 55–87 (2014)
24. Liu, H., Ning, R., Teng, Z., Liu, J., Zhou, Q., Zhang, Y.: Evaluating the logical reasoning ability of ChatGPT and GPT-4. arXiv preprint arXiv:2304.03439 (2023)
25. Mauw, S., Oostdijk, M.: Foundations of attack trees. In: Won, D.H., Kim, S. (eds.) ICISC 2005. LNCS, vol. 3935, pp. 186–198. Springer, Heidelberg (2006). https://doi.org/10.1007/11734727_17
26. OpenAI: GPT-4 technical report (2023)

27. Pinchinat, S., Acher, M., Vojtisek, D.: ATSyRa: an integrated environment for synthesizing attack trees. In: Mauw, S., Kordy, B., Jajodia, S. (eds.) GraMSec 2015. LNCS, vol. 9390, pp. 97–101. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29968-6_7
28. Pinchinat, S., Schwarzentruber, F., Lê Cong, S.: Library-based attack tree synthesis. In: Eades III, H., Gadyatskaya, O. (eds.) GraMSec 2020. LNCS, vol. 12419, pp. 24–44. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62230-5_2
29. Sallam, M.: ChatGPT utility in healthcare education, research, and practice: systematic review on the promising perspectives and valid concerns. Healthcare **11**(6), 887 (2023)
30. Scao, T.L., et al.: Bloom: a 176B-parameter open-access multilingual language model. arXiv preprint arXiv:2211.05100 (2022)
31. Schneier, B.: Attack trees: modeling security threats. Dr. Dobb's J. Softw. Tools **24**(12), 21–29 (1999)
32. Shen, X., Chen, Z., Backes, M., Zhang, Y.. In ChatGPT we trust? Measuring and characterizing the reliability of ChatGPT. arXiv preprint arXiv:2304.08979 (2023)
33. Vigo, R., Nielson, F., Nielson, H.R.: Automated generation of attack trees. In: Proceedings of CSF 2014, pp. 337–350. IEEE (2014)
34. Wideł, W., Audinot, M., Fila, B., Pinchinat, S.: Beyond 2014: formal methods for attack tree-based security modeling. ACM Comput. Surv. (CSUR) **52**(4), 1–36 (2019)
35. Xiong, W., Lagerström, R.: Threat modeling - a systematic literature review. Comput. Secur. **84**, 53–69 (2019)
36. Zhang, X., Zhang, Z., Ji, S., Wang, T.: Trojaning language models for fun and profit. In: Proceedings of EuroS&P, pp. 179–197. IEEE (2021)

BERT Fine-Tuning the Covid-19 Open Research Dataset for Named Entity Recognition

Shin Thant[1](✉), Teeradaj Racharak[2], and Frederic Andres[3]

[1] Asian Institute of Technology, Khlong Nueng, Thailand
`shinthant@alumini.ait.asia`
[2] School of Information Science, Japan Advanced Institute of Science and Technology, Nomi, Japan
`racharak@jaist.ac.jp`
[3] National Institute of Informatics, Tokyo, Japan
`andres@nii.ac.jp`

Abstract. This study employs the widely used Large Language Model (LLM), BERT, to implement Named Entity Recognition (NER) on the CORD-19 biomedical literature corpus. By fine-tuning the pre-trained BERT on the CORD-NER dataset, the model gains the ability to comprehend the context and semantics of biomedical named entities. The refined model is then utilized on the CORD-19 to extract more contextually relevant and updated named entities. However, fine-tuning large datasets with LLMs poses a challenge. To counter this, two distinct sampling methodologies are proposed to apply on each dataset. First, for the NER task on the CORD-19, a Latent Dirichlet Allocation (LDA) topic modeling technique is employed. This maintains the sentence structure while concentrating on related content. Second, a straightforward greedy method is deployed to gather the most informative data of 25 entity types from the CORD-NER dataset. The study realizes its goals by demonstrating the content comprehension capability of BERT-based models without the necessity of supercomputers, and converting the document-level corpus into a source for NER data, enhancing data accessibility. The outcomes of this research can shed light on the potential progression of more sophisticated NLP applications across various sectors, including knowledge graph creation, ontology learning, and conversational AI.

Keywords: Large language models · BERT · NER · Document level entity extraction · CORD-19 · CORD-NER · Dataset sampling

1 Introduction

The advent of Large Language Models (LLMs) such as BERT and GPT has heralded a new era in the field of Natural Language Processing (NLP), with groundbreaking achievements demonstrated in tasks such as Named Entity Recognition (NER) [1,2], text classification [3], text summarization [4], and sentiment analysis [5]. Our research pivots on the most consequential biomedical subject of

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 261–275, 2023.
https://doi.org/10.1007/978-981-99-7969-1_19

recent times, Covid-19 The challenges presented by the CORD-19 dataset [14], a considerable corpus with a complex document-level structure, can often impede direct access. However, various innovative information systems have been devised to address this, capable of transmuting literature into data that machines can interpret. These include sophisticated methods such as Named Entity Recognition (NER) [6], knowledge graph (KG) [7], and text summarization [4]. The effective implementation of such systems can help democratize access to crucial information, thereby accelerating scientific advancements and improving public health response strategies.

Our research is primarily focused on the pivotal problem of Named Entity Recognition (NER), a fundamental task in Natural Language Processing (NLP), and employs the Distil-BERT model in its approach. The cruciality of NER in NLP is not to be understated, given its applicability across a wide spectrum of advanced applications, encompassing areas like Knowledge Graphs and Ontology. Hence, our research emphasis is clearly set on tackling the NER problem, thereby advancing the state-of-the-art in this essential NLP task.

In this study, we employ CORD-NER [6], an earlier CORD-19 dataset of named entities, to fine-tune BERT. Subsequently, this refined model is applied to extract named entities from the corpus of CORD-19 literature. Nevertheless, the significant computational resources needed to fine-tune BERT on vast datasets can be prohibitive, particularly for researchers with limited computational capabilities. To mitigate this challenge, we implement two distinct sampling techniques on each dataset. We commence with the greedy selection of data subsets from 25 entities of the CORD-NER dataset, followed by the fine-tuning of this prepared NER dataset with Distil-BERT. The second part of the study effectively samples the CORD-19 dataset using a topic modeling method, namely, Latent Dirichlet Allocation (LDA) [8]. We then generate a comprehensive NER dataset by applying the fine-tuned model to the sampled CORD-19 dataset. The resultant named entity dataset has the potential to be extended for other advanced applications.

In sum, this study aims to tackle the Named Entity Recognition problem for the COVID-19 open research dataset. We achieve this goal by fine-tuning BERT on the CORD-NER dataset to better comprehend the context and enhance with the biomedical semantics of named entities. We investigate the effectiveness of different sampling techniques for fine-tuning BERT. Our research results provide a valuable resource for NER on the CORD-19 document-level corpus.

2 Related Work

2.1 Bidirectional Encoder Representations from Transformers

Large language models (LLMs) have revolutionized natural language processing tasks. Bidirectional Encoder Representations from Transformers (BERT) [9] is one such LLM that has achieved state-of-the-art performance on several natural language processing tasks including NER [1,2], and text summarization [4].

BERT is a transformer-based model that uses a pre-trained language representation to generate context-aware representations of input text. Fine-tuning BERT on specific NER datasets has shown promising results [1,2]. The study applies BERT on the CORD-19 literature corpus without supercomputers, which could have significant implications for natural language processing research.

2.2 Document-Level Entity Extraction

Document-level entity extraction has been a topic of interest in the natural language processing community for several years [6,10,11]. Named entity recognition (NER) is the task of extracting entities such as people, organizations, and locations from a given text. Several approaches have been proposed for NER, including rule-based, statistical, and machine-learning methods [12,13]. With the emergence of large-scale language models like BERT, there has been renewed interest in using these models for NER tasks [1,2].

Named Entity Recognition is a sub-task in the construction of a knowledge graph. Giles et al. [10] applied a TERMite commercial NER engine to recognize the respective entity types of the entities obtained from the CORD-19 dataset. X. Wang et al. [6] created a NER dataset, called CORD-NER, with 75 entity types based on the earliest version of the CORD-19 dataset. This CORD-NER dataset is then used as training data for entity recognition in this study. Wu et al., 2020 [11] tried to extract the entities of the 'PERSON' and 'ORG' entity types in the CORD-19 dataset using Python built-in library named 'Stanza'.

2.3 Dataset Sampling

Dataset sampling is an important aspect of NER tasks, especially when dealing with large corpora such as CORD-19 [14]. Various approaches have been utilized for dataset sampling [15], including random sampling, stratified sampling, and cluster-based sampling. In this research, two different sampling approaches are used for CORD-19 and CORD-NER. For CORD-19, a Latent Dirichlet allocation algorithm is used to focus on closely related contents, whereas, a simple greedy approach is used in case of CORD-NER to collect the most informative data of 25 entity types. These sampling techniques are designed to make the NER process more efficient and effective, especially when dealing with large datasets.

Researchers also tried to cluster the literatures based on their content similarity. A sort of statistical modeling called topic modeling is used to identify the abstract topics that appear in a corpus of documents. Latent Dirichlet Allocation (LDA) is commonly used to build a topic model from titles and abstracts of the research corpus [16]. Different versions of the pre-trained model BERT can also be applied in paper clustering [4].

3 Methodology

Our methodology consists of two main stages. fine-tuning the pre-trained BERT model on the NER dataset (in Sect. 3.1) and extracting named entities from

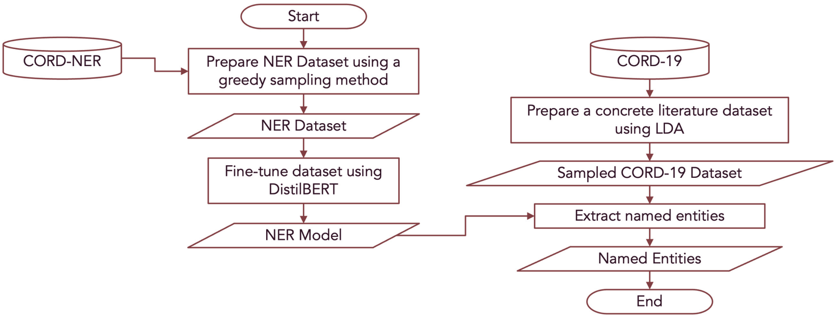

Start
CORD-NER
Prepare NER Dataset using a greedy sampling method
NER Dataset
Fine-tune dataset using DistilBERT
NER Model
CORD-19
Prepare a concrete literature dataset using LDA
Sampled CORD-19 Dataset
Extract named entities
Named Entities
End

Fig. 1. Workflow of the proposed system.

Table 1. Examples of some annotated entities in CORD-NER

CORONAVIRUS	SUBSTRATE	IMMUNE_ RESPONSE	SIGN_OR_ SYMPTOM
sars-cov-2	blood	immune cells	cough
cov	saliva	innate immune	diarrhoea
covid-19	urine	immunization	wheezing

the Covid-19-related literatures using the fine-tuned model (in Sect. 3.2). Each section involves the dataset preparation step (sampling). The detailed methodology is discussed in the next sections. The overview workflow of the proposed system is illustrated in Fig. 1.

3.1 Named Entity Recognition

This section discusses the details of the NER dataset, training model, and sampling method. The NER process is performed by fine-tuning the BERT model on the selective types of the most informative entities of the CORD-NER dataset.

NER Dataset. The CORD-NER is defined based on the earliest version of the CORD-19 dataset published in March 2020. The CORD-19 dataset which is used as the main corpus in this research was published in March 2022. CORD-NER has 75 fine-grained entity types. Examples of some annotated entities in CORD-NER are illustrated in Table 1.

Deep Learning Model for NER. There are different methodologies that can provide a well-trained NER model, such as CNN, LSTM, Bi-LSTM, GPT-n, and BERT. Based on our preliminary study, we have decided to employ BERT in this study. The fine-tuning process requires huge processing power and takes time since the NER dataset is big and the BERT model has millions of parameters. We discuss our strategy to handle this problem in the next section.

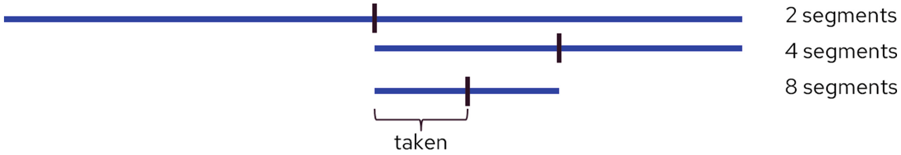


Fig. 2. Illustration of segmenting CORD-NER into 8 parts

Dataset Sampling. Since the CORD-NER dataset is big, it will require a huge processing power to fine-tune all data. Therefore, a subset of the dataset is used to fine-tune the BERT transformer. We propose to experiment with two approaches to dataset sampling for CORD-NER in this study.

Sampling Method - 1. The first and simple attempt is to segment the dataset into specific parts (2, 4, 8, 16, ..., 256) and take the part with the most information (entity counts for each type). An example segmentation work is illustrated in Fig. 2. The idea behind this sampling method can be described as follows:

1. Segmentation: the CORD-NER dataset is segmented into two equal parts.
2. Selection: the statistical information (number of documents, number of sentences, number of entity types, overall number of entities) on segmented parts are compared. And the segment that contains more information is selected.
3. Segmentation: then, the selected segment is divided into half again. And followed by Selection as in step 2.
4. This process is carried out until a specified segment threshold is reached.

Sampling Method - 2. The CORD-NER has 75 entity types: 10 covid-19 related types (e.g. Coronavirus, Material, Immune response), 18 Scispacy types (e.g. Loc, Cardinal, Quantity) [18], and 47 UMLS types (e.g. Cell, Bacterium, Social Behavior) [17]. To handle the unbalance distribution of entities for each type and sample a compact NER dataset, 15 biomedical-related types and 10 general types are selected manually. The selected types along with respective categories are shown in Fig. 3. Out of 25 types, the 5 Spacy types are written in abbreviations. FAC type represents for buildings, airports, highways, bridges, etc. GPE represents countries, cities, and states. LOC includes Non-GPE locations, mountain ranges, and bodies of water. Nationalities, religious, or political groups are in NORP. ORG includes companies, agencies, institutions, etc.

The balanced dataset for the selected 25 types is performed in the following procedure: (The process is illustrated with 15 entity types in Table 2.)

1. Given the CORD-NER dataset, we first filter sentences that only include selected 25 types. If we utilize the sentences that also have ignored types, we have to delete those types and this might affect the meaning and structure of sentences (cf. the ‘Selected Types’ column in Table 2).
2. Sort entities count of the filtered corpus in descending order.

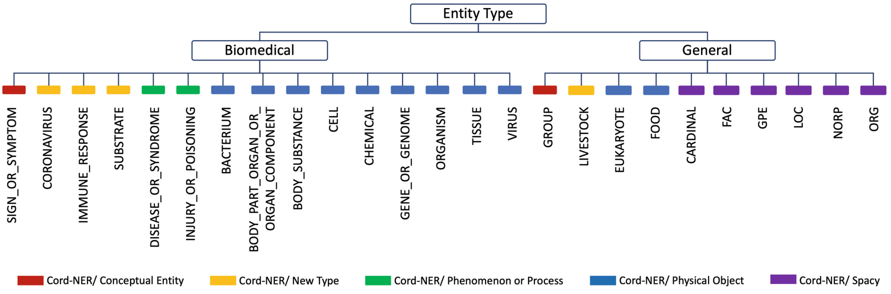


Fig. 3. Categorization of 25 selected Entity Types

3. Cut off the smallest N types to take all and get all sentences involving selected types. The smallest five types are marked as base types and all the entities of these types will be taken (cf. the ‘Smallest types’ column in Table 2).
4. Count entities in the cut-off dataset. Sort again as in step 2.
5. Fill entities with lower counts by adding more sentences that contain those types. [’added_1’ column].
6. Continue steps 4 and 5 until the desired entity count is satisfied. In the current case, the desired entity range is [2000, 20000] and it is satisfied at the added_1 stage.

3.2 Entity Extraction

The fine-tuned model is applied to the corpus of biomedical literatures, CORD-19. The CORD-19 dataset is sampled by using a topic modeling method, LDA, to reduce the size of a big corpus while maintaining sentence structure in a co-related topic. Then, entities and respective types are extracted. The NER process is finalized by a simple post-processing step.

Sampling on the CORD-19 Dataset. The CORD-19 dataset [14] used in the proposed system was published on 5 March 2022. This dataset is composed of 902,589 articles and encompasses various information like ‘source’, ‘title’, ‘doi’, ‘abstract’, ‘publish_time’, ‘authors’, ‘journal’, ‘pdf_json_files’. For the purposes of our system, only the ‘title’, ‘abstract’, and ‘publish time’ will be utilized. To determine dominant topics within the extracted literature, the standard Latent Dirichlet Allocation (LDA) approach [8] is employed. LDA is a generative probabilistic model typically used for collections of discrete data, such as text corpora, wherein each word is considered a blend of an underlying set of topics, and each topic is a blend of a set of topic probabilities. We then select one of the most intriguing topics, and the papers related to this topic form the dataset for the proposed system.

Table 2. Illustration of Balancing NER Dataset

Entity Type	All	Selected Types	Smallest types	Added_1
INJURY_OR_POISONING	3738	1839	1839	1839
BACTERIUM	8360	4839	4839	4839
FAC	10295	4251	4251	4251
FOOD	11036	4994	4994	4994
LOC	17186	8246	8246	8246
SIGN_0R_SYMPTOM	24386	13375	172	2811
BODY_SUBSTANCE	35133	15411	238	3127
SUBSTRATE	40037	18642	378	3138
NORP	41078	19405	519	3277
IMMUNE_RESPONSE	50361	27375	477	3281
LIVESTOCK	59597	29933	1087	1648
BODY_PART_ORGAN_OR _ORGAN_COMPONENT	85111	45502	997	3899
VIRUS	100482	47697	932	3546
CORONAVIRUS	104440	48031	647	3357
TISSUE	113554	57972	1192	2418

NER Post-Processing. Once the fine-tuning process is completed, the model is employed to classify abstracts from the refined corpus. The model's output will be an entity and its associated type. Two prefixes (B-, I-) are used for each recognized type, where (B-) denotes the start of an entity and (I-) indicates a subsequent entity. The use of such prefixes allows the identification of entities composed of multiple words. Subsequently, post-processing is performed on these recognized entities and types to yield finalized and clearly named entities. The principles behind the NER post-processing are illustrated in Fig. 4. The (I-) prefix cannot exist independently; it is only accepted if it matches the type of its preceding entity, otherwise, the identified (I-) entity is omitted. All entities with (B-) prefixes are considered. If a (B-) prefix is already matched with an (I-), it is not considered as a standalone word. If not, the (B-) entity is recognized as a single word of its corresponding type.

4 Result and Discussion

4.1 Named Entity Recognition

Named Entity Recognition (NER) is the process of extracting entities along with their respective types from the input sentence. For example, for the sentence "Pfizer effective reducing incidence severe SARS-COV 2 hypoxia critical illness death", NER will provide some pairs of entities and types such as (Pfizer, vaccine), (SARS-COV 2, covid-19) and (illness, disease). To perform this process, the BERT model is fine-tuned on the existing CORD-NER dataset.

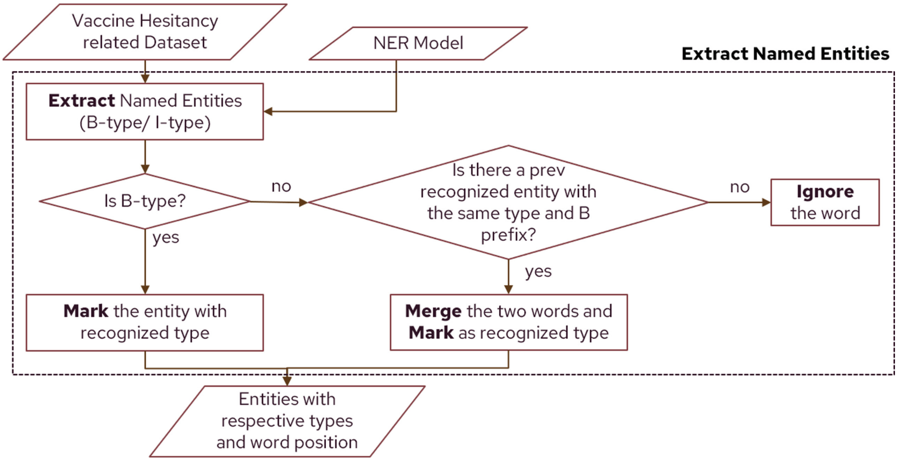

Vaccine Hesitancy related Dataset
NER Model
Extract Named Entities
Extract Named Entities (B-type/ I-type)
Is B-type?
no
yes
Is there a prev recognized entity with the same type and B prefix?
no
yes
Ignore the word
Mark the entity with recognized type
Merge the two words and Mark as recognized type
Entities with respective types and word position

Fig. 4. Workflow diagram for post-processing of recognized named entities.

NER Dataset Preparation. The BERT model will recognize word by word. In order to keep two or more words that can be merged to be an entity, we have to add some labels (B- as start of an entity, I- as a subsequent entity). Therefore, the CORD-NER Dataset is reformatted to a suitable format for the fine-tuning of the BERT model, as depicted in Table 3. The Sampling Method-1, detailed in Sect. 3.1, is applied to the CORD-NER Dataset. DistilBERT is fine-tuned using two segmentation sizes: 512 and 32, and with a single epoch. The associated performance metrics and runtime are documented in Table 4. However, this approach possesses two key limitations:

1. Given that segmentation is conducted on sequential data, potential bias towards similar domain topics might arise.
2. The selection procedure employs a greedy algorithm, implying that the predominant types are determined solely based on a segmented portion, not on the entire dataset.

To address these shortcomings, we propose to employ our Sampling Method-2. This approach places emphasis on the entity types, in contrast to the count-centric focus of the first method. Within the CORD-NER Dataset, a total of 75 entity types are present. The entity count for each type exhibits a significant imbalance. Hence, the dataset is balanced by selecting 25 crucial entity types and adjusting the count of unique entities within each type to fall between 2,000 and 20,000, as outlined in Sect. 3.1, using sampling method-2 under the dataset sampling subsection. The entity distribution for the selected types is depicted in Fig. 5. The balanced dataset comprises a total of 43,432 sentences.

Table 3. Sample NER Dataset for fine-tuning

doc no.	Sentence No.	Word	Tag
1	1	pfizer	B-VACCINE
1	1	effectively	O
1	1	reduces	O
1	1	severe	O
1	1	SARS-COV	B-CORONAVIRUS
1	1	2	I-CORONAVIRUS

Table 4. Greedy Segmentation Result of CORD-NER Dataset on DistilBERT

Segment	# of Sentences	Precision	Recall	F1-score	Runtime
32-Seg	105,591	0.7255	0.5977	0.6554	2 hr 28 min
512-Seg	6,407	0.4729	0.3067	0.3721	7 min

Fine-Tuning BERT Model for NER. The balanced CORD-NER dataset (resulting from Sect. 4.1) is fine-tuned using a smaller version of the BERT model, i.e., the distilbert-base-uncased. Three different BERT transformers are trained and compared (see Table 5). The experiment made use of the NER model provided by the Simple Transformers package. An allocation of 80% of the dataset was made for training, with the remaining 20% set aside for testing purposes. The model is trained for 30 epochs at a learning rate of 1e-4, resulting in a training duration of approximately 22 h and 14 minutes. The trend of loss during the fine-tuning task across each epoch is graphically represented in Fig. 6. Performance metrics for the 20% randomly selected test data demonstrated a precision of 0.7494, recall of 0.7366, and an F1 score of 0.7429.

The fine-tuned model performed well during the training (i.e., loss ≈ 0.00) but the accuracy of testing on unseen random test data was around 75%. This suggests that the fine-tuned model is overfitted. Eight simple techniques to prevent overfitting are discussed in [19]. They are Hold-out, Cross-validation, Data augmentation, Feature selection, L1 / L2 regularization, Removing layers/number of units per layer, Dropout, and Early stopping. In our proposed system, two techniques had already been done in the dataset preparation stage: (1) Hold-out which is an approach for separation of training and testing parts, and (2) Feature selection. We will consider other techniques to obtain a better model in future.

4.2 Entity Extraction

The latest CORD-19 dataset [14], published on March 2022, consists of 902,589 papers in terms of 19 attributes: 'cord_uid', 'sha', 'source_x', 'title', 'doi', 'pmcid', 'pubmed_id', 'license', 'abstract', 'publish_time', 'authors', 'journal', 'mag_id', 'who_covidence_id', 'arxiv_id', 'pdf_json_files', 'pmc_json_files', 'url', and 's2_id'.

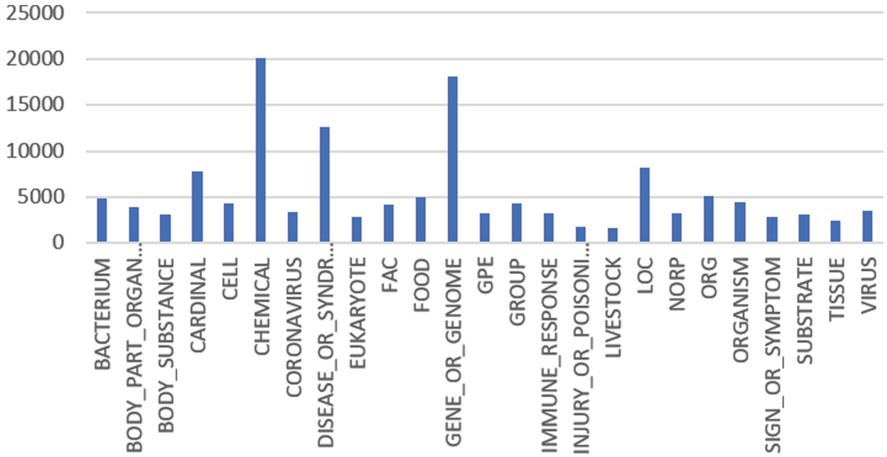


Fig. 5. Entity distribution of the selected types

Table 5. Comparison of three BERT Transformers

Model	# of Epoch	Precision	Recall	F1-score	Runtime
BERT	30	0.7458	0.6483	0.6948	33 hr 30 min
DistilBERT	30	0.7494	0.7366	0.7429	22 hr 14 min
Biomed_roberta	30	0.7396	0.6418	0.6872	34 hr 6 min

Three of 19 attributes are extracted for this study, i.e., ‘title’, ‘abstract’, and ‘publish_time’. Papers published before 2020 are filtered out in order to focus on recent research. Also, 396,374 papers are left after removing null and duplicate files. Then, the vaccine-related dataset is extracted following the procedure presented in the next section.

Dataset Preprocessing. The abstracts from the filtered dataset are taken and preprocessed. The stop words are removed from the abstract sentences. The pre-defined stop words list is acquired from the Natural Language Toolkit (NLTK), a built-in library provided in Python. There are 179 words in the pre-defined stop words list. Some of them are ‘about’, ‘an’, ‘didn’t’, ‘each’, ‘hers’, and ‘most’. Then, the data are tokenized. Tokenization is a way of separating a piece of text into smaller units called ‘tokens’. Then, non-English words are removed by checking whether each word is an alphabet or not. Finally, words with single characters are also removed. The list of vaccine-related words including ‘vaccinated’, and ‘vaccination’ is searched through the cleaned abstracts. A total number of 30,208 vaccine-related papers are found after this process. Table 6 shows the statistical information of filtering papers for the dataset.

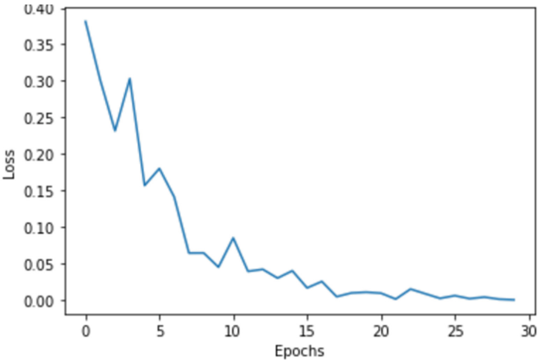


Fig. 6. Performance graph on fine-tuning NER model

Table 6. Statistics of step-by-step filtering on the CORD-19 dataset

Dataset Status	Number of papers
The CORD-19 dataset	902589
After removing null and duplicate files	554654
After filtering papers published from 2020	396374
After filtering papers related to vaccine	30208

Literature Clustering. The obtained vaccine-related dataset is clustered into specific topics. The Latent Dirichlet Allocation (LDA) method [8] is used to do topic modeling. The abstract data are lemmatized and chunked into uni-gram and bi-gram word phrases of 'NOUN', 'ADJ', 'VERB', and 'ADV'. Examples of extracted phrases include 'urgently_need', 'immune_response', and 'infection'.

LDA considers each document as a collection of topics, and each topic is a collection of keywords. Once the number of topics is provided, the algorithm rearranges the topics distribution within the documents and keywords distribution within the topics to obtain a good composition of the topic-keywords distribution. Note that each topic's name is not defined by the algorithm. We have to define each topic by looking at the dominant terms under the specific topic.

The coherence value defines the accuracy of the LDA model. We can adjust the number of topics and compare the result coherence value to select the optimal topic number. The higher the coherence value, the better the clustering. The data is trained for the topic numbers from three to ten. The result coherence scores are plotted via a line chart in Fig. 7. Although eight is the highest score, there is a gradual improvement from topic numbers four to six. Therefore, we analyzed the topics under six clusters.

We train LDA based on the configuations: $random\_state = 98$, $chunksize = 200$, $passes = 30$, $iterations = 150$, $decay = 0.7$, and $num\_topics = 6$. Using LDA,

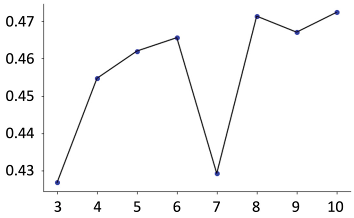


Fig. 7. LDA: Coherence Score w.r.t. Topic Number

Table 7. Sample clustering result on five papers

Paper No.	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
0	0	0.68626	0	0	0.30411	0
1	0	0	0.17671	0.81582	0	0
2	0	0.03487	0.0854	0.14731	0.72960	0
3	0	0.01404	0	0.37329	0.56058	0.0477
4	0	0	0	0.92077	0.07459	0

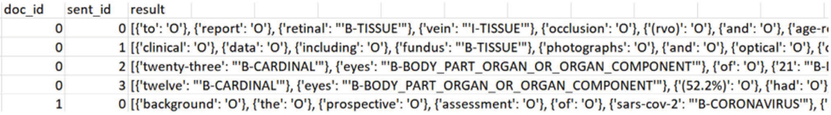
the probability of each cluster for each paper is obtained. The sample data on clustering on six topics of five papers are shown in Table 7.

According to the results shown in Table 8, the papers under Topic 4 are selected to be used as the core dataset.

Post-processing Recognized Named Entities. The CORD-NER dataset also provides information for two-word entities by identifying them using two prefixes (B-) and (I-). The procedure to clean the types has been discussed in NER Post-Processing under Sect. 3.2. Results are illustrated in Fig. 8. In this example, only the words 'submacular hemorrhage' will be combined as the 'DISEASE_OR_SYNDROME' type. Other entities with the prefix (I-) are discarded. Other entities with the prefix (B-) are taken together with respective entity types. The finalized NER dataset is accessible through IEEE Dataportal [20]. The overall implementation of this research is available online [21].

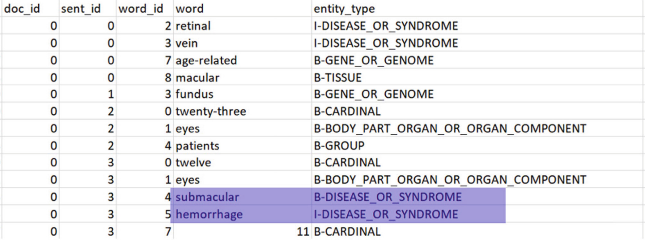
Table 8. Statistic of clustered papers (sorted in preferred topic order)

Topic	# papers	Dominant terms	Possible topic
4	6179	participant, survey, vaccine_ hesitancy, acceptance	research on vaccine hesitancy
2	6394	protein, mutation, spike_ protein, sequence	Biomedical studies in vaccine production
3	995	review, clinical_trial, research	research on clinical trials of vaccines
5	6916	model, transmission, epidemic	general topic on vaccination
1	5251	dose, day, month, mrna	Analysis on vaccination Status
6	4473	pneumonia, complication, syndrome, bcg_vaccination	general domain



doc_id	sent_id	result
0	0	[{'to': 'O'}, {'report': 'O'}, {'retinal': "'B-TISSUE'"}, {'vein': "'I-TISSUE'"}, {'occlusion': 'O'}, {'(rvo)': 'O'}, {'and': 'O'}, {'age-r
0	1	[{'clinical': 'O'}, {'data': 'O'}, {'including': 'O'}, {'fundus': "'B-TISSUE'"}, {'photographs': 'O'}, {'and': 'O'}, {'optical': 'O'}, {'
0	2	[{'twenty-three': "'B-CARDINAL'"}, {'eyes': "'B-BODY_PART_ORGAN_OR_ORGAN_COMPONENT'"}, {'of': 'O'}, {'21': "'B-I
0	3	[{'twelve': "'B-CARDINAL'"}, {'eyes': "'B-BODY_PART_ORGAN_OR_ORGAN_COMPONENT'"}, {'(52.2%)': 'O'}, {'had': 'O'}
1	0	[{'background': 'O'}, {'the': 'O'}, {'prospective': 'O'}, {'assessment': 'O'}, {'of': 'O'}, {'sars-cov-2': "'B-CORONAVIRUS'"}, {'

(a) Predicted Result



doc_id	sent_id	word_id	word	entity_type
0	0	2	retinal	I-DISEASE_OR_SYNDROME
0	0	3	vein	I-DISEASE_OR_SYNDROME
0	0	7	age-related	B-GENE_OR_GENOME
0	0	8	macular	B-TISSUE
0	1	3	fundus	B-GENE_OR_GENOME
0	2	0	twenty-three	B-CARDINAL
0	2	1	eyes	B-BODY_PART_ORGAN_OR_ORGAN_COMPONENT
0	2	4	patients	B-GROUP
0	3	0	twelve	B-CARDINAL
0	3	1	eyes	B-BODY_PART_ORGAN_OR_ORGAN_COMPONENT
0	3	4	submacular	B-DISEASE_OR_SYNDROME
0	3	5	hemorrhage	I-DISEASE_OR_SYNDROME
0	3	7	11	B-CARDINAL

(b) Cleaned Predicted Result

Fig. 8. Illustration of NER Post-processing

5 Conclusion

In conclusion, this research presents a method for document-level named entity recognition (NER) using BERT, a large language model (LLM), on the CORD-19 dataset. The study utilizes different sampling approaches for the two datasets, with the Latent Dirichlet allocation algorithm applied to CORD-19 and a simple greedy approach used to collect the most informative data of 25 entity types for CORD-NER. By extracting named entities from the CORD-19 dataset, a baseline dataset for further advanced applications such as knowledge base and ontology is established. Furthermore, the study demonstrates the possibility of utilizing the content understanding ability of LLMs on low-performance machines without using supercomputers, which is an important aspect of making NER accessible to a wider range of researchers. Overall, this research provides a valuable contribution to the field of NER and LLMs, and it has implications for a wide range of applications, from information retrieval and knowledge management to natural language processing and artificial intelligence.

Acknowledgment. The authors would like to thank Dr. Chutiporn Anutariya for the fruitful discussion, and her inspirational comments. We would also like to thank the National Institute of Informatics (Tokyo, Japan) for the support of this research. We would like to express all our gratitude to Nicolas Greneche and Prof. Christophe Cerin (the Institut Galilée - LIPN UMR CNRS 7030, UNIVERSITÉ SORBONNE PARIS NORD) for their helpful advice and cooperation in the use of MAGI cloud. This work was also partially supported by JSPS KAKENHI Grant Number JP22K18004.

References

1. Scherbakov, V., Mayorov, V.: Finetuning BERT on partially annotated NER corpora. arXiv. (2022). https://doi.org/10.48550/arXiv.2211.14360
2. Park, Y.I., Lee, M., Yang, G., Park, S.J., Sohn, C.: Biomedical text NER tagging tool with web interface for generating BERT-based fine-tuning dataset. Appl. Sci. **12**, 12012 (2022)
3. Balkus, S.V., Yan, D.: Improving short text classification with augmented data using GPT-3. ArXiv, abs/2205.10981 (2022)
4. Kieuvongngam, V., Tan, B., Niu, Y.: Automatic text summarization of COVID-19 medical research articles using BERT and GPT-2. ArXiv, abs/2006.01997 (2020)
5. Maltoudoglou, L., Paisios, A., Papadopoulos, H.: BERT-based conformal predictor for sentiment analysis. In Conformal and Probabilistic Prediction and Applications, pp. 269–284. PMLR (2020)
6. Wang, X., Song, X., Guan, Y., Li, B., Han, J.: Comprehensive named entity recognition on CORD-19 with distant or weak supervision. ArXiv, abs/2003.12218 (2020)
7. Pestryakova, S., et al.: CovidPubGraph: a FAIR knowledge graph of COVID-19 publications. Sci. Data **9**, 389 (2022)
8. Blei, D.M., Ng, A., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2001). https://doi.org/10.1016/B978-0-12-411519-4.00006-9
9. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. ArXiv (2019). https://doi.org/10.48550/arXiv.1810.04805
10. Giles, O., Huntley, R.P., Karlsson, A., Lomax, J., Malone, J.: Reference ontology and database annotation of the COVID-19 Open Research Dataset (CORD-19). bioRxiv (2020). https://doi.org/10.1101/2020.10.04.325266
11. Wu, J., Wang, P., Wei, X., Rajtmajer, S.M., Giles, C.L., Griffin, C.: Acknowledgement entity recognition in CORD-19 papers. In: SDP, pp. 10-19 (2020). https://doi.org/10.18653/v1/2020.sdp-1.3
12. Popovski, G., Kochev, S., Korousic-Seljak, B., Eftimov, T.: FoodIE: a rule-based named-entity recognition method for food information extraction. Int. Conf. Pattern Recogn. Appl. Meth. **12**, 915 (2019)
13. Dekhili, G., Sadat, F.: Hybrid statistical and attentive deep neural approach for named entity recognition in historical newspapers. In: Conference and Labs of the Evaluation Forum (2020)
14. Wang, L.L., et al.: CORD-19: the COVID-19 open research dataset. ArXiv (2020)
15. 5 Probabilistic Training Data Sampling Methods in Machine Learning. https://towardsdatascience.com/5-probabilistic-training-data-sampling-methods-in-machine-learning-460f2d6ffd9. Accessed 1 July 2023
16. Liu, J., et al.: Tracing the pace of COVID-19 research: topic modeling and evolution. Big Data Res. **25**, 100236–100236 (2021). https://doi.org/10.1016/j.bdr.2021.100236
17. Unified Medical Language System(UMLS). https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/index.html. Accessed 8 July 2023
18. SpaCy models for biomedical text processing. https://allenai.github.io/scispacy/. Accessed 8 July 2023
19. David Chuan-En Lin, 8 Simple Techniques to Prevent Overfitting. https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d. Accessed 1 July 2023

20. Thant, S., Anutariya, C., Andres, F., Racharak, T.: BERT fine-tuned CORD-19 NER dataset, IEEE Dataport (2023). https://doi.org/10.21227/m7gj-ks21
21. ShinThant3010, 'ShinThant3010/Deep-Learning-based-KG-for-Covid19-Vaccination: Deep Learning based KG for Covid19 Vaccination'. Zenodo, 02 November 2023. https://doi.org/10.5281/zenodo.10066965, https://github.com/ShinThant3010/Deep-Learning-based-KG-for-Covid19-Vaccination

Data Science and AI in Healthcare

Optimizing Heart Disease Classification: Exploring the Impact of Feature Selection and Performance of Machine Learning Algorithms

Aga Maulana[1], Farrasa Rani Faisyal[1], Faris Khowarizmi Tarmizi[1],
Taufik Fuadi Abidin[1](✉), and Hammam Riza[2]

[1] Department of Informatics, Faculty of Mathematics and Natural Sciences,
Universitas Syiah Kuala, Banda Aceh, Indonesia
`{agamaulana,faris,taufik.abidin}@usk.ac.id`
[2] Research Center for Artificial Intelligence and Cyber Security, National Research and Innovation Agency, Jakarta, Indonesia
`hammam.riza@brin.go.id`

Abstract. This study aimed to predict heart disease using machine learning models and feature selection techniques. The Cleveland Clinic Heart Disease dataset was used. Some feature selection methods were applied to identify relevant features, including Recursive Feature Elimination (RFE) and Least Absolute Shrinkage and Selection Operator (LASSO). Multiple machine learning models, such as Random Forest, Logistic Regression, and Support Vector Machines, were trained and evaluated for their performance in classifying heart disease. The results showed that the combination of Random Forest with RFE achieved the highest performance, exhibiting the highest accuracy (91%), precision (90%), recall (90%), and F-score (90%). Furthermore, the model with RFE and Random Forest demonstrated a superior discriminatory ability with an AUC Score of 92%. This research has implications for improving heart disease diagnosis and developing predictive models for early detection.

Keywords: Heart disease · Feature selection · Machine learning

1 Introduction

Cardiovascular diseases, including heart disease, have become a significant global health problem that affects a large portion of the population. Early detection and accurate classification of heart disease are crucial for facilitating timely medical interventions and reducing the risk of serious complications. In recent years, integrating feature selection techniques and machine learning algorithms has shown the ability to enhance the overall performance of the classification models of heart disease [1].

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 279–287, 2023.
https://doi.org/10.1007/978-981-99-7969-1_20

The rapid advancement of technology and the availability of large-scale medical datasets have opened opportunities to gain valuable insights from complex data, even for data with high dimensionality. Feature selection techniques aim to identify the most informative and relevant subset of features from a given dataset. By reducing the data's dimensionality, feature selection helps improve classification accuracy, enhance computational efficiency, and reduce the risk of overfitting [2].

On the other hand, machine learning algorithms offer computational models capable of acquiring knowledge from data and making predictions or decisions without requiring explicit programming. When combined with appropriate feature selection techniques, these algorithms can effectively classify heart disease based on the selected features. K-Nearest Neighbors (KNN), Logistic Regression, Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Random Forest are some prominent machine learning algorithms used in disease classification [3].

This research aims to analyze the performance of classification models for heart disease by leveraging feature selection techniques and machine learning algorithms. The study investigates the impact of feature selection on classification accuracy and compares the performance of various machine learning algorithms in heart disease classification.

2 Related Work

Many researchers have worked on these research areas. Dissanayake et al. [4] analyzed various feature selections, such as filter, wrapper, and embedded methods, to obtain the most valuable insights into feature selection methods in heart disease classification.

Modak et al. [5] employed a multilayer perceptron and achieved an accuracy of 87.70%. They evaluated their approach on Cleveland, Hungarian, Switzerland, Long Beach, and Statlog datasets. Sarah et al. [6] utilized logistic regression and reported accuracy percentages for the Cleveland dataset, achieving 87.70% and 85.25% accuracy on different occasions. Nguyen et al. [7] took a different approach by using Logistic Regression, SVM, Naive Bayes, and Decision Trees, with their evaluation focusing on the Cleveland dataset, achieving an accuracy rate of 83.50%. Latha et al. [8] experimented with Multilayer Perceptron, Random Forest, Naïve Bayes, and Bayes Net achieving an accuracy of 84.49% on the Cleveland dataset. In addition, Sujatha et al. [9] analyzed the performance of Decision Tree, Random Forest, Naive Bayes, and SVM using feature selection methods in categorizing heart disease. The results give a better understanding of the most effective machine-learning techniques in classifying heart disease by incorporating feature selection methods during preprocessing.

Meanwhile, Li et al. [10] studied the use of machine learning for heart disease. They developed a method to accurately classify heart disease categories and successfully recognize associated patterns and clinical characteristics using appropriate machine learning algorithms such as Neural Networks, Random Forests, and Decision Trees with relevant features.

3 Materials and Methods

The flow of this study starts with data acquisition, preprocessing, feature selection, machine learning models, and performance analysis. It demonstrates the systematic process of gathering and preparing the data, selecting relevant features, training different machine learning models, and evaluating their performance.

3.1 Dataset

The Cleveland Clinic Heart Disease Dataset [11] has been widely used in heart disease research and classification. This dataset was collected from several other medical institutions at the Cleveland Clinic Foundation. The dataset encompasses 303 patients and 14 attributes. The target variable in this dataset represents a heart disease situation, with values ranging from 0 (no heart disease) to 4 (severe heart disease). For this study, the dataset is adapted into binary classification, with 0 against 1-2-3-4 indicating the presence or absence of the disease.

3.2 Data Preprocessing

The dataset was split into 80:20 for training and testing before applying machine learning modeling techniques. Subsequently, the dataset is processed to address missing values, normalize numerical features, and encode categorical variables. These steps ensure that the dataset is appropriately formatted for the analysis.

3.3 Feature Selection

Creating a good machine learning model performance requires carefully selecting features to avoid overfitting. Hence, various techniques are used to determine the most valuable and relevant subset of features for classifying heart disease. Methods such as the Least Absolute Shrinkage and Selection Operator (LASSO) [12] and Recursive Feature Elimination (RFE) [13] ensure that only the most essential features are used in the model, making it easier to understand and interpret the results.

3.4 Machine Learning Models

We employed various machine learning models, including KNN, Logistic Regression, Random Forest, ANN, and SVM [14–16]. These models were trained using subsets of features determined by the feature selection techniques mentioned earlier. We compared the performance of different feature selection techniques with each model to determine the best combination to achieve optimal results in categorizing heart disease.

3.5 Performance Evaluations

The performance of the models was assessed using commonly used evaluation metrics, i.e., accuracy, precision, recall, and F-score. These metrics measure the effectiveness of feature selection techniques and machine learning algorithms in heart disease classification problems. We calculate weighted averages of these metrics to ensure more robust evaluation.

The formula of accuracy, precision, recall, and F-score are presented in Eqs. 1,2,3 and 4, respectively. TP is the true positive, TN represents the true negative, FP means the false positive, and FN denotes the false negative. These metrics measure the model's performance and help assess the ability of the model to classify heart disease accurately.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$Fscore = \frac{\sum_{i=1}^{n} w_i \cdot F_1^i}{\sum_{i=1}^{n} w_i} \tag{4}$$

where:

$Fscore$ is the weighted average F-score

n is the number of classes (in this case, two classes)

w_i is the weight assigned to class i

F_1^i is the F-score for class i

4 Results and Discussions

4.1 Results of Feature Selections

This study uses two feature selection methods: LASSO Feature Selection and RFE. LASSO selects features by modifying the absolute coefficient value of the features. Features with zero coefficient values are eliminated from the feature set. In the selected feature set, those with high coefficient values are included. However, sometimes, LASSO may select irrelevant features and include them in the subset of features.

RFE is a feature selection method in data analysis that gradually reduces the number of features to build a more effective model. This method is commonly used in machine learning and regression. The RFE process begins by building a model with all available features. Then, the weight or importance of each feature is evaluated, for example, using regression coefficients or scores of other feature influences. Features with the lowest weights are considered less significant and are removed from the model. This step is repeated until the desired number of features or a specified termination criterion is reached. For example, a certain number of features can be selected, or a specific threshold value can be used for feature importance.

For the LASSO Feature Selection method, the features are ‘chol’ and ‘thalach,’ and two features were chosen. For the RFE method, the selected features are ‘sex’, ‘cp’, ‘restecg’, ‘fbs’, ‘thalach’, ‘oldpeak’, ‘exang’, ‘ca’, ‘slope’, and ‘thal’. In this case, a total of 10 features were selected. The LASSO method resulted in a more aggressive feature elimination, selecting only two features, while the RFE method retained a more extensive set of features with a total of 10.

4.2 Model Performance

In this study, we trained a model using selected subsets of features and compared its performance when a model was trained using all features. The purpose was to evaluate the model’s performance after feature selection.

The results of the model trained using all features are presented in Table 1. It is clearly shown that the Random Forest model achieved perfect scores during the training but lower during the testing than the Logistic Regression and ANN models. The SVM model performed relatively lower on the training and testing sets than the other models.

Table 2 displays the performance of models trained using subsets of features selected by LASSO. The Random Forest model obtained the highest scores on the training and testing sets, indicating superior performance. The result suggests that the Random Forest model is better for classifying heart disease using the selected feature subset than other methods.

Table 3 presents the performance results of models trained using subsets of features selected by RFE. The Random Forest model achieved the highest accuracy on the training set but experienced a slight decrease in performance on

Table 1. Performance of the model trained using all available features

Model	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)
KNN	Training	76	76	76	76
	Testing	69	69	69	69
Logistic Regression	Training	86	86	86	85
	Testing	**89**	**89**	**89**	**89**
Random Forest	Training	**100**	**100**	**100**	**100**
	Testing	84	84	84	84
ANN	Training	81	82	81	80
	Testing	87	87	87	87
SVM	Training	66	67	66	64
	Testing	70	73	70	69

Table 2. Performance of the model trained using Lasso

Model	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)
KNN	Training	71	71	71	71
	Testing	70	70	70	70
Logistic Regression	Training	69	69	69	69
	Testing	75	76	75	75
Random Forest	Training	**100**	**100**	**100**	**100**
	Testing	**77**	**77**	**77**	**77**
ANN	Training	63	63	63	61
	Testing	62	65	62	59
SVM	Training	69	70	69	68
	Testing	72	74	72	71

Table 3. Performance of the model trained using RFE

Model	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)
KNN	Training	81	82	81	81
	Testing	79	79	79	79
Logistic Regression	Training	85	85	85	85
	Testing	89	89	89	89
Random Forest	Training	**100**	**100**	**100**	**100**
	Testing	**91**	**90**	**90**	**90**
ANN	Training	85	86	85	84
	Testing	85	85	85	85
SVM	Training	68	69	68	67
	Testing	69	70	69	68

the testing set. In contrast, the Logistic Regression and ANN models consistently performed well on both sets. The SVM model had the lowest performance initially but improved on the testing set. These findings emphasize the diverse performance of different machine learning models and their relevance for heart disease classification tasks.

By comparing the performance results, the combination of Random Forest and RFE achieved the highest model accuracy, precision, recall, and F-score performance. RFE, as a feature selection technique, systematically identifies and retains the most informative features, which reduces noise and enhances the model's focus on relevant information. When coupled with Random Forest, which excels in capturing complex relationships and patterns in the data while mitigating overfitting through ensemble learning, this combination results in a powerful synergy. The Random Forest model's inherent feature importance assessment aligns well with RFE's feature selection, ensuring that the most critical variables are considered. Moreover, Random Forests handle non-linearity effectively, often prevalent in real-world datasets. As a result, the combined approach benefits from both techniques' strengths, enabling it to deliver better accuracy, precision, recall, and F-score performance in classification tasks. This finding indicates that RFE selects a subset of features that provide sufficient information about the feature and target variable.

On the other hand, the model trained using feature subsets selected by LASSO exhibited poorer performance than the model trained without feature selection and the model trained using the features selected by RFE. It indicates that LASSO selected only two features, limiting its ability to capture the relationships between features and the target variable. Consequently, the model trained with such a restricted feature subset lacked the information to classify heart disease accurately.

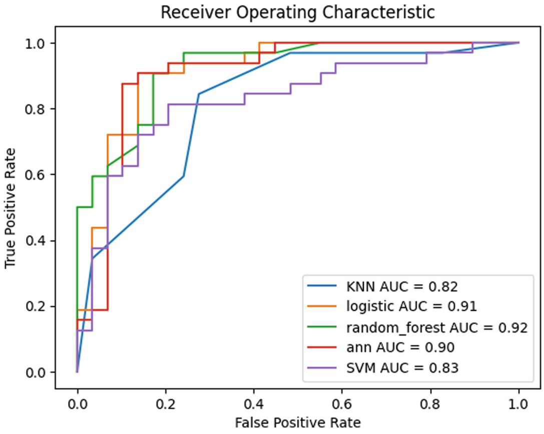


Fig. 1. ROC Curve and AUC Score of RFE.

As RFE demonstrated the best performance, we analyzed the ROC curve more by visualizing and calculating the AUC score (Fig. 1). Remarkably, the RFE with the Random Forest combination achieved the highest AUC score of 0.92. The result indicates that the model can accurately predict individuals the presence or absence of heart disease. The high AUC score confirms that the selected feature subset effectively classifies heart disease.

5 Conclusions

This study successfully predicted heart disease using feature selection and machine learning techniques. The combination of RFE achieved the highest performance, demonstrating the effectiveness of RFE in selecting informative features for accurate classification. The model with RFE and Random Forest also obtained the highest AUC score, indicating solid discriminatory ability.

It is essential to acknowledge the limitations of this study. This study was conducted on the Cleveland Clinic Heart Disease dataset, which may not fully represent the general population's diversity of heart disease cases. Also, this study did not consider other factors, such as lifestyle, genetics, or biomarkers, potentially impacting heart disease prediction. Future research should address these limitations to understand heart disease classification better.

References

1. Alizadehsani, R., Habibi, J., Gopalakrishnan, V.: Predicting heart disease using feature selection and classifier ensemble. Comput. Biol. Med. **89**, 15–22 (2017)
2. Pankaj, N., Kaur, G., Jain, A., Sharma, A.: Prediction of heart disease using feature selection and machine learning algorithms. Int. J. Eng. Adv. Technol. (IJEAT) **8**(6), 84–89 (2019)
3. Zhang, W., Sun, Y., Wang, Y., Liu, X.: The feature selection and classification of heart disease based on random forest. J. Phys. Conf. Ser. **1942**(1), 012052 (2021)
4. Dissanayake, K., Md Johar, M.G.: Comparative study on heart disease prediction using feature selection techniques on classification algorithms. Appl. Comput. Intell. Soft Comput. **2021**, 1–17 (2021)
5. Modak, S., Abdel-Raheem, E., Rueda, L.: Heart disease prediction using adaptive infinite feature selection and deep neural networks. In: International Conference on Artificial Intelligence in Information and Communication (ICAIIC), pp. 235–240 (2022)
6. Sarah, S., Gourisaria, M.K., Khare, S., Das, H.: Heart disease prediction using core machine learning techniques—a comparative study. In: Advances in Data and Information Sciences. LNNS, vol. 318, pp. 247–260. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-5689-7_22
7. Nguyen, K., et al.: Heart disease classification using novel heterogeneous ensemble. In: IEEE EMBS International Conference on Biomedical and Health Informatics (BHI), pp. 1–4 (2021)
8. Latha, C.B., Jeeva, S.C.: Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques. Inform. Med. Unlocked **16**, 100203 (2019)
9. Sujatha, P., Mahalakshmi, K.: Performance evaluation of supervised machine learning algorithms in prediction of heart disease. In: IEEE INOCON, pp. 1–7 (2020)
10. Li, J.P., Haq, A.U., Din, S.U., Khan, J., Khan, A., Saboor, A.: Heart disease identification method using machine learning classification in e-healthcare (2020)
11. Kaggle, Cleveland clinic heart disease dataset (2019). https://www.kaggle.com/datasets/aavigan/cleveland-clinic-heart-disease-dataset. Accessed 13 May 2023
12. Emmert-Streib, F., Dehmer, M.: High-dimensional LASSO-based computational regression models: regularization, shrinkage, and selection. Mach. Learn. Knowl. Extract. **1**(1), 359–383 (2019)
13. Misra, P., Yadav, A.S.: Improving the classification accuracy using recursive feature elimination with cross-validation. Int. J. Emerg. Tech. **11**(3), 659–665 (2020)
14. Maulana, A., Noviandy, T.R., Idroes, R., Sasmita, N.R., Suhendra, R., Irvanizam, I.: Prediction of kovats retention indices for fragrance and flavor using artificial neural network. In: 2020 International Conference on Electrical Engineering and Informatics (ICELTICs), pp. 1–5. IEEE (2020)

15. Noviandy, T.R., Maulana, A., Emran, T.B., Idroes, G.M., Idroes, R.: QSAR classification of beta-secretase 1 inhibitor activity in Alzheimer's disease using ensemble machine learning algorithms. Heca J. Appl. Sci. **1**(1), 1–7 (2023)
16. Maulana, A., et al.: Machine learning approach for diabetes detection using fine-tuned XGBoost algorithm. Infolitika J. Data Sci. **1**(1), 1–7 (2023)

Enhancing Data Science Interoperability: An Innovative System for Managing OpenEHR Structures

Miguel Dias, Regina Sousa, Júlio Duarte, Hugo Peixoto, António Abelha, and José Machado(✉)

ALGORITMI/LASI, University of Minho, 4710-059 Braga, Portugal
jmac@di.uminho.pt

Abstract. The concept of e-Health is increasingly used in the healthcare industry, referring to the development and application of software and hardware solutions for efficient collection, storage, manipulation, and communication of data, to improve healthcare delivery. Data Science Interoperability is critical to create a data repository from which to implement artificial intelligence models that support the decision-making process. The lack of interoperability in Health Information Systems (HIS) has been a significant challenge. The need for systems that promote interoperability between HIS within the same institution or even between HIS from different institutions is a worldwide concern. A system that contributes to improving interoperability in healthcare through the use of the openEHR standard is proposed with this paper. The system provides a way to manipulate clinical data by creating an artifact built with React and NextJS that allows the conversion of openEHR standardized data into a JSON object. The artifact being a web application presents a new way for users to check openEHR data, while the API can be used by developers to work with openEHR data in a more accessible and supported way. The results of this research and engineering effort have been successful in presenting a new approach to implementing yet another tool to help healthcare professionals and biomedical software engineers.

Keywords: openEHR · Health Information Systems · Data Science Interoperability

1 Introduction

Healthcare professionals have historically faced many challenges in maintaining patient safety due to the chaotic and constantly evolving hospital environment. Patient safety should always be the highest priority in any healthcare facility, so there is a continuous quest to improve healthcare services and avoid delays in information exchange that can lead to serious adverse reactions [17]. To ensure effective clinical decision making, the demand for auxiliary tools to support clinical actions has become evident. In today's healthcare sector, information

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 288–299, 2023.
https://doi.org/10.1007/978-981-99-7969-1_21

technology plays a critical role in improving patient care, especially in organizing clinical data. However, efficient storage and organization of data comes with its own set of challenges. Due to the increasing propagation of large amounts of data, interoperability between different Hospital Information Systems (HIS) becomes exponentially more difficult [4].

As a result of the ongoing inefficiencies with the propagation of healthcare data, there is a need for a more simplified and well-supported data organization system. An artifact is proposed with a system that organizes data into a more concise and supported format to assist both healthcare professionals and biomedical IT engineers in developing applications using healthcare data. Using openEHR archetype files, these are converted into a valid JSON format and integrated into a prototype web application, meeting the initial need for data organization [11]. The JSON format has established itself as a long-standing standard in Web development due to its efficiency and lightness, making it an ideal choice for storing and sharing data across multiple systems. Converting files to this structure will allow for more efficient data propagation and easier development in future projects.

The following paper begins by providing a concise background on the health data topic area, providing background information to establish the context of the research. It then describes the research methodologies applied to the project, describing the specific approaches and techniques employed in conducting the study. The results and their analysis are presented in the results and discussion sections, providing a comprehensive analysis of the findings and their implications.

2 Background

In recent years, the increasing volume of clinical data has increased the demand for efficient and organized information management. Since the 1960 s, hospitals have used computerized systems that integrate administrative, financial, and clinical functions to effectively collect, organize, and analyze clinical data [18].

Arguably, one of the biggest contributors to modern clinical data storage has been the introduction of the Electronic Health Record (EHR). EHRs are digital versions of a patient's medical history that contain information about their medical conditions, medications, or test results. A plethora of advantages have come from replacing paper records with electronic ones, the increased speed and efficiency of data sharing, as a result, has helped improve patient care and enhance analysis of population health trends [2,3] with a notable decrease in adverse reactions due to lack of data on a patient. While EHRs have really helped in data propagation, there are still certain inefficiencies that need to be fixed, especially when it comes to data exchange between different HIS [4].

2.1 Interoperability in Healthcare

Interoperability, as defined by the Cambridge dictionary, can be described as 'The ability to work together with other systems or equipment.' In the healthcare context, achieving high levels of interoperability is of utmost importance. The challenge lies in enabling multiple systems from different Health Information Systems (HIS) to interpret health-related data, mainly due to the significant amounts of data being produced [8].

Different levels of interoperability can be identified, with semantic interoperability, for example, involving the creation of consistently defined domains, terminologies, or concepts that are shared among multiple HIS [7].

According to the European Commission, achieving semantic interoperability depends on establishing agreements on standards, information models, terminologies, and semantic definitions for sharing data [6]. Factors such as cultural, legal, and social considerations within organizations, regions, or countries may influence the implementation of systems that aim to achieve semantic interoperability.

2.2 Health Data Standards

For EHRs to function properly, certain national and international health data standards must be created to facilitate interoperability of electronic records across various hospital information systems. These guidelines emerged as a group effort to provide standardized EHRs. From that effort, a plethora of guidelines have emerged over the years. HL7 is one of the pioneering frameworks developed to standardize EHR data, using XML syntax to represent data that is human readable and can be easily exchanged and interpreted by computers [5].

The organization of clinical images has been made easier with the introduction of DICOM standards to manage clinical imaging exam results produced from scanned medical exams while the SNOMED standard has been created to provide standards for clinical definitions [10,16].

Arguably, one of the most well-known healthcare data standards today is openEHR. With the goal of creating a fully open source EHR system that can share interoperable data, the openEHR has been characterized by a unique dual model approach with a first layer representing all the ontology and data structures used, while the second layer is the main source of patient information. A representation of this layered model can be seen in Fig. 1 [1].

The development of healthcare data management has been greatly aided by these standards [12]. As a result of their creation, healthcare providers and researchers are now able to provide more efficient and coordinated care for patients through increased interoperability, streamlined clinical image management, and standardized clinical definitions.

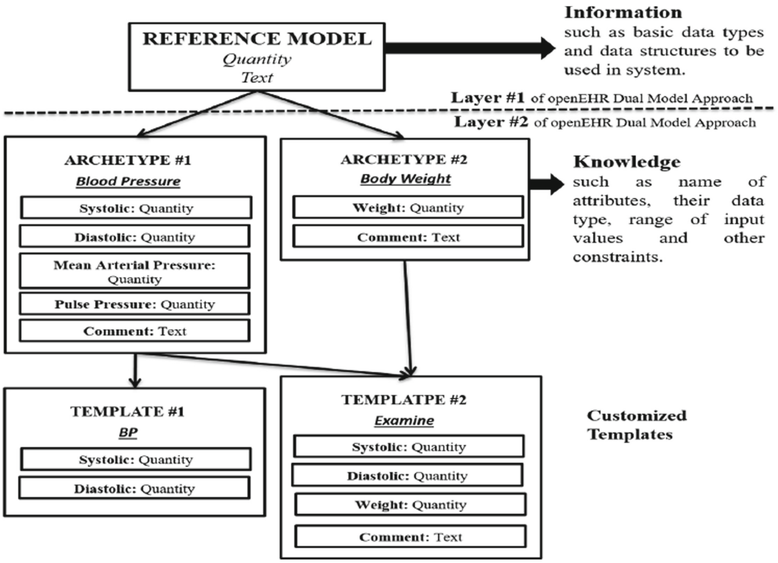


Fig. 1. The openEHR reference model. Taken from [1].

2.3 Archetype Definition Language

The Archetype Definition Language (ADL) was developed by the openEHR foundation to standardize the definition of archetypes and their constituent parts. There are two main flavors of this syntax that can be found in the archetype file, according to the official docs. Commonly, the header, concept, description, and ontology sections are represented using the syntax of Data ADL (dADL). The syntax is illustrated in Fig. 2.

```
language
    original_language = <[ISO_639-1::en]>
    translations = <
        ["nb"] = <
                language = <[ISO_639-1::nb]>
                author = <
                        ["name"] = <"Silje Ljosland Bakke">
                        ["organisation"] = <"Helse Vest IKT AS">
                        ["email"] = <"silje.ljosland.bakke@helse-vest-ikt.no">
                >
        >
    >
```

Fig. 2. An example showing an archetype language section.

What makes this syntax stand out in the ADL document is the usage of arrows to represent object blocks, Fig. 3. Multiple of these objects in turn create a tree branch like structure with multiple pathways, similar to how JSON functions. The definition portion of the document is represented using a modified version of the previous syntax, the constraint ADL or simply cADL syntax. Here the arrows are replaced with curly braces and the equal signs with the "matches" or "is_in" keywords [13]. The use of arrows to represent object blocks sets this syntax apart from others in the ADL document. Multiple such objects generate a tree-like structure with multiple paths, much like JSON. Constraint ADL, or cADL for short, is an updated version of the previous syntax used to represent the document's definition. In this case, brackets take the place of arrows, and "matches" and "is_in" are used in place of the equals sign [13]. There are other distinctions that need to be clarified:

- Constraint identifiers or archetype classes are written in upper case. Attributes are followed after a "matches" keyword an open curly brace all written in lower case (unless it's a string type represented inside quotations marks).
- Certain added keywords like "cardinality", or "occurrences" to add additional constraint instructions.
- The usage of regular expressions in "allow_archetype" to instruct the inclusion of more archetype files.
- Nodes are represented using the "at" or "ac" prefix followed by a numerical code inside brackets.

```
definition
    CLUSTER[at0000] matches {    -- Address
        items cardinality matches {1..*; unordered} matches {
            ELEMENT[at0001] occurrences matches {0..*} matches {
                    -- Address line
                value matches {
                    DV_TEXT matches {*}
                }
            }
            ELEMENT[at0002] occurrences matches {0..*} matches {
                    -- Town
                value matches {
                    DV_TEXT matches {*}
                }
            }

        }
    }
```

Fig. 3. An example showing an archetype definition section.

3 Research Methodologies

In order to carry out the development, some studies and tests were conducted to gain a thorough understanding of the project's base requirements. Research methods that have been shown to aid in the decision-making process during the design phase were used to carry out this preliminary effort. The research question that motivated the creation of the artifact is as follows.

"Will this new approach provide a solution for the significant challenges faced by professionals in the medical industry with the increasing complexity of propagation of healthcare record data?"

The two methodologies chosen will be presented as follows.

3.1 Design Science Research

In the realm of information science research methodologies, the Design Science Research (DSR) methodology emerges as an optimal choice for addressing the problem at hand within the provided case study. DSR encompasses the creation and evaluation of practical solutions or artefacts in real-world contexts, making it a widely recognized framework in the realm of IT product development [15]. Its structured approach facilitates targeted actions and offers a systematic framework for addressing specific issues. A model for the DSR approach, designed by Peffers is structured as followed [14]:

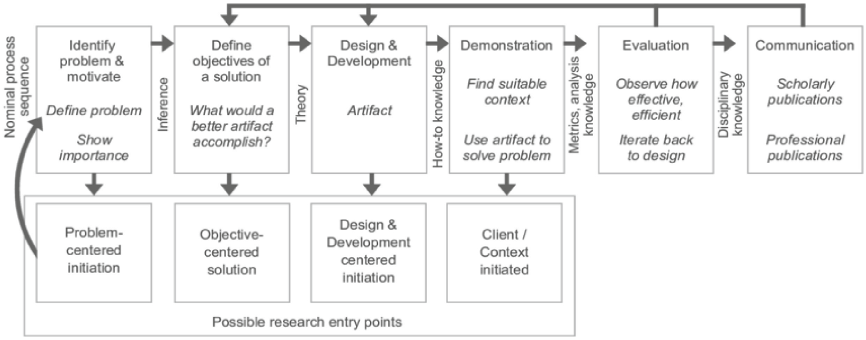

Nominal process sequence
Identify problem & motivate
Define problem
Show importance
Inference
Define objectives of a solution
What would a better artifact accomplish?
Theory
Design & Development
Artifact
How-to knowledge
Demonstration
Find suitable context
Use artifact to solve problem
Metrics, analysis knowledge
Evaluation
Observe how effective, efficient
Iterate back to design
Disciplinary knowledge
Communication
Scholarly publications
Professional publications
Problem-centered initiation
Objective-centered solution
Design & Development centered initiation
Client / Context initiated
Possible research entry points

Fig. 4. Design Science Research. Adapted from [14].

The primary deliverable of a DSR project to create a new health app is the app itself, along with documentation of key design decisions and the outcomes of usability testing. These components, when put together, represent the final product of the design process and are meant to solve a certain health issue or meet a certain need. Because of its centrality, the application undergoes stringent testing to determine whether or not it successfully addresses the specified issue and fulfills the expectations of its target audience.

Researchers and developers of IT products alike can benefit from the DSR methodology's robust approach to tackling difficult problems. Insights into how to solve a problem can be gleaned from the development and testing of actual artifacts or solutions in the real world. The DSR methodology is applied to the creation of a health app that successfully addresses user issues and fulfills their requirements.

3.2 Waterfall Development

Choosing how to manage the design process and any associated tasks is crucial in the creation of new software. Factors outside of a developer's control, such as shifting client requirements and preferences or market conditions, can have a significant impact on the initial stages of conceptualizing new software.

The Waterfall technique, shown in Fig. 5, is a tried and true software development methodology. It's organized in a linear fashion, with one stage preceding another until both are complete. The step-by-step approach fits well with the overall small scope and requirements of this project, despite being seen as more restrictive than other popular methodologies like the agile methodology [9]. It is possible that a new approach to planning and development would be required for a massively expanded version of this artifact with more developers involved in the decision-making process.

Following the gathering of client inputs for the purpose of identifying and defining software requirements during the initial planning phase, the next step involves designing the overall architecture of the software while considering the technologies that are best suited for the task at hand, and this is followed by the thorough implementation of the ideas that have been accumulated throughout the preceding stages. After that, the software developers will write all of the code based on the ideas that were discussed earlier, and there will not be much of a change in the design occurring in between. The testing phase doesn't begin until after there is a functional product, and its primary purpose is to fix bugs in the programming or design choices that weren't properly implemented. The software is then deployed and made available to the end users, presuming that the verification of its functionality was successful.

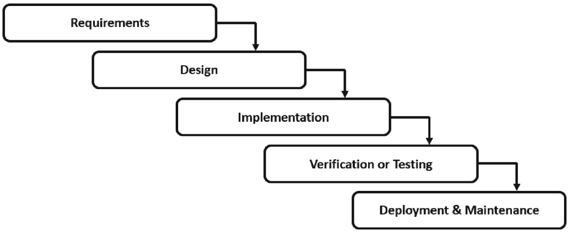


Fig. 5. Waterfall development schema.

4 Results

4.1 Conversion Algorithm

An algorithm was utilized to successfully convert openEHR archetype data into JSON. This algorithm first extracts particular parameters from within the document, and then stores the data in a key-value pair format. Archetype data consists of information that is organized in chunks, specifically concerning data related to the header, concept, language, description, and definition, as well as an ontology section; these are all compiled into a single JSON object at the end. As an illustration, the language section shown in Fig. 6 is what a dADL syntax section ultimately gets compiled into. Other examples include:

```
{
"language": {
    "original_language" : "en",
    "translations" : {
        "nb" : {
            "author" : {
                "name" : "Silje Ljosland Bakke",
                "organisation" = "Helse Vest IKT AS",
                "email" = "silje.ljosland.bakke@helse-vest-ikt.no"
                }
            }
        }
    }
}
```

Fig. 6. An example showing an archetype language section converted to JSON.

In order to ensure full compliance with the stringent formatting requirements of JSON, the arrows that represent objects have been changed to look like curly braces, brackets have been removed from all identifiers, and they are now only used to represent lists that contain more than one attribute. In order to accurately represent string types in attributes, quotation marks were implemented. For this particular illustration, the only information that will be required is that regarding the languages that are available, any translations, and the author(s) of said translations. Certain parameters, such as the ISO code and the mention of the language parameter in the section devoted to translation, were omitted in order to cut down on instances of redundant information.

When you get to the definition section, the process becomes a little bit more complicated, and it is not sufficient to merely replace or add symbols in order to accurately represent this data. Following the first two lines of a definition section is a list of items, after which comes a set of parameters that have been added in relation to the reference model representing the class (rmType), the occurrences, and information about the initial node. As was mentioned earlier, nodes are embodied in the form of a code, and the ontology section of this document contains information pertaining to the nodes themselves. Because of this very reason, the ontology section was parsed first in order to ensure that the information was correctly extracted and placed in the definition section. The remainder of the object tree will have its data related to the nodes that follow after the items section has been filled in with that data. The example that is presented in Fig. 7 contains ELEMENT leaf nodes that are equipped with a “value matches” parameter. This parameter is utilized to recognize multiple text data types. These properties get attributed with empty objects whose purpose is to contain data that users fill out while using any web app. These objects’ purpose is to contain the data that users fill out.

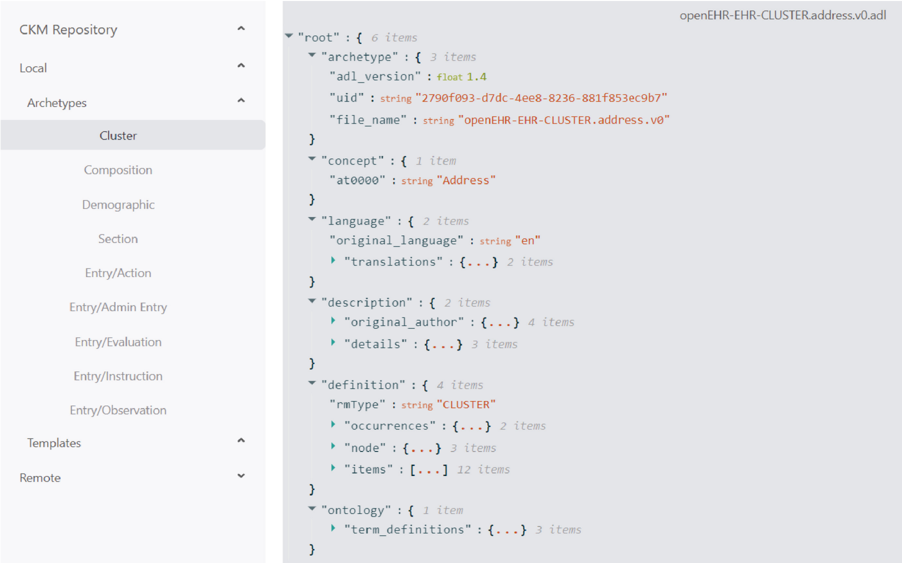
```
{
"definition": {
    "rmType":"CLUSTER",
    "occurrences":{"lowerOccurence":"1","upperOccurence":"*"},
    "node":{"code":"at0000",
        "text":"A title",
        "description":"A description"},
        "items":[{
            "rmType":"ELEMENT",
            "occurrences":{
                "lowerOccurence":"0","upperOccurence":"*"},
            "node":{
                "code":"at0001","text":"","description":""},
            "value_matches":[{
                "dataType":"DV_TEXT","value":{"value":null}}]
            },{
            "rmType":"ELEMENT",
            "occurrences":{
                "lowerOccurence":"0","upperOccurence":"*"},
            "node":{
                "code":"at0002","text":"","description":""},
            "value_matches":[{
                "dataType":"DV_TEXT","value":{"value":null}}]
            }
        ]
    }
}
```

Fig. 7. An example showing an archetype definition section converted to JSON.

4.2 Web Application

The application is made up of two primary parts: a backend section that is linked to a Github repository that houses a variety of template and archetype files, and a frontend section that was designed with React and the bootstrap library.

The frontend structure follows a general schema, featuring a sidebar that displays the different types of archetype files and a body section that showcases the files present in the repository. Upon clicking on a file, a subsequent GET request is triggered to retrieve its content. The content is then processed using the previously developed algorithm for conversion. Once the conversion is completed, the file's content is displayed in the body section, presented in a JSON format. The frontend interface empowers users to interact seamlessly with the files, easily allowing the visualization of any parameter and attribute, including any added archetypes with the "allow_archetype" keyword facilitating a smooth and intuitive user experience.



```
▾ "root" : { 6 items
  ▾ "archetype" : { 3 items
     "adl_version" : float 1.4
     "uid" : string "2790f093-d7dc-4ee8-8236-881f853ec9b7"
     "file_name" : string "openEHR-EHR-CLUSTER.address.v0"
  }
  ▾ "concept" : { 1 item
     "at0000" : string "Address"
  }
  ▾ "language" : { 2 items
     "original_language" : string "en"
    ▸ "translations" : {...} 2 items
  }
  ▾ "description" : { 2 items
    ▸ "original_author" : {...} 4 items
    ▸ "details" : {...} 3 items
  }
  ▾ "definition" : { 4 items
     "rmType" : string "CLUSTER"
    ▸ "occurrences" : {...} 2 items
    ▸ "node" : {...} 3 items
    ▸ "items" : [...] 12 items
  }
  ▾ "ontology" : { 1 item
    ▸ "term_definitions" : {...} 3 items
  }
```

Fig. 8. Application window after the retrieval and conversion to JSON of an archetype file.

5 Discussion

Support from emerging technologies is becoming an increasingly important factor in the delivery of healthcare services within institutional settings. As a direct result of this progression, new hospitalization support systems have emerged, which significantly contribute to an improvement in the standard of care provided to patients. The creation of an artifact with the purpose of enhancing the organization and archiving of clinical data and minimizing any adverse reactions that may result from improper data propagation in a clinical work environment was the primary impetus behind the work that has been discussed throughout the entirety of this paper.

The initial research question that was formulated served as the impetus for the development of the envisioned artifact. This was accomplished by researching a solution that could be most effectively implemented to improve this particular problem in healthcare data. This research made use of the information obtained from extensive research that was documented in the background section. The process of development included writing the preliminary descriptions and carrying out the essential steps; in particular, the development of a prototype platform and an envisioned REST API that makes use of a conversion algorithm for ADL files.

A web application that can retrieve files and display data in real time was used to validate the API's functionality. As a result, a standardized method for high-speed data transfer across any web platform was successfully established. However, despite the fact that the API will serve as the basis for future application development, in order for this solution to be effective, the API will need to be constantly updated so that it takes into account the most recent ADL syntax updates that are provided by openEHR. The JSON conversation algorithm that is currently being used needs to be updated in the future so that it can accept more versions of ADL, particularly the most recent 2.0 version. The web application also has a significant amount of room for improvement in terms of both its functionality and the quality of the experience it provides to users.

The prototype can only connect to a single repository on GitHub at the moment; however, future developments may give users the option to connect and interact with the online storage file systems of their choice. In addition, because mobile devices are becoming increasingly widespread, there is a market for the eventual development of a specialized mobile application. If these specific areas of development are addressed, the artifact will be able to realize its full potential and significantly contribute toward overcoming the data science interoperability challenge.

6 Conclusion

In this essay it is emphasized the importance of e-Health and the need for data science interoperability in the healthcare industry. Moreover, it shows an openEHR-based solution that enables interoperability and ensures a way to facilitate clinical data management using React and NextJS. The efforts present in this research have culminated in a novel strategy that will help healthcare practitioners and biomedical software engineers in achieving healthcare interoperability. Indeed, the main purpose of this study is to improve the management of clinical data and its preservation. This way one will be able to reduce, the unpleasant effects caused by incorrect data propagation. It is important to highlight that this study shows the steps to create a platform, illustrating a standardized way for high-speed data transfer across online platforms in the healthcare domain. By addressing these areas of improvement, one can fully realize its potential and make significant contributions to overcoming the challenge of data science interoperability in healthcare.

Acknowledgements. This work has been supported by FCT (Fundacão para a Ciência e Tecnologia) within the R&D Units Project Scope: UIDB/00319/2020.

References

1. Batra, S., Sachdeva, S., Bhalla, S.: Entity attribute value style modeling approach for archetype based data. Information **9**(1), 2 (2017)
2. Birkhead, G.S., Klompas, M., Shah, N.R.: Uses of electronic health records for public health surveillance to advance public health. Annu. Rev. Public Health **36**, 345–359 (2015)
3. Caligtan, C.A., Dykes, P.C.: Electronic health records and personal health records. In: Seminars in Oncology Nursing, vol. 27, pp. 218–228. Elsevier (2011)
4. Dash, S., Shakyawar, S.K., Sharma, M., Kaushik, S.: Big data in healthcare: management, analysis and future prospects. J. Big Data **6**(1), 1–25 (2019)
5. Goossen, W., Langford, L.H.: Exchanging care records using HL7 V3 care provision messages. J. Am. Med. Inform. Assoc. **21**(e2), e363–e368 (2014)
6. Guijarro, L.: Semantic interoperability in egovernment initiatives. Comput. Stand. Interfaces **31**(1), 174–180 (2009)
7. Heiler, S.: Semantic interoperability. ACM Comput. Surv. **27**(2), 271–273 (1995)
8. Iroju, O., Soriyan, A., Gambo, I., Olaleke, J., et al.: Interoperability in healthcare: benefits, challenges and resolutions. Int. J. Innov. Appl. Stud. **3**(1), 262–270 (2013)
9. Lawal, A., Ogbu, R.C.: A comparative analysis of agile and waterfall software development methodologies. Bakolori J. Gen. Stud. **11**(2), 1–2 (2021)
10. Mustra, M., Delac, K., Grgic, M.: Overview of the DICOM standard. In: 2008 50th International Symposium ELMAR, vol. 1, pp. 39–44. IEEE (2008)
11. Oliveira, D., Coimbra, A., Miranda, F., Abreu, N., Leuschner, P., Machado, J., Abelha, A.: New approach to an openEHR introduction in a portuguese healthcare facility. In: Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S. (eds.) WorldCIST'18 2018. AISC, vol. 747, pp. 205–211. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-77700-9_21
12. Oliveira, D., Ferreira, D., Abreu, N., Leuschner, P., Abelha, A., Machado, J.: Prediction of COVID-19 diagnosis based on openEHR artefacts. Sci. Rep. **12**, 12549 (2022)
13. openEHR Foundation: Archetype Definition Language (ADL) 1.4 Specification. https://specifications.openehr.org/releases/AM/latest/ADL1.4.html#_cadl_constraint_adl. Accessed 03 Aug 2023
14. Peffers, K., Rothenberger, M., Tuunanen, T., Vaezi, R.: Design science research evaluation. In: Peffers, K., Rothenberger, M., Kuechler, B. (eds.) DESRIST 2012. LNCS, vol. 7286, pp. 398–410. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29863-9_29
15. Sonnenberg, C., vom Brocke, J.: Evaluation patterns for design science research artefacts. In: Helfert, M., Donnellan, B. (eds.) EDSS 2011. CCIS, vol. 286, pp. 71–83. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33681-2_7
16. Stearns, M.Q., Price, C., Spackman, K.A., Wang, A.Y.: SNOMED clinical terms: overview of the development process and project status. In: Proceedings of the AMIA Symposium, p. 662. American Medical Informatics Association (2001)
17. Vaismoradi, M., Tella, S., Logan, P.A., Khakurel, J., Vizcaya-Moreno, F.: Nurses' adherence to patient safety principles: a systematic review. Int. J. Environ. Res. Public Health **17**(6), 2028 (2020)
18. Wittson, C.L., Benschoter, R.: Two-way television: helping the medical center reach out. Am. J. Psychiatry **129**(5), 624–627 (1972)

A Machine Learning Approach to Evaluating the Relationship Between Dental Extraction and Craniofacial Growth in Adolescents

Guillermo Hernández[1,2](✉), Alfonso González-Briones[1,2](✉), José Machado[3](✉), Pablo Chamoso[1,2](✉), and Paulo Novais[3](✉)

[1] BISITE Research Group, University of Salamanca, Edificio Multiusos I+D+i,
37007 Salamanca, Spain
[2] Air Institute, IoT Digital Innovation Hub, Carbajosa de la Sagrada,
37188 Salamanca, Spain
{guillehg,alfonsogb,chamoso}@usal.es
[3] Algoritmi Center/LASI, University of Minho, 4710-057 Braga, Portugal
{jmac,pjon}@di.uminho.pt

Abstract. There may be multiple reasons for tooth extraction, such as deep cavities, an infection that has destroyed an important portion of the tooth or the bone that surrounds it, or for orthodontic reasons, such as the lack of space for all the teeth in the mouth. In the case of orthodontics, however, there is a relationship between tooth extraction and the craniofacial morphological pattern. The purpose of this study is to establish whether such a relationship exists in adolescents and to evaluate it and to serve as a tool to support medical decision making. Machine Learning techniques can now be applied to datasets to discover relationships between different variables. Thus, this study involves the application of a series of Machine Learning techniques to a dataset containing information on orthodontic tooth extraction in adolescents. It has been discovered that by following simple rules it is possible to identify the need of treatment in 98.7% of the cases, while the remaining can be regarded as "limited cases", in which an expert's opinion is necessary.

Keywords: Machine Learning · Craniofacial Morphological Growth · Dental Extraction · Orthodontic treatments

1 Introduction

To successfully complete orthodontic treatment, tooth extraction may sometimes be necessary. The main reasons for this procedure are: to solve negative osteodental discrepancy (severe dental crowding), camouflage a horizontal skeletal discrepancy (class III or class II) or vertical discrepancy (vertical growth pattern or skeletal open bite), to improve the facial aesthetics of the patient in cases of protrusion or lip incompetence, etc. [7]. The present study determines

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 300–313, 2023.
https://doi.org/10.1007/978-981-99-7969-1_22

whether tooth extraction is necessary by examining the craniofacial morphology and the growth pattern. Orthodontists consider different factors when identifying negative osteodental discrepancy. Thus, the rules identified in this paper could support orthodontists' in making decisions. The following factors have been identified:

- Tooth factors: the size of the teeth (macrodontia), number of teeth (supernumerary), pathology (tooth decay, periodontal disease), abnormal position (ectopias, the inclination of the incisors).
- Arcade factors: narrow arcade, osteodental severe discrepancy, changes in the midline.
- Occlusal factors: open bite, molar, and canine class, increased overjet.
- Facial factors: aesthetics protrusive facial profile and vertical growth patterns.
- General factors: patient low growth potential and low degree of patient cooperation.

The presence of crowded and irregular anterior teeth is the most cited reason for seeking orthodontic treatment [25]. Crowding has been associated with vertical growth, lower incisor eruption, and increased vertical dentoalveolar eruption. One might expect crowding and facial divergence to be associated because divergence increases anterior vertical dentoalveolar eruption. Hyperdivergence results in the retroclination of the incisors, which may cause crowding by reducing the arch length.

The following techniques have been employed in this research for data analysis: statistical techniques such as mutual information and correlation analysis; visualization techniques such as kernel density estimation; non-supervised techniques such as association rules; and supervised learning techniques such as decision trees. A series of other techniques have been evaluated, however, it has been decided not to use them due to poor performance or complexity that did not manage to overcome the results obtained.

The remainder of the paper is structured as follows: the next section presents a brief review of related studies, as well as the craniofacial growth patterns that have been considered in this research and the corresponding data set. Section 3 presents some of the widely used techniques to analyse data in the dental sector. Section 4 describes the proposed analysis process. Section 5 presents the results and the conclusions drawn from the conducted research.

2 Biological Background

The experiments have been performed on a set of 303 patients. Nine craniofacial growth patterns were calculated for those patients. This is a group of patients with severe malocclusions presenting a negative osteodental discrepancy of more than 6 mm in the lower arch. These patients were undergoing orthodontic treatment in two private clinics.

The identifiers and patterns used to conduct this study will now be presented. The variables used are shown in parentheses.

- Tooth factors: the size of the teeth (macrodontia), number of teeth (supernumerary), pathology (tooth decay, periodontal disease), abnormal position (ectopias, the inclination of the incisors).
- Age (age): nominal variable of patients between 12 and 15 years of age. The treatment begins at this age.
- Sex (sex): numeric variable where 0 corresponds to a male and 1 to a female.
- Jaw Plane (JaPl) (Tweed, 1946) [27]: numeric variable, in degrees.
  The mandibular plane angle: formed by the mandibular plane and the Frankfort horizontal plane. The Frankfort plane: the point is drawn from the porion point to the orbital point. It is the basic horizontal reference line on the cephalometric tracing.
  Mandibular plane: a tangent to the lower edge of the mandibular connecting the Me point with the lowest point of the mandibular branch. It is a reference that describes the morphology and/or mandibular position. The junction of the front end with the facial plane forms the gnathion cephalometric point.
  Standard value: 26° at 9 years. Decreases 0.3° per year. S. D.: ± 4°.
  Interpretation: Low values correspond to brachyfacial patients, with a squared jaw. A high value corresponds to a dolichofacial biotype, and the jaw morphology indicates a very obtuse angle, a clockwise mandibular implant, or a combination of both.
- Vert Ricketts (verRic) (Ricketts, 1956) [21]: numeric variable, in S.D. Facial biotype is determined by the lateral cephalometric VERT index. The type and degree of the vertical growth of the lower third of the face are established numerically by this index. The growth is caused by the anterior or posterior rotation of the mandible. Other parameters such as gender and age are also considered. To obtain the result, five angles are taken into account to position the jaw: face depth, jaw plane, facial axis, lower face height, and jaw arch.
  To Ricketts, VERT is the vertical variation coefficient. It is obtained by calculating the arithmetic mean of the difference between the patient's value and the normal value for that age, divided by the standard deviation. Dolicho standard deviations are characterized by a negative sign. Deviations regarding brachy take a positive sign. If they remain within the norm, the value is 0. To calculate the facial biotype, it is first necessary to add the results of the 5 cephalogram measures. Finally, the resulting number is divided by 5,
  - Low to -1.5: Severe dolicho
  - From -1.5 to -1: Moderate dolicho
  - From -1 to -0.5: Soft dolicho
  - From -0.5 to 0.5: Mesofacial
  - From 0.5 to 1: Soft brachyfacial
  - From 1 to 1.5: Moderate brachyfacial
  - Over 1.5: Severe brachyfacial
- Jarabak Spheres (JaSp): numeric variable in % [16]. (Figure 1). Jarabak identifies three types of growth according to the general growth direction
  A) Counter-clockwise.
  B) Clockwise.
  C) Direct down.

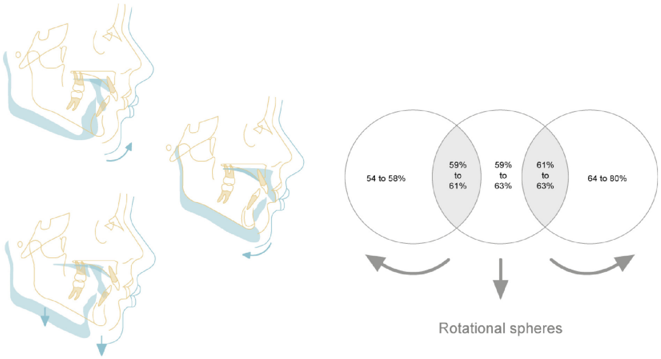


Fig. 1. Rotational Spheres (from Jarabak and Fizzell, 1971)

Patients placed in areas A and C have different growth characteristics, as illustrated in the diagram in Fig. 1. In the case of patients in A and C, there is no doubt as to whether the growth pattern is dolichofacial or brachyfacial. However, the growth percentage of the patients in B corresponds to what Jarabak called vertical growth; this type of growth may develop either dolichofacial or barchyfacial pattern, which is determined by the muscles (Fig. 2).

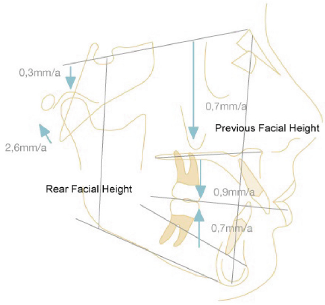
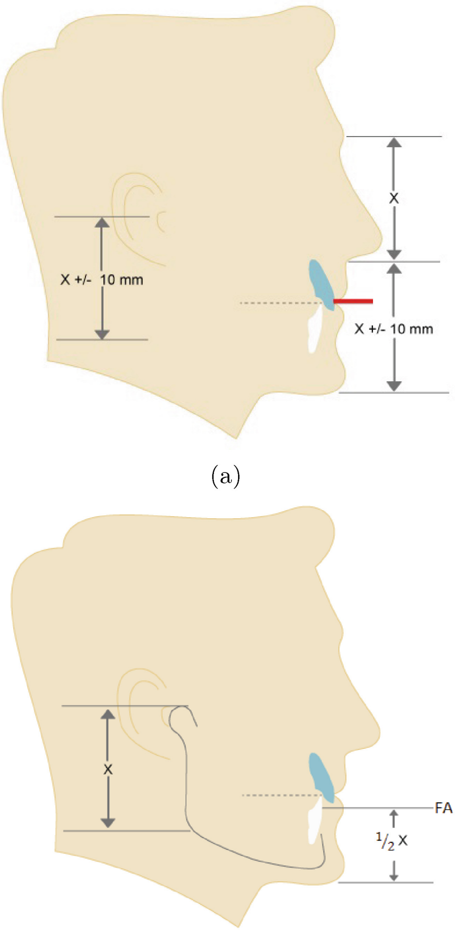


Fig. 2. Rear / Previous Facial Height (from Jarabak and Fizzel, 1971)

- Previous Facial Height (PrFH) / Rear Facial Height (ReFH): numeric variable in mm [3]. The anterior and posterior numerical facial height values are linear, not angular, and should be proportional to define a balanced face shape. Several reference points are used to measure them: Glabella, subnasal, soft chin, and condylar gonion. Their thirds should be symmetrical as shown in Fig. 3 A).
  The FA Point is used as a reference for hard tissue. The FA point is also called lower incisor Crown. The distance between this point and the hard tissue chin should be half the lower posterior third. If the face is balanced, the length of

the ramus should be twice the height of the symphysis measured from FA to the hard chin (Fig. 3 B). If the length of the branch is greater, facial growth will be horizontal, otherwise, it will be vertical.
- Facial Pattern (FacPat): nominal variables can take the following values, 0 for brachy, 1 for meso and 2 for dolicho.
- Type of Treatment (class): Nominal variable where 0 indicates no extraction of tooth and 1 indicates that there has been an extraction.



(a)

(b)

Fig. 3. Anterior /posterior relationship

3 Data Analysis in Dental Sector

Data analysis is a science that uncovers patterns. Machine Learning, or Deep Learning techniques are used to analyze data. The application of Deep Learning techniques requires a large amount of data, and makes it difficult to apply to the current study. The areas to which these techniques can be applied are diverse, ranging from energy optimization and industry 4.0 processes to education and medical care [6,9,10,22]. Data analysis in the dental sector has been a common practice for decades and it has been used to treat different diseases, and dental and mouth-related problems [28], as well as other general problems [19]. Statistical methodologies have made it possible to extract knowledge for better diagnosis and treatment [4,8].

Initial studies obtained data from questionnaires that were answered by patients [1,13]. Collecting a large amount of information from patients was difficult and there was a possibility that the patients' answers were imprecise or misleading. Consequently, the results of those studies were not entirely reliable. Fortunately, recent advances in computer technology have simplified the process of collecting data from patients. Moreover, current tools can accurately capture and store information about patients without having to rely on patients' opinions.

Similarly, data processing takes much less time; the advances in computing have increased the processing speed and the ability to apply artificial intelligence algorithms. Therefore, there is a tendency to use this type of methodology in the dental sector.

This is why machine learning methodologies have become more prevalent in the dental sector and have been applied in numerous studies in recent years. For example, machine learning-based methodologies have been employed in the development of a recommendation system for dental care, [15], dental age estimation [26] or for the study of the factors influencing the prognosis of a dental implant [11,18].

Among the multiple technologies and methodologies for data analysis, a specific subset has finally been chosen and used in this study. This subset combines statistical methodologies with unsupervised and supervised learning methodologies. More specifically, mutual information [5], correlation analysis [12], kernel density estimation [24], association rules [14], and decision trees [20].

4 Analysis

The sample comprised 303 subjects aged between 12 and 15 years, of which 136 were men and 167 were women. The dataset was divided into 3 groups by biotypes or facial patrons: 75 with a brachyfacial pattern, 91 with a mesofacial pattern, and 137 with a dolichofacial pattern.

Both patients and their legal representatives (parents in all cases) signed an agreement by which they provided (disinterestedly) the data for this study, and could not disseminate them for other purposes. For this reason, the dataset cannot be made public.

The demographic characteristics of the dataset are represented in Fig. 4. It is justified below that this factor does not have a significant influence on the class (within the range of values in the dataset in the case of age).

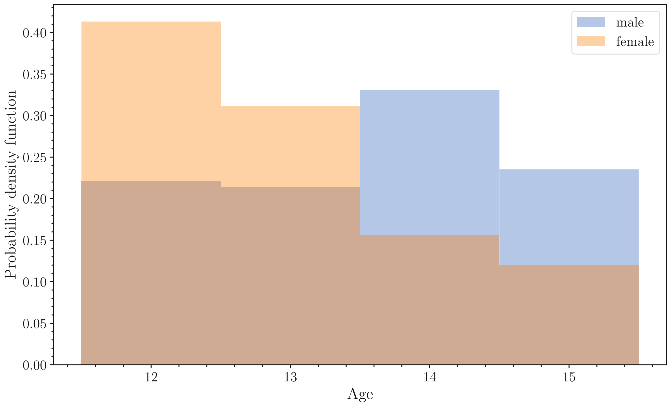


Fig. 4. Gender and age distributions in the dataset.

Figure 5 shows the values of the mutual information [17,23] of each of the attributes in the dataset with the target variable, ordered by such values. As can be seen there, age and gender do not have any significant influence on the class, which suggests that the demographic characteristics of the sample (Fig. 4) do not influence the study. On the other hand, it is observed that most of the attributes have a marked relationship with the class, giving rise to the represented ranking, which justifies trying to predict the presence or absence of treatment using a subset of attributes.

The two most relevant features in the mutual information analysis can provide a partial explanation of the presence or absence of treatment. These characteristics were chosen systematically for simplicity of extraction for this first study. This can be observed in Fig. 6, where the distributions for each of the series practically do not overlap.

Since the purpose of the research is to assist in medical diagnosis, and the physician must weigh the importance of the various errors, it is difficult to reduce the performance of the proposed system to a single metric. Instead, several alternatives are offered, as well as the confusion matrix which provides a complete evaluation of the system. However, it is important to realize that the attributes present in the dataset are not independent of each other and that some of them may provide the same information about the target variable. A simple means of analyzing this relationship is by using Pearson's correlation coefficient between the pairs of attributes, as represented in Fig. 7. It is convenient to analyze the correlation of the variables involved in order to identify possibly redudant information, which hinders the performance of the models and makes their interpretation difficult.

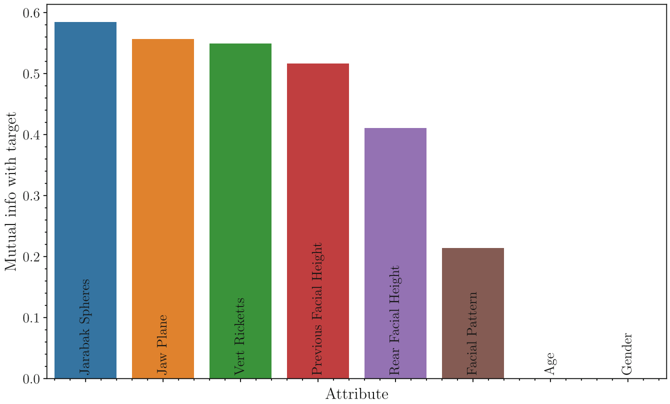


Fig. 5. Mutual information for each of the attributes in the dataset with the presence or absence of treatment.

It can be seen from the correlation matrix that several strongly related variables are worth studying independently. Figure 8 illustrates the interrelations between the different variables, as well as the ability to separate the classes they provide in pairs.

Before proceeding to supervised modeling, the dataset has been explored for association rules. Association rules are found by analysing data for frequent if/then patterns, thanks to which it is possible to assess their reliability. The metrics used in the search for association rules include: Support, which measures the frequency with which the consequent of the rule appears when the consequent appears. Confidence, which indicates the number of times both the antecedent and the consequent appear in the dataset. Lift, measures the extent to which rule frequency is greater than it would be in a statistically independent condition; values greater than one indicate an increase from that hypothesis. In this study, the widely used Apriori algorithm, proposed by Agrawal and Srikant in 1994 [2], has been employed for association rules extraction.

To obtain a set of useful rules, the search has been restricted to those with confidence greater than 0.5 and with a lift greater than 1.1, using only categorical attributes. The resulting rules are summarized in Tab. 9. Although it is also possible to extract rules by bucketing continuous attributes, the resulting rules would not provide as much knowledge about the dataset.

As it can be seen in, there are strong relationships between the facial pattern and the class, as well as the Dolicho pattern and the presence of treatment. The other facial patterns are associated with the absence of treatment.

So far, the analysis results show that there are multiple relationships between the attributes and the presence or absence of treatment. To determine, numerically, the probability that a patient requires treatment, it is necessary to adequately represent the knowledge. This is difficult because of how the attributes are interrelated. Fortunately, multiple supervised learning algorithms can offer

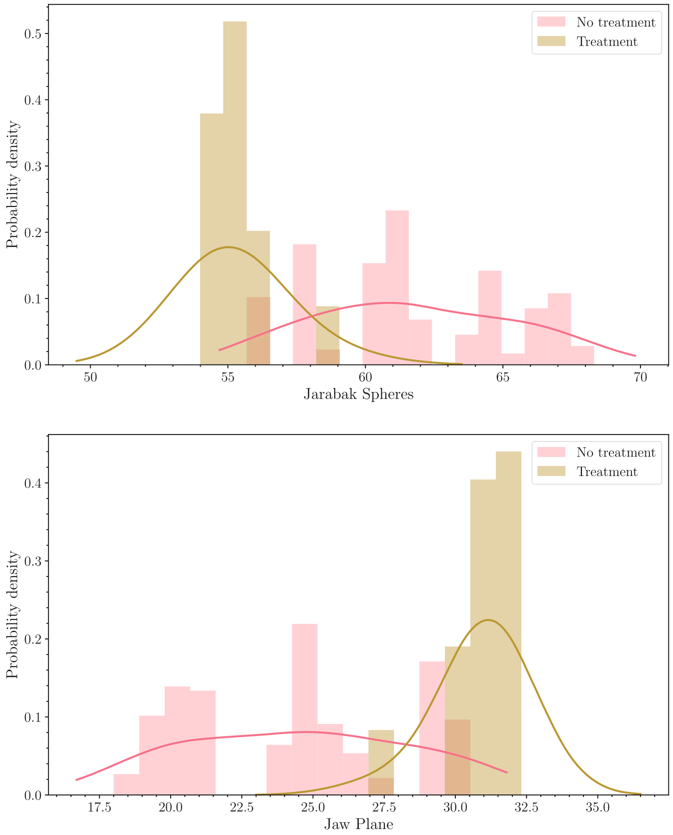


Fig. 6. Probability distributions of the two attributes that are the most relevant to the target variable. The figure shows histograms (shaded regions), as well as the kernel density estimator with manually chosen bandwidths, to illustrate the probability distribution, distinguishing the distributions according to class (legend).

a solution to this problem. Decision trees integrate statistical and mathematical techniques to facilitate the selection of optimal criteria for dataset description. Thanks to its training, the decision trees algorithm can provide human-understandable representations. Figure 10 shows a decision tree which was fit with the dataset. It can be observed that the limit value of the Vert Rickets variable is -1.25, making it possible to identify the majority of cases in which treatment is necessary. In cases where the value crosses the threshold, the algorithm considers the value of the rear facial height and can identify some of the cases in which treatment is necessary (less than 70.0, left node). Nevertheless, several limit cases remain in which the algorithm cannot determine whether treatment is necessary or not (more than 70.0, right node).

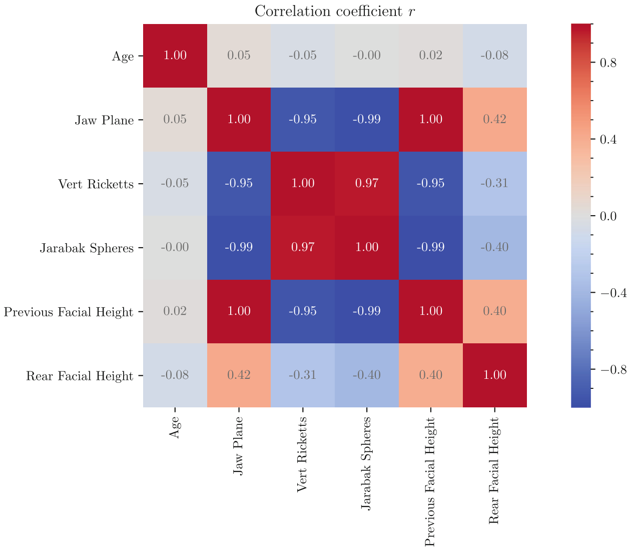


Fig. 7. Pearson correlation coefficients between pairs of variables in the dataset.

Despite the limit cases, the tree is a highly optimal model for representing all the data because it only classified 4 patients (1.32 %) erroneously. The actual evaluation of the tree must be made using a procedure which avoids biasing due to evaluating the tree with the training data itself. To prevent this, a 5-fold evaluation has been carried out, the tree has been clinically trained with 4/5 of the dataset and evaluated on the remaining 1/5. The confusion matrix obtained with this method is

$$\begin{pmatrix} 205 & 4 \\ 0 & 94 \end{pmatrix},$$

where rows refer to the actual class (presence and absence of treatment) and columns refer to the prediction of the tree (in the same order).

It can be seen that the evaluation results are consistent with those of the tree shown in Fig. 10. Thus, the algorithm's "failure rate" is estimated to be approximately 1.32 %. Failure occurs when the algorithm misclassifies the patients, suggesting that there is no need for treatment while in reality, an expert would have recommended treatment.

Another possible interpretation of the misclassified instances is that the "impure" leaf node in Fig. 10 (bottom right) actually represents cases where the decision is not so clear and that both results may be acceptable in the expert's opinion.

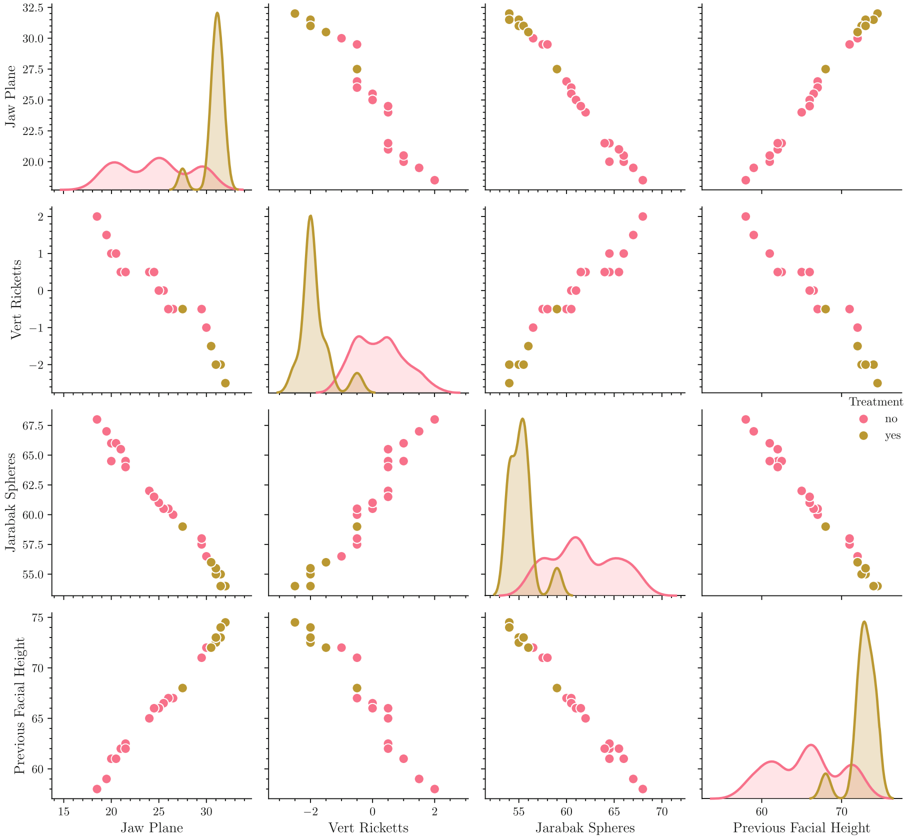


Fig. 8. Representation of variables in pairs in relation to class (color). The plots in the diagonal of the grid represent the distributions using a kernel density estimator.

	antecedent	consequent	support	confidence	lift
2	Facial Pattern = dolicho	Treatment = yes	0.287129	0.635036	2.046979
3	Treatment = yes	Facial Pattern = dolicho	0.287129	0.925532	2.046979
1	Facial Pattern = brachy	Treatment = no	0.247525	1.000000	1.449761
4	Facial Pattern = meso	Treatment = no	0.277228	0.923077	1.338241
0	Facial Pattern = brachy	Gender = male	0.125413	0.506667	1.128824

Fig. 9. Association rules extracted by the Apriori algorithm [2] using the criteria detailed in the main text (lift > 1.1, confidence > 0.5).

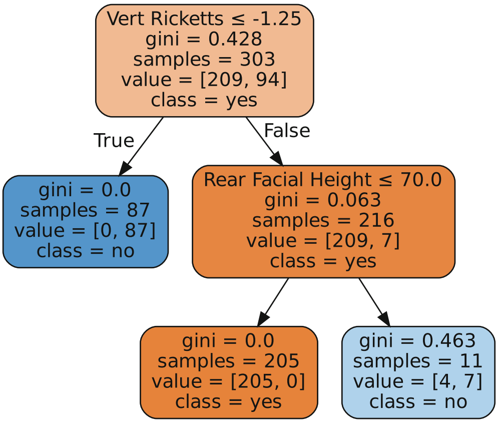


Fig. 10. Decision tree trained with the whole dataset. The first line of a non-leaf node indicates the decision criterion. The gini value describes the tree impurity (Gini impurity). All other information in the nodes can be interpreted directly.

5 Conclusions

The purpose of this study was to investigate whether there is a correlation between the craniofacial morphology and the need for tooth extraction as part of orthodontic treatment. According to our findings, non-extractive therapy is preferable. This is because the extraction of premolars tends to rotate the jaw counter-clockwise due to the closure of post-wedge mesializing molars. Counter-clockwise rotation makes the chin project forward which worsens the appearance of the profile; it also tends to close the bite, to reduce the lower face height, and to retract the lip protrusion.

Our analysis has shown that there are multiple relationships between the attributes and the presence or absence of treatment. To determine, numerically, the probability that a patient requires treatment, it is necessary to adequately represent the knowledge. This is difficult because of how the attributes are interrelated. However, the analysis has shown that it is easy to control the depth of the bite and the incisive and lip protrusion in patients with a mesofacial pattern. Thus, it is possible to have good treatment control with or without dental extractions because in these patients. Machine Learning techniques have been applied to identify the relationship between craniofacial parameters and the need for tooth extraction. By applying simple rules, it has been possible to identify the need of treatment in 98.7 % of the cases, while the remaining can be regarded as “limit cases”, which must be evaluated by a human expert.

Acknowledgement. The work of José Machado and Paulo Novais has been supported by FCT - Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020

References

1. Agerberg, G., Carlsson, G.E.: Chewing ability in relation to dental and general health: analyses of data obtained from a questionnaire. Acta Odontol. Scand. **39**(3), 147–153 (1981)
2. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference Very Large Data Bases, VLDB, vol. 1215, pp. 487–499 (1994)
3. Andrews, L.: The six elements of orofacial harmony. Andrews J. **1**, 13–22 (2000)
4. Bearden, E., Robinson, K., Deis, M.: A statistical analysis of dental hygiene students' grades in online and on-campus courses and performance on the national board dental hygiene exams. J. Dental Hygiene **76**(3), 213–217 (2002)
5. Bennasar, M., Hicks, Y., Setchi, R.: Feature selection using joint mutual information maximisation. Expert Syst. Appl. **42**(22), 8520–8532 (2015)
6. Candanedo, I.S., Nieves, E.H., González, S.R., Martín, M.T.S., Briones, A.G.: Machine learning predictive model for industry 4.0. In: Uden, L., Hadzima, B., Ting, I.-H. (eds.) KMO 2018. CCIS, vol. 877, pp. 501–510. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95204-8_42
7. Dimberg, L., Arnrup, K., Bondemark, L.: The impact of malocclusion on the quality of life among children and adolescents: a systematic review of quantitative studies. Eur. J. Orthod. **37**(3), 238–247 (2014)
8. Edgar, H.J.H.: Dentitions, distance, and difficulty: a comparison of two statistical techniques for dental morphological data. Dental Anthropol. **17**(2), 55–62 (2004)
9. González-Briones, A., Hernández, G., Corchado, J.M., Omatu, S., Mohamad, M.S.: Machine learning models for electricity consumption forecasting: a review. In: 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), pp. 1–6. IEEE (2019)
10. González-Briones, A., Hernández, G., Pinto, T., Vale, Z., Corchado, J.M.: A review of the main machine learning methods for predicting residential energy consumption. In: 2019 16th International Conference on the European Energy Market (EEM), pp. 1–6. IEEE (2019)
11. Ha, S.R., et al.: A pilot study using machine learning methods about factors influencing prognosis of dental implants. J. Adv. Prosthodont. **10**(6), 395–400 (2018)
12. Härdle, W.K., Simar, L.: Canonical correlation analysis. In: Applied Multivariate Statistical Analysis. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-45171-7_16
13. Helöe, L.A.: Comparison of dental health data obtained from questionnaires, interviews and clinical examination. Eur. J. Oral Sci. **80**(6), 495–499 (1972)
14. Heritage, J., McDonald, S., McGarry, K.: Integrating association rules mined from health-care data with ontological information for automated knowledge generation. In: Chao, F., Schockaert, S., Zhang, Q. (eds.) UKCI 2017. AISC, vol. 650, pp. 3–16. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-66939-7_1
15. Hung, M., et al.: Development of a recommender system for dental care using machine learning. SN Appl. Sci. **1**(7), 785 (2019)
16. Jarabak, J.R., Fizzell, J.A.: Technique and Treatment with the Light-wire Appliances: Light Differential Forces in Clinical Orthodontics. Mosby (1963)
17. Kraskov, A., Stögbauer, H., Grassberger, P.: Erratum: estimating mutual information [phys. rev. e 69, 066138 (2004)]. Phys. Rev. E **83**(1), 019903 (2011)
18. Lee, J.H., Kim, D.H., Jeong, S.N., Choi, S.H.: Detection and diagnosis of dental caries using a deep learning-based convolutional neural network algorithm. J. Dent. **77**, 106–111 (2018)

19. Locker, D., Shapiro, D., Liddell, A.: Negative dental experiences and their relationship to dental anxiety. Community Dent. Health **13**(2), 86–92 (1996)
20. Ma, L., Destercke, S., Wang, Y.: Online active learning of decision trees with evidential data. Pattern Recogn. **52**, 33–45 (2016)
21. Ricketts, R.M.: Planning treatment on the basis of the facial pattern and an estimate of its growth. Angle Orthod. **27**(1), 14–37 (1957)
22. Rivas, A., Fraile, J.M., Chamoso, P., González-Briones, A., Rodríguez, S., Corchado, J.M.: Students performance analysis based on machine learning techniques. In: Uden, L., Liberona, D., Sanchez, G., Rodríguez-González, S. (eds.) LTEC 2019. CCIS, vol. 1011, pp. 428–438. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20798-4_37
23. Ross, B.C.: Mutual information between discrete and continuous data sets. PLoS ONE **9**(2), e87357 (2014)
24. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Routledge (2018)
25. Svedström-Oristo, A.L., Pietilä, T., Pietilä, I., Vahlberg, T., Alanen, P., Varrela, J.: Acceptability of dental appearance in a group of finnish 16-to 25-year-olds. Angle Orthod. **79**(3), 479–483 (2009)
26. Tao, J., et al.: Dental age estimation: a machine learning perspective. In: Hassanien, A.E., Azar, A.T., Gaber, T., Bhatnagar, R., F. Tolba, M. (eds.) AMLTA 2019. AISC, vol. 921, pp. 722–733. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-14118-9_71
27. Tweed, C.H.: The frankfort-mandibular plane angle in orthodontic diagnosis, classification, treatment planning, and prognosis. Am. J. Orthod. Oral Surg. **32**(4), 175–230 (1946)
28. Williams, A., Bower, E., Newton, J.: Research in primary dental care part 6: data analysis. Br. Dent. J. **197**(2), 67 (2004)

AI and Creativity

Generative Design Optimization for Tourism-Oriented Souvenir Development: Blending Art and Science into Tourism Industry

Somlak Wannarumon Kielarova[1,2](✉) and Prapasson Pradujphongphet[1,2]

[1] iD3-Industrial Design, Decision and Development Research Unit, Faculty of Engineering, Naresuan University, Phitsanulok 65000, Thailand
somlakw@nu.ac.th
[2] Department of Industrial Engineering, Faculty of Engineering, Naresuan University, Phitsanulok 65000, Thailand

Abstract. In this paper we bring the benefits of computer-aided design algorithm into art and design to develop souvenirs as tourism products. The paper proposes a new methodology to design and develop souvenirs using interactive-generative design method in conceptual phase of design. Parametric design, shape grammar, and genetic algorithm are integrated into an interactive design platform to accomplish a tourism-oriented labelling process in product design. Conceptual design is an early stage of design process, where designers require exploring numbers of feasible design ideas. The proposed method enables supporting designers to automatically generate a large variety of alternative design ideas. Design exploration and design evaluation are enhanced by the way of the interaction of designers with the process. The design solutions are presented in the form of three-dimensional models, which can be further used in the successive phases in design and production processes. A specific case study of tourism-oriented souvenir development in Phitsanulok, a city and one of the attractive historical-cultural destination in Thailand, is illustrated as an example of concept shape design in souvenir development.

Keywords: Generative Design · Genetic Algorithm · Interactive Generative Design · Shape Grammar · Souvenir Development · Tourism Product · Tourism-Oriented Labelling · Tourism Industry

1 Introduction

Tourism is an economic contributor to Thailand. In this paper, we create interactive-generative design algorithm to create souvenirs as tourism products. The souvenir could be considered as a part of the travel experience, generate high income, and promote tourism. From the preliminary survey, most tourists bought souvenirs to symbolize and memorialize those experiences in the destinations, when they return home.

Souvenir product development employs the same process as general product development. It begins with conceptual design, through detailed design, prototyping, until

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 317–324, 2023.
https://doi.org/10.1007/978-981-99-7969-1_23

production. In conceptual design stage, the designer explores feasible product shapes according to several customer needs and product specifications.

The paper proposes a new methodology to design souvenirs using interactive-generative design method in the conceptual design phase, which integrates destination identity, folk art identity and industrial design conditions into the computer-aided design tool. Parametric design, genetic algorithm and shape grammar are integrated into a single platform to accomplish a tourism-oriented labelling process in souvenir product design.

The paper organized into five sections. The next section provides the related theories and research works. In Sect. 3, the proposed methodology is described. Section 4 explains the experimental results and discussions. Finally, this research work is summarized in the last section.

2 Literature Review

In this research, several related theories are reviewed as follows:

2.1 Parametric Design

Parametric design is a design process in which design attributes and components are parameterized and then formed according to algorithmic processes. The input parameters are fed into the algorithms to generate the design outcome. This method uses parameters and rules to determine the relationship between design intent and design response [1]. Parametric designs have been used in various purposes and obviously used in architectural design; for example, search tool [2], design exploration [3], design generation [4], design optimization [5], and product serialization [6].

2.2 Interactive Generative Design

Generative Design (GD) system is a computer-based tool that employs computational capabilities to be mainly used for supporting designers in design exploration and design generation. GD enables the designer to explore a large solution space, which provides a larger range of design alternatives than by manual production [7]. GD also assists the designer to automate some parts of design process [8]. GD has been used in various design applications, for example, coffee maker [9], chair [10], Coca Cola bottles [11], jewelry design [12], etc.

Interactive-generative design (IGD) is a method in which allows human to work with the generative design [13]. These works are such as selection, evaluation, giving direction, etc. Therefore, interactive generative design can generate and achieve the desired solutions based on user inputs and interactions.

2.3 Genetic Algorithm

In GA a population of individuals or phenotypes is evolved toward better solutions as part of the optimization problem. Each individual has a set of properties called chromosomes or genotype, which can be mutated and crossed over. Solutions can be encoded in binary code, or real numbers [14].

The evolutionary process starts from a population of randomly generated individuals. It is an iterative process with the population with each iteration called a generation. In each generation, the fitness of every individual in the population is evaluated. The fitter individuals are stochastically selected from the current population, and each individual's genome is modified by recombination and mutation, to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. GA typically terminates when either a maximum number of generations has been reached, or a satisfactory fitness level has been achieved for the population.

2.4 Shape Grammar

Shape grammar creates a design according to sets of shape rules, by performing calculations with shapes [15, 16]. The process of shape grammar begins with giving a shape and iterating it with a set of shape rules. New shape is created by finite composition of geometric elements (such as points, lines, curves, and planes). Those elements can be applied several times with any transformation. Shape grammar was designed by formalizing the spatial relations between their elements. Shape rules are applied to describe transformations of shapes [17]. The reasons that using shape grammar in this research, because shape grammar is easily modified to create a new design language by the small changes in shape rules [18].

2.5 Tourism Product Development

Tourism product consists of five elements: physical plant, service, hospitality, freedom of choice and involvement [19]. To create a tourism product, it is not just simply combining those five elements together. Nevertheless, it needs harmonious integration and interaction among all the elements [19].

Tourism product development, in this research, it is scoped in souvenir products in which uses the same process as general product design and development. Product development process begins with conceptual design, through detailed design, prototyping, until production [20]. Nevertheless, in souvenir product developments they are associated with destinations and their folk tales, histories, and local materials. The guidelines for the souvenir development that reflecting the tourism destinations have been suggested in [21].

3 An Interactive Generative Design for Souvenir Product Development

The development of an interactive generative design tool integrated into the conceptual phase of souvenir product design consists of eight main steps:

(1) The identification of features and relationships of souvenir products in destinations. We need to collect information and data associated to the attraction of destinations for tourists, which typically depend on physical and environmental attributes and socio-cultural attributes of destination. Product design market analysis is needed to understand the expectations of tourists, as well as, to analyze the market whether is mass tourism or niche tourism, and to define the target groups. The mentioned information is used to capture the features and their relationships that represent the specific destination.

(2) The specification of the souvenir product to be designed by the proposed tool.

After the feature and their relationships were identified, we establish a set of target specifications of souvenir product. In this step, we obtain the broad characters and the metrics of the products.

(3) The creation of preliminary description on the model structure of the selected product. The model structure is a representation scheme of the target product shape. We use simple geometric shapes like as boxes and spheres to represent major parts that establish the whole target product shape. The main aim of setting the model structure is to control the positions and the shapes of the product parts.

(4) The creation of product vocabulary and product shape grammar. The product vocabulary is constructed based on the product structure and product shape grammar. The product vocabulary allows changing the positions of the major parts into the product structure. The product shape grammar with the uses of shape rules can create new product shapes.

(5) The interaction between designer and computer system. We place human designer at the center of the proposed design system to collaboratively performs his/her tasks with the system. Designer provides a set of inputs and utilizes the generated shapes to easily initialize his conceptual design.

(6) The generation of product shapes. We apply shape grammar method to generate new product shapes. It needs to define the initial shape, the scope, and limitations of shape grammar of the selected souvenir product. It also needs to define shape grammar rules that consist of initial rules, shape transformation rules, and termination rules [13].

(7) The optimization of positions of points and curves. We use genetic algorithm (GA) to optimize the positions of points and curves to automatically generate the product shapes and the design elements.

(8) The selection process. In the final stage, the IGD presents the generated product shapes to users or designers. Users can either make final decision to choose some desired shapes or start a new generation trial.

4 A Case Study of Souvenir Product Development in Phitsanulok

The proposed system was developed using Grasshopper plug-in [22] and Rhinoscript in Rhinoceros software [23]. By incorporating between parametric design and shape grammar, the proposed interactive-generative design tool is developed to accomplish a tourism-oriented labelling process in souvenir product design.

In the first step, we collect information and data associated to the attraction of destinations, here is Phitsanulok, a city and one of the attractive historical-cultural destination in Thailand. We also considered the famous tradition and culture in the destination. From the data collection and analysis, we found that one of the famous temples, Phra Si Ratana Temple, is one of the most favorite destinations in Phitsanulok that tourists always come to visit. It is one of the symbolic places of Phitsanulok. We have studied the feasible elements that can represent the temple. One of the most interesting elements is the Thai motif on the main entrance door of the vihara, as shown in Fig. 1.

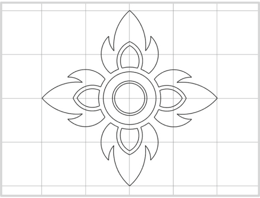


Fig. 1. Thai motif selected to use in this souvenir development.

In product design market analysis, this destination is famous and can be considered as mass tourism. We applied the feature of the Thai motif in Fig. 1, to develop souvenir. We develop the souvenir product in forms of jewelry.

The souvenir pendant was designed and developed the product structure and its initial shape, shown in Fig. 2.

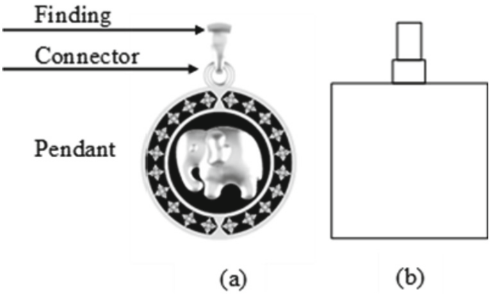


Fig. 2. (a) product structure of souvenir pendant; (b) its initial shape

Shape grammar of the souvenir pendant with the uses of shape rules can create new product shapes. An example of shape grammar application is illustrated in Fig. 3. Shape grammar development includes the construction of shape rules, control mechanisms, and object representation. Shape grammars and shape rules in this study were derived from the experimental investigations of souvenir jewelry design.

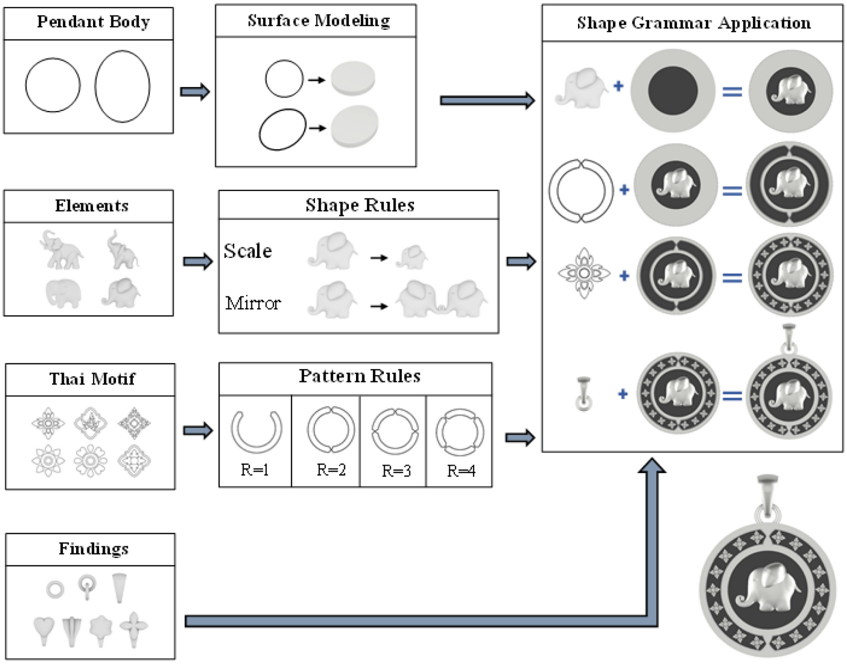


Fig. 3. An example of the defined shape grammar of souvenir pendant.

The proposed IGD system can generate number of various product shapes according to the defined shape grammar. Genetic algorithm (GA) is used to optimize the positions of points and curves in automatically generating the design elements. The IGD performs the generation of the souvenir shapes according to the input parameters such as the symbolic-tourism parameters relating to the target destination and user-based evaluation

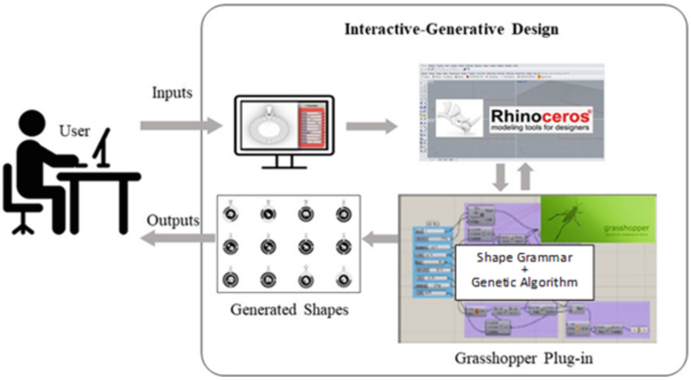


Fig. 4. The proposed interactive-generative design system for generating the Phitsanulok souvenir pendants.

and selection. The IGD presents the generated product shapes to users or designers. Users can either make final decision to choose some preferred shape or start a new generation trial. The described process illustrated in Fig. 4.

From the experimental case study, the feedbacks from five designers confirm the practicality of the method. It has potential to enhance the system by more testing and it can extend to other product design applications.

5 Conclusions

This paper proposed a new method for designing and developing souvenir product. The method is developed based on interactive-generative design process, which considers product shape as the strategic parameters of the tourism-oriented souvenir development. The proposed IGD system is used for souvenir design and development with the limited functionality design, like jewelry product. The IGD has been tested using Phitsanulok souvenir pendant design. The proposed design system considers both qualitative and quantitative parameters within the single framework. The computational time used in the design computing is reasonable with varying by the different users and set of inputs. The future research will include the tests of the proposed system with different tourism attributes. Furthermore, the proposed framework can be applied to other design problems.

Acknowledgement. This research work was supported by Naresuan University (NU) and National Science, Research and Innovation Fund (NSRF)" Grant No. R2566B068. The authors would like to gratefully thank all participants for their collaborations in this research.

References

1. Woodbury, R.: Elements of Parametric Design, 1st edn. Routledge, Oxon (2010)
2. Motta, E.: Reusable Components for Knowledge Modelling. IOS Press, Amsterdam (1999)
3. Turrin, M., Buelow, P.V., Stouffs, R.: Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms. Adv. Eng. Inform. **25**, 656–75 (2011)
4. Monizza, G.P., Raucha, E., Matt, D.T.: Parametric and generative design techniques for mass-customization in building industry: a case study for glued-laminated timber. In: 27th CIRP Design 2017, pp. 392–397. Procedia CIRP (2017)
5. Holzer, D., Hough, R., Burry, M.: Parametric design and structural optimisation for early design exploration. Int. J. Archit. Comput. **5**, 625–643 (2007)
6. Sun, B., Huang, S.: Realizing product serialization by Grasshopper parametric design. In: 7th International Forum on Industrial Design, IOP Conference Series Materials Science Engineering (2019)
7. Krish, S.: A practical generative design method. Comput. Aided Des. **43**, 88–100 (2011)
8. Singh, V., Gu, N.: Towards an integrated generative design framework. Des. Stud. **33**, 185–207 (2012)
9. Agarwal, M., Cagan, J.: A blend of different tastes: the language of coffeemakers. Environ. Plann. B. Plann. Des. **25**, 205–226 (1998)
10. Brintrup, A.M., Ramsden, J., Takagi, H., Tiwari, A.: Ergonomic chair design by fusing qualitative and quantitative criteria using interactive genetic algorithms. IEEE Trans. Evol. Comput. **12**, 343–354 (2008)

11. Ang, M.C., Chau, H.H., Mckay, A., Pennington, A.D.: Combining evolutionary algorithms and shape grammars to generate branded product design. In: Gero, J.S. (ed.) Design Computing and Cognition 2006, pp. 521–39. Springer Netherlands, Dordrecht (2006). https://doi.org/10.1007/978-1-4020-5131-9_27
12. Sansri, S., Kielarova, S.W.: Multi-objective shape optimization in generative design: art deco double clip brooch jewelry design. In: Kim, K., Kim, H., Baek, N. (eds.) IT Convergence and Security 2017. LNEE, vol. 449. Springer, Singapore. https://doi.org/10.1007/978-981-10-6451-7_30
13. Kielarova, S.W., Pradujphongphet, P., Bohez, E.L.J.: New interactive-generative design system: hybrid of shape grammar and evolutionary design - an application of jewelry design. Adv. Swarm Comput. Intell. **9140**, 302–313 (2015)
14. Whitley, D.: A genetic algorithm tutorial. Stat. Comput. **4**, 65–85 (1994)
15. Stiny, G., Gips J.: Shape Grammars and the Generative Specification of Painting and Sculpture. IFIP Congress 71, pp. 125–35. Yugoslavia (1971)
16. Tapia, M.: A visual implementation of a shape grammar system. Environ. Plann. B. Plann. Des. **26**, 59–73 (1999)
17. Prats, M., Lim, S., Jowers, I., Garner, S.W., Chase, S.: Transforming shape in design: observations from studies of sketching. Des. Stud. **30**, 503–520 (2009)
18. Al-kazzaz, D.A., Bridges, A.H.: A framework for adaptation in shape grammars. Des. Stud. **33**, 342–356 (2012)
19. Smith, S.L.J.: The tourism product. Ann. Tour. Res. **21**, 582–595 (1994)
20. Ulrich, K., Eppinger, S., Yang, M.C.: Product Design and Development. 7th edn. McGraw Hill (2019)
21. Sangchumnonga, A., Nubwandeeb, T., Lertpatcharapongc, R., Sangpiroond, T.: Souvenir product development to promote tourism in Klong Rang Chorakhe, Thailand. Int. J. Innov. Creat. Change **14**, 1228–1244 (2020)
22. Grasshopper-Algorithmic Modeling for Rhino. http://www.grasshopper3d.com/. Accessed 20 April 2022
23. Rhinoceros 3D Modelling Software. https://www.rhino3d.com/. Accessed 20 April 2022

A Computer Scientist's View on Machine Learning and Creativity

Maarten H. Lamers(✉)

Creative Intelligence Lab, LIACS, Leiden University, Leiden, The Netherlands
m.h.lamers@liacs.leidenuniv.nl

Abstract. The recent surge in deployment and use of generative machine learning models has sparked an interest in the relationships between AI and creativity, or more specifically into the question and debate of whether machines can exhibit human-level creativity. This is by no means a new discussion, going back in time decades if not centuries. The debate was approached from multiple angles, and a general consensus was not yet reached. In this position paper, we present the long-standing debate as it formed across various fields such as cognitive science, philosophy, and computing, approaching it mainly from a historical perspective. Along the way we identify how the various views relate to recent developments in machine learning models and argue our own position regarding the question of whether machines can exhibit human-level creativity. As such we aim to involve computer scientists and AI practitioners into the ongoing debate.

Keywords: machine learning · creativity · artificial intelligence · generative AI

1 Introduction

It will be clear to anyone following the news or browsing the internet that generative artificial intelligence is causing quite a stir at this moment in time. Within this category of AI systems fall text-based large language models (LLM's). But also text-to-image generative AI models, that generate images when prompted with text, or generators that create images within a particular class without requiring prompt specification (e.g. thispersondoesnotexist.com). Music generation AI enables the automated composition of music, from textual prompts or other forms of instruction. And altogether, other output modalities can also be generated with contemporary machine learning models.

Automated generation of various forms of content is not new. Under the nomen of *procedural content generation*, many systems were built and models proposed in the decades past. From automated creation of visual artworks (Cohen 2016; Colton 2012) to procedurally generated games and gameplay (Cook et al. 2017), and too much else to enumerate. So, what makes for the current humdrum and media attention?

Firstly, the algorithms that underly generative AI systems have changed in the last few years, just as one could argue that AI itself has changed in the same period. In the decades before 2010 relatively much of AI work was about algorithms other than

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 325–335, 2023.
https://doi.org/10.1007/978-981-99-7969-1_24

based on neural network theory. Expert systems, evolutionary computation, swarm optimization, or custom-designed algorithms were commonly used for models underlying generative tools and other real-world applications. Since the advent of deep (neural network) learning, most of the AI landscape has become about such algorithms. It may even appear from the media that this class of machine learning systems is the only one worthy of consideration.

Notwithstanding continued research into other algorithms, deep learning models have become the *de facto* norm for AI model choice, even though also within this class of algorithms, very many model choices are available. This is in part fueled by their general availability to businesses and individuals. Moreover, the size or potential complexity of these models has increased to a level at which their functioning is very impressive and even challenging human imagination. It appears as if any conceivable pattern hidden in data can be modelled by deep neural network learning models, that can subsequently be used to generate new output to any degree of realism and detail.

Similarly, the high quality of output further fuels a tendency to project human qualities onto them. Early psychological experiments have demonstrated this innate human tendency to project intention and emotion onto abstract artefacts (e.g. Heider & Simmel 1944). Now image generators can autonomously produce pictures much better than average humans can, and LLMs like GPT-4 appear to understand difficult problems. Moreover, since they are initiated by goal-directed prompting, not stepwise specification of operation (a distinction that we computer scientists recognize from old debates surrounding procedural, functional and logical programming paradigms), our interaction with them appears more human than we tend to expect from machines. Resultingly, it is easy to instinctively attribute to them human qualities as understanding, intention, reasoning, and *creativity*.

It is upon the assignment of the latter quality to machine learning models that this paper is focused: creativity. But what is creativity? A seemingly simple question to ask, but not to answer. A quick query on Google teaches us that countless definitions of creativity exist. Many focus around the idea of making something new. Whereas others speak about problem solving. Quite many, but not all, touch on the aspect of doing or making something that holds value. And words that occur often in definitions of creativity include *unique*, *original*, *artistic*, *imaginative*, plus a handful more. Clearly, no generally agreed-on definition for creativity exists, which makes us, computer scientists, slightly nervous. How can we assess or determine whether machine learning models and products convey or possess a notion for which we do not have a definition? This is an issue that we will work around later in the paper.

In the remainder of the paper we dive further into the question of *whether (learning) machines can be creative*. This debate is by no means new, it goes back in time decades if not centuries. Yet we observe that the debate does not commonly include views from a modern computer science perspective. Nor is it commonly discussed in computer science circles.

So, let's formulate the aim of our contribution. We present to a computer science and AI audience the long-standing debate around the question of whether machines can be creative, as it formed across various fields such as cognitive science, philosophy, and computing. We approach this mainly from a historical perspective, and along the way we identify how the various views in this debate relate to recent developments in machine learning models. We also argue our own position regarding the question of whether machines can exhibit human-level creativity. As such we aim to involve computer scientists and AI practitioners into the ongoing debate. Just to be clear, we do not propose new AI models or build a new application thereof. Moreover, we do not present empirical findings in support of computer creativity, or against it for that matter.

In Sect. 2 we take a step back in time, to trace some thoughts on this question that were posed before, not only by scientists but also by artists. Then in Sect. 3, drawing from cognitive psychology theory and some philosophy, we attempt to deal with the question more constructively. Finally, we argue our own position on the issue when considering modern machine learning models.

2 A Historical Perspective on (Learning) Machines and Creativity

It is an oft-mentioned fact that Ada Lovelace (1815–1852) wrote about the possibility of machines acting creatively, even before proper programmable machines existed. Some later claims were made about Lovelace in relation to her understanding of computing principles and programming of which the validity is debated (Swade 2015; Charman-Anderson 2020). Nonetheless, all agree that she was one of the first to make, in the words of historian Doron Swade (2015), "the leap from calculation to computing". This means that she recognized that a machine capable of number calculations could also manipulate symbols that represent things beyond only quantity, things that are part of our world, such as words in a text, features of people, or notes of music. And in this context Lovelace (Menabrea & Lovelace 1843) famously stated that the computer "might compose elaborate and scientific pieces of music of any degree of complexity or extent", yet it "has no pretensions whatever to originate anything. It can do whatever we know how to order it to perform." Clearly, in the view of Lovelace, any apparent creativity emanating from a computer program should be attributed to its programmer, because only after they deconstructed the exact creative process could it be programmed facsimile into the computer.

Within the context of generative art, this tension in attribution of creativity is a much-encountered phenomenon: is it the generative machine or its maker that is creative, or both? Interestingly, this question itself inspired interpretations of famous artworks, if not the artworks themselves. For example, kinetic sculptor Jean Tinguely's *Cyclograveur* (1959) is interpreted by us as asking exactly this question. When its instable bicycle-like contraption (Fig. 1) is operated, it draws an image on paper. Does the creativity apparent from that act of drawing originate from the machine, the person riding it, or its maker Tinguely?

But let's return to Lovelace's conjecture about computers only being able to create what we know exactly how to. In response to it, Alan Turing (1950) proposed that this may not hold when computers could be programmed to learn. That if computers could learn, they might possibly (paraphrasing Lovelace) originate something without us knowing how to order it to perform. In Turing's words, "the evidence available to Lady Lovelace did not encourage her to believe that [computers] had [the property of learning]. It is quite possible that the machines in question in a sense got this property."

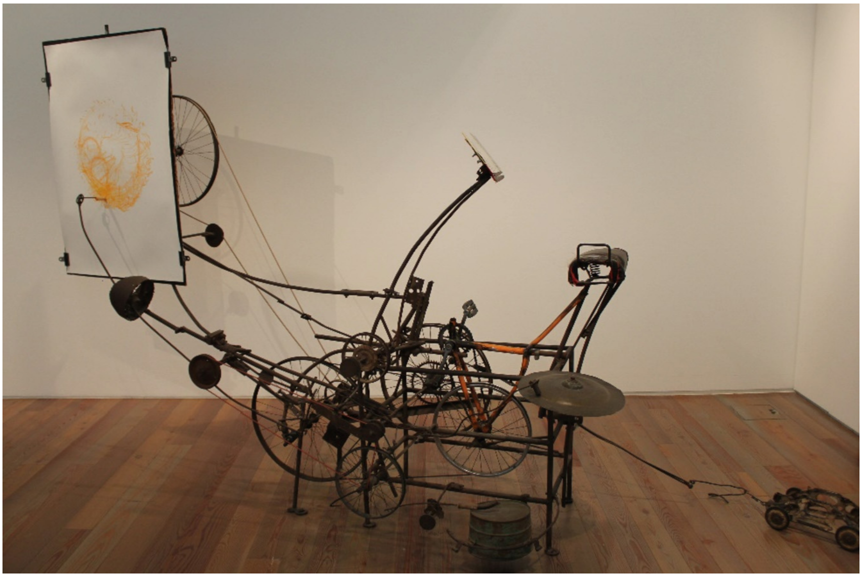


Fig. 1. *Cyclograveur* kinetic sculpture by Jean Tinguely (1959). Photo by Tim Sheerman-Chase (2013), with permission.

Indeed, modern machine learning models can generate complex images and music, without a need to specify exactly how, to understand how this process works within ourselves, or even to replicate the human creative processes at all. As such, it appears that Turing was correct about the potential of machine learning in generative AI. Nonetheless, Turing did not state whether learning machines could be considered *creative.*

In 1972 AI pioneers Marvin Minsky and Seymour Papert posed the importance of machines "learning to learn better", that "could develop as rapidly as human minds rather than requiring evolutionary epochs" (Minsky & Papert 1972). One could imagine that generative adversarial networks (GANs) (Goodfellow et al. 2014) would fit this description. In such systems, two machine learning models effectively contest in an "arms race", a competition with no absolute goal, only the relative goal of staying ahead of the competitor, leading to rapid learning improvement in both models. Such GANs underly generative AI tool thispersondoesnotexist.com and the initial version of OpenAI's DALL·E. Minsky later proposed the concept as a potential enabling condition for machine creativity, by stating that "we might see creativity happen in machines, once we begin to travel down the road of making machines that learn—and learn to learn better." (Minsky 1982).

But perhaps the strongest claim that AI could be truly creative was made by Douglas Hofstadter. In his view, intelligence is a sufficient condition for creativity. This he beautifully phrased as "I see creativity and insight, for machines no less than for people, as being intimately bound up with intelligence, and so I cannot imagine a noncreative yet intelligent machine." (Hofstadter 1982) Although he links creativity immediately to intelligence, it does not provide an operational proof for creativity in machines, since it defers the issue to the establishing of intelligence in machines, which as we know is a minefield not yet cleared of mines. Also, it allows for the existence of creative machines that are not necessarily intelligent, although his phrasing of them being "intimately bound up" appears to suggest otherwise. As such, Hofstadter appears to suggest necessity besides sufficiency of the intelligence condition for creativity.

Interestingly, Hofstadter links creative ability also to having access to stored representations of concepts and their mutual interrelations, or as he put it "having creativity is an automatic consequence of having the proper representation of concepts in a mind. It is not something you add on afterward." (Hofstadter 1982) The reason we find this interesting is that it brings to mind a well-known and often mentioned event in the history of AI: the 1956 Dartmouth Summer Research Project on Artificial Intelligence. The proposal for the Dartmouth project, co-authored by Marvin Minsky, is generally credited for coining the term "artificial intelligence". (McCarthy et al. 1955) Communication leading up to the event stated that the participants would "concentrate on a problem of devising a way of programming a calculator to form concepts and to form generalizations." (Kline 2011).

We observe and conclude that this possibly first statement about the nature of AI, is all the more relevant nowadays. Modern heuristic machine learning models induce concepts and their relations from data in a process of logical inference. The uncanny ability of modern text-to-image generators to combine different concepts into many seemingly logical visual constellations is an example of their ability to "form concepts" from data and interrelate them, and to "form generalizations" from those concepts. As such, these systems align exactly with the Dartmouth project's view of what AI should be and Hofstadter's sufficient condition for creativity. Therefore, if Hofstadter's conjecture is correct, modern heuristic machine learning models *are creative* as an automatic consequence of having representation of concepts.

As a historical sidenote, it is interesting to observe how the proposal for the Dartmouth project (McCarthy et al. 1955) also mentions creativity as an aspect of AI to be studied. It proposed that a key to creative thinking may lie in the "injection of some randomness" that "must be guided by intuition". Although randomness is indeed a ubiquitous element of modern heuristic generative AI, its role plays no significant part in the larger contemporary debate surrounding AI and creativity.

To summarize, the historical perspective offers insight into issues surrounding the question of whether machines can be creative. We observed that speculation of learning being important for machine creativity (Turing 1950; Minsky 1982) appears affirmed by the current advent of strong generative machine learning models. Also, we drew novel parallels between the original, yet fundamental, notion of AI as formulated by McCarthy et al. (1955) and creativity's reliance on intelligence and representation of concepts, as conjectured by Hofstadter (1982). From that, we conclude that if Hofstadter's conjecture is correct, then modern machine learning models must be creative by consequence. Nonetheless, the historical perspective does not truly, beyond conjecture, answer the question of whether learning machines can be creative. So, let's continue.

3 Boden's Categorization and Modern Machine Learning

How can one answer the question of whether learning machines can be creative, without a proper definition or operational test of creativity? This is similar to the problems faced when attempting to define *intelligence*, be it artificial or not. Russell & Norvig in their famous textbook (2020) avoid defining AI and instead identified four different versions of what it could mean. For each, they identified research goals and methods used. This provides categories into which activities that people call "AI" can be placed. We take a similar approach towards understanding machine creativity without forcing an overall definition onto it, by recognizing categories of potentially creative machine activities. Such categories enable comparison of generative AI models and identification of model requirements or features that appear in some categories but not others. We believe this is a fruitful method towards understanding a complex and heterogeneous topic.

The categorization we apply was proposed by cognitive scientist Margaret Boden (1998). From the view that creative ideas are novel, surprising, and valuable, she identifies two dimensions along which creative actions can be classified. The first dimension relates to the outcome of creativity, the *idea* (e.g. an image, a music composition, a plan, etc.), and identifies two levels of novelty. An agent (human or machine) may create ideas that are only novel to the mind of the agent, an ability Boden calls *P-creativity*, P for psychological. However, if the agent can devise ideas that are novel with respect to history (insofar as can be assessed) their ability she calls *H-creativity*, H for historical. H-creativity is logically a special case of P-creativity.

Boden's second dimension concerns *how* the idea was generated. On this dimension we will build our arguments further. Boden identifies three ways along which novel ideas are generated. The first is "combinational creativity" and involves finding interesting novel combinations of existing ideas that share inherent conceptual structures. Examples mentioned by her are writing poetic imagery and analogy making (incidentally a field in which Hofstadter did much of his research). This way of being creative relies strongly on having structured knowledge of concepts and their many possible connections, e.g. a rose and blood both are red, both are organic products, one is part of the human body but the other is not, the word "rose" can also mean "to have got up" and rhymes with "prose", etcetera.

Boden (1998) presents examples of computer systems that exhibit combinational creativity. These are symbolic systems built on rules, facts and some form of analogy finding engine. The knowledge and rules are captured into structures alike semantic networks, while an analogy finding engine then proposes novel associations between concepts encoded in these structures. But how does this approach compare to the modern generative AI systems that rekindled the debate about machine creativity? The latter group is built on heuristic machine learning methods, such as deep neural networks, to infer concepts from data. But they do not construct identifiable symbolic structures to capture these concepts in ways that can be searched by an analogy finding engine, nor do they contain an identifiable analogy finding engine. Therefore, we conclude that modern machine learning methods are not likely to exhibit combinational creativity in the intended sense of the term.

The other two ways to generate novel ideas, says Boden, rely on the notion of "conceptual spaces": existing, accepted and appreciate styles of thinking or ideas. For example, an accepted style of music or poetry, like jazz compositions or limerick type poems. But also, a particular style of clothing would be a conceptual space. "Exploratory creativity" is creating novel ideas by exploring such a conceptual space. Resulting ideas may be novel but satisfy the general rules or patterns that characterize the conceptual space from which they were drawn. Writing a novel limerick would be an example of exploratory creativity, but also making a Western style group portrait, as many USA tourist locations offer.[1]

We observe that current generative machine learning models generate output using the exploratory creativity approach. The conceptual space from which their ideas are drawn is captured in their model parameters and is the result of training on data. The parameters essentially describe the patterns found to be salient within that data. These patterns can be seen as concepts, as they describe feature relations (e.g. word order structure or facial feature structures) abstracted from high-dimensional lower-order features (e.g. syllables in a text or pixel values in a portrait photo). Novel ideas are generated according to the general patterns captured in the model, that characterize the original conceptual space from which the data were drawn (e.g. limerick poems or portrait photos).

Some state (e.g. Du Sautoy quoted by Clarke 2022) that the combinational approach to creativity is manifested by modern generative machine learning models, such as DALL·E 2. This is based on the observation that they can produce output that combines multiple concepts into interesting hybrid ideas. When prompted, DALL·E 2 for example creates images of virtually any combination of ideas. Nevertheless, we do not see this ability as combinational creativity. For that, the systems must find interesting combinations of concepts themselves based on similarities of their inherent conceptual features, instead of producing combinations for which they were explicitly prompted by a human. DALL·E 2's output is the result of prompt-directed *exploratory* creativity in a conceptual space that is very high-dimensional, very large, trained on very diverse images and texts, and as such virtually limitless in generating visual complexity. Any appearance of combinational creativity is introduced through prompts that initiate (or guide, if you will) the image generation, by specifying the concept combinations to make, and as such must be attributed to the human.

The third and last method of generating novel ideas is transformational creativity. It occurs when novel valued ideas are generated that transform a pre-existing conceptual space. Resulting ideas may relate to a pre-existing conceptual space, but do not themselves match the pattern that defines it, and are even inconceivable within it (Boden 2003). Popular examples of truly transformational human artists are Piet Mondriaan, Marcel Duchamp[2], and The Beatles. Their ideas transformed the existing and appreciated styles of works at their respective times in history (which makes them also H-creative, as you may remember).

Surely randomly generating ideas will ensure novel and surprising ones that do not match a pre-existing conceptual space's patterns, since random behavior is defined as not originating from an observable pattern. But random ideas are generally not considered

---

[1] Google "Western style group portrait" to see what we mean.

[2] Although we acknowledge the ongoing debate concerning the uncredited, and possibly large, contributions to his work by Elsa von Freytag-Loringhoven (Spalding & Thompson 2014).

valuable to humans, as Boden (1998) requires of creative ideas, and as McCarthy et al. (1955) established in the proposal for the Dartmouth project. For AI models to properly display combinational and transformational creativity they must evaluate potential ideas for interest or value to their intended application domain, ensuring their value in that domain. So, if we wish AI models to be creative according to human standards, then the value of their ideas to humans must be assessed *within* the model itself. And since currently no general and functional model of human appreciation exists to execute this task, automated combinational and transformational creativity are not exhibited by current machine learning systems. They are limited to exploratory creativity for generation of ideas, since appreciation by humans is inherently captured in the conceptual space that they inferred from patterns in the training data.

4 Is Transformational H-creativity a Bridge Too Far?

We dare say that transformational H-creativity along human standards is the holy grail of generative AI. It captures the elusive quality that historically renowned artists have displayed. With all the information that is available on the internet, assessing whether a generated idea is novel and H-creative appears to be automatable in the form of a matching task. Assessing its value to humans is a different matter, as we just argued. This issue was identified by Boden (1998) who attributes its difficulty to "the richness of human associative memory, and the difficulty of identifying our values and of expressing them in computational form."

But is it fundamentally impossible to automate the assessment of value to humans for a novel idea? Given that values and associations differ across time and cultural groupings, this problem becomes more difficult if we aim for a general model, or more easy if we limit its temporal and cultural context. Nonetheless, we have no reason to believe that it would be impossible. The difficulty lies not in explicit identification of human values, as conjectured by Boden. Modern machine learning models do not rely on explicit coding of phenomena or associations from their application domain, but on the algorithmic inference of them from data. Although properly inferring the relations between an idea's features and its appreciation by humans may be complicated, yet it appears not beyond the algorithmic capabilities of modern machine learning models. Models such as GPT-4 and DALL·E 2 have successfully captured phenomena and associations from data that appear much more detailed and complex, although this is of course no formal argument.

This reduces the problem of automating human value assessment through machine learning to obtaining the right training data. And let us say immediately: we don't have it. Nonetheless, we can draw inspiration from LLMs apparent capability to produce motivations for humans to like something. When we asked ChatGPT "Why do we appreciate the art of Piet Mondriaan?", it provided reasons based on the art's historical context, but also based on its stylistic patterns. Surely no fundamental objection exists against machine learning models inferring more detailed phenomena and associations that capture human value assessment, be it for a particular class of ideas, cultural context, or historical timeframe. And given the abundance of reviews that we post on the web, many including both quantitative value expressions and textual motivations thereof, we expect automated human value assessment a feasible possibility for machine learning models. If we are correct in this prediction, no objection remains against machine learning models being fundamentally capable of transformational H-creativity at the human level.

5 Conclusions

We observed that the context of modern machine learning models was not yet fully integrated in the discussion concerning relationships between AI and machine creativity, and in particular the fundamental question of whether machines can exhibit human-level creativity. To fix this, we approached the issue from two angles.

First, taking a computer science historical perspective, we identified insights as developed over time. We observed that speculation by Turing and Minsky of learning being important for machine creativity appears affirmed by the current rise of strong generative machine learning models. Also, we concluded how notions formulated by McCarthy et al. (1955) when coining the term "artificial intelligence" match to large degree modern (deep neural network) machine learning models, perhaps more so than other more classical models that fly under the flag of AI. This is important, because McCarthy and colleagues viewed AI as the collective efforts to computationally form concepts and to generalize from them. As such, we showed how this view nicely aligns with Hofstadter's speculation that creativity automatically follows from having intelligence and proper representation of concepts. This in turn led us to conclude that if Hofstadter's conjecture is correct, then modern machine learning models must be creative by consequence. This however does not tell us how AI can be creative.

For that we took a second perspective, namely that of Boden's categorizations of creativity, and combined that with our understanding of modern (deep neural network) machine learning models. The absence in these models of both identifiable (symbolic) structures that capture concepts and methods of searching for novel combinations among these concepts, led us to conclude that modern machine learning methods are not likely to exhibit combinational creativity in Boden's intended sense of the term. Any appearance of automated combinational creativity, as others have claimed to observe in popular generative AI models, is introduced through prompts that specify which concept combinations to make. As such we conclude that these apparent acts of combinational creativity must be attributed to the human user, not the AI model, which we argued works within Boden's paradigm of exploratory creativity.

We then posed that human-level transformational H-creativity is the unattained holy grail of generative AI. Just as Boden and others have also concluded, this requires a general and functional model of human appreciation, to be used for automated evaluation of novel ideas. We argued (although not formally) that building such a model through logical inference from data may be difficult, but it appears not beyond the algorithmic capabilities of modern machine learning models. Furthermore, we concluded that no fundamental objection exists against machine learning models inferring detailed patterns that capture human value assessment for ideas. Lastly, after we proposed that human-written online reviews may provide a rich source of both quantitative and qualitative data on which to base this inference, we postulated the fundamental capability of transformational creativity for machine learning models.

With this position paper we trust to have shown new connections between the philosophy of AI, the study of creative agency, and modern machine learning models. As computer scientists we feel the need to be involved in the discussion of computational creativity, and to involve other computer scientists based on their knowledge of machine learning, algorithmic fundamentals, and even computing history. We argued our personal position in the debate, in the hope to forward the discussion of whether AI can be creative. But also, to connect the discussion to practitioners and academics of AI and machine learning.

References

Boden, M.A.: Creativity and artificial intelligence. Artif. Intell. **103**(1–2), 347–356 (1998). https://doi.org/10.1016/s0004-3702(98)00055-1

Boden, M.A.: The Creative Mind: Myths and Mechanisms. Routledge Publishing, London (2003)

Charman-Anderson, S.: Ada Lovelace: a simple solution to a lengthy controversy. Patterns **1**(7), 100118 (2020). https://doi.org/10.1016/j.patter.2020.100118

Clarke, L.: When AI Can Make Art – What Does It Mean for Creativity? The Guardian (2022). https://www.theguardian.com/technology/2022/nov/12/when-ai-can-make-art-what-does-it-mean-for-creativity-dall-e-midjourney. Accessed 7 July 2023

Cohen, P.: Harold Cohen and AARON. AI Mag. **37**(4), 63–66 (2016). https://doi.org/10.1609/aimag.v37i4.2695

Colton, S.: The painting fool: stories from building an automated painter. In: McCormack, J., d'Inverno, M. (eds.) Computers and Creativity, pp. 3–38. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31727-9_1

Cook, M., Colton, S., Gow, J.: The ANGELINA videogame design system—part I. IEEE Trans. Comput. Intell. AI Games **9**(2), 192–203 (2017). https://doi.org/10.1109/TCIAIG.2016.2520256

Goodfellow, I., et al.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K. (eds.) Advances in Neural Information Processing Systems, vol. 27 (NIPS 2014), pp. 2672–2680 (2014)

Heider, F., Simmel, M.: An experimental study of apparent behavior. Am. J. Psychol. **57**(2), 243–259 (1944). https://doi.org/10.2307/1416950

Hofstadter, D.: Can inspiration be mechanized? Sci. Am. **247**(3), 18–31 (1982)

Kline, R.: Cybernetics, automata studies, and the Dartmouth conference on artificial intelligence. IEEE Ann. History Comput. **33**(4), 5–16 (2011). https://doi.org/10.1109/MAHC.2010.44

McCarthy, J., Minsky, M., Rochester, N., Shannon, C.: A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence (1955). https://raysolomonoff.com/dartmouth/boxa/dart564props.pdf. Accessed 7 July 2023

Menabrea, L., Lovelace, A.: Sketch of the Analytical Engine Invented by Charles Babbage... With Notes by the Translator. Translated by Ada Lovelace. In: Taylor, R. (ed.), Scientific Memoirs **3**, pp. 666–731. Richard and John E. Taylor Publishers, London (1843)

Minsky, M., Papert, S.: Artificial Intelligence Progress Report. MIT Artificial Intelligence Memo AIM-252, 1972/01/01. https://dspace.mit.edu/bitstream/handle/1721.1/6087/AIM-252.pdf. Accessed 7 July 2023

Minsky, M.: Why people think computers can't. AI Mag. **3**(4), 3–15 (1982). https://doi.org/10.1609/aimag.v3i4.376

Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. 4th edn. Pearson (2020)

Spalding, J., Thompson, G.: Did Marcel Duchamp Steal Elsa's Urinal? The Art Newspaper (2014). https://www.theartnewspaper.com/2014/11/01/did-marcel-duchamp-steal-elsas-urinal. Accessed 7 July 2023

Swade, D.: Charles Babbage and Ada Lovelace: Two Visions of Computing. Presentation at the Ada Lovelace Symposium, Celebrating 200 Years of a Computer Visionary, 2015/12/17, University of Oxford. https://podcasts.ox.ac.uk/charles-babbage-and-ada-lovelace-two-visions-computing. Accessed 7 July 2023

Turing, A.: Computing machinery and intelligence. Mind **59**(236), 433–460 (1950). https://doi.org/10.1093/mind/LIX.236.433

Data Science and AI in Education

Performance Analysis of Models Used to Predict Failure in Secondary School

Sofia Jordão, Dalila Durães(✉), and Paulo Novais

ALGORITMI Centre/LASI, University of Minho, Guimarães, Portugal
`pg47489@alunos.uminho.pt`, `{dad,pjon}@di.uminho.pt`

Abstract. According to data made available by Pordata for the year 2021, about 8.3% of medium school students in Portuguese schools fail or drop out of the educational system, another 9.8% of students in this situation are still in basic education. Since education is one of the pillars of a country's development, it is important to understand the reasons behind these statistics and discover what leads students to such failure in order to try to mitigate these results. In order to do so, it is necessary to acquire data about the students, thus emerging the area of Educational Data Mining. Early prediction of school failure can be the key piece of the effort to avoid it. So, this paper present a comparative study of machine learning models to indicate the best model to predict school failure.

Keywords: Artificial Intelligence · Machine Learning · School failure

1 Introduction

Education is a process of acquiring knowledge, skills, values, and habits through various forms of learning. It is essential for human development and is one of the pillars for the integration of an individual into society. The dropout rate in education and training has been the main indicator, at the European level, of the evolution of educational systems, as its centrality is recognized in today's societies, for economic competitiveness, social cohesion, and equal opportunities [1].

If a student is at risk of failure, it is important to identify the underlying cause and treat it to improve their chances of success. The challenge is to develop technologies to predict the risk of student failure in advance in order to prevent it. Thus emerged the concept of Educational Data Mining (EDM), which focuses on the application of data mining techniques in the field of education. EDM is mainly used to develop accurate models that predict student performance in order to improve learning experiences, presenting great potential to improve the quality of educational institutions and systems [2].

The use of machine learning models in education has become increasingly prevalent in recent years and presents great potential to significantly improve the decision-making process and increase student success. However, there is a

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 339–348, 2023.
https://doi.org/10.1007/978-981-99-7969-1_25

constant debate about which models are most effective in the context of education. Given this focus, the main research question to be addressed is: Which machine learning models achieve better results in the field of education?

This paper discusses a study conducted to determine the most effective models for preventing failure in secondary education. The results of different models are compared to determine the best approach.

The paper is structured into six main sections. The second section provides an overview of the current state-of-the-art in preventing school failure. The dataset with the pre-processing and methodology are explain on Sect. 3. Section 4 explain the Experiments applied. The results of the study are presented and discussed on Sect. 5, leading to a conclusion at Sect. 6.

2 Background

Educational Decision Support Systems (EDSS) are computer-based systems that help educators and educational organizations make informed decisions about teaching and learning. An EDSS may use data mining techniques to analyze student performance data in order to identify patterns and trends that can be used to support decision making related to their learning. It may also use data visualization tools to present this data in an intuitive manner, allowing educators to make informed decisions.

Although EDSS has been widely studied in higher education, its capabilities for primary and secondary education have been little explored. In higher education, EDSS has been mostly used to support resource allocation and learning analytics, as well as to analyze student performance data to identify factors contributing to their success [2,3].

Phauk Sokkhey and colleagues presented a new web-based system for predicting low-performing students using data mining techniques in their study [4]. A dataset with 43 features was created through a carefully designed questionnaire that covered personal information, domestic factors, individual factors, school factors, and score records. The authors used various filter-based selection methods, including a new method combining Chi-square and Mutual information, to create an optimal set of features for easier training. However, the authors did not disclose the features in this optimal set. Four different algorithms, including K-Nearest Neighbour, Hybrid C5.0, Hybrid RF, and an Improved Deep Belief Network optimization approach built by the authors, were tested to build the model. Hybrid RF was the best-performing algorithm for both the normal and optimal sets, achieving an accuracy of 99.95% and 99.98%, respectively. Additionally, the proposed filter-based selection method produced the best results.

J. Wang and colleagues [5]conducted a study to identify the key factors that contribute to academic success of students in the United States from their first year of school up to the end of high school. They analyzed a dataset that included information on school enrollment, student finances, and academic performance. Using a Random Forest-based data mining model, the authors achieved a mean square error of 0.910, outperforming other algorithms like Lasso and ElasticNet.

The authors determined that the number of students enrolled in a class and the location of the school were the most significant factors influencing students' grades.

S. Begum *et al.* [6] conducted a study to predict students' performance by implementing algorithms. Two datasets were created for this purpose: one for mathematics, with 395 examples, and another for Portuguese classes, with 649 entries. Three algorithms were developed along with an optimized version of each. The algorithms used were Boosting, Bagging, and Random subspace algorithms. Genetic Algorithms were developed to optimize each algorithm. Boosting was found to be the best algorithm for predicting success in both mathematics and Portuguese, with an accuracy of 98.11% and 96.82% respectively.

3 Dataset

Regarding the original dataset, it consisted of 123 features containing information ranging from more general information about the student, such as gender, age and grades, to more personal information such as their satisfaction with their grades and their level of school involvement. Table 1 present some details of the dadta from the dataset.

Before working with the dataset, it was necessary to pre-process it in order to ensure data quality.

Firstly, since this data comes from a questionnaire conducted by students under the age of 16, it was necessary to ensure that only data to which we were allowed to use would be used. After that selection we were left with 466 students data. These students encompass five academic years, with each class consisting of approximately 20 students. To elaborate, the 5th and 6th class have five classes each, totaling approximately 200 students, evenly distributed. As for the 7th, 8th, and 9th grades, there are a collective 266 students, divided into classes of roughly 20 students each.

The dataset underwent various data preparation steps following the CRISP-DM framework [7]. Initially, efforts were made to handle missing values and identify duplicate entries. Subsequently, the data needed to be transformed into numerical format to make it compatible for model input. Redundant data within the dataset was also removed.

After a first analysis of the dataset, we found that there were 28 columns with null values. The first ones consisted of how many days a week the students had tutoring and what type of tutoring they had. The reason why there were so many null values was because most students do not have tutoring and therefore left the question blank. The solution then was to fill in these null values with 0. There was also one student who did not answer what their grade was in Portuguese and Mathematics for the 1st period. In order to not discard the remaining information provided, these missing values were replaced by the average grade in both subjects. Furthermore, the remaining columns were mostly empty and did not provide significant information, so they were discarded.

Table 1. Dataset features example

Personal Data	Socio-biographical Data	School Data	Questionnaire
Name	School Allowance level	Name of the Course	"If it's important for students to show that they're smarter"
Gender	House Situation	Name of Subject	"Whether it's important to avoid showing that you're having difficulties with your work"
Age	Father's Educational Qualifications	Subject Grade	"If you hand in your homework on time"
Naturality	Mother's Educational Qualifications	Average Grade	"Participates actively in class"
Town	Transport for School	Self-Evaluation	"Do you tutor? If yes, how many days a week do you go and what type of tutoring?"
School Year	Parents' marital status	School Situation	"How satisfied are you with your grades?"

In the survey, there were multiple questions related to private tutoring, including whether students received it and how frequently. This information was consolidated into a single column, where "1" denoted students without tutoring, "2" indicated those receiving tutoring once a week, and so forth. Columns like "username" and "class," primarily used for student identification, were dropped. Eliminating these columns not only removed irrelevant information for the model but also ensured the students' anonymity was preserved.

Upon examining the grade distribution, it was noted that the data was imbalanced. To address this, oversampling techniques, as described in [8], were applied using the RandomOverSampler from the imblearn.over_sampling library.

The resulting dataset was split into two subsets: one for predicting Portuguese grades and another for predicting Math grades. Both subsets contained the same demographic information, but the Portuguese dataset exclusively included final Portuguese grades as the target variable, while the Math dataset focused solely on students' final Math grades.

To tune the models'hyperparameters, GridSearch was used which improved its performance. So, for the 123 initial features, we obtain only 6 important features for obtain the level of student risks. The final resume of the final dataset is presented on Table 2.

Table 2. Final Dataset Features

Feature	Name
1	Results of Portuguese on 2nd Trimester
2	Results of Math on 2nd Trimester
3	satisfaction of the student about the results
5	Assessment of the students on the questionnaires
6	Student involvement in school

3.1 Methodology

This subsection explains the methodology for predict student failure. It includes the pre-processing steps and the model used for prediction. A visual representation of the entire system is provided in Fig. 1. The system architecture comprises three main components, namely packet input, pre-processing, and ML model.

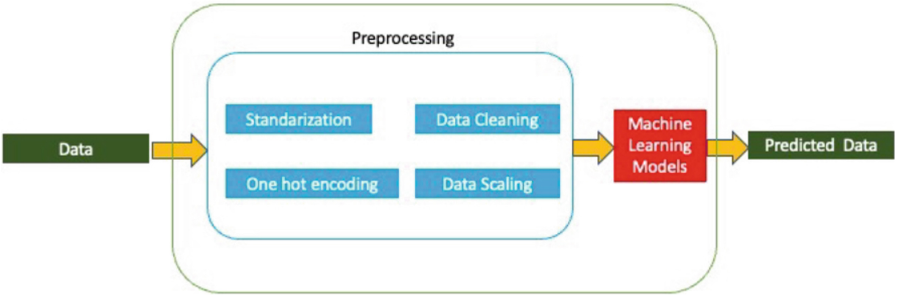


Fig. 1. Methodology for predicted failure.

To begin with, the dataset values were standardized to ensure that they were in the correct format. The data was cleaned by removing any corrupted values. The next step involved implementing one-hot encoding due to the presence of multiple values in the categorical features. This process resulted in the creation of N binary columns, where N represents the number of distinct values in a given column. A value of 1 in a binary column indicated the presence of that value in the current instance, while a value of 0 indicated its absence. Prior to this step, data cleaning was conducted to eliminate any corrupt or irrelevant values from the dataset. The four step in our pre-processing involved data scaling, which we achieved using the Min-Max scaler. Scaling is a technique that allows for the transformation of the values of numeric variables in a dataset into a common scale. Finally, it was applied the machine learning models which one with the predicted data.

4 Experiments

This section details the experiments that were carried out to confirm the effectiveness of the approach outlined in Fig. 1. We provide an overview of the experimental setup, including the methodology used to validate our models. Additionally, we present the metrics used and illustrate the outcomes of the experiments.

4.1 Metrics

To make sure that our predictive models are effective, we assess their performance using F1-score, Precision, and Recall. Since our dataset is imbalanced, where accuracy alone could lead to inaccurate results, these metrics are crucial. F1-score, Precision, and Recall offer a more precise evaluation of the model's performance by addressing the imbalance in the dataset and guaranteeing a just evaluation.

4.2 Machine Learning Models

Once the dataset was preprocessed, it was time to use machine learning models to predict students at risk. Remembering that the target class is the cluster. Each student, based on the previously selected features, will be assigned a value from 0 to 4, where 0 and 1 represent students at high risk of school failure and 2 to 3 represent students at low risk. Taking into consideration the type of data to be worked with and the objective of this work, supervised learning models were used. The selected models were Hybrid Random Forest [9], Decision Tree [10], XGBoost, and Boosting [11], as they are the most commonly used in this domain according to the literature.

For the Hybrid Random Forest model it was used the Random Forest Classifier algorithm as well as the Principal Component Analysi also knowed as PCA, and cross validation [9]. As for Boosting algorithms it was used Gradient Boosting and AdaBoost [11].

In all the models, Grid Search was used in order to find the best parameters for each model. Cross-validation, with 10 folds, was also executed in all the models to ensure that there was no overfitting.

5 Results and Discussion

In this section, the results of each model for each cluster will be presented, displaying their confusion matrix and a classification report with the values of the metrics precision, recall, and F1-score.

The first model to be developed was the hybrid random forest. For this model the first step was to standardize the data, after that it was applied the PCA algorithm. After doing so the Random Forest Classifier algorithm was applied while using Grid Search for the best parameters. The second model to be developed was the decision tree. The third model to be developed was the

Table 3. Classification report - Cluster 0

Model	Precision	Recall	F1-score	Support
Hybrid Random Forest	0.98	0.98	0.98	44
Decision Tree	0.91	0.89	0.90	44
XGBoost	0.98	1.00	0.99	44
Gradient Boosting	0.95	0.93	0.94	44
AdaBoosting	0.94	0.77	0.85	44
accuracy			0.93	44

Table 4. Classification report - Cluster 1

Model	Precision	Recall	F1-score	Support
Hybrid Random Forest	0.95	0.95	0.95	40
Decision Tree	0.86	0.95	0.90	40
XGBoost	1.00	0.97	0.99	40
Gradient Boosting	0.93	0.97	0.95	40
AdaBoosting	0.78	0.95	0.85	40
accuracy			0.96	169

XGBoost Classifier. The fourth model to be developed was the Gradient Boosting Classifier and the fifth model to be developed was the ADA Boost Classifier.

When it comes to predicting the students with higher risk of school failure, students which belong in both 0 or 1 cluster, we can observe by the Tables 3 and 4 that XGBoost obtained overall better metrics. Also by observing all its results on the prediciton of different clusters we can verify that this model achieved an overall accuracy of 99% before cross validation. And approximately 96,4% after cross validation (Tables 5 and 6).

As for predicting students with lower risk of school failure, belonging in cluster 2 or 3, we can obverse that XGBoost still maintains overall better metrics than the other methods. For cluster 3, hybrid random forest model obtained equal metrics to XGBoost but when compared to the other cluters, we verify that this model only achieved an overall accuracy of 96% before cross validation. And approximately 97% after cross validation.

Regarding the other models, such as Decision Tree, Gradient Boosting and AdaBoost, they achieved an overall accuracy of 94,5%, 94,7% and 89,4% after cross validation, respectively.

Table 5. Classification report - Cluster 2

Cluster	Precision	Recall	F1-score	Support
Hybrid Random Forest	0.93	0.95	0.94	40
Decision Tree	1.00	0.88	0.93	40
XGBoost	0.98	1.00	0.99	40
Gradient Boosting	0.97	0.88	0.92	45
AdaBoosting	0.91	0.97	0.94	40
accuracy			0.92	169

Table 6. Classification report - Cluster 3

Cluster	Precision	Recall	F1-score	Support
Hybrid Random Forest	1.00	0.98	0.99	45
Decision Tree	0.94	0.98	0.96	45
XGBoost	1.00	0.98	0.99	45
Gradient Boosting	0.94	1.00	0.97	45
AdaBoosting	0.98	0.89	0.93	45
accuracy			0.99	169

5.1 Which Machine Learning Models Achieve Better Results in the Field of Education?

With a focus on the research question outlined in Sect. 1 of the Introduction, it appears that XGBoost demonstrated the highest accuracy and overall metrics, reaching 99%, making it the ideal model to use for this type of dataset.

By utilizing the six features of the dataset, which include the Portuguese and Math on second-trimester results, student satisfaction with their performance, student assessments, and total score of involvement in school, we can determine the level of risk for each student.

These findings suggest that, in practical terms, the XGBoost model is the optimal choice for predicting school failure in both Portuguese and Mathematics performance across all clusters.

5.2 Future Work

Concerning the implementation of machine learning algorithms for successfully predicting students at risk of failing school, it is crucial to take action based on these insights. For future work, it would be beneficial to further investigate this study by developing a decision support system that incorporates the predictions made. By utilizing decision-making methods, this system can assist decision-makers in taking appropriate actions in this regard. Given that the problem involves multiple criteria and a predetermined set of alternatives, decision-making methods are mentioned in order to determine the optimal alternative.

These methods take into account multiple criteria during the selection process, enabling a comprehensive evaluation that considers various factors when making the final decision. The purpose of the work would involve conducting a detailed analysis of multi-criteria decision-making methods, considering that our problem relies on multiple criteria. Furthermore, it would entail implementing these methods using the original dataset along with the predictions made. Subsequently, the results obtained would serve as a guideline for decision-makers to take appropriate actions and assist students in improving their performance. The proposed architecture would be as follows:

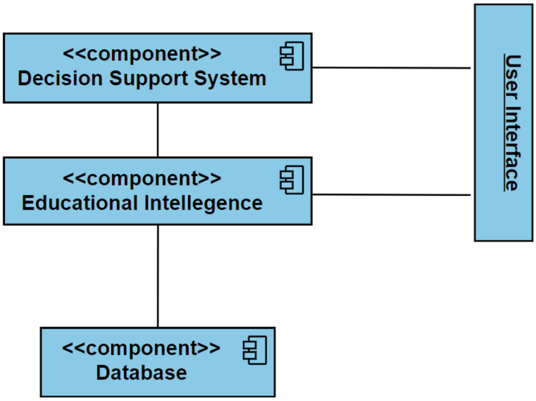


Fig. 2. Caption

The database refers to the one mentioned in Sect. 3. The Educational Intelligence refers to the machine learning method implemented, XGBoost, as we concluded in the previous section. This method would generate a new dataset that combines the original dataset with an additional column containing the predictions obtained from the model. Finally, these results would be applied to new multi-criteria decision-making methods, which would produce a ranking list of students from higher to lower risk of failing. This list would enable decision-makers to take appropriate actions and interventions for students based on their level of risk.

6 Conclusions

The objective of this paper is to examine a study that aimed to identify the most efficient models for preventing failure in secondary education. Based on the background study, a dataset was used for the study.

After making the pre-processing data of the dataset, we had the initial 117 features, but only six were found to be significant in determining the level of risk for students.

It was applied 5 machine learning models: random forester, decision tree, XGBoost, gradient boos, and ADA Boost. After comparing the outcomes of various models to determine which approach is the most effective, we conclude that XGBoost had the best results. That is the anwser to the research question.

For Futute work, and based on the level of the risk of the student, we intend to create a decision model to alert. The alert can be for teacher, parents and/or psicologist depending on the level of the risk.

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

References

1. Grifo, A., Marques, J.L.: A influência da governação europeia da Educação nas políticas educativas portuguesas. Sensos-e **10**(1), 91–102 (2023)
2. Rashad, H.M., Aly, W.M., Hegazy, O.F.: An educational data mining system for advising higher education students. IJIE **7**(10) (2013)
3. Carneiro, D., Novais, P., Durães, D., Pego, J.M., Sousa, N.: Predicting completion time in high-stakes exams. Futur. Gener. Comput. Syst. **92**, 549–559 (2019)
4. Sokkhey, P., Takeo, O.: Developing web-based support systems for predicting poor-performing students using educational data mining techniques. Int. J. Adv. Comput. Sci. Appl. **11**, 23–32 (2020). https://doi.org/10.14569/IJACSA.2020.0110704
5. Wang, J., Tang, J., Ling, Z., Luo, J.P.: Analysis of the K12 education of united states using machine learning and data mining techniques. Intell. Internet Things Eng. ICBAIE **2021**, 93–96 (2021). https://doi.org/10.1109/ICBAIE52039.2021.9389887
6. Begum, S., Padmannavar, S.: Genetically optimized ensemble classifiers for multiclass student performance prediction. Int. J. Intell. Eng. Syst. **15** (2022). https://doi.org/10.22266/ijies2022.0430.29
7. Azevedo, A., Santos, M.F.: KDD, SEMMA and CRISP-DM: a parallel overview. IADS-DM (2008)
8. Mohammed, R., Rawashdeh, J., Abdullah, M.: Machine learning with oversampling and undersampling techniques: overview study and experimental results. In: 2020 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, pp. 243–248 (2020). https://doi.org/10.1109/ICICS49469.2020.239556
9. Sokkhey, P., Okazaki, T.: Hybrid machine learning algorithms for predicting academic performance. Int. J. Adv. Comput. Sci. Appl. **11**(1), 32–41 (2020)
10. Kabra, R.R., Bichkar, R.S.: Performance prediction of engineering students using decision trees. Int. J. Comput. Appl. **36**(11), 8–12 (2011)
11. Giannakas, F., Troussas, C., Krouska, A., Sgouropoulou, C., Voyiatzis, I.: XGBoost and deep neural network comparison: the case of teams' performance. In: Cristea, A.I., Troussas, C. (eds.) ITS 2021. LNCS, vol. 12677, pp. 343–349. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-80421-3_37

Navigating the Implications of AI in Indonesian Education: Tutors, Governance, and Ethical Perspectives

Daphne Wong-A-Foe(✉)

Cultural Anthropology and Development Sociology (CADS), Leiden University, Leiden, The Netherlands
d.l.wong-a-foe@fsw.leidenuniv.nl

Abstract. The rapid advancement of Artificial Intelligence (AI) has opened up new frontiers of technological possibilities, yet the traditional education system in Indonesia has largely remained entrenched in conventional practices. Despite Indonesia's recognition for its vibrant AI innovation scene and its diverse student population, there is a significant gap in exploring the multifaceted implications of AI in, for, and by education within the country. This paper aims to address this gap by delving into three key areas: AI tutors, governance, and (virtue-based) ethical considerations. Drawing insights from both global and Islamic literature, we first examine the discourse surrounding AI tutors within Indonesia's education system. Next, we discuss the potential applications of AI in governance, including the role of the government and the emergence of AI-related education in Indonesia. Thirdly, we contemplate an ethical framework encompassing issues of inequality, public policy, and Islamic-based principles. Throughout, this paper emphasizes the critical importance of examining these three facets of AI's impact in education. Ultimately, this research raises the intriguing question of how education and AI will mutually shape each other in the future, urging further exploration of this dynamic relationship.

Keywords: DS & AI Education · AI Tutors · AI Governance · AI Ethics · Indonesian Education · Islamic Context

1 Introduction

The recent advancements in Artificial Intelligence (AI) tools, such as ChatGPT (Chat Generative Pre-Trained Transformer), have illuminated the significant potential for AI to revolutionize educational practices on a global scale. However, this development has also evoked mixed reactions among educators, reflecting the ongoing discourse concerning the implications of AI in education. Consequently, some educational institutions have taken measures to restrict its use, concerned that students may excessively rely on ChatGPT for automated generation of academic assignments [1]. Nevertheless, as García-Peñalvo argued [2],

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 349–360, 2023.
https://doi.org/10.1007/978-981-99-7969-1_26

these attempts to impose restrictions or prohibitions are unlikely to effectively discourage students from using the AI-tool. Instead, there is an anticipation that ChatGPT will assume a vital role in the writing process, comparable to the transformative impact of computers and calculators on the fields of science and mathematics [3]. The unanticipated possibilities exemplify a paradigm shift in the way education is delivered, fostering a more tailored and efficient learning environment for students and educators alike.

Moving beyond the global perspective, it is crucial to consider the specific challenges and opportunities that AI presents in different cultural and educational contexts. Existing literature on AI applications in education primarily reflect a Western perspective [2,4,7], which may not fully account for the cultural and religious nuances specific to countries like Indonesia. As it is the world's fourth most populous country, Indonesia faces distinct challenges in providing quality education to its vast and diverse population [5]. Hence, this paper delves into the multifaceted implications of AI in Indonesian education, acknowledging the importance of examining the unique socio-cultural, economic, and educational milieu of the Southeast Asian nation. It aims to provide insights and recommendations to help Indonesian policymakers, educators, and other stakeholders better understand how AI can effectively address the region-specific education challenges.

Furthermore, Indonesia is also recognized as the world's most populous Muslim nation [6]. It boasts a significant network of more than 50,000 Islamic schools, which signifies the enduring influence of religious teachings on the lives and Islamic orientation of its citizens [5]. Consequently, when delving into the ethical implications of AI from an Islamic standpoint, it becomes imperative to critically examine the educational philosophy, curriculum, and pedagogy employed by Islamic schools in Indonesia. In this regard, this research paper builds upon the framework rooted in the *Maqāṣid* (objectives) theory, emphasizing the utilization of AI for meaningful feedback rather than focusing solely on efficiency gains. By doing so, this paper seeks to offer valuable insights into the convergence of AI ethics and Islamic perspectives, thereby fostering a deeper comprehension of how AI can be ethically integrated within the Indonesian Islamic education system.

To provide a comprehensive analysis of the implications of AI in Indonesian education, this paper will explore three key focus areas: AI tutors, -governance applications, and -ethical considerations. Firstly, the paper will examine the role of AI tutors, examining their role and impact on educational practices, drawing upon pertinent global and Islamic studies. Secondly, the study will delve into the potential applications of AI in governance, encompassing the government's involvement in curriculum development, integration of AI systems within religious institutions, optimization of administrative tasks, and educational policies. Lastly, the paper will discuss the ethical considerations surrounding the use of AI and education in Indonesia, including inequality, public policy, and Islamic-based ethics. By exploring these facets, this study aims to offer valuable insights and future directions to facilitate the successful integration of AI and education within the Indonesian milieu.

2 AI Tutors

2.1 A Brief Overview

In the realm of computer technology, the use of AI tutors in Western classrooms has gained momentum since the 1980s. A notable example is 'GPTUTOR', a digital AI tutor introduced in 1984 to offer personalized instruction and feedback to students studying geometry [7]. GPTUTOR's design aimed to diagnose students' challenges and provide tailored learning experiences that matched their individual capabilities, which was highly advantageous during that period [8]. Research conducted by Schofield et al. [9] examining the impact of GPTUTOR on classroom behavior revealed that students exhibited increased effort and engagement in tasks, while teachers allocated more time to assist struggling students, interacted more collaboratively with students, and prioritized students' efforts when assessing their performance.

Nowadays, AI tutors have become prominent as educational aids, tutors, and facilitators of peer learning, manifesting in both embodied (physical) and disembodied (digital) forms across classrooms worldwide [10]. For instance, embodied social robots are now being utilized in preschools and elementary schools to assist educators, such as 'Mr. Robot', teaching students about electric circuits in Palestine [11], 'Kebbi', focusing on imparting knowledge about shapes and colors in England [12], and 'Sligo', providing French language instruction in Ireland [13]. Conversely, disembodied AI tutors encompass digital platforms, such as the 'Duolingo' app that facilitates language practice for second language learners, and the 'iTalk2Learn' online platform that supports elementary students in mathematics learning.

This phenomenon extends not only to the Global North's focus on science subjects but also to the Islamic world, where chat-bots are employed for educational purposes in Islamic teachings. Notably, 'Muslim Pro' exemplifies this trend, functioning as a religious mobile application specifically designed for the practice of Islamic prayer. The app provides users with information on prayer times and the *Qiblah* (direction towards the Kaaba in Mecca) based on their specific location, offering a convenient tool for religious observance [14]. Additionally, Iran serves as an intriguing case study, showcasing the integration of Islamic AI tutors in embodied educational settings. For instance, bots like 'Alice' have been introduced in classrooms to educate children about the concept of *Hijab* (headscarf) [15]. Another bot named 'NOA' autonomously recites the *Salaat* (Islamic prayers) and performs the ritual acts while facing the *Qiblah*. These bots contribute to a more engaging and less unsettling approach to Islamic education for children in Iran, as discussed in the study by Alemi et al. [15].

However, despite the increasing usage of various types of AI tutors, there is still considerable hesitancy among teachers, students, and even parents to fully embrace them. According to a study by Hao [16], there is a prevailing belief that human tutors are more effective in assisting students with their learning compared to AI tutors. More evidently, the Covid-19 pandemic has further highlighted the strong yearning for interpersonal interaction among the majority of students [17], leading to important reflections on the effectiveness of robotic (religious) teaching practices.

2.2 Through a Religious Lens

From a religious perspective, the idea that the effectiveness of teachings can transcend the need for a human intermediary and include automated systems is encapsulated in the Latin concept of *ex opere operato* (from the work performed). This notion posits that the ritual or teaching itself possesses intrinsic value and efficacy, challenging the conventional belief that human involvement is indispensable for successful administration. In Islamic theology, a parallel concept is the *Niyyah* (intention), which recognizes the profound significance of one's intention, even outweighing actions considered *haram* (prohibited) [18]. By considering both the *ex opere operato* and the importance of *Niyyah*, a more comprehensive understanding of the role of AI technology and human agency in religious teaching practices can be attained.

In the realm of Islamic education, a distinct emphasis is placed on pedagogical approaches that differs from those found in secular educational settings [19]. Islamic teaching practices places strong emphasis on moral and ethical virtues such as honesty, integrity, compassion, and the significance of prayer. These fundamental values are imparted by religious educators known as *ustadz* or *kyai* [5]. Additionally, personalization stands as another pivotal facet of Islamic education, with instructors treating each student as a unique individual and adjusting their teaching approaches accordingly. In parallel to this, empirical research has demonstrated that various AI chat-bot voices cater to and are preferred by different individuals. This suggests that AI tutor voices can be tailored to accommodate the diverse characteristics and needs of each student [20]. This prompts us to question how Islamic education in Indonesia incorporates personalized teaching practices and how AI technologies can be harnessed to enhance this personalized approach.

2.3 Implications in Indonesian Education

To understand the Indonesian education system, it is necessary to realize that navigating the delicate balance between conservative beliefs and adjusting to societal demands can be a challenge in Indonesia. The country is officially secular, but both its *Pancasila* (national ideology) and the 1945 Indonesian Constitution uphold the religious belief in one supreme God [21]. For Muslims, this reaffirms the concept of *Tauhid*, emphasizing the indivisible oneness of God. As a result, religiosity is deeply ingrained in numerous facets of their daily lives, including education, as evidenced by the incorporation of religious instruction into the national curriculum across the country [5]. Nevertheless, when it comes to the potential conflict between Islamic teachings and scientific discoveries, the majority of Islamic schools in Indonesia adopt a stance that embraces the compatibility of Islamic beliefs and modern science. These schools do not perceive scientific findings as contradictory to their religious teachings. Instead, they believe that Islamic principles and scientific knowledge can coexist harmoniously without conflicts arising between them. This approach adopted by Islamic schools in Indonesia safeguards the continuity and progress of Islamic education by avoiding obstacles that may emerge from discord between religious dogma and scientific advancements [22].

However, even though Indonesia seems to be thriving as an AI innovation ecosystem [23] with its diverse student demographic [24], little to none is known about the multifaceted implications of AI tutors and Indonesian education, especially when considering ethical and religious considerations. In an effort to shed light on this matter, a recent study conducted by Sumakul [25] investigated the use of AI apps in their English as a Foreign Language (EFL) classes. The study involved the trial implementation of two AI apps, namely 'Plot Generator' [https://www.plot-generator.org.uk/] for creative writing classes and 'Elsa' [https://elsaspeak.com/en/] for pronunciation classes. The findings of the study revealed that both Indonesian students and teachers expressed positive perceptions regarding the incorporation of AI technology in their EFL classroom. This pioneering research offers valuable insights into the application and impact of AI tutors within the Indonesian educational context. However, further qualitative and longitudinal exploration is necessary to fully comprehend the broader implications and potential challenges associated with the adoption of AI tutors in Indonesia's education system.

Subsequently, we will discuss Indonesia's history of technological nationalism and current efforts towards AI-dependent technologies for national development in order to demonstrate the country's willingness to embrace new technologies and its ability to adapt to changing technological landscapes.

3 Technology as an Ally

3.1 Historical Control and Classification

Historically, the Indonesian government has also sought to use technology as a means of classifying and controlling its citizens through education. For example, the government has implemented a national curriculum that aims to promote a specific understanding of Islam and Indonesian identity [5] through IT-based access to education [26]. The state has also used technology to monitor and control the dissemination of finances and information in schools, including through the regulation of textbooks and other learning materials [27].

Notably, the success of ICT implementation in Indonesian education was also determined by several governmental factors. One significant obstacle to the widespread use of computers among the Indonesian population was their limited proficiency in English, which resulted in a reluctance to embrace computer technology [28]. To address this issue and promote ICT literacy, the government issued Presidential Decree (No. 2/2001) which led to the development of computer applications in the Indonesian language. Further, the 2004 national program 'Indonesia, Go Open Source!' (IGOS) and the 2009 policy letter of the State Ministry for State Apparatus (No: SE/01/M.PAN/3/2009), which concerned proper software utilization and open source software (OSS), played a significant role in accelerating and encouraging public internet usage, as well as improving ICT infrastructure within the education sector curriculum [26].

3.2 AI Governance in Indonesian Education

Similarly, in efforts to foster the successful integration of AI in Indonesian education, the *Kementerian Agama* (Ministry of Religious Affairs) recently launched a digitization program in 2021 as part of the 'Making Indonesia 4.0' state project. This program introduced AI systems, namely the Madrasah Self Evaluation (EDM) for administration and evaluation, and e-RKAM (Electronic-Based Madrasah Work Plan and Budget) for e-planning and e-budgeting, across madrasahs (Islamic elementary schools) in Indonesia [19]. In other words, this initiative addresses governance in educational institutions through the implementation of AI. According to Hamruni and Suwartini [19], this incorporation of AI in madrasahs aligns with Islamic morality, as it upholds principles of trustworthiness, non-maleficence, fairness, and beneficence. Consequently, it can be inferred that the use of AI governance in Islamic education is no different from its application in secular education within the Indonesian context.

Moreover, as Indonesia undergoes rapid technological transformation, the need for high-quality education and training in this field becomes increasingly important. To tackle this, Indonesia introduced its own National Strategy on Artificial Intelligence 2020–2045 *(Strategi Nasional Kecerdasan Artifisial 2020–2045)*, placing great emphasis on Indonesia's AI innovation, research, and development in the education sector [29]. This 25-year strategy outlines the government's vision to shift from a resource-dependent economy to one that is innovation-driven, making Indonesia the second member of the Association of Southeast Asian Nations (ASEAN) after Singapore to do so [30]. At the local level, this national strategy has led to the initiation of projects such as the AI-dependent '100 Smart Cities' initiative [31]. These efforts have also attracted international attention, with China providing funding to AI firms and strengthening cybersecurity partnerships with Indonesia, while the United States primarily invests in private industries to support AI development in the country [29].

In the realm of education, AI startup Tokopedia has collaborated with Universitas Indonesia (UI) to establish the 'Tokopedia-UI AI Center of Excellence' in Jakarta [29], for example. Additionally, foreign companies like Huawei, Alibaba Cloud Indonesia, and Amazon Web Services have introduced cloud training programs targeting Indonesian students [29]. These privately funded programs cater to the demands of employers and have contributed to reducing the exclusivity of the local public university system. Furthermore, several Indonesian public universities have embraced the European-funded initiative Erasmus+ to provide local students with the opportunity to pursue the Data Science & Artificial

Intelligence (DS&AI) master's program [https://dsai-project.eu/site/], namely *Institut Teknologi Bandung* (ITB), *Universitas Syiah Kuala* (USK), and *Universitas Sumatera Utara* (USU). Additionally, *Universitas Islamic Negeri Syarif Hidayatullah Jakarta* (UIN-JKT), the first Islamic State University in Indonesia, is currently offering AI courses under the Indonesian translation *Kecerdasan Buatan* as part of its Informatics bachelor's program. These initiatives and partnerships signify the growing importance of AI in shaping the future of education in Indonesia.

However, it is noteworthy that the mentioned projects seem to have a secular orientation, lacking explicit reference or connection to religion, despite Indonesia's strong ties to religiosity. This raises questions about how Indonesian-Muslims navigate the challenges posed (if any) by emerging technologies and new educational programs.

4 Ethical Considerations

As AI technologies are being integrated into (religious) educational practices in Indonesia, it is essential to critically examine the ethical implications associated with their use within the Indonesian context. This section delves into three key dimensions: inequality, public policy, and Islamic ethics, as they pertain to the unique socio-cultural and religious dynamics of the Indonesian education system.

4.1 AI and Inequality

Firstly, the issue of inequality arises from concerns regarding the potential contribution of AI technologies to the deepening of existing disparities in educational access and outcomes within Indonesia. Emergent AI technologies often require advanced education and skills, leading to a growing education premium and increased inequality. Despite facing similar demands for technological change, not all countries experience the same level of inequality. Unlike before, where certain countries held a technological lead while others caught up [32], AI's integration knows no geographical or other boundaries. This dynamic has significant implications for global competition and geopolitical struggles. In the context of Indonesia, with its diverse population encompassing various socio-economic backgrounds and geographic locations, ensuring inclusive and equitable access to AI-enabled educational opportunities becomes paramount. By critically examining the implications of AI in education through an equity lens, researchers and policymakers can contribute to the development of educational systems that empower all learners and foster a more equitable and just society, placing emphasis on the needs of marginalized and underserved learners in Indonesia.

4.2 AI and Public Policy

Moreover, the formulation of public policy is of utmost importance in addressing the ethical considerations associated with the integration of AI in education. One significant area that requires attention is academic integrity, as highlighted by Sullivan et al. [33]. The rapid development of AI systems like ChatGPT has elicited concerns among academics and industry leaders, leading to calls for a temporary halt in the development of such powerful AI systems [https://futureoflife.org/open-letter/pause-giant-ai-experiments/]. While organizations like UNESCO have provided guidance on leveraging AI to enhance educational quality in line with Sustainable Development Goal 4 [34], existing frameworks often fail to comprehensively address the unique cultural and institutional context of Indonesia. For instance, the UNESCO guidelines propose the integration of AI literacy and AI ethics into existing programs and courses. However, given Indonesia's diversity and the presence of distinct socio-cultural and religious nuances, it is crucial to develop locally contextualized ethical frameworks that effectively address the challenges and opportunities arising from the AI era in education. Therefore, considering that the majority of Indonesians identify as Muslims, this study consequently delves into the examination and exploration of Islamic-based ethics as a viable framework for navigating the complex ethical dilemmas posed by AI in Indonesia.

4.3 AI and Islamic Ethics

Nowadays, researchers are increasingly exploring virtue-based ethics as a framework for addressing ethical dilemmas in the AI era [35]. Notably, even the Vatican has recently issued guidelines on AI ethics to serve as a guiding reference for individuals following the Catholic faith [36]. This development raises the question of whether the Islamic world can also unite and create a virtue-based handbook, fostering ethical guidelines specific to their values and principles in relation to AI, particularly within the domain of education.

Within the context of Islamic ethics, the principles of *Shari'ah* (Islamic law) seek to promote the common good and prevent actions that may harm individuals and society in all facets of human life [18]. Consequently, these Islamic principles extend to the use and creation of AI applications in education. Recently, Raquib et al. [37] proposed a valuable framework grounded in Islamic ethics known as the *Maqāṣid* (objectives) theory. This theory categorizes the advantages obtained from achieving educational objectives into the following: (1) essentials, (2) needs and necessities, and (3) enhancements. This classification emphasizes the importance of giving priority to student well-being (essentials) and educational progress (needs and necessities), rather than focusing solely on technological advancements (enhancements). For example, when assessing the use of AI in educational processes, such as grading or administrative tasks, the application of Islamic objectives ethics would involve evaluating whether the implementation of AI serves the essentials or needs and necessities of education. If the primary focus of using AI is to enhance efficiency without significantly improving student learning outcomes or overall educational experiences, it may be considered unnecessary and fall under the category of enhancements.

Hence, in the context of AI tutors, the Islamic ethics framework based on the *Maqāṣid* theory would prompt educators to avoid using AI tutors if they pose significant risks to students' academic progress or well-being. For example, if an AI tutoring system lacks the capability to recognize and address students' emotional needs or struggles, resulting in an adverse impact on their overall learning experience, the use of such AI tutors would be considered inappropriate. In such instances, the involvement of human educators becomes crucial to ensure the holistic development and support of students throughout their educational journey. By carefully considering these distinctions and applying the *Maqāṣid* theory, educators and policymakers can strike a balance between leveraging AI's potential and safeguarding the best interests of students within the Indonesian-Muslim context.

In conclusion, through an examination of these ethical considerations encompassing inequality, public policy, and Islamic ethics, it becomes evident that AI's integration in Indonesian education warrants careful evaluation and adherence to ethical principles. Such considerations are essential for ensuring equitable access, protecting academic integrity, and aligning technological advancements with the values and goals of education within the country.

5 Discussion and Future Directions

All in all, this paper has discussed the various applications of AI in the field of education, specifically focusing on AI tutors for science-based subjects and Islamic education. The research encompasses the utilization of AI tutors in language learning, mathematics, plot-generation, and Quran recitation tools, drawing upon examples from both global educational contexts and those specific to Islamic education. However, despite Indonesia's evident interest in AI technological advancements, as demonstrated by the aforementioned government-led projects, there remains a lack of understanding regarding the implications of AI applications in the Indonesian education system. Consequently, this paper sought to shed light on the practical implementation of AI in education in Indonesia, with particular attention to three fundamental aspects: firstly, ensuring the fair and equitable distribution of technological advancements and educational opportunities through an equity lens; secondly, establishing well-defined guidelines for AI usage in alignment with the recommendations set forth by UNESCO, to which Indonesia is a member-state; and lastly, developing an Islamic-based ethical framework that adequately addresses the unique needs and perspectives of the country's Muslim majority population.

However, the focus of this discussion should not solely be on the members of the current generation, who are already more adept in the use of AI technology than their elders. Instead, it should also encompass future generations who rely on AI technologies as their primary mode of communication, central platform for academic tasks, entertainment, and exploring contemporary issues. Nevertheless, it is essential to recognize that political and economic power will still predominantly reside with adults. Consequently, it becomes crucial for the upcoming generation to understand and adapt their interactions to effectively navigate society. However, the lack of skills or knowledge required to communicate with individuals who hold different values and attitudes further complicates the traditional conflicts between youth and their elders. Moreover, these divisions extend beyond generational disparities and encompass any group with differing characteristics. Thus, this paper emphasizes the need for future generations to develop the necessary skills to bridge these gaps and navigate a society where technological advancements and differing perspectives are prevalent.

In terms of future directions for Indonesia, it is valuable to consider the advancements made in AI education in other countries. For example, China has made significant progress in AI education, with companies like Squirrel AI pioneering a learning center where students and teachers rely solely on the latest technologies for instruction, eliminating the need for physical blackboards and notebooks [16]. This approach aims to shift the focus away from test-centric education and prioritize physical, moral, and artistic aspects. China's exploration of AI education has the potential to influence global educational practices and serves as a valuable case study for Indonesia.

Further, to ensure effective utilization of AI tools, proper training should be provided to educate educators and students on the functions of AI, including how to evaluate accuracy and information and how to track queries as discussed in the UNESCO guidelines. This paper suggests that educators should allow the use of ChatGPT and initiate its use, as students are likely to use it regardless [2]. Allowing students to use the tool gives them an equal chance to develop ideas and improve their writing [38]. To illustrate this, an example of reverse-engineering a ChatGPT-based exam could be given, showcasing how the tool can be used to stimulate critical thinking, creativity, and problem-solving.

In light of these considerations, further research and examination in the Indonesian context are necessary to comprehensively investigate the implications of AI in (religious) teaching practices within classrooms, accounting for diverse religious perspectives, theological considerations, and the intricate interplay between intention, actions, and spiritual rewards within different religious frameworks.

In conclusion, as we witness the potential of AI technologies like ChatGPT to effectively address the limitations of traditional exams, we find ourselves at a pivotal juncture in the evolution of education. To navigate this transformative path successfully, it is imperative that we actively engage in qualitative and longitudinal research within the context of AI and education in Indonesia. This paper serves as a preliminary step, offering valuable insights and emphasizing the urgent need for further investigations. Only through such rigorous investigation can we fully understand the multifaceted implications, benefits, and challenges that AI integration presents to our educational landscape. By embracing AI tools, offering comprehensive training, and dedicating ourselves to profound research in diverse educational settings, including religious education, we pave the way for an enlightened future of learning.

References

1. Lucas, R.: New York City Schools Ban ChatGPT to Head Off a Cheating Epidemic. Gizmodo. https://gizmodo.com/new-york-city-schools-chatgpt-ban-cheating-essay-openai-1849949384. Accessed 4 July 2023
2. García-Peñalvo, F.J.: The Perception of Artificial Intelligence in Educational Contexts After the Launch of ChatGPT: Disruption or Panic? (2023)
3. McMurtrie, B.: AI and the future of undergraduate writing. The chronicle of higher education (2022). https://www.chronicle.com/article/ai-and-the-future-of-undergraduate-writing. Accessed 4 July 2023
4. Zembylas, M.: A decolonial approach to AI in higher education teaching and learning: strategies for undoing the ethics of digital neocolonialism. Learn. Media Technol. **48**(1), 25–37 (2023)
5. Alwasilah, A.C.: Islam, Culture, and Education: Essays on Contemporary Indonesia. Rosda, (2013)
6. World Population Review (2023). https://worldpopulationreview.com/country-rankings/muslim-population-by-country. Accessed 4 May 2023
7. Anderson, J.R., Boyle, C.F., Yost, G.: The geometry tutor. In: Proceedings of the 9th International Joint Conference on Artificial Intelligence (1985)
8. Anderson, J.R.: Proposal to the Carnegie Corporation to Support Demonstration and Development of a Geometry tutor. Unpublished proposal, Carnegie-Mellon University, Pittsburgh, PA (1984)
9. Schofield, J.W., Evans-Rhodes, D., Huber, B.R.: Artificial intelligence in the classroom: the impact of a computer-based tutor on teachers and students. Soc. Sci. Comput. Rev. **8**(1), 24–41 (1990)
10. Chen, M., Liu, F., Lee, Y.H.: My tutor is an AI: the effects of involvement and tutor type on perceived quality, perceived credibility, and use intention. In: Degen, H., Ntoa, S. (eds.) HCII 2022. LNCS, vol. 13336, pp. 232–244. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-05643-7_15
11. Hams, M.: Robot Teacher Introduced in Gaza Classroom (2021). https://www.i24news.tv/en/news/middle-east/palestinian-territories/1639688788-robot-teacher-introduced-in-gaza-classroom. Accessed 10 June 2023
12. Kohli, D.: An Andover Preschool Hired an Unusual Teacher's Aide: a Robot (2021). https://www.bostonglobe.com/2021/12/06/business/an-andover-preschool-teacher-is-robot/. Accessed 10 June 2023
13. McDonagh, M.: Sligo Schoolchildren's New Teacher will be Nao - a Robot (2021). https://www.irishtimes.com/news/education/sligo-schoolchildrens-new-teacher-will-be-nao-a-robot-1.4659247 Accessed 10 June 2023
14. Bellar, W.: Rituals: prayer app rituals: how Islamic participants engage with technological and religious affordances in Muslim pro. In: Digital Religion, pp. 150–158. Routledge (2021)
15. Alemi, M., et al.: Social robotics, education, and religion in the Islamic world: an Iranian perspective. Sci. Eng. Ethics **26**, 2709–2734 (2020)
16. Hao, K.: China has started a grand experiment in AI education. It could reshape how the world learns. MIT Technol. Rev. **123**(1) (2019)
17. Reuter, P.R., Forster, B.L., Kruger, B.J.: A longitudinal study of the impact of COVID-19 restrictions on students' health behavior, mental health and emotional well-being. PeerJ **9**, e12528 (2021)
18. Popova, B.: Islamic philosophy and artificial intelligence: epistemological arguments. Zygon® **55**(4), 977–995 (2020)

19. Hamruni, Suwartini, S.: Artificial intelligence in global Islamic education. IOSR J. Human. Soc. Sci. (IOSR-JHSS), 39–49 (2022)
20. Xu, K.: First encounter with robot alpha: how individual differences interact with vocal and kinetic cues in users' social responses. New Media Soc. **21**(11–12), 2522–2547 (2019)
21. Morfit, M.: Pancasila: the Indonesian state ideology according to the new order government. Asian Surv. **21**(8), 838–851 (1981)
22. Tan, C.: Educative Tradition and Islamic Schools in Indonesia (2014)
23. Das, K., et al.: The digital archipelago: how online commerce is driving Indonesia's economic development. McKinsey Company, 1–72 (2018)
24. Satya, V.E.: Strategi Indonesia menghadapi Industri 4.0. Info. Singkat **10**(9), 19–24 (2018)
25. Sumakul, D.T.Y., Hamied, F.A., Sukyadi, D.: Artificial intelligence in EFL classrooms: friend or foe? LEARN J. Lang. Educ. Acquisition Res. Netw. **15**(1), 232–256 (2022)
26. Hermawan, H.D., Deswila, N., Yunita, D.N.: Implementation of ICT in education in Indonesia during 2004–2017. In: 2018 International Symposium on Educational Technology (ISET). IEEE (2018)
27. Rosser, A.: Beyond Access: Making Indonesia's Education System Work (2018)
28. Yuhetty, H.: ICT and Education in Indonesia (2002). Accessed 19 Sept 2012
29. Goode, K., Kim, H.M.: Indonesia's AI promise in perspective. Center Secur. Emerg. Technol. **28** (2021)
30. Sri Saraswati Wisjnu, W., Ismunandar, I., Anto, S.N.: Strategi Nasional Kecerdasan Artifisial Indonesia 2020–2045 (2020)
31. Ministry of Communication and Informatics. Langkah Menuju "100 Smart City" (2017). https://kominfo.go.id/content/detail/11656/langkah-menuju-100-smart-city/0/sorotan_media
32. Clark, C.M., Gevorkyan, A.V.: Artificial intelligence and human flourishing. Am. J. Econom. Sociol. **79**(4), 1307–1344 (2020)
33. Sullivan, M., Kelly, A., McLaughlan, P.: ChatGPT in higher education: considerations for academic integrity and student learning. J. Appl. Learn. Teach. **6**(1) (2023). https://doi.org/10.37074/jalt.2023.6.1.17
34. Sabzalieva, E., Arianna, V.: ChatGPT and Artificial Intelligence in Higher Education: Quick Start Guide (2023)
35. Vallor, S.: Technology and the Virtues a Philosophical Guide to a Future Worth Wanting. Oxford University Press, Oxford (2018)
36. Flahau, J., et al.: Ethics in the Age of Disruptive Technologies: An Operational Roadmap (2023). https://www.scu.edu/institute-for-technology-ethics-and-culture/itec-handbook/. Accessed 4 July 2023
37. Raquib, A., et al.: Islamic virtue-based ethics for artificial intelligence. Discover Artif. Intell. **2**(1), 11 (2022)
38. Halaweh, M.: ChatGPT in Education: Strategies for Responsible Implementation (2023)

Author Index

A
Abelha, António 288
Abidin, Taufik Fuadi 279
Adriat, Riza 103
Al Afghani Edsa, Said 35
Al-Assad, Muhammad Hafez 185
Andres, Frederic 261
Ardianto, Randy 103

B
Bhatti, Muhammad Omer Farooq 198

C
Cabral, Jorge 3
Candra, Yudi 185
Cao, Lu 170
Chak-Fu Wong, Jeff 43
Chamoso, Pablo 300
Chatnuntawech, Itthi 111
Costa, Miguel 3

D
Dailey, Matthew N. 198, 228
Dawood, Rahmad 185
De Silva, Tiloka 59
Den, Simone A. Ten 170
Dias, Miguel 288
Duarte, Júlio 288
Durães, Dalila 339

E
Ekpanyapong, Mongkol 228
Esichaikul, Vatcharaporn 88

F
Faisyal, Farrasa Rani 279
Fitria, Maya 185

G
Gadyatskaya, Olga 245
Gomes, Tiago 3
González-Briones, Alfonso 300

H
Hernández, Guillermo 300
Hewawalpita, Supun 59

J
Jahanbanifard, Mehrdad 213
Javanmardi, Shima 213
Jongsma, Siem 170
Jordão, Sofia 339

K
Keshan, Sandupa 59
Kielarova, Somlak Wannarumon 317
Kurniawan, Kanada 103
Kushadiwijayanto, Arie Antasari 103

L
Lamers, Maarten H. 325
Lhasiw, Nunthawat 127
Liyanage, Sanduni 59

M
Machado, José 288, 300
Maulana, Aga 279
Mawitagama, Bawun 59
Monteiro, João 3
Muangkote, Nipotepat 18
Muchallil, Sayed 185

N
Nguyen Quoc, Khang 157
Nguyen Quynh, Anh 157
Novais, Paulo 300, 339

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Singapore Pte Ltd. 2023
C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 361–362, 2023.
https://doi.org/10.1007/978-981-99-7969-1

P
Papuc, Dalia 245
Passier, Robert 170
Peixoto, Hugo 288
Pinto, Sandro 3
Pradujphongphet, Prapasson 317

Q
Quach, Luyl-Da 157

R
Racharak, Teeradaj 261
Ranaweera, Uvini 59
Rasool, Mehran 142
Rawira, Panunsiya 88
Regmi, Biplav S. 228
Rehman, Mariam 142
Riza, Hammam 279
Roza, Silvia 185

S
Safder, Sadaf 142
Sampurno, Joko 103
Sanglerdsinlapachai, Nuttapong 127
Schwach, Verena 170
Silpasuwanchai, Chaklam 111
Sinsamersuk, Akraradet 111
Soingern, Nutapol 111
Soomlek, Chitsutha 74
Sousa, Regina 288
Sukkasem, Peeradon 74
Sunat, Khamron 18, 35

T
Tanantong, Tanatorn 127
Tang, Xiaoqin 213
Tarmizi, Faris Khowarizmi 279
Tavares, Adriano 3
Thant, Shin 261
Tran Ngoc, Hoang 157

V
Verbeek, Fons J. 170
Verbeek, Fons. J. 213

W
Watthaisong, Tanatip 18
Wong, Po-Chai 43
Wong-A-Foe, Daphne 349