Volume 49 Number 3 October 2025 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Special Issue:

SoICT 2024

Guest Editors:

Huynh Thi Thanh Binh,
Ichiro Ide,
Minh Triet Tran

Editorial:

ACM Turing Award

1977

Editorial Boards

Informatica is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Higher Education, Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

Executive Editor – Editor in Chief
Matjaž Gams
Jožef Stefan Institute Jamova 39, 1000
Ljubljana, Slovenia
Phone: +386 1 4773 900
matjaz.gams@ijs.si
http://dis.ijs.si/mezi

Editor Emeritus
Anton P. Železnikar
Volaričeva 8, Ljubljana, Slovenia
s51em@lea.hamradio.si

Executive Associate Editor - Technical Editor
Drago Torkar
Jožef Stefan Institute Jamova 39, 1000
Ljubljana, Slovenia
Phone: +386 1 4773 900
drago.torkar@ijs.si

Executive Associate Editor - Deputy Technical Editor
Tine Kolenik
Paracelsus Medical University, Salzburg
amsinformatica@ijs.si

Production Editors
Gašper Slapničar and Blaž Mahnič
Jožef Stefan Institute Jamova 39, 1000
Ljubljana, Slovenia

Editorial Board
Juan Carlos Augusto (Argentina)
Vladimir Batagelj (Slovenia)
Francesco Bergadano (Italy)
Marco Botta (Italy)
Pavel Brazdil (Portugal)
Andrej Brodnik (Slovenia)
Ivan Bruha (Canada)
Wray Buntine (Finland)
Zhihua Cui (China)
Aleksander Denisiuk (Poland)
Hubert L. Dreyfus (USA)
Jozo Dujmović (USA)
Johann Eder (Austria)
George Eleftherakis (Greece)
Ling Feng (China)
Vladimir A. Fomichov (Russia)
Maria Ganzha (Poland)
Sumit Goyal (India)
Marjan Gušev (Macedonia)
N. Jaisankar (India)
Dariusz Jacek Jakóbczak (Poland)
Dimitris Kanellopoulos (Greece)
Dimitris Karagiannis (Austria)
Samee Ullah Khan (USA)
Hiroaki Kitano (Japan)
Igor Kononenko (Slovenia)
Miroslav Kubat (USA)
Ante Lauc (Croatia)
Jadran Lenarčič (Slovenia)
Shiguo Lian (China)
Suzana Loskovska (Macedonia)
Ramon L. de Mantaras (Spain)
Natividad Martínez Madrid (Germany)
Sanda Martinčić Ipšić (Croatia)
Angelo Montanari (Italy)
Pavol Návrat (Slovakia)
Jerzy R. Nawrocki (Poland)
Nadia Nedjah (Brasil)
Franc Novak (Slovenia)
Marcin Paprzycki (USA/Poland)
Wiesław Pawłowski (Poland)
Ivana Podnar Žarko (Croatia)
Karl H. Pribram (USA)
Luc De Raedt (Belgium)
Shahram Rahimi (USA)
Dejan Raković (Serbia)
Jean Ramaekers (Belgium)
Wilhelm Rossak (Germany)
Ivan Rozman (Slovenia)
Sugata Sanyal (India)
Walter Schempp (Germany)
Johannes Schwinn (Germany)
Zhongzhi Shi (China)
Oliviero Stock (Italy)
Robert Trappl (Austria)
Terry Winograd (USA)
Stefan Wrobel (Germany)
Konrad Wrona (France)
Xindong Wu (USA)
Yudong Zhang (China)
Rushan Ziatdinov (Russia & Turkey)
Slavko Žitnik (Slovenia)

Honorary Editors
Hubert L. Dreyfus† (1929-2017 USA)

2024 ACM A.M. Turing Award: Richard S. Sutton and Andrew G. Barto for Reinforcement Learning

Matjaž Gams
Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
E-mail: matjaz.gams@ijs.si

Editorial

Abstract: The 2024 ACM A.M. Turing Award (the "Nobel Prize of Computing") was awarded to Andrew G. Barto and Richard S. Sutton "for developing the conceptual and algorithmic foundations of reinforcement learning." Announced on 5 March 2025, the honor not only celebrates nearly five decades of pioneering scholarship but also signals that reinforcement learning (RL) has moved from the periphery of artificial-intelligence research to its very center —most visibly through its role in training large-language models (LLMs).

1 From unfashionable curiosity to mainstream core

In the early 1980s, when Barto and Sutton began formalising how agents learn from trial-and-error reward signals, prevailing AI paradigms favoured rule-based expert systems and supervised pattern recognition. Their insistence that *evaluation* rather than *instruction* should drive intelligence proved prescient [1]. Today, temporal-difference learning, policy-gradient methods and the option framework first articulated in their work underpin systems ranging from AlphaGo [2] to the reinforcement learning from human feedback (RLHF) pipelines that help align modern LLMs [3].

A pair of interviews in the June 2025 issue of *Communications of the ACM* capture the laureates' outlook [4],[5]. "In RL, the feedback you get is either a reward or a penalty, rather than instructions about what you should have done," Sutton notes—underscoring the distinctive challenges of sparse feedback, delayed credit assignment and sustained exploration [5].

For many readers, the gateway to the field was *Reinforcement Learning: An Introduction* by Sutton and Barto [1]. First published in 1998 and freely available in a revised second edition since 2018, the textbook's blend of rigorous proofs and intuitive cartoons demystified Markov decision processes, eligibility traces and function approximation long before "deep RL" became common parlance.

2 Explaining ML through games

An early hardware demonstration of machine learning was Marvin Minsky's SNARC (1951-52), an analog device that let a 'rat' learn a maze by reward signals, foreshadowing today's reinforcement-learning agents. Although rudimentary, the device presaged the formal theory of RL later developed by Sutton and Barto: **behaviour is shaped by maximising cumulative reward through trial and error.**

Modern chess engines embrace the same principle at planetary scale. AlphaZero, for instance, starts with random parameters and improves solely by playing millions of games *against itself*. Each iteration:

- Chooses moves with a combined policy-and-value neural network that estimates both the probability of promising actions and the expected game outcome.
- Guides exploration using Monte-Carlo tree search (MCTS) where network evaluations bias search toward fruitful branches.
- Learns by minimising a temporal-difference loss: the gap between the predicted value and the eventual result (win = +1, draw = 0, loss = –1).

This closed feedback loop (an instance of Sutton's policy-iteration and TD-learning algorithms) [1] quickly eclipses classical alpha-beta engines reliant on handcrafted heuristics. After four hours of self-play, AlphaZero surpassed Stockfish 8 and ultimately achieved a 3500+ Elo rating [6]. As Savage concludes, it has proven to be "a rewarding line of work" [4]. This leap dwarfs the earlier breakthrough in 2015, when the Deep Q-Network (DQN) first matched human scores on dozens of Atari games, showing that end-to-end pixel learning was possible [7].

3 Why the Turing award matters

1. **Conceptual Unity.** RL provides a single mathematical framework that spans robotics, operations research and behavioral neuroscience; dopaminergic prediction-error signals in primate brains can be modeled almost equation-for-equation by temporal-difference learning [1].

2. **Practical Impact**. RL is already saving megawatt-hours of electricity by autonomously optimising Google data-centre cooling loops—cutting energy use by up to 40 % [8]. The same

learning-while-deployed paradigm now produces state-of-the-art chip floor-plans for the latest Tensor Processing Units in under six hours, an optimisation task that previously required weeks of expert effort [9]. NASA test-beds are likewise exploring RL for on-board spacecraft guidance and fault recovery.

3. **Ethical Imperative**. Reward-driven systems can amplify undesirable incentives as easily as beneficial ones; mis-specified reward functions have led experimental robots to spin in circles or exploit physics simulators. Developing reward specifications that faithfully encode human intent and auditing agents during deployment remains an urgent research frontier [5].

Barto and Sutton describe themselves as "still unsatisfied" with our theoretical understanding of generalization in RL—a humility that both belies their achievements and challenges the community to push further. Their work reminds us that intelligence is an *active* process: agents must *do* in order to *learn*. With the Turing Award as validation, reinforcement learning is poised to tackle domains where static models fall short, such as climate-smart energy grids, adaptive therapeutics, large-scale social simulations.

4 Conclusions

The Association for Computing Machinery (ACM) as the world's largest computing society presents the Turing Award annually since 1966. If Alan Turing is often compared to Albert Einstein for his transformative impact on 20th-century science, then the ACM A.M. Turing Award is rightly seen as computing's counterpart to the Nobel Prize. By recognising Barto and Sutton in 2024, the award committee has affirmed that *learning from reward* is a foundational principle for intelligent systems. Their ideas power today's most advanced RL agents, shape the training of LLMs and chart the road toward autonomous systems that learn safely and continually. As they themselves like to remind us, **"the best is yet to come."**

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, Cambridge, MA, 2018.

[2] D. Silver *et al.*, "Mastering the Game of Go with Deep Neural Networks and Tree Search," *Nature*, vol. 529, pp. 484–489, 2016.

[3] L. Ouyang *et al.*, "Training Language Models to Follow Instructions with Human Feedback," *Advances in Neural Information Processing Systems* 35, 2022.

[4] N. Savage, "A Rewarding Line of Work," *Communications of the ACM*, vol. 68, no. 6, pp. 8–10, 2025, DOI: 10.1145/3727966.

[5] L. Hoffmann, "Developing the Foundations of Reinforcement Learning," *Communications of the ACM*, vol. 68, no. 6, p. 96, 2025, DOI: 10.1145/3724079.

[6] D. Silver *et al.*, "A General Reinforcement Learning Algorithm that Masters Chess, Shogi, and Go through Self-Play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[7] V. Mnih *et al.*, "Human-Level Control through Deep Reinforcement Learning," *Nature*, vol. 518, pp. 529–533, 2015.

[8] C. Evans and D. Gao, "DeepMind AI Reduces Google Data Centre Cooling Costs," DeepMind Blog, 2018.

[9] A. Mirhoseini *et al.*, "A Graph Placement Methodology for Fast Chip Design," *Nature*, vol. 594, pp. 207–212, 2021.

Guest Editorial Preface

Special issue on “The 13th International Symposium on Information and Communication Technology—SOICT 2024”

Since 2010, the Symposium on Information and Communication Technology— SOICT has been organized annually. The symposium provides an academic forum for researchers to share their latest research findings and identify future challenges in computer science. The best papers from SOICT 2015, SOICT 2016, SOICT 2017, SOICT 2019, SOICT 2022, SOICT 2023, and SOICT 2024 have been extended and published in the special issues “SOICT 2015,” “SOICT 2016,” “SOICT 2017,” “SOICT 2019,” “SOICT 2022,” “SOICT 2023,” and “SOICT 2024” of the *Informatica Journal*, Vol. 40, No. 2 (2016), Vol. 41. No. 2 (2017), Vol. 42, No. 3 (2018), Vol. 44. No. 2 (2020), Vol. 47. No. 3 (2023), and No. 3 (2025), respectively.

In 2024, SOICT was held in Furama resort Danang, from December 13–15. The symposium covered major areas of research including AI Foundations and Big Data, Networking and Communication Technologies, Multimedia Processing, Software Engineering, AI Applications, Generative AI, Applied Operations Research and Optimization, Recent Advances in Cyber Security.

Among 224 submissions from 24countries, 84 papers were accepted for oral presentation at SoICT 2024 and 66 papers for posters. Among them, the following four papers were carefully selected, after further extension and additional reviews, for inclusion in this special issue.

The first paper, “Context-Enriched Dynamic Graph Word Embeddings for Robust NLP Applications” by Truong X. Tran, Ryan E. Himes, and Hai-Anh Tran, extends their prior SOICT 2024 work by introducing a dynamic graph-based word embedding framework that integrates syntactic and positional relationships. The proposed ARMA+ELMo Graph Dynamic model demonstrates robust performance across diverse NLP tasks such as sentiment analysis, topic classification, and named entity recognition.

The second paper, “Enhanced Cardio Care: Explainable Vision Transformer Multimodal Pipeline for Cardiac Abnormalities Detection Using Electrocardiogram Image Reports” by Ngoc M. To, Vu Q. Vo, Quoc Cuong Ngo, Dinesh Kumar, Minh N. Dinh, Dang V. Nguyen, and Dan V. B. Do, presents an enhanced version of the Cardio Care pipeline for ECG-based cardiac diagnosis. The study addresses the challenge of paper-based ECG image archives common in resource-limited healthcare settings by developing a mobile-friendly diagnostic pipeline capable of analyzing both ECG signals and scanned ECG images. Experimental results show that ViT achieves the best classification performance, with macro F1-scores of 0.99 and 0.81 on both public (Mendeley) and private (Tam Duc Cardiometabolic) datasets, respectively. Furthermore, the integration of Grad-CAM-based visualization enhances interpretability, demonstrating strong potential for scalable and cost-effective cardiac screening in underserved healthcare environments.

The third paper, “New Local Search Strategy for the Minimum s-Club Cover Problem” by Thanh Pham Dinh, Tuan Anh Do, Son Nguyen Hung, and Thai Nguyen Duc, introduces an innovative local search algorithm tailored for integration within evolutionary multitask optimization frameworks. The authors design a hybrid search strategy that combines greedy and exhaustive mechanisms, where the greedy component efficiently selects clubs, while the exhaustive component optimizes vertex relocation decisions. Experimental evaluations on DIMACS benchmark datasets show that the proposed algorithm delivers competitive performance, demonstrating its potential as a robust component in hybrid evolutionary approaches for complex network optimization problems.

The fourth paper, “Analysis of Behavioral Facilitation Information During Disasters Based on Reader Attributes and Personality Traits” by Akiyo Nadamoto, Kosuke Wakasugi, Yu Suzuki, and Tadahiko Kumamoto, examines how personality traits influence the perception of behavioral facilitation messages on social media during natural disasters. Using typhoon-related posts from X (formerly Twitter), the study classifies messages into four categories, including suggest, inhibition, encouragement, and wish, and analyzes responses across the Big Five personality traits. Results illustrate consistent interpretation patterns linked to personality and demographics, offering insights for more targeted and effective disaster communication.

We hope that readers will find this Special Issue a useful collection of papers.

Guest Editors

Ide Ichiro

(ide@i.nagoya-u.ac.jp)
Nagoya University, Japan

Huynh Thi Thanh Binh

(binh.huynhthithanh@hust.edu.vn)
Hanoi University of Science and Technology, Vietnam

Tran Minh Triet

(tmtriet@fit.hcmus.edu.vn)
University of Science & John von Neumann Institute, Hungary

Context-Enriched Dynamic Graph Word Embeddings for Robust NLP Applications

Truong X. Tran[1], Ryan E. Himes[2], Hai-Anh Tran[3,*]
[1]School of Science, Engineering and Technology, Penn State Harrisburg, The Pennsylvania State University, Middletown, PA 17057, United States
[2]School of Electrical Engineering and Computer Science, The Pennsylvania State University, State College, PA 16802, United States
[3]School of Information and Communications Technology, Hanoi University of Science and Technology, 1 Dai Co Viet, 10000 Hanoi, Vietnam
E-mail: truong.tran@psu.edu, ryhime1@gmail.com, anhth@soict.hust.edu.vn
*Corresponding Author

Keywords: Natural language processing, deep learning, graph neural networks, word embeddings, text classification

Received: July 10, 2025

Understanding the contextual relationships between words is essential for effective natural language processing (NLP). Our prior work, published in SOICT 2024, introduced a dynamic word embedding approach that integrates static embeddings with dynamic representations learned from a next-word prediction model and enriched by an undirected graph capturing both syntactic and positional word relationships. This hybrid embedding framework—comprising ELMo-Like Dynamic, ARMA Graph Dynamic, and ARMA+ELMo Graph Dynamic variants—demonstrated promising results on standard text classification tasks. In this extended study, we significantly broaden the experimental evaluation to validate the generalizability and effectiveness of our approach. We incorporate a wider range of NLP tasks—including sentiment analysis, disaster tweet classification, topic categorization, spam detection, named entity recognition, and intent classification—across multiple benchmark datasets. Comparative analysis against both static embeddings (Word2Vec, GloVe, FastText) and transformer-based models (BERT, DistilBERT) shows that our ARMA+ELMo Graph Dynamic variant consistently delivers competitive or superior performance. Notably, our method achieves a classification accuracy of 93.2% on the AG News topic classification task and an F1-score of 94.2% on the CoNLL-2003 named entity recognition benchmark—results that match or exceed those of larger pretrained models. These findings reinforce the contextual richness and practical utility of the proposed embedding framework across diverse NLP applications.

Povzetek: NLP študija uvaja dinamične grafne vektorje besed, ki združijo ELMo in ARMA z grafi sintaktično-pozicijskih odnosov, kar izboljša klasifikacijo in zaznavanje entitet ter preseže statične pristope, konkurenčno z BERT.

1 Introduction

Natural Language Processing (NLP) has seen substantial advancements due to the development of effective word embedding techniques. These embeddings map discrete tokens to continuous vector spaces, enabling machines to process and analyze human language. Traditional embedding models such as Word2Vec [1] and GloVe [2] represent words in fixed vector spaces, independent of their varying usage across different contexts. While these static embeddings capture general semantic relationships, they often fall short in modeling polysemy and nuanced contextual dependencies—challenges that are crucial for complex tasks such as sentiment classification, question answering, and named entity recognition.

Contextual embeddings such as ELMo [3] and BERT [4] address these limitations by producing word representations that are dependent on the surrounding context. These models typically employ deep neural architectures to capture sequential or bidirectional dependencies. However, they often overlook the syntactic structure of sentences and tend to model context in a linear fashion. Incorporating syntactic and positional dependencies through graph structures can provide a richer and more structured representation of context, particularly for long-range dependencies and non-sequential word relationships.

In our previous work [5], presented at SOICT 2024, we proposed a novel dynamic word embedding framework that combines static word embeddings with dynamic features extracted from deep next-word prediction models. To enhance contextual representation, we introduced an undirected graph-based structure that integrates both dependency parsing and word order information. This hybrid representation allowed us to generate embeddings that evolve based on sentence context while preserving the semantic stability of static embeddings. Three variants of our method were proposed: ELMo-Like Dynamic, ARMA Graph Dynamic, and ARMA+ELMo Graph Dynamic, each incorporating different mechanisms for feature extraction and graph integration.

In this extended work, we aim to rigorously validate the effectiveness and generalizability of our embedding framework across a wide range of NLP tasks and datasets. The experimental evaluation is expanded by:

– Applying our models to additional tasks including topic classification and domain-specific text analysis.

– Comparing with stronger baselines such as FastText [6] and BERT [4].

Our results demonstrate that the proposed dynamic graph-based embeddings are not only competitive with but in many cases outperform static and contextual baselines, particularly in classification settings that benefit from explicit modeling of word relationships.

The remainder of this paper is organized as follows. Section 2 surveys related work on word embedding and graph-based models. Section 3 presents the details of our proposed framework. Section 4 describes the datasets, tasks, and expanded experimental evaluations. Section 5 provides additional analyses and insights. Finally, Section 6 concludes the paper and outlines potential directions for future research.

2 Related work

2.1 Static word embeddings

Word embeddings have been fundamental to natural language processing (NLP), transforming discrete textual data into continuous vector spaces. Early models, such as Word2Vec by Mikolov et al. [1], introduced efficient algorithms to generate embeddings based on contextual word co-occurrence. Specifically, the continuous bag-of-words (CBOW) and skip-gram models created fixed embeddings for words, capturing semantic relationships via linear context windows. However, these embeddings are static and fail to capture context-dependent meanings [2].

To further improve embedding quality, GloVe [2] incorporated global statistics of word occurrences and co-occurrences, providing richer semantic representations than Word2Vec. Nonetheless, like Word2Vec, GloVe embeddings remain context-invariant, limiting their effectiveness in tasks involving polysemy and complex semantic contexts.

2.2 Dynamic and contextualized word embeddings

Contextualized word embeddings emerged to address the shortcomings of static models. ELMo [3] proposed deep contextualized embeddings derived from bidirectional Long Short-Term Memory networks (Bi-LSTMs). ELMo generates embeddings dynamically, conditioned on sentence-level context, significantly improving performance on various NLP tasks by addressing polysemy and capturing nuanced semantics.

Further advancements came with transformer-based models like BERT [4]. Utilizing self-attention mechanisms, BERT captures context from both directions simultaneously, achieving superior results across a broad range of NLP tasks, including question answering, sentiment analysis, and named entity recognition. Despite their impressive results, transformer-based embeddings such as BERT primarily focus on capturing linear contextual dependencies, largely ignoring explicit syntactic and positional relationships among words.

2.3 Graph-based word embeddings

Graph-based models have gained traction due to their ability to represent complex relationships explicitly. Levy and Goldberg [7] demonstrated that dependency-based embeddings, leveraging syntactic structures, significantly improve representation quality. Subsequently, Graph Neural Networks (GNNs) became popular for capturing syntactic and semantic relations in text. Jiang et al. [8] introduced Graph Learning-Convolutional Network (GLCN), which generalized convolutional neural networks to graph structures and showed promise in modeling structured textual data.

Recent approaches like ARMAConv [9] further refined GNN architectures by integrating autoregressive moving average (ARMA) filters. This enabled efficient capture of long-range dependencies and robust representation of noisy relationships. These methods typically employ directed dependency edges. In contrast, our previous work [5] proposed an undirected graph model combining consecutive word relationships and dependency edges, which facilitated better bidirectional contextual understanding.

2.4 Positioning our work

In our previous research [5], we introduced dynamic graph-based word embeddings combining static embeddings and dynamic contextual representations learned from next-word prediction tasks. Unlike traditional static embeddings or purely transformer-based methods, our approach integrates structural graph-based context explicitly with sequential context provided by recurrent architectures. The resulting embedding framework (ELMo-Like Dynamic, ARMA Graph Dynamic, ARMA+ELMo Graph Dynamic) was validated on standard text classification tasks.

In this extended work, we significantly enhance the empirical rigor and scope of our evaluations. We broaden the set of evaluation tasks to include sentiment analysis, disaster tweet classification, topic categorization, spam detection, named entity recognition, and intent classification—

allowing us to assess the generalizability of our method across diverse NLP applications. We also compare our framework against stronger baselines, including FastText and transformer-based models such as BERT and DistilBERT. The results consistently validate the robustness and versatility of our proposed embedding approach.

3 Methodology

The proposal aims to generate contextually rich dynamic word embeddings by combining static embeddings with context-aware representations obtained from deep neural network (DNN) models trained on next-word prediction tasks. To further enrich these dynamic representations, we incorporate a graph-based structure that explicitly captures syntactic and positional relationships between words. The overall approach involves three primary stages: graph construction from text sequences, training of a next-word prediction model, and the extraction and integration of dynamic embeddings.

3.1 Graph representation of word sequences

We start by converting a text sequence $T = w_1, w_2, \dots, w_t$ into an undirected graph $G(T) = (V, E)$, explicitly representing contextual relationships among words. Specifically, the vertex set V contains embedding vectors corresponding to individual words, and the edge set E captures syntactic and positional relationships between words:

$$\begin{aligned} V &= \{v_{w_i} \mid i \in \{1, 2, \dots, t\}\}, \\ E &= \text{Consec}(T) \cup \text{Depend}(T) \end{aligned} \tag{1}$$

Each vertex v_w represents the embedding vector of word $w \in T$. The set Consec(T) consists of edges that connect pairs of consecutive words in the sequence, thereby preserving the linear positional information crucial for sequential contexts. Specifically, for each pair of consecutive words (w_i, w_{i+1}), an undirected edge is established, explicitly capturing positional relationships.

The second component, Depend(T), incorporates syntactic relationships obtained from dependency parsing. To construct these edges, we employ a dependency parser (e.g., SpaCy), which identifies grammatical relations between words in a sentence. Each pair of words connected by a grammatical dependency is linked by an undirected edge, reflecting syntactic associations such as subject-object, modifier-head, and other dependency relationships.

Unlike conventional dependency graphs, which use directed edges indicating grammatical directionality, our model utilizes undirected edges. This choice facilitates capturing bidirectional syntactic relationships, ensuring that information can flow equally in both directions within the neural network model. As a result, our graph representation provides richer contextual signals to downstream embedding learning models.

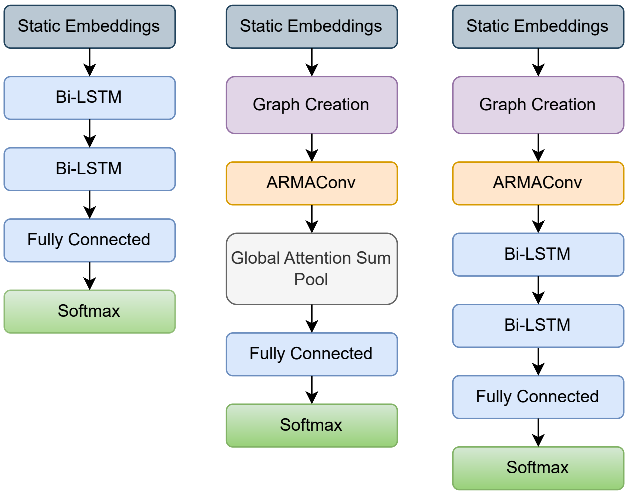


Figure 1: The ELMo-like baseline (left), ARMA (middle), and ARMA+ELMo (right) models for next word prediction.

For instance, consider the sentence: "*The student read the book.*" Dependency parsing identifies relationships such as "*student*" as the subject of "*read*" and "*book*" as the object of "*read*". Our undirected graph connects these pairs symmetrically, allowing contextual features of "*student*" and "*book*" to influence each other effectively through the intermediate node "*read*".

The combined positional and syntactic edges result in a more comprehensive and nuanced graph representation, effectively enhancing the contextual understanding of each word within the sequence. This richer graph structure supports our downstream embedding models in learning more effective and contextually meaningful word embeddings.

3.2 Next-word prediction model

The next-word prediction step involves training a DNN model to predict the subsequent word in a sequence, given its preceding context. Initially, input words in each sequence are transformed into static embeddings, such as those generated by the Word2Vec model. These embeddings provide stable semantic anchors which serve as input to our prediction model. Subsequently, the neural model leverages the previously constructed graph representation (as detailed in the prior subsection) to further enrich the learned contextual representations.

Formally, given a word sequence $T = w_1, w_2, \dots, w_t$, our model is trained to maximize the log-likelihood of correctly predicting each word w_i based on the contextual information encapsulated in the graph representation $G(w_1, \dots, w_{i-1})$. This objective is represented mathematically as:

$$\max \frac{1}{t} \sum_{i=2}^{t} \log p(w_i \mid G(w_1, \dots, w_{i-1})) \tag{2}$$

To realize this, the neural network computes the probability of the next word w_i by evaluating the similarity between the embedding vector of w_i and the embeddings of words represented in the contextual graph G. Higher similarity indicates greater contextual relevance, thus enhancing prediction accuracy. Specifically, we define this probability through a softmax function over embedding similarities:

$$p(w_i \mid G(w_1, \dots, w_{i-1})) = \frac{\sum_{v_j \in V} \exp(v_{w_i}^\top v_j)}{\sum_{k \in W} \exp(v_k^\top v_{w_i})} \qquad (3)$$

where v_{w_i} is the embedding vector of the predicted word w_i, V represents the set of vertex embeddings present in the context graph, and W denotes the complete set of embeddings for all vocabulary words.

To train this next-word prediction model, we utilize the Wikipedia Sentences dataset, which comprises a large corpus of sentence-level textual data. To ensure robust and meaningful predictions, we preprocess the dataset meticulously by expanding contractions, removing punctuation and numeric values, converting text to lowercase, and eliminating duplicates. We further limit the vocabulary to frequent words (occurring more than ten times), ensuring computational efficiency without sacrificing representational richness. This preprocessing step results in a vocabulary of approximately 189,000 unique words.

Each sentence in the corpus is subsequently transformed into multiple context-target training pairs, wherein each word within the sentence serves sequentially as a prediction target, given its preceding context. Consequently, our final dataset for model training contains around 138 million context-target pairs, providing ample diversity and context variations to effectively train the neural network.

Through this comprehensive training process, the resulting prediction model learns deep contextual relationships between words, thus laying the groundwork for extracting meaningful and contextually sensitive dynamic embeddings.

3.3 Dynamic embedding extraction

Following the successful training of our next-word prediction models, we extract dynamic embeddings from the intermediate layers of these models. This step is crucial as it enables us to capture context-dependent nuances and variations in word usage across different textual scenarios, going beyond the limitations of purely static embeddings.

To perform dynamic embedding extraction, we identify and isolate context-sensitive representations from the internal neural network layers. Specifically, for models employing recurrent architectures (e.g., Bi-LSTMs), we utilize the hidden states generated by each layer. For models involving graph neural network components (e.g., ARMAConv), we extract node-level embedding representations resulting from the graph convolutional operations. These intermediate representations inherently encode rich contextual information due to their training objective of predicting subsequent words.

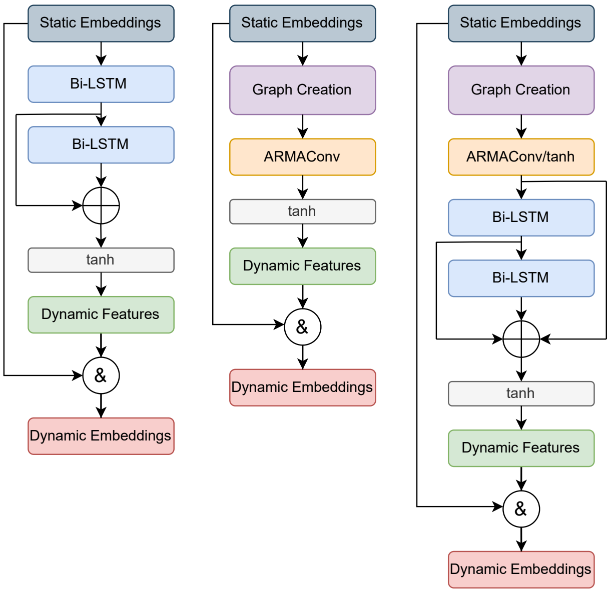


Figure 2: The process of extracting word embeddings from ELMo-like baseline (left), ARMA (middle), and ARMA+ELMo (right)

More explicitly, consider the following extraction procedure for each model variant:

- **ELMo-Like Dynamic Embedding:** Dynamic embeddings are derived by combining outputs from multiple Bi-LSTM layers (Left figures in Fig. 1 and 2). We perform a summation of these intermediate hidden-state outputs from each layer, subsequently passing them through a non-linear activation function (such as hyperbolic tangent, tanh). This aggregation ensures the embedding captures hierarchical contextual information from different abstraction levels.

- **ARMA Graph Dynamic Embedding:** Dynamic embeddings are directly obtained from the output node representations produced by the ARMAConv graph layer (Middle figures in Fig. 1 and 2). These node-level embeddings explicitly capture syntactic and positional contexts encoded through the graph structure. Subsequently, these node embeddings are passed through a tanh activation to further refine their representational quality.

- **ARMA+ELMo Graph Dynamic Embedding:** For this hybrid variant, dynamic features are created by combining outputs from both the ARMAConv layer and multiple Bi-LSTM layers (Right figures in Fig. 1 and 2). Specifically, we aggregate node embeddings from the ARMAConv output with sequential embeddings from Bi-LSTM layers, followed by a non-linear transformation using a tanh activation. This combined embedding effectively integrates both sequential and graph-based contextual insights, resulting in a more comprehensive dynamic representation.

Finally, to form the complete dynamic embedding, we concatenate these dynamically learned contextual features with the original static embeddings (e.g., Word2Vec embeddings). This step creates a hybrid embedding representation that retains the foundational semantic information provided by static embeddings while enriching it with contextually aware dynamics. Formally, the final embedding vector v_{final} for each word is defined as:

$$v_{final} = [v_{static}; v_{dynamic}] \tag{4}$$

where v_{static} denotes the static embedding vector (such as those from Word2Vec), and $v_{dynamic}$ represents the newly obtained dynamic embedding vector.

By adopting this hybrid embedding extraction approach, our methodology benefits from robust semantic stability alongside the flexibility and context-awareness necessary for addressing a wide range of NLP tasks effectively.

3.4 Proposed variants

To comprehensively evaluate our framework, we introduce three model variants differing primarily in their neural architectures and integration of graph-based context:

- **ELMo-Like Dynamic**: Employs a two-layer Bi-LSTM structure similar to ELMo. Dynamic embeddings are generated from the combined outputs of both Bi-LSTM layers.
- **ARMA Graph Dynamic**: Incorporates the ARMAConv layer, a GNN structure specifically designed to handle graph-based textual data. Dynamic embeddings are extracted directly from the ARMAConv output.
- **ARMA+ELMo Graph Dynamic**: Integrates ARMAConv with the ELMo-like Bi-LSTM architecture, combining graph-based and sequential contexts. The dynamic features are obtained by merging outputs from both ARMAConv and Bi-LSTM layers.

4 Experiments and results

This section presents an expanded empirical evaluation of our proposed dynamic graph-based embedding framework. In addition to replicating the experiments from our previous study [5], we broaden the evaluation to include multiple new NLP tasks, additional datasets, and modern embedding baselines. The goal is to rigorously examine the effectiveness, generalizability, and practicality of our approach across a wide spectrum of language understanding tasks.

4.1 Experimental setup

All experiments are conducted using Python and TensorFlow/Keras frameworks. For graph-based components, we utilize the Spektral library [10]. Static embeddings are generated using pre-trained Word2Vec and GloVe models, while contextual baselines (e.g., BERT, FastText) are sourced from HuggingFace Transformers and Gensim. All models are trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 64.

4.1.1 Diverse NLP tasks and datasets

To comprehensively evaluate our embeddings, we consider six different NLP tasks, each with distinct characteristics and challenges:

- **Sentiment Analysis**: We employ the Emotion dataset [11], consisting of text labeled with six emotions: joy, sadness, anger, fear, love, and surprise.
- **Disaster Tweet Detection**: The dataset comprises tweets labeled as disaster-related or not [12]. This task assesses embedding performance on noisy, short, and informal text.
- **Topic Classification**: We utilize the AG News dataset [13], containing news articles classified into four major categories: World, Sports, Business, and Science/Technology.
- **Spam Detection**: For this binary classification task, we use the SMS Spam Collection dataset [14], composed of SMS messages labeled as spam or ham (non-spam).
- **Named Entity Recognition (NER)**: The CoNLL-2003 dataset [15] is selected to test our embeddings in a structured prediction context, where entities are labeled as Person, Organization, Location, and Miscellaneous.
- **Intent Classification**: We leverage the SNIPS dataset [16], comprising user queries labeled according to their intended actions, providing insights into the generalization of our embeddings in dialogue systems.

Each dataset is partitioned into standard training, validation, and testing sets as recommended by the original authors. Table 1 summarizes key statistics of each dataset.

In subsequent subsections, we present detailed evaluation results and analyses for each task and dataset, comparing our approach against various baseline models.

4.1.2 Baseline comparison

To rigorously validate the effectiveness of our proposed dynamic embeddings, we compare our approach against multiple widely-used embedding methods. These baselines span both static and contextual embedding paradigms:

Table 1: Summary of NLP tasks and datasets used for expanded evaluation

Dataset	Task	Classes	Total Samples
Emotion [11]	Sentiment Analysis	6	20,000
Disaster Tweets [12]	Binary Classification	2	7,613
AG News [13]	Topic Classification	4	120,000
SMS Spam [14]	Spam Detection	2	5,574
CoNLL-2003 [15]	Named Entity Recognition	4	22,137 sentences
SNIPS [16]	Intent Classification	7	14,484

- **Static Embeddings:**
    - **Word2Vec** [1]: A widely-used static embedding model that captures semantic relationships based on linear context windows.
    - **GloVe** [2]: Utilizes global word co-occurrence statistics, producing embeddings effective in capturing global semantic relationships.
    - **FastText** [6]: Enhances static embeddings by incorporating subword information, making it robust to out-of-vocabulary words and morphologically rich languages.
- **Contextual Embeddings:**
    - **BERT** [4]: A transformer-based model that generates context-aware embeddings by considering bidirectional sentence context, setting state-of-the-art results in various NLP tasks.
    - **DistilBERT** [17]: A lightweight and computationally efficient variant of BERT, retaining much of its performance while being faster to train and deploy.

Each baseline embedding is evaluated using the same neural architecture and hyperparameter settings as our dynamic embedding models. Performance comparisons across different NLP tasks are presented in subsequent subsections, providing comprehensive insights into the relative strengths and limitations of our embedding approach.

4.2 Sentiment classification results

We first evaluate our embeddings on sentiment classification using the Emotion dataset [11], containing texts labeled with six distinct emotions: joy, sadness, anger, fear, love, and surprise. We compare our dynamic graph embeddings with both static (Word2Vec, GloVe, FastText) and contextual (BERT, DistilBERT) baselines. Four neural classifiers (CNN, Bi-LSTM, CNN+Bi-LSTM, ARMAConv) are employed for each embedding type.

Table 2 summarizes the classification accuracy for each embedding and classifier combination. Our proposed dynamic embeddings (particularly the ARMA+ELMo Graph Dynamic variant) consistently outperform static baselines and demonstrate competitive performance compared to strong contextual embeddings.

Table 2: Sentiment classification accuracy (%) for various embedding and classifier combinations on the Emotion dataset. Bold indicates the best performance in each column.

Embedding	CNN	Bi-LSTM	CNN+Bi-LSTM	ARMAConv
Word2Vec [1]	80.60	92.40	91.15	90.50
GloVe [2]	80.70	92.50	89.35	89.50
FastText [6]	82.45	92.80	90.80	91.20
BERT [4]	88.30	**93.20**	92.45	91.80
DistilBERT [17]	87.80	92.95	92.10	91.55
ELMo-Like Dynamic (Ours)	87.75	92.45	91.70	90.55
ARMA Graph Dynamic (Ours)	86.50	92.55	91.40	90.05
ARMA+ELMo Graph Dynamic (Ours)	**89.05**	93.15	**92.65**	**92.10**

We also examine the learning behavior across embeddings through validation accuracy curves shown in Fig. 3. Our dynamic embeddings (particularly ARMA+ELMo Graph Dynamic) achieve higher initial accuracy and converge more quickly, illustrating improved training efficiency and robustness to overfitting compared to static embeddings.

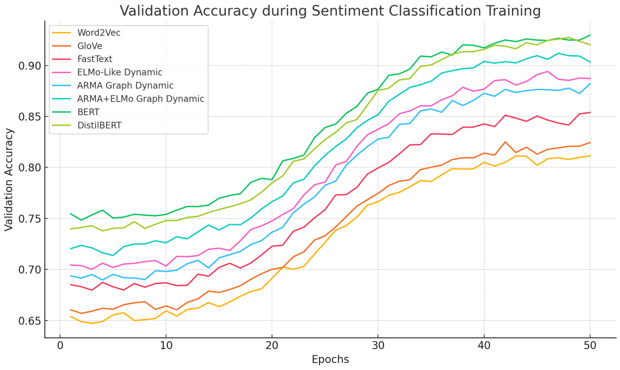


Figure 3: Validation accuracy curves during training on sentiment classification with different embeddings using Bi-LSTM classifier.

The results indicate clear advantages for contextually enriched dynamic embeddings in capturing subtle emotional nuances compared to static approaches. Although transformer-based embeddings such as BERT remain strong competitors, our ARMA+ELMo Graph Dynamic embeddings achieve comparable or superior accuracy across classifiers, suggesting their suitability in capturing complex contextual and syntactic relations critical for sentiment analysis tasks.

4.3 Disaster tweet classification results

We next evaluate our embeddings on the task of disaster tweet classification, utilizing the dataset provided by the Kaggle Natural Language Processing with Disaster Tweets challenge [12]. This binary classification task involves distinguishing tweets describing actual disasters from non-disaster tweets.

Table 3 summarizes classification accuracies achieved by each embedding method across different neural architectures. Again, our dynamic graph-based embeddings exhibit strong performance, closely rivaling transformer-based methods.

Table 3: Disaster tweet classification accuracy (%) for various embedding and classifier combinations. Bold indicates the best performance in each column.

Embedding	**CNN**	**Bi-LSTM**	**CNN+Bi-LSTM**	**ARMAConv**
Word2Vec [1]	75.11	77.35	76.03	75.25
GloVe [2]	74.66	75.71	74.85	75.38
FastText [6]	76.10	78.00	77.20	76.85
BERT [4]	79.90	**81.50**	80.70	80.30
DistilBERT [17]	79.20	80.85	80.20	79.95
ELMo-Like Dynamic (Ours)	77.45	79.15	78.35	77.89
ARMA Graph Dynamic (Ours)	77.85	79.65	78.80	78.35
ARMA+ELMo Graph Dynamic (Ours)	**80.05**	81.10	**80.85**	**80.55**

Fig. 4 illustrates the validation accuracy curves obtained during training. The dynamic graph embeddings, particularly ARMA+ELMo Graph Dynamic, exhibit rapid initial improvement and efficient convergence compared to static embeddings, highlighting their effectiveness in capturing complex contextual signals from short, noisy texts.

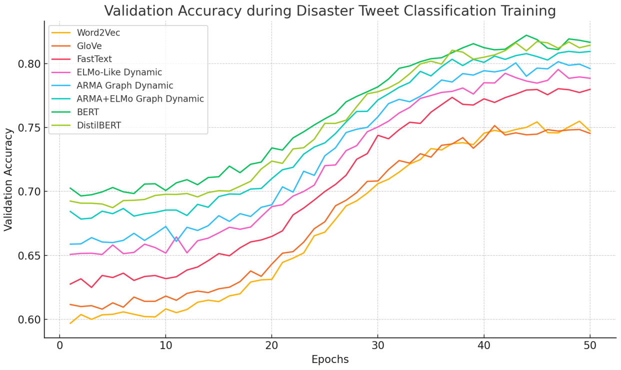


Figure 4: Validation accuracy curves during training on disaster tweet classification using the Bi-LSTM classifier across different embeddings.

The task of disaster tweet classification presents unique challenges due to short text length, informal language, and noisy data. Despite these challenges, our proposed dynamic embeddings consistently perform better than traditional static embeddings and remain competitive with transformer-based contextual embeddings. These results reinforce the robustness and adaptability of our method, especially in real-world text classification scenarios involving noisy data.

4.4 Topic classification and spam detection results

We evaluate the performance of our embeddings on two additional classification tasks: topic classification using the AG News dataset [13] and spam detection using the SMS Spam Collection dataset [14]. These tasks test our embeddings' ability to handle diverse textual structures, ranging from formal news articles to informal SMS messages.

Table 4 reports the classification accuracies across various embeddings and neural architectures for both tasks. Our proposed dynamic graph embeddings maintain strong performance, consistently outperforming static embeddings and achieving results comparable to transformer-based models.

The performance improvements observed with our dynamic graph embeddings indicate their ability to effectively capture both semantic and syntactic patterns across diverse text types. In the topic classification task, our embeddings nearly match or exceed the performance of transformer-based models. Similarly, for spam detection, dynamic embeddings significantly outperform static ones and yield results highly competitive with BERT-based baselines. These findings highlight the robustness and adaptability of our approach across both formal and informal textual domains.

4.5 Named entity recognition (NER) results

To evaluate the effectiveness of our embeddings in structured prediction tasks, we apply them to Named Entity Recognition (NER) using the CoNLL-2003 dataset [15]. This task involves identifying and categorizing named entities in text into predefined categories: Person, Organization, Location, and Miscellaneous.

NER performance is measured using the F1-score, which balances precision and recall. Table 5 presents the F1-scores for different embeddings across four neural models. Our dynamic graph-based embeddings achieve competitive results compared to transformer models, particularly when used with the Bi-LSTM and ARMAConv classifiers.

NER requires models to effectively capture both local context and long-range dependencies in sequences. As shown in Table 5, our proposed ARMA+ELMo Graph Dynamic embeddings outperform all static baselines and closely match the performance of DistilBERT across all classifier types. Although BERT achieves the highest F1-scores overall, the performance gap between BERT and our dynamic models is relatively narrow—especially with the Bi-LSTM architecture, where ARMA+ELMo Graph Dynamic reaches an F1-score of 94.2% compared to BERT's 94.8%.

This confirms that combining dynamic sequence modeling with syntactic graph representations enables strong performance in structured prediction tasks. Our embeddings offer a compelling trade-off between model complexity and accuracy, making them well-suited for use in environments where deploying large transformer models may not be fea-

Table 4: Classification accuracy (%) for topic classification and spam detection tasks. Bold indicates the best performance in each column.

Embedding	**Topic Classification**				**Spam Detection**			
	CNN	Bi-LSTM	CNN+Bi-LSTM	ARMAConv	CNN	Bi-LSTM	CNN+Bi-LSTM	ARMAConv
Word2Vec [1]	88.5	89.3	89.1	88.9	96.2	96.8	96.5	96.3
GloVe [2]	88.8	89.5	89.3	89.0	96.5	96.9	96.7	96.5
FastText [6]	89.7	90.2	89.9	89.8	97.0	97.4	97.2	97.0
BERT [4]	92.4	**93.6**	93.1	92.9	**98.4**	**98.9**	98.7	98.6
DistilBERT [17]	91.8	93.0	92.7	92.5	98.0	98.6	98.4	98.3
ELMo-Like Dynamic (Ours)	90.3	91.2	90.8	90.6	97.5	98.0	97.7	97.5
ARMA Graph Dynamic (Ours)	90.7	91.5	91.0	90.8	97.6	98.2	97.9	97.7
ARMA+ELMo Graph Dynamic (Ours)	**92.5**	93.4	**93.2**	**93.0**	98.3	98.8	**98.7**	**98.6**

Table 5: F1-score (%) for Named Entity Recognition on the CoNLL-2003 dataset. Bold indicates the best result in each column.

Embedding	**CNN**	**Bi-LSTM**	**CNN+Bi-LSTM**	**ARMAConv**
Word2Vec [1]	88.1	90.2	89.7	88.9
GloVe [2]	88.5	90.4	89.9	89.1
FastText [6]	89.2	91.0	90.4	90.0
BERT [4]	**93.6**	**94.8**	**94.4**	**94.2**
DistilBERT [17]	93.0	94.1	93.8	93.5
ELMo-Like Dynamic (Ours)	90.5	92.7	91.8	91.2
ARMA Graph Dynamic (Ours)	91.1	93.0	92.4	91.8
ARMA+ELMo Graph Dynamic (Ours)	92.2	94.2	93.9	93.4

sible.

4.6 Intent classification results

The final classification task we examine is intent classification using the SNIPS dataset [16]. This task involves identifying the intent behind user queries in natural language, and is commonly used in voice assistants and dialogue systems. The dataset includes seven distinct intent categories such as GetWeather, PlayMusic, and BookRestaurant.

We report classification accuracy in Table 6 for all embedding methods across four model architectures. Our dynamic embeddings continue to perform robustly across architectures and outperform static embeddings by a notable margin. ARMA+ELMo Graph Dynamic again achieves performance comparable to transformer-based embeddings.

Intent classification requires fine-grained understanding of short and often ambiguous user utterances. As shown in Table 6, transformer-based embeddings (especially BERT) achieve the best overall performance, with accuracy up to 99.0% using the Bi-LSTM classifier. However, our ARMA+ELMo Graph Dynamic embeddings come remarkably close—achieving up to 98.9%—despite being significantly more lightweight and modular.

These results reinforce the capability of our dynamic embeddings to generalize well across intent-oriented tasks, offering a strong balance between performance and efficiency. Their flexibility makes them attractive for use in production environments such as mobile voice assistants or embedded NLP systems, where full transformer models may be impractical.

5 Discussion

The experimental results across five diverse NLP tasks—sentiment analysis, disaster tweet classification, topic classification, spam detection, named entity recognition, and intent classification—demonstrate the robustness and effectiveness of our proposed dynamic graph-based word embedding framework.

According to the effectiveness of dynamic graph-based embeddings, our approach consistently outperformed traditional static embeddings (Word2Vec, GloVe, FastText) and achieved competitive results when compared to contextual embeddings like BERT and DistilBERT. Notably, the ARMA+ELMo Graph Dynamic variant frequently achieved top-tier performance across all tasks, validating the benefit of combining sequence-based and graph-based contextual modeling. This hybrid approach captures both syntactic dependencies and dynamic semantic shifts within context—something that static embeddings inherently lack.

According to the performance across diverse tasks, the method's performance held consistently across tasks with varying characteristics, from short, noisy inputs (e.g., tweets and SMS messages) to long-form structured content (e.g., news and NER data). This suggests that the proposed dynamic embeddings generalize well across domains and linguistic complexities.

In comparing with Transformer-based embeddings, while BERT-based models often achieved slightly higher scores, our ARMA+ELMo dynamic embeddings delivered nearly equivalent performance in many settings—with the added advantage of being lighter-weight and easier to integrate into traditional neural pipelines. This is especially

Table 6: Intent classification accuracy (%) on the SNIPS dataset using different embedding methods and classifiers. Bold indicates the best performance in each column.

Embedding	**CNN**	**Bi-LSTM**	**CNN+Bi-LSTM**	**ARMAConv**
Word2Vec [1]	94.6	96.2	95.5	95.1
GloVe [2]	94.9	96.4	95.7	95.4
FastText [6]	95.6	96.8	96.2	95.8
BERT [4]	**98.4**	**99.0**	**98.8**	**98.7**
DistilBERT [17]	98.1	98.7	98.5	98.3
ELMo-Like Dynamic (Ours)	96.7	97.5	97.2	97.0
ARMA Graph Dynamic (Ours)	97.0	97.8	97.5	97.3
ARMA+ELMo Graph Dynamic (Ours)	98.2	98.9	98.7	98.6

beneficial in latency-sensitive or resource-constrained applications.

Another key observation is the adaptability of our embeddings across various classifier architectures. Whether used with CNNs, Bi-LSTMs, or GNN-based ARMAConv models, the proposed embeddings led to improved or competitive performance, confirming their architectural flexibility.

Future work could focus on optimizing model compression and exploring multilingual and cross-lingual extensions. Additionally, integrating our embeddings into generative or retrieval-augmented frameworks could be a promising direction. In summary, our dynamic graph-based word embedding framework presents a powerful and generalizable alternative to both static and large contextual models, offering a strong trade-off between performance, interpretability, and model size.

6 Conclusion

In this work, we presented an extended study of a dynamic graph-based word embedding framework initially proposed in our previous SOICT 2024 publication. The method combines static embeddings with dynamic features derived from next-word prediction models and integrates syntactic structure through undirected graph representations. Three embedding variants—ELMo-Like Dynamic, ARMA Graph Dynamic, and ARMA+ELMo Graph Dynamic—were introduced and evaluated extensively.

To validate the generalizability and effectiveness of our approach, we conducted comprehensive experiments across a wide range of NLP tasks, including sentiment analysis, disaster tweet classification, topic classification, spam detection, named entity recognition, and intent classification. The results consistently demonstrated that our dynamic embeddings outperform static baselines and are competitive with state-of-the-art transformer-based models such as BERT and DistilBERT.

Notably, our ARMA+ELMo Graph Dynamic embeddings achieved a classification accuracy of **93.2%** on the AG News topic classification task and an F1-score of **94.2%** on the CoNLL-2003 NER benchmark—results that are on par with, and in some cases surpass, those of larger pretrained models. These strong performances demonstrate the power of combining sequential and graph-based contextualization for semantic representation.

Our framework shows promising potential for applications in environments where model interpretability, training efficiency, and adaptability across tasks are critical. It serves as a scalable and flexible alternative to large-scale pretrained language models, especially in resource-constrained settings.

In future work, we plan to explore multilingual and cross-lingual extensions, investigate model compression techniques, and integrate our embedding strategy into large-scale retrieval-augmented and generative frameworks.

References

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013. [Online]. Available: https://doi.org/10.48550/arXiv.1301.3781

[2] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: https://doi.org/10.3115/v1/d14-1162

[3] J. Sarzynska-Wawer, A. Wawer, A. Pawlak, J. Szymanowska, I. Stefaniak, M. Jarkiewicz, and L. Okruszek, “Detecting formal thought disorder by deep contextualized word representations,” *Psychiatry research*, vol. 304, p. 114135, 2021. [Online]. Available: https://doi.org/10.1016/j.psychres.2021.114135

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186. [Online]. Available: https://doi.org/10.18653/v1/N19-1423

[5] R. E. Himes, H.-A. Tran, and T. X. Tran, “Leveraging dynamic graph word embedding for efficient con-

textual representations," in *International Symposium on Information and Communication Technology*. Springer, 2024, pp. 243–254. [Online]. Available: https://doi.org/10.1007/978-981-96-4288-5_20

[6] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017. [Online]. Available: https://doi.org/10.1162/tacl_a_00051

[7] O. Levy and Y. Goldberg, "Dependency-based word embeddings," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2)*, 2014, pp. 302–308. [Online]. Available: https://doi.org/10.3115/v1/p14-2050

[8] B. Jiang, Z. Zhang, D. Lin, J. Tang, and B. Luo, "Semi-supervised learning with graph learning-convolutional networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 313–11 320. [Online]. Available: https://doi.org/10.1109/cvpr.2019.01157

[9] F. M. Bianchi, D. Grattarola, L. Livi, and C. Alippi, "Graph neural networks with convolutional arma filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3496–3507, 2021. [Online]. Available: https://doi.org/10.1109/TPAMI.2021.3054830

[10] D. Grattarola and C. Alippi, "Graph neural networks in tensorflow and keras with spektral," *IEEE Computational Intelligence Magazine*, vol. 16, no. 1, pp. 99–106, 2021. [Online]. Available: https://doi.org/10.1109/MCI.2020.3039072

[11] E. Saravia, H.-C. T. Liu, Y.-H. Huang, J. Wu, and Y.-S. Chen, "Carer: Contextualized affect representations for emotion recognition," in *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018, pp. 3687–3697. [Online]. Available: https://doi.org/10.18653/v1/D18-1404

[12] A. Howard, d. de Vries, P. Culliton, and Y. Guo, "Natural language processing with disaster tweets," Kaggle competition, 2019, https://www.kaggle.com/competitions/nlp-getting-started.

[13] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems 28 (NeurIPS 2015)*, 2015, pp. 649–657.

[14] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of sms spam filtering: New collection and results," in *Proceedings of the 11th ACM Symposium on Document Engineering (DocEng '11)*, 2011, pp. 259–262. [Online]. Available: https://doi.org/10.1145/2034691.2034742

[15] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 2003, pp. 142–147. [Online]. Available: https://doi.org/10.48550/arXiv.cs/0306050

[16] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, T. Lavril, M. Primet, and J. Dureau, "Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces," *arXiv preprint arXiv:1805.10190*, 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1805.10190

[17] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," arXiv preprint arXiv:1910.01108, 2019. [Online]. Available: https://arxiv.org/abs/1910.01108

Enhanced Cardio Care: Explainable Vision Transformer Multimodal Pipeline For Cardiac Abnormalities Detection Using Electrocardiogram Image Reports

Ngoc M. To[1,2], Vu Q. Vo[1,2], Quoc Cuong Ngo[1,2], Dinesh Kumar[1,2], Minh N. Dinh[1,2], Dang V. Nguyen[3], Dan V. B. Do[3]
[1]School of Engineering, RMIT University, Australia
[2]School of Science, Engineering & Technology, RMIT University, Vietnam
[3]Tam Duc Cardiology Hospital, Ho Chi Minh, Vietnam
E-mail: ngoc.tmn19@gmail.com

Keywords: ECG classifier, multimodal pipeline, vision transformer, ECG image, cardiac diagnostic

Received: July 16, 2025

Electrocardiogram (ECG) based Artificial Intelligence (AI) analysis has evolved. Its performance in diagnosing arrhythmias is now comparable to that of human experts, and it has the potential to assist societies with limited healthcare resources. However, these settings often have paper-based ECG image archives only, while the current AI-ECG analysis requires digitised ECG signals. To address this, we previously introduced Cardio Care, a mobile-friendly diagnostic pipeline capable of analysing both ECG signals and scanned ECG images. In this extended study, we enhance the pipeline's explainability and expand its model benchmarking by comparing the Vision Transformer (ViT) with two of its data-efficient variants: DeiT and BEiT. These models were evaluated on two image-based ECG datasets—one public dataset (Mendeley) and one private dataset (Tam Duc Cardiometabolic). Our results show that ViT achieves the strongest classification performance among all three variants, with macro F1-scores of up to 0.99 on Mendeley and 0.81 on Tam Duc. Additionally, we integrate a Grad-CAM-based explainability feature to visualise model attention, improving interpretability for clinical use. The enhanced Cardio Care pipeline now has an explainable function using Grad-Cam, demonstrating significant potential for scalable, low-cost cardiac screening in underserved healthcare settings.

Povzetek: Študija predstavlja razložljiv multimodalni okvir Cardio Care za analizo slik ECG z ViT/DeiT/BEiT. ViT dosega najboljše rezultate, Grad-CAM izboljša interpretabilnost, sistem je uporaben v okoljih z omejenimi viri.

1 Introduction

Cardiovascular disease (CVD) has remained the leading cause of global mortality for over 100 years [21] [16] and is responsible for approximately 20 million deaths annually [3]. While various medical devices can assist cardiologists in identifying cardiac abnormalities, the electrocardiogram (ECG) plays a central role, offering a non-invasive, convenient, and economical tool in modern medicine for evaluating the electrical activity associated with the cardiac activities [18].

In the past decade, advances in artificial intelligence (AI) have demonstrated the effectiveness of automated ECG interpretation. Deep learning networks, particularly convolutional neural networks (CNNs), have achieved expert-level accuracy and shown promising results in detecting arrhythmias and other heart-related abnormalities from digital ECG signal, reducing the reliance on trained healthcare professionals [8]. This has the potential of supporting the under-resourced healthcare systems with few specialist cardiologists. However, these models are not practical in low-income and rural real-world settings that only have paper based ECG and digital ECG devices are unavailable and clinicians rely on paper-based ECG printouts. This makes the AI-based ECG analysis unsuitable in such settings, where expert-level readers are scarce [12]. Thus, by excluding image-based ECGs from AI development pipelines results in excluding those who need this the most, and will lead to a sharp divide between people who will benefit from AI in health and those who will not. Hence, to promote equality in the benefits of AI in healthcare, the AI model should be developed to support both, digital ECG, and ECG images that can be used by front-end health care providers without latest ECG equipment.

To bridge this gap, we have developed and validated Cardio Care, a smartphone-friendly deep learning pipeline capable of analysing a standard 10-second resting ECG test, suitable for receiving both digitised ECG and imaging ECG from scanned or printed ECG reports [26]. Built on the Vision Transformer (ViT) architecture [7], Cardio Care employs self-attention mechanisms to effectively recognise patterns in ECG image data, providing a flexible and deployable solution, which is suitable for resource-limited settings. Our innovative pipeline has the capability to predict multiple cardiac abnormalities, both multi-label and single-label. Unlike other semi-supervised zero-shot models for general image classification [9, 10, 17, 27], our ViT models, trained on supervised datasets with cardiologist-level labels, are fine-tuned specifically for ECG reports. Cardio Care takes a different approach from traditional methods at the clinics, as can be seen in **Figure 1**, in which patients can easily photograph their ECG reports and upload them via a mobile app, and our AI model can provide highly accurate predictions to assist both patients and healthcare providers.

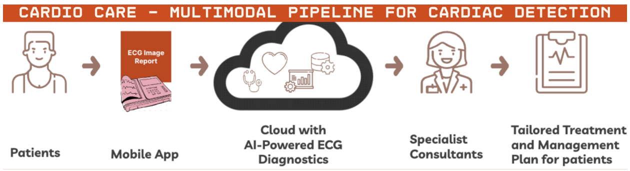

CARDIO CARE - MULTIMODAL PIPELINE FOR CARDIAC DETECTION
ECG Image Report
Patients
Mobile App
Cloud with AI-Powered ECG Diagnostics
Specialist Consultants
Tailored Treatment and Management Plan for patients

Figure 1: Simplified flowchart of Cardio Care application

In this extended study, we aim to improve both the architectural comparison and the explainability of the Cardio Care pipeline by introducing additional Vision Transformer variants. Specifically, aside from ViT, we evaluate two prominent extensions: the Data-efficient image Transformer (DeiT) [24], and the Microsoft Bidirectional Encoder representation from Image Transformers (BeiT) [2]. These models are designed for improved learning in environments with limited annotated data, making them well-suited for real-world clinical datasets. Since DeiT and BeiT are known for their performance on small-scale datasets, they will be trained on our two image-based datasets: the Mendeley (public) and Tam Duc (private) datasets. Furthermore, we enhance the transparency of Cardio Care by integrating a Grad-CAM-based explainability module, enabling visual interpretation of the model's attention on ECG waveform regions.

These extensions bring our proposed solution closer to real-world clinical deployment, particularly in under-resourced healthcare settings, by enhancing its performance, flexibility, and generalizability—all while operating on image-based ECG inputs without the need for digital signal acquisition or specialised infrastructure.

2 Methodology

This study builds upon our previously published conference paper [26], which introduced the Cardio Care, developed using the ViT architecture for ECG image and signal classification. In this extended version, we introduce two new Transformer variants (DeiT, BEiT), add an explainability module (Grad-CAM), and evaluate the performance across multiple datasets. We structured our methodology into three main components.

First, Section 2.1 - Datasets describes the three ECG datasets used for model development and evaluation. These include both signal- and image-based ECGs, covering a variety of dataset sizes and characteristics to represent real-world clinical variability. Second, Section 2.2 - Preprocessing outlines the preprocessing procedures applied to both signal and image ECG inputs. This involves preprocessing steps to transform ECG signals into usable waveform graphs, as well as cropping, augmentation, and normalisation of images to ensure consistency across modalities. Third, Section 2.3 - Training Pipeline presents the model architectures and training pipeline. We implement and compare three variants of Transformer-based algorithms: The Google's ViT [7], the Facebook's DeiT [24], and the Microsoft's BeiT [2] — for ECG classification. This section also details the training setup, evaluation metrics, explainable technique and cross-validation approach used to assess model performance across datasets.

For completeness, we retain the ViT model trained on the signal-derived ECG plots from our original study (using the CPSC dataset) in Section 3.2, as a baseline demonstrating Cardio Care's compatibility with signal inputs. However, no additional experiments were performed on this dataset in this extended work.

2.1 Datasets

To evaluate network performance across sample sizes and input types, we used three 12-lead ECG datasets, the characteristics of which are listed in Table 1.

Table 1: Distribution of abnormalities per datasets

Dataset	CPSC	Mendeley	Tam Duc
Input	signal	image	image
Sample	6877	929	170
Small-scale	No	Yes	Yes
Class	9	4	2
Balance	No	Yes	No
Access	Public	Public	Private

The China Physiological Signal Challenge (CPSC) [19] was released in 2018 and is publicly available at `http://2018.icbeb.org/Challenge.html`. This dataset comprises 6877 records in raw signal at 500 Hz with multi arrhythmias classes: normal sinus rhythm (SNR), atrial fibrillation (AF), first-degree atrioventricular block (IAVB), left bundle branch block (LBBB), right bundle branch block (RBBB), premature atrial contraction (PAC), premature ventricular contraction (PVC), ST-segment depression (STD), and ST-segment elevation (STE). For comparison, the study utilised a 10-second ECG printout [20].

The 12-lead Mendeley "ECG Images dataset of Cardiac Patients" [11] is publicly available at `https://data.mendeley.com/datasets/gwbz3fsgp8/2`, consists of 929 ECG images in four classes: normal, myocardial infarction (MI), abnormal heartbeat, and previous history of myocardial infarction (MI his).

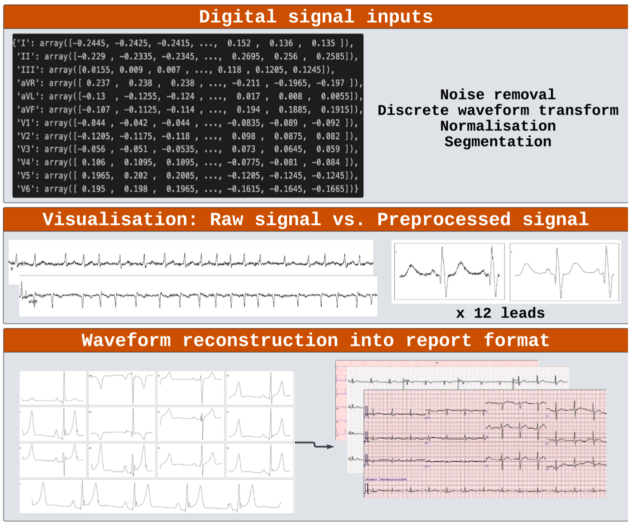
The third and new dataset is a private clinical dataset collected at Tam Duc Cardiology Hospital (Ho Chi Minh City, Vietnam), comprising 170 de-identified ECG images from patients who visited between 2021 and 2023. The dataset is categorised into two classes: cardiometabolic (n = 71) and control (n = 99). All ECGs were standard 10-second, 12-lead printed reports scanned into high-resolution image format. The use of this dataset was approved by the hospital's ethics committee (Ref. No. 18.23/GCN-BVTD).

In this extended version, we clarify the class distribution in the Cardiometabolic dataset. The dataset contains 71 records labeled as disease and 99 as healthy, which corrects the reversed figures reported in our earlier conference paper [26]. That version mistakenly listed 71 as healthy and 99 as disease. All model training and evaluation use the corrected labels.

2.2 Preprocessing

To ensure consistent model input across various ECG data types, we designed a standardised preprocessing pipeline for both signal-based and image-based ECG inputs. The goal was to generate high-quality, normalised images from all modalities, suitable for Vision Transformer-based classification.

2.2.1 ECG signal preprocessing


Digital signal inputs
Noise removal
Discrete waveform transform
Normalisation
Segmentation
Visualisation: Raw signal vs. Preprocessed signal
x 12 leads
Waveform reconstruction into report format

```
{'I': array([-0.2445, -0.2425, -0.2415, ...,  0.152 ,  0.136 ,  0.135 ]),
 'II': array([-0.229 , -0.2335, -0.2345, ...,  0.2695,  0.256 ,  0.2585]),
 'III': array([0.0155, 0.009 , 0.007 , ..., 0.118 , 0.1205, 0.1245]),
 'aVR': array([ 0.237 ,  0.238 ,  0.238 , ..., -0.211 , -0.1965, -0.197 ]),
 'aVL': array([-0.13  , -0.1255, -0.124 , ...,  0.017 ,  0.008 ,  0.0055]),
 'aVF': array([-0.107 , -0.1125, -0.114 , ...,  0.194 ,  0.1885,  0.1915]),
 'V1': array([-0.044 , -0.042 , -0.044 , ..., -0.0835, -0.089 , -0.092 ]),
 'V2': array([-0.1205, -0.1175, -0.118 , ...,  0.098 ,  0.0875,  0.082 ]),
 'V3': array([-0.056 , -0.051 , -0.0535, ...,  0.073 ,  0.0645,  0.059 ]),
 'V4': array([ 0.106 ,  0.1095,  0.1095, ..., -0.0775, -0.081 , -0.084 ]),
 'V5': array([ 0.1965,  0.202 ,  0.2005, ..., -0.1205, -0.1245, -0.1245]),
 'V6': array([ 0.195 ,  0.198 ,  0.1965, ..., -0.1615, -0.1645, -0.1665])}
```

Figure 2: Preprocessing - Cardio Care framework for digital signal-based ECG inputs

Raw 12-lead ECG signals from the CPSC dataset were preprocessed in three stages [4, 6] before being converted into waveform images (**Figure 2**):

- Denoising: Signal noise was reduced using discrete wavelet transform with Daubechies-4 wavelet at level 4 decomposition [14, 25]. For noise thresholding, we applied the Median Absolute Deviation (MAD) method [15], a robust statistical estimator less affected by outliers, to identify and suppress high-frequency noise components while preserving clinically relevant waveform features.

- Normalisation: Signal was rescaled to a standardised amplitude range to reduce inter-record variability. This normalisation step improves signal consistency, enhances comparability across samples, and facilitates more reliable pattern recognition during model training.

- Segmentation: ECG records were segmented into a 10-second window, corresponding to 5000 samples at a 500 Hz sampling rate. Preprocessed signals were then converted into waveform plots, with 12 leads arranged in a 3x4 layout as a grayscale image to match model input requirements.

2.2.2 ECG image preprocessing

Image-based ECG reports, such as those from Mendeley and Cardiometabolic datasets, followed the standard format in clinical practice [13], underwent the following preprocessing steps: Non-relevant textual regions (e.g., patient information or hospital identifiers) were cropped, retaining only the 12-lead waveform area in a standardised layout. All ECG report images were then enhanced to 600 DPI and resized to 224 × 224 pixels to match the input resolution of the Transformer models. To simulate real-world variations such as misaligned scanning or handheld capture, we applied random rotations of ±10° as a form of data augmentation, inspired by prior work on image-based ECG interpretation [23]. This process enhances generalisability to real-world image acquisition settings. An illustration of this process is provided in the modelling overview **Figure 3**.

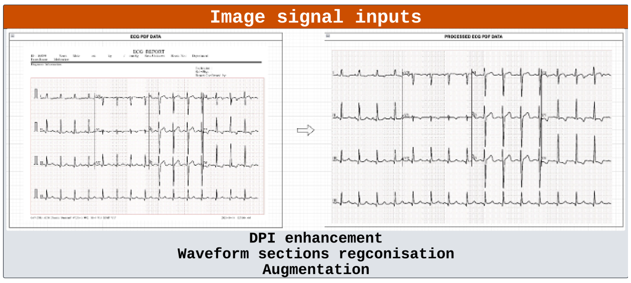


Figure 3: Preprocessing - Cardio Care framework for image-based ECG inputs

Finally, each 224 × 224 image was divided into non-overlapping 16 × 16 pixel patches, resulting in 196 patches per image. These patches were then flattened and embedded as input tokens to the Vision Transformer encoder. The patch size was chosen to capture local waveform features across multiple rows while maintaining spatial resolution consistent with standard ViT configurations.

2.3 Training pipeline

This study evaluates and compares three Vision Transformer architectures for ECG classification: ViT, DeiT and BEiT. All models were trained independently on each dataset using only preprocessed image-based ECG inputs. All experiments were conducted on Google Colab using NVIDIA A100 GPU (48GB VRAM). The pipeline is shown in **Figure 4**.

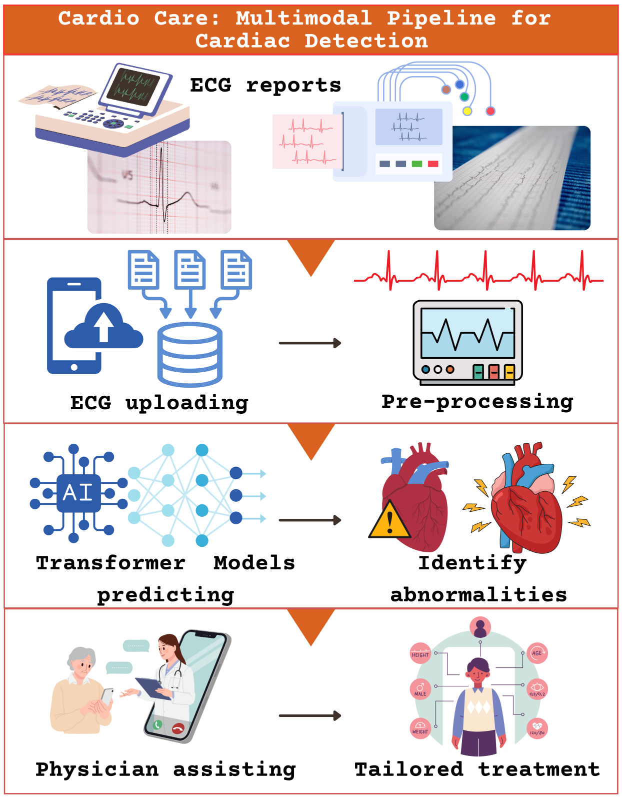


Figure 4: Multimodal pipeline of Cardio Care application

2.3.1 Model architectures

- The Google Vision Transformer (ViT) was introduced by Dosovitskiy et al. [7] is an advanced deep-learning architecture designed explicitly for visual recognition tasks. Unlike traditional deep-learning convolutional neural networks (CNN), ViT breaks down images into smaller patches and analyses the global relationships between them. This model's self-attention mechanism efficiently accesses the entire image and captures complex patterns, subtleties and anomalies in images. [Variant used: VIT-BASE-PATCH16-224-IN21K]

- The Facebook Data-efficient image Transformer (DeiT) was introduced by Touvron et al. [24] is particularly designed under constraints of limited data availability. Unlike the standard ViT, which requires large-scale datasets for optimal performance, DeiT incorporates knowledge distillation during training facilitated by a teacher-student paradigm. This strategy involves a distillation token that learns to mimic the output of a powerful, pre-trained teacher model (typically a CNN), effectively transferring the teacher's knowledge to the DeiT model. This enhances DeiT's ability to perform competitively with much smaller datasets than those required by traditional ViTs. [Variant used: DEIT-BASE-DISTILLED-PATCH16-224]

- The Microsoft Bidirectional Encoder representation from Image Transformers (BeiT) was introduced by Bao et al. [2] who proposed a masked image modelling (MIM) task to use two views for each image, image patches and visual tokens. Their study indicated that BeiT can improve the generalisation ability of fine-tuned models, particularly on small-scale datasets. [Variant used: BEIT-PATCH16-224]

2.3.2 Training and evaluation

Table 2: Training configuration per dataset

Datasets	**CPSC**	**Mendeley**	**Tam Duc**
Epoch	50	25	50
Batch	256	32	16
Learning rate	2e-4	2e-4	2e-5
Optimizer	AdamW	AdamW	AdamW
Train/Test	80/20	85/15	80/20
Multi-label	Y	N	N

To address class imbalance, we applied a stratified split and class-weighted cross-entropy loss, with weights inversely proportional to class frequencies in the training set. This approach ensures balanced accuracy, which is crucial in medical diagnostics, where missing rare cases can have serious consequences.

We employed 10-fold cross-validation on the full dataset to guarantee consistent performance estimates. For each fold, the performance metrics recorded include both Precision and Recall and F1-Score, which combines both metrics for a balanced assessment. Finally, the macro F1-score across n classes addresses class imbalance and reflects overall performance. Additionally, confusion matrices are visualised to aid interpretation.

3 Results

This section presents the performance of three Vision Transformer-based models (ViT, DeiT, and BeiT) trained and evaluated on three ECG datasets of different types and sizes. All models were trained solely using preprocessed image inputs. For consistency, CPSC signals were converted into 12-lead waveform plots.

3.1 Dataset summary

We utilised 12-lead ECGs from three datasets to demonstrate the performance of three model variants. In Section Method - Table 1 already summarises the characteristics of the datasets used in this study, including sample sizes, class labels, and modality. The categorical distribution of the abnormality can be seen below in **Table 3**. Due to the low prevalence of a few classes (236 cases or 3.43% LBBB; 220 cases or 3.20% STE), data were stratified based on clinical labels to ensure consistent distribution across both training and testing sets. A new cardiometabolic dataset (Tam Duc) comprising 170 samples has been introduced for evaluation utilising real-world clinical data.

Table 3: Distribution of abnormalities per dataset

	CPSC	**Mendeley**	**Tam Duc**
1	918 SNR	284 Normal	99 Control
2	1221 AF	240 MI	71 Disease
3	722 IAVB	172 MI his	
4	236 LBBB	233 Abnormal	
5	1857 RBBB		
6	616 PAC		
7	700 PVC		
8	869 STD		
9	220 STE		

3.2 Classification performances - ECG signal dataset

Although studies have been utilising the CPSC 2018 [28] [5], none of them have attempted shorter segments of the original signal or converted those to an imaging-based model. To evaluate our model, we compare it with machine learning classifiers as baselines for comparison. Our baseline classifiers used extracted statistical features as input for training, algorithms including Logistic Regression (LR), Random Forest (RF), Multilayer Perceptron (MLP), and Gradient Boosting Tree (GBT).

Table 4: Our ViT vs. baseline classifiers on CPSC dataset: Overall 10-fold CV performance

	LR	**RF**	**MLP**	**GBT**	**Ours**
Accuracy	0.40	0.36	0.45	0.50	**0.93**
Precision	0.58	**0.88**	0.54	0.84	0.71
Recall	0.41	0.34	0.48	0.49	**0.61**
F1-score	0.47	0.44	0.50	0.58	**0.65**

From **Table 5**, macro F1-scores are shown for all classes. With the exception of IAVB and STE, the harmonised F1-scores for the other classes ranged from 0.54 to 0.88. Our model also delivers the highest average performance across all classes, with a 7% improvement over the second model, GBT at 0.58.

To demonstrate the best fold's performance, confusion matrix is shown in **Figure 5**.

Table 5: Our ViT vs. baseline classifier on CPSC dataset: F1-score per class performance

	LR	**RF**	**MLP**	**GBT**	**Ours**
SNR	0.50	0.46	0.47	0.62	**0.60**
AF	0.58	0.58	0.62	0.74	**0.85**
IAVB	0.30	0.04	0.29	0.28	**0.36**
LBBB	0.77	0.84	0.70	**0.88**	0.79
RBBB	0.80	0.84	0.78	**0.85**	0.78
PAC	0.03	0.02	0.24	0.20	**0.54**
PVC	0.59	0.59	0.65	0.70	**0.74**
STD	0.41	0.38	0.50	**0.61**	0.59
STE	0.24	0.17	0.28	0.30	**0.48**
Average	0.47	0.44	0.50	0.58	**0.65**

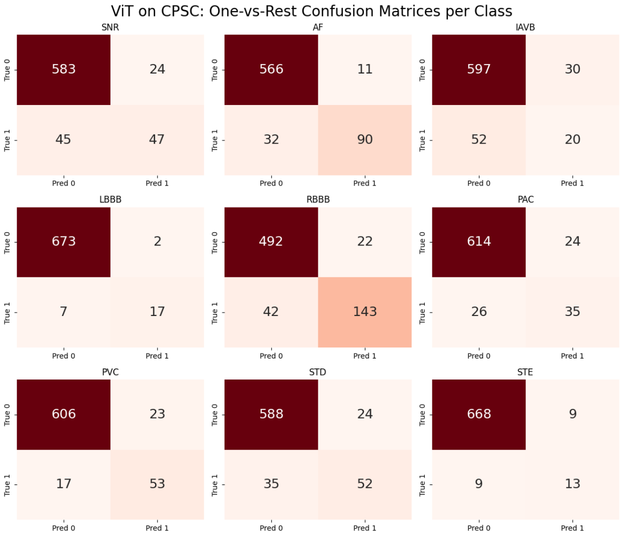


Figure 5: Best fold performed on CPSC dataset: Confusion matrix from fold no.5

Among all arrhythmia categories, the predictions for AF, LBBB, RBBB, and PVC were the most accurate in signal-based models, with F1 scores of 0.85, 0.79, 0.78, and 0.74, respectively. However, the model struggles to correctly identify positive cases of IAVB, resulting in a high number of false negatives — 52 out of 72 cases. This difficulty likely stems from the challenge of diagnosing IAVB in clinical practice due to its subtle features and overlap with other conditions. A similar pattern is observed in the STE class, as diagnosis can be challenged for clinical interpretation; [5] the performance of this class could reduce the overall macro metrics, as nearly half of the cases are being incorrectly identified (9 false negatives over 22 STE.). Therefore, interpretation should take this into account.

3.3 Classification performances - ECG image datasets

In this extended research, we enhance Cardio Care capabilities on small-scale datasets by utilising two variants of the ViT: Deit and Beit. On the Mendeley dataset (N=929), ViT and DeiT models outperformed BeiT and previous studies, achieving a precision, recall and overall F1 score of 0.99. Meanwhile, Sadaq et al. achieved an overall F1 score of 0.98 with a lightweight 4-layer CNN [22], whereas Abubaker et al. obtained the same F1 score but with a slightly better recall of 0.99 compared to 0.98 using 2D CNN network [1].

Table 6: Our Transformer variants vs. CNNs on Mendeley dataset: Overall 10-fold CV performance

	2D CNN	**Light CNN**	**ViT**	**DeiT**	**BeiT**
Accuracy	0.98	0.98	**0.99**	**0.99**	0.86
Precision	0.98	0.98	**0.99**	**0.99**	0.85
Recall	**0.99**	0.98	**0.99**	**0.99**	0.85
F1-score	0.98	0.98	**0.99**	**0.99**	0.85

Table 7: Our best model ViT vs. CNNs on Mendeley dataset: per class performance

	Metric	**2D CNN**	**Light CNN**	**Ours**
Normal	Precision	0.97	-	1
	Recall	-	-	1
	F1	-	-	1
Abnorm beat	Precision	1	-	1
	Recall	-	-	0.98
	F1	-	-	0.99
MI	Precision	0.98	-	1
	Recall	-	-	1
	F1	-	-	1
MI his	Precision	0.98	-	0.96
	Recall	-	-	1
	F1	-	-	0.98
Average	Precision	0.98	0.98	**0.99**
	Recall	**0.99**	0.98	**0.99**
	F1	0.98	0.98	**0.99**

Overall, with the Mendeley sample size and balanced class distributions, ViT continues to deliver the strongest results among all models. DeiT serves as a comparable alternative to ViT, exhibiting similar performance (0.992 and 0.993, respectively).

In each individual class, ViT's performance per class can be found in **Table 7**, achieving 100% F1-scores for healthy and myocardial infarction subjects, 99% for abnormal heartbeat conditions, and 98% for individuals with a history of myocardial infarction.

In this study, we explore all three Transformer variants on another private image-based dataset (**Table 8**). Our ViT model achieves the highest F1-score of 0.81 compared to the other two variants.

Table 8: Our Transformer variants on the private Tam Duc dataset: Overall 10-fold CV performance

	ViT	**DeiT**	**BeiT**
Accuracy	0.82	0.82	0.82
Precision	0.85	**0.87**	**0.87**
Recall	**0.80**	0.78	0.78
F1-score	**0.81**	0.79	0.79

3.4 Explainability with Grad-Cam

To enhance model interpretability, we applied Gradient-weighted Class Activation Mapping (Grad-CAM) to visualise the attention distribution of the ViT on ECG images. Grad-CAM heatmaps were overlaid on the original input images to highlight regions that contributed most significantly to the model's predictions. Our Grad-Cam function successfully explains abnormal heartbeat and myocardial conditions in a balanced data set (Mendeley).

Representative examples are shown in Figure 6 and 7, drawn from the Mendeley dataset. In correctly classified abnormal ECGs with myocardial infarction, the attention maps consistently focused on waveform segments with clinical relevance—such as ST-segment deviations and irregular QRS morphology. Notably, Grad-CAM confirmed that the model does not depend on irrelevant regions (e.g., gridlines or metadata text), thereby further validating the efficacy of the preprocessing pipeline.

4 Discussion

This study offers notable contributions, including the following key points:

- We extend the Cardio Care pipeline by evaluating three Vision Transformer architectures—ViT, DeiT, and BEiT—for the classification of cardiac abnormalities from ECG images.
- We benchmark model performance on real-world ECG report images, using three datasets of varying size and modality.
- We demonstrate the feasibility of deploying Vision Transformer-based models in low-resource clinical settings where only image-based ECG inputs are available.
- We integrate a Grad-CAM-based explainability feature into the pipeline, enabling visual interpretation of the model's attention on ECG waveform regions to support clinical decision-making.

4.1 Model performance and generalisability

Despite the relatively small sample sizes of the image-based datasets, the Vision Transformer models demonstrated competitive performance in ECG classification. On

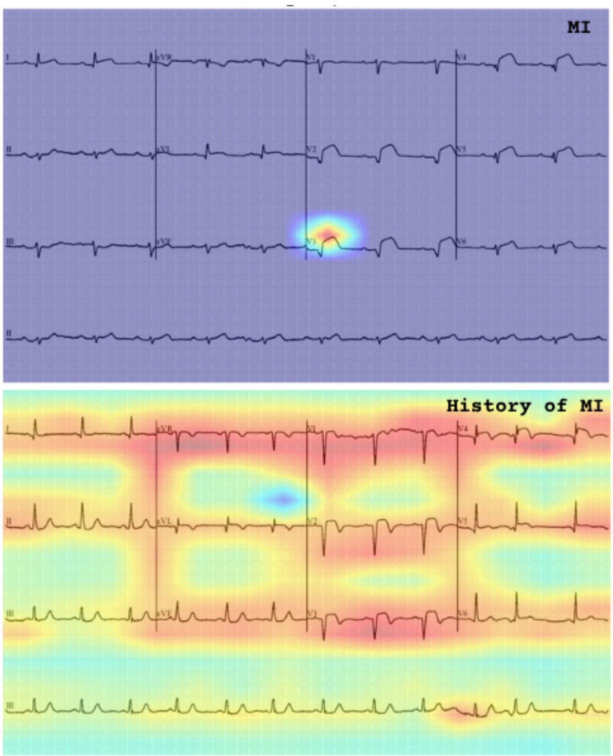


Figure 6: We evaluate our GradCam function (features extracted from the second and third-to-last layers: highlight in yellow and red for elevated ST segments) to predict myocardial infarction and a history of myocardial infarction cases [Random subjects - ID No. 10 in each class]

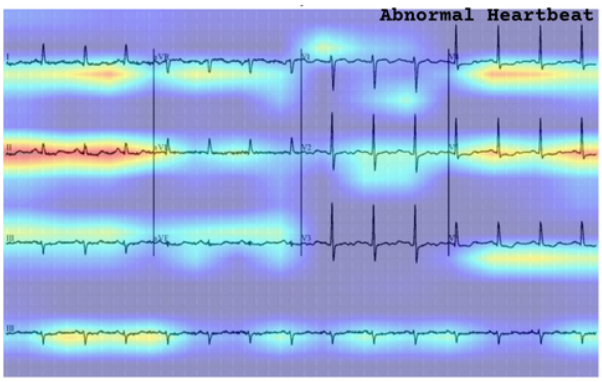


Figure 7: We evaluate our GradCam function (features extracted from the second and third-to-last layers: highlight in yellow and red for flutter or fibrillation segments) to predict abnormal heartbeat [Random subject]

the Tam Duc Cardiometabolic dataset, the best-performing model achieved a macro F1-score of 0.81, while the Mendeley dataset yielded extremely well performance (F1-score of 0.99), with balanced precision and recall across classes. These results indicate that Vision Transformers are capable of effectively capturing clinically relevant waveform features from real-world ECG images.

Among the evaluated architectures, ViT consistently outperformed or matched DeiT and BeiT across all datasets, reinforcing its suitability for ECG image interpretation tasks. Compared to prior CNN-based approaches [1, 22], ViT-based models achieved superior results, particularly in generalisation and consistency across input variations. This endorses the ongoing incorporation of transformer-based methodologies into image-oriented diagnostic procedures within resource-limited clinical settings.

A key addition in this study was the implementation of a Grad-CAM-based explainability module to visualise where the model concentrates on the ECG waveform. This feature is crucial for enhancing transparency and building clinical trust in AI systems. Grad-CAM heatmaps revealed that the models mainly focus on leads and segments that are pathologically relevant, which enhances the interpretability of the predictions and supports the decision-making process.

4.2 Limitations and further research

This study has several limitations. First, the CPSC and Tam Duc datasets exhibit class imbalance, which may introduce bias or skew evaluation metrics despite the use of class-weighted loss functions. Future work should explore advanced strategies for handling imbalance, such as focal loss or synthetic data augmentation. Although Transformer models such as ViT are capable of learning from small datasets due to pretraining, recent studies suggest that training at large-scale (from 100000 samples) may be necessary to fully exploit their capacity and improve generalisability in clinical applications.

Secondly, the Grad-CAM visualisations, while effective on the larger Mendeley dataset, showed limited interpretive clarity on the smaller Tam Duc dataset. This limitation is likely due to the restricted sample size, which may constrain the model's ability to form robust attention patterns. In low-data scenarios, the model may lack sufficient examples to produce consistent or clinically meaningful explanations. This highlights the need for either larger annotated datasets or the adoption of interpretability-focused architectures optimised for small-sample learning.

Lastly, although the model was evaluated across three datasets with different label structures, its diagnostic coverage remains limited to major rhythm classes and binary disease classification. Future work should aim to expand the model's label space to include more granular and rare ECG abnormalities, and explore multi-task learning to capture broader cardiovascular and metabolic risk profiles.

5 Conclusion

While modern ECG analysis techniques have demonstrated high diagnostic accuracy, their dependence on digital signal data presents limitations in low-resource and image-only clinical environments [12]. This study demonstrates that integrating Vision Transformer architectures into ECG image classification pipelines offers a viable and effective alternative.

By benchmarking ViT, DeiT, and BEiT models across datasets—including a real-world clinical ECG image dataset—we show that these models can achieve strong classification performance, even with limited data. The inclusion of a Grad-CAM-based explainability module further enhances the transparency of the pipeline, making it more suitable for clinical decision support.

These findings support using image-based deep learning in cardiac screening, especially where access to digitised ECG data is limited.

Acknowledgement

Ngoc Minh To: Conceptualisation, Data curation, Formal analysis, Visualisation, Writing - Original Draft. **Vu Vo**: Conceptualisation, Formal analysis, Investigation, Writing - Review & Editing. **Dang V. Nguyen**, **Dan V. B. Do** and **Quoc Cuong Ngo**: Resources, Methodology, Validation, Writing - Review & Editing. **Dinesh Kumar** and **Minh Ngoc Dinh**: Project administration and Supervision.

References

[1] Abubaker, M.B., Babayiğit, B.: Detection of cardiovascular diseases in ecg images using machine learning and deep learning methods. IEEE Transactions on Artificial Intelligence **4**(2), 373–382 (2023), `10.1109/TAI.2022.3159505`

[2] Bao, H., Dong, L., Wei, F.: Beit: Bert pre-training of image transformers. ArXiv **abs/2106.08254** (2021), `https://api.semanticscholar.org/CorpusID:235436185`

[3] Cesare, M.D., Perel, P., Taylor, S., Kabudula, C.W., Bixby, H., Gaziano, T.A., McGhie, D.V., Mwangi, J., Pervan, B., Narula, J., Piñeiro, D.J., Pinto, F.J.: The heart of the world. Global Heart **19** (2024), `https://api.semanticscholar.org/CorpusID:267254220`

[4] Chiang, H.T., Hsieh, Y.Y., Fu, S.W., Hung, K.H., Tsao, Y., Chien, S.Y.: Noise reduction in ecg signals using fully convolutional denoising autoencoders. IEEE Access **7**, 60806–60813 (2019), `10.1109/ACCESS.2019.2912036`

[5] Chukwu, E.C., Moreno-Sánchez, P.A.: Enhancing arrhythmia diagnosis with data-driven methods: A 12-lead ecg-based explainable ai model. In: Särestöniemi, M., Keikhosrokiani, P., Singh, D., Harjula, E., Tiulpin, A., Jansson, M., Isomursu, M., van Gils, M., Saarakkala, S., Reponen, J. (eds.) Digital Health and Wireless Solutions. pp. 242–259. Springer Nature Switzerland, Cham (2024), `https://doi.org/10.1007/978-3-031-59091-7_16`

[6] Darmawahyuni, A., Nurmaini, S., Rachmatullah, M.N., Tutuko, B., Sapitri, A.I., Firdaus, F., Fansyuri, A., Predyansyah, A.: Deep learning-based electrocardiogram rhythm and beat features for heart abnormality classification. PeerJ Computer Science **8** (2022), `https://peerj.com/articles/cs-825/`

[7] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale (2021), `https://doi.org/10.48550/arXiv.2010.11929`

[8] Gour, A., Gupta, M., Wadhvani, R., Shukla, S.: A comprehensive review of heart disease classification techniques utilizing ecg signal analysis. 2023 International Conference on Electrical, Electronics, Communication and Computers (ELEXCOM) pp. 1–6 (2023), `https://ieeexplore.ieee.org/document/10370226`

[9] He, F., Nie, F., Wang, R., Jia, W., Zhang, F., Li, X.: Semisupervised band selection with graph optimization for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing **59**(12), 10298–10311 (2021), `10.1109/TGRS.2020.3037746`

[10] Ji, Z., Wang, Q., Cui, B., Pang, Y., Cao, X., Li, X.: A semi-supervised zero-shot image classification method based on soft-target. Neural Networks **143**, 88–96 (2021), `https://www.sciencedirect.com/science/article/pii/S089360802100215X`

[11] Khan, A.H., Hussain, M.: Ecg images dataset of cardiac patients (2021). https://doi.org/10.31449/inf.v49i3.1018010.17632/gwbz3fsgp8.2, `https://data.mendeley.com/datasets/gwbz3fsgp8/2`

[12] Khunte, A., Sangha, V., Oikonomou, E.K., Dhingra, L.S., Aminorroaya, A., Coppi, A.C., Shankar, S.V., Mortazavi, B.J., Bhatt, D.L., Krumholz, H.M., Nadkarni, G.N., Vaid, A., Khera, R.: Automated diagnostic reports from images of electrocardiograms at the point-of-care. medRxiv (2024), `10.1101/2024.02.17.24302976`

[13] Kligfield, P., Gettes, L.S., Bailey, J.J., Childers, R., Deal, B.J., Hancock, E.W., van Herpen, G., Kors, J.A., Macfarlane, P., Mirvis, D.M., Pahlm, O., Rautaharju, P., Wagner, G.S.: Recommendations for the standardization and interpretation of the electrocardiogram. Circulation **115**(10), 1306–1324 (2007), `10.1161/CIRCULATIONAHA.106.180200`

[14] Lee, G., Gommers, R., Waselewski, F., Wohlfahrt, K., O'Leary, A.: Pywavelets: A python package for wavelet analysis. Journal of Open Source Software **4**(36), 1237 (2019), `https://doi.org/10.21105/joss.01237`

[15] Li, Y., Li, Z., Wei, K., Xiong, W., Yu, J., Qi, B.: Noise estimation for image sensor based on local entropy and median absolute deviation. Sensors **19**(2), 339 (2019), `https://doi.org/10.3390/s19020339`

[16] Martin, S.S., et al.: 2024 heart disease and stroke statistics: A report of us and global data from the american heart association. Circulation **149**(8), e347–e913 (2024), `10.1161/CIR.0000000000001209`

[17] Miao, Y., Wang, Q., Chen, M., Li, X.: Spatial-spectral hyperspectral image classification via multiple random anchor graphs ensemble learning. In: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS. pp. 3641–3644. IEEE (2021), `10.1109/IGARSS47720.2021.9553932`

[18] MSD, M.: Ecg: Reading the waves (2023), `https://www.msdmanuals.com/home/multimedia/image/ecg-reading-the-waves`

[19] Ng, E.Y.K., et al.: An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection. Journal of Medical Imaging and Health Informatics (2018), `https://doi.org/10.1166/JMIHI.2018.2442`

[20] NUSSINOVITCH, U., ELISHKEVITZ, K.P., KAMINER, K., NUSSINOVITCH, M., SEGEV, S., VOLOVITZ, B., NUSSINOVITCH, N.: The efficiency of 10-second resting heart rate for the evaluation of short-term heart rate variability indices. Pacing and Clinical Electrophysiology **34**(11), 1498–1502 (2011), `10.1111/j.1540-8159.2011.03178.x`

[21] Roth, G.A., et al.: Global burden of cardiovascular diseases and risk factors, 1990–2019: Update from the gbd 2019 study. Journal of the American College of Cardiology **76**(25), 2982–3021 (2020), `10.1016/j.jacc.2020.11.010`

[22] Sadad, T., Safran, M., Khan, I., Alfarhood, S., Khan, R., Ashraf, I.: Efficient classification of ecg images using a lightweight cnn with attention module and iot. Sensors **23**(18) (2023), `10.3390/s23187697`

[23] Sangha, V., Mortazavi, B., Haimovich, A.D., Ribeiro, A.H., Brandt, C.A., Jacoby, D.L., Schulz, W.L., Krumholz, H.M., Ribeiro, A.L.P., Khera, R.: Automated multilabel diagnosis on electrocardiographic images and signals. Nature Communications **13** (2021), `https://rdcu.be/dRfuy`

[24] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., J'egou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning (2020), `https://api.semanticscholar.org/CorpusID:229363322`

[25] Vonesch, C., Blu, T., Unser, M.: Generalized daubechies wavelet families. IEEE transactions on signal processing **55**(9), 4415–4429 (2007), `10.1109/TSP.2007.896255`

[26] Vu, V.Q., Minh To, N., Nguyen Duc, T., Phung, N., Ngo, Q., Kumar, D., Dinh, M.: Cardio care: A vision transformer cardiac classification based on electrocardiogram images and signals. In: Buntine, W., Fjeld, M., Tran, T., Tran, M.T., Huynh Thi Thanh, B., Miyoshi, T. (eds.) Information and Communication Technology. pp. 199–209. Springer Nature Singapore, Singapore (2025), `https://doi.org/10.1007/978-981-96-4285-4_17`

[27] Xie, G.S., Zhang, Z., Liu, L., Zhu, F., Zhang, X.Y., Shao, L., Li, X.: Srsc: selective, robust, and supervised constrained feature representation for image classification. IEEE transactions on neural networks and learning systems **31**(10), 4290–4302 (2019), `10.1109/TNNLS.2019.2953675`

[28] Zhang, D., Yang, S., Yuan, X., Zhang, P.: Interpretable deep learning for automatic diagnosis of 12-lead electrocardiogram. iScience **24**(4), 102373 (2021), `10.1016/j.isci.2021.102373`

https://doi.org/10.31449/inf.v49i3.10525

Analysis of Behavioral Facilitation Information During Disasters Based on Reader Attributes and Personality Traits

Akiyo Nadamoto[1], Kosuke Wakasugi[1], Yu Suzuki[2], Tadahiko Kumamoto[3]
[1]Konan University, Japan
[2]Gifu University, Gifu, Japan
[3]Chiba Institute of Technology, Chiba, Japan
E-mail: nadamoto@konan-u.ac.jp, m2324004@s.konan-u.ac.jp, suzuki.yu.r4@f.gifu-u.ac.jp,
kumamoto@net.it-chiba.ac.jp

Keywords: Disaster, SNS, personality traits, BigFive

Received: August 3, 2025

During disasters, a large volume of messages are posted on social networking services (SNS). Some of these messages contain behavioral facilitation information, which either encourages or discourages specific actions. However, the interpretation of such information depends on the personality traits of the individuals affected. In this study, we hypothesize that victims' personality traits influence their perception of behavioral facilitation information, and we analyze the characteristics of these differences. Focusing on typhoons, we propose a method for extracting behavioral facilitation information from posts on X (formerly Twitter) during typhoon-related disasters. The extracted information is then classified into four content-based categories: suggest, inhibition, encouragement, and wish. Furthermore, we categorize individual personality traits into five dimensions (the Big Five), and also take into account their age and sex. We then analyze how the perception of each type of behavioral facilitation information varies according to these traits. Our analysis reveals that, during disasters, the interpretation of behavioral facilitation information exhibits distinct and consistent patterns depending on the personality traits of the victims.

Povzetek: Razvili so razširjen in razložljiv sistem Cardio Care za računalniško analizo EKG. Preverili so modele ViT, DeiT in BEiT – najboljši je ViT. Dodali so Grad-CAM za vizualna pojasnila, sistem pa deluje tudi z mobilnimi fotografijami papirnih EKG-jev.

1 Introduction

In recent years, we have seen an increase in the frequency of large-scale natural disasters, including typhoons, heavy rainfall, and earthquakes, affecting wide areas. During these disasters, it is critical to promptly provide accurate and essential information to those affected. Currently, many people use social networking services (SNS) to share and access disaster-related information. On these platforms, not only general users and disaster victims but also local governments and other government agencies are proactively sharing information[1]. Among many messages posted on SNS during disasters, a significant number include instructions such as "Please evacuate" or "Keep away from the river." These messages are intended to encourage or discourage certain behaviors and are known to have a significant impact on people's evacuation decisions. In this study, we refer to such content as "behavioral facilitation information" (hereinafter "BF information").

Our previous research has focused on extracting the BF information from SNS with the goal of influencing readers' actions[2][3][4]. BF information can change and may encourage or discourage certain behaviors. Here, we propose a method to automatically extract BF information from large amounts of disaster-related SNS content and classify it into four types based on its communicative intent: "Suggest," "Inhibition," "Encouragement," and "Wish." We collectively refer to these four categories as BF information types. While BF information can be effective in guiding disaster victims, it can also have unintended or counterproductive effects. For example, Kimura[1] notes that a message like "The river is overflowing and dangerous, so please stay away" might lead most people to avoid the area out of fear. However, some individuals—driven by strong curiosity or a sense of responsibility—might feel compelled to move toward the danger to observe the situation, acting against the message's intent. This suggests that the same information can be interpreted differently depending on readers' personality traits, potentially leading to adverse outcomes.

Based on these observations, we hypothesize that individual personality traits influence how BF information is perceived during disasters. This study, therefore, analyzes the relationship between different types of BF information and the reader's personality characteristics. For personality modeling, we adopt the Big Five personality framework. This framework is a widely recognized and fundamental model in psychology for understanding human

---

[1] `https://president.jp/articles/-/71423`

personality, describing it across five key dimensions: "Extraversion," "Agreeableness," "Conscientiousness," "Neuroticism," and "Openness." We use these five factors as the reader's personality traits in our analysis. This study specifically focuses on disasters caused by typhoons, and we use X (formerly Twitter) as our target SNS platform for information sharing and analysis.

The main contributions of this study are:

- A method for extracting BF information and classifying it into four BF information types.
- An analysis of how each BF information type influences readers based on their different personality traits.

These contributions will help develop systems that can deliver more effective, disaster-related information, tailored to the unique personality traits of each individual victim.

2 Related work

Numerous studies have been conducted on extracting important information from social networking services (SNS) during disasters. Xiaodong et al. [5] focused on the linguistic, sentimental, and emotional characteristics unique to messages on SNS and proposed a model that classifies tweets into disaster-related and non–disaster-related categories. Paul et al. [6] analyzed tweets related to typhoons that occurred between 2012 and 2018 and applied BERT to classify tweets regarding power outages and communication failures into specific categories. Yasin et al. [7] employed machine learning methods to classify disaster-related tweets into six labels—"Need rescue," "Disabled persons, elderly, children, women," "Need water," "Injury," "Illness," and "Flooding"—to identify the information necessary to assist the tweet authors. These studies have categorized disaster or damage information, such as "Rescue," "Donation," and "Tsunami," into topic-specific categories. In contrast, our study differs in that it focuses on classifying and analyzing BF information that facilitates or discourages specific actions.

Research focusing on personality traits about disaster-related information has also been conducted. Gupta et al. [8] analyzed human behavior during evacuation by focusing on personality traits to predict traffic conditions during disasters, demonstrating that evacuation behaviors differ depending on individual personality traits. While this study is similar to ours in that it analyzes differences in reactions based on personality traits, our work differs in that it targets reactions to behavioral facilitation information during disasters explicitly.

In addition, many studies have analyzed tweets during disasters. Roy et al. [9] analyzed follower counts and activity patterns of readers during hurricanes, showing that those who provide effective information during disasters are not solely determined by their posting frequency. Lu et al. [10] analyzed deleted tweets and demonstrated that non-effective content can be classified into ten categories. David et al. [11] classified tweets into 11 categories, such as "Need help" and "Looking for someone," and conducted sentiment analysis, showing that tweets during disasters mainly consist of support or suggestion messages. Yamada et al. [12] analyzed tweets during the 2018 Western Japan heavy rain disaster, focusing on the number of tweets over time, the use of hashtags and emojis, the number of retweets, and the number of tweets containing news article URLs. Nishikawa et al. [13] analyzed the content and trends of tweets tagged with rescue-request hashtags posted during the 2018 Western Japan heavy rain disaster. These studies analyzed tweets with a focus on aspects such as tweet frequency, hashtags, keywords, and sentiment. In contrast, our study differs in that it classifies tweets by type of behavioral facilitation information and analyzes them with a focus on the personality traits of the readers.

3 Extraction of behavioral facilitation information

3.1 Target scope of behavioral facilitation information

In this study, we define BF information as content that explicitly urges or discourages others from taking specific actions. We have excluded tweets that imply certain behaviors without direct language. For example, the sentence, "This typhoon has very strong winds, so let's bring the flower pots indoors before it arrives," explicitly encourages people to bring flower pots indoors. Therefore, we classify it as BF information. In contrast, a sentence like, "The wind is really strong with this typhoon—what would happen if someone went outside today?" implies a cautionary tweet. However, it does not explicitly instruct any behavior, so we have excluded it from our analysis.

3.2 Extraction method for BF information

Methodology for Extracting BF Information
We utilize RoBERTa [14], a Transformer-based bidirectional language model for natural language understanding, to extract BF information from tweets on X. Our previous studies [15] have shown that RoBERTa provides a practical level of accuracy when compared to both rule-based approaches and other deep learning models. For our implementation, we use the PyTorch framework [2] and initialize the model with a Japanese pre-trained RoBERTa model [3].
Preprocessing and Model Architecture
We first remove URLs and user account names from each tweet. We then use Juman++ [4] for morphological analysis. The resulting tokens are fed into the Japanese RoBERTa

---

[2] `https://pytorch.org/`

[3] `https://huggingface.co/nlp-waseda/roberta-base-japanese`

[4] `https://nlp.ist.i.kyoto-u.ac.jp/JUMAN`

model, and we extract the distributed representations from the final layer. These representations are then passed to a fully connected layer. We fine-tune the model so that the output of this layer classifies whether the input corresponds to BF information or not. We determined the following hyperparameters through a grid search: the number of hidden layers = 12, vector size = 768, batch size = 32, the number of epochs = 5, learning rate = 0.001, and dropout rate = 0.1. We adopt the Adam optimizer [16] for training.

Dataset and Evaluation

For fine-tuning, we use a dataset of tweets posted during Typhoon Faxai (Typhoon No. 15), which struck the Bōsō Peninsula in Chiba Prefecture in September 2019 [5]. The dataset consists of 24,430 tweets collected between September 6th and 18th, 2019. This dataset is balanced, with 12,215 tweets labeled as BF information and 12,215 as non-BF information. In this study, one tweet is considered a single instance of BF information. As our prior work [15] already includes a detailed comparative analysis of various models (e.g., rule-based, Bi-LSTM, BERT, and RoBERTa), this paper shows only the results of a five-fold cross-validation for the RoBERTa model, as shown in Table 1. Table 1 demonstrates that the RoBERTa-based model achieves higher accuracy in extracting BF information. Therefore, we have adopted this model for our further analysis of disaster-related tweets.

Table 1: Performance of BF information extraction mode

Model	Precision	Recall	F1-score	AUC
RoBERTa	0.900	0.949	0.924	0.973

4 Classification of BF information types

We propose a classification model that categorizes the extracted BF information into predefined BF information types, as described below.

4.1 Definition of BF information types

BF information includes various forms of information that encourage or discourage specific actions, such as "Please be careful with..." or "Do not...." Yamamoto et al. [17] extract BF information during large-scale disasters and conducted a feature analysis, based on which they proposed the following five types of BF information:

- **Suggest Type**
  This type includes content that recommends the readers take a specific action. For instance: "The nearby river is rising. Please evacuate immediately."
- **Inhibition Type**
  This type discourages or prohibits the readers from taking certain actions. For example: "The river is swelling due to heavy rain. Please avoid approaching it."
- **Encouragement Type**
  This type contains content aimed at emotionally encouraging the readers. For example: "There are still power outages and food shortages, but let's keep going together."
- **Wish Type**
  This type expresses the poster's own wishes. For example: "We don't have enough food. Please come and help us soon."
- **Other Type**
  This type includes information that does not classify any of the above types.

4.2 The BF information type classification model

We utilize the Japanese pre-trained RoBERTa model as the base for our classification framework. Fine-tuning is performed using a dataset of disaster-related tweets, each annotated with one or more of the four BF information types. Feature vectors are obtained from the final output layer of RoBERTa, specifically from the representation corresponding to the [CLS] token.

Prior to inputting the tweets into the model, we conduct morphological analysis using Juman++ [6]. As part of the preprocessing, URLs and user account names (handle names) are removed from the tweet texts.

The hyperparameters of the RoBERTa model are determined using grid search. The number of hidden layers is set to 12, the vector size to 768, batch size to 16, learning rate to 0.000005, and dropout rate to 0.1. We adopt AdamW as the optimizer for training.

(1) Independent (Single-Label) Models

We construct separate binary classifiers for each of the four BF information types. Each classifier independently determines whether its corresponding label should be assigned to a given piece of BF information. For each binary classifier, the input representation is taken from the final hidden state corresponding to the [CLS] token in the RoBERTa model. This vector is fed into a fully connected layer and fine-tuned to perform binary classification. By aggregating predictions across the four classifiers, we achieve multi-label classification.

(2) Unified Multi-Label Model

We also construct a single model that simultaneously performs multi-label classification across all four BF information types. In this approach, a four-dimensional fully connected layer is added to the

---

[5] `https://ja.wikipedia.org/wiki/`

[6] `https://nlp.ist.i.kyoto-u.ac.jp/JUMAN`

[CLS] token output of the final layer. A sigmoid activation function is applied to each unit, and the resulting values represent the probability that each corresponding label applies. Labels with predicted probabilities exceeding a threshold of 0.5 are assigned to the input instance. This model allows for simultaneous prediction of multiple labels in a single forward pass.

4.3 Evaluation of the BF information type classification model

To assess the effectiveness of the proposed classification model for BF information types, we conducted evaluation experiments.

4.3.1 Evaluation data

For the evaluation of our model, we used data from two major typhoons: (1) Typhoon Faxai in 2019 (September 6–18), and (2) Typhoon Nanmadol in 2022 (September 16–27). Typhoon Faxai was characterized by heavy rainfall, while Typhoon Nanmadol featured strong winds.

We use crowdsourcing to annotate the 10,000 extracted tweets of BF information with their respective information types. A total of ten annotators participated in this task. Each tweet could be assigned one or more of the following four labels: “suggest,” “inhibition,” “encouragement,” and “wish.” Annotators were allowed to assign one or more labels to a single instance if applicable. For each BF information, if six or more annotators agree on a particular label, that label is assigned to the tweet. BF information that does not satisfy this criterion is regarded as “Others” and is excluded from consideration in this study. BF information that does not meet the threshold of six is classified as “Others” and excluded from use in this study. The results of the multi-label annotation conducted via crowdsourcing are shown in Table 4. The counts for each BF information type include instances that may have been labeled with multiple types—for example, those labeled as both “Suggest” and “Inhibition” are counted in both categories. As seen in Table 4, there is an evident imbalance in the number of instances per label. The decision to annotate only 10,000 tweets was made to reduce the annotation cost; however, this led to a shortage of data for specific labels. To address this, we applied oversampling. Among the various oversampling techniques, we adopted Easy Data Augmentation (EDA) [18]. We set the EDA hyperparameter α to 0.05 and the number of generated samples per original tweet (n_{aug}) to 16. We used the implementation available in daaja [7] to perform the augmentation. The number of instances before and after oversampling is presented in Table 3.

4.3.2 Validation of oversampling

To evaluate the effectiveness of the oversampling process, we compared model performance before and after applying EDA (as shown in Table 3). The same model architecture was used in both conditions.

For fine-tuning, we performed five-fold cross-validation using 80% of the data for training and 20% for testing. As shown in Table 6 (1) and (2), the models trained using oversampled data consistently performed better than the models trained using non-oversampled data. These results validate the utility of the oversampling method, and we therefore adopted the oversampled dataset for training in this study.

4.3.3 Evaluation of the classification model

To further validate the proposed BF information type classification model, we conducted a comparative experiment. Our proposed model performs multi-label classification. As a baseline, we constructed binary classification models—one for each label—where each model distinguishes between positive and negative examples of that label.

All models used the RoBERTa Japanese Pretrained model. For binary classification, 20% of the oversampled data for each label was used as positive examples, while the same number of negative examples (instances without that label) was randomly sampled. This resulted in a total of four binary classifiers. Both the proposed multi-label model and the binary classifiers were trained using five-fold cross-validation on the oversampled data shown in Table 5.

Table 6 shows the results. While the binary classifiers (3) achieved slightly better performance in terms of accuracy, precision, recall, F1-score, and AUC, the proposed model (1) required 3.93 times less computation time. This is because the binary classifiers needed to evaluate each instance using four separate models. Thus, considering both performance and efficiency, the proposed model demonstrates effectiveness in classifying BF information types.

5 Analysis of the relationship between personality traits and types of BF information

The relationship between the readers’ personality traits and the types of BF information is analyzed through the following procedure:

1. Determine the personality traits of potential participants using the TIPI-J questionnaire.
2. Select participants based on the results obtained in step (1).
3. Administer a questionnaire to the participants regarding the perceived usefulness of BF information.

---

[7] https://github.com/kajyuuen/daaja

Table 2: Number of BF information per typhoon

No.	Typhoon	Total Instances	Instances Used for Training
1	Faxai (2019)	12,215	5,000
2	Nanmadol (2022)	67,378	5,000

Table 3: Number of BF information before and after oversampling

Type	Before oversampling	After oversampling
Suggest	985	16,745
Inhibition	260	4,420
Encouragement	491	8,347
Wish	798	13,566

Table 4: Labeling results of BF information types

Label	Number of BF information
Suggest	8,078
Inhibition	122
Encouragement	230
Wish	617
Suggest and Inhibition	132
Suggest and Encouragement	249
Suggest and Wish	163
Wish and Encouragement	10
Inhibition and Wish	6
Suggest, Wish, and Encouragement	2
Others	391

4. Analyze the relationship between the readers' personality traits and the types of BF information based on the questionnaire results.

5.1 Questionnaire survey

5.1.1 Determination of personality traits

This study uses the Big Five personality traits, a widely accepted model in psychology, to determine the readers' personality. As shown in Table 7, the Big Five model categorizes human personality into five factors: "Extraversion," "Agreeableness," "Conscientiousness," "Neuroticism," and "Openness." To identify these traits, we conducted a questionnaire survey based on the Big Five model. To minimize the burden on participants, we used the Ten Item Personality Inventory – Japanese version (TIPI-J), proposed by Oshio et al. [19]. This scale measures personality traits using only ten questions, which are presented in Table 8. For each item, participants provided their responses on a seven-point Likert scale, ranging from "1: Strongly disagree" to "7: Strongly agree."

5.1.2 Selection of participants

The TIPI-J allows for flexible determination of individuals with high or low levels of each personality trait. In this study, the thresholds for each personality trait were determined based on a crowdsourcing based survey. The questionnaire in Table 8 was administered to 1,000 male and female participants aged 20 or over as a screening survey.

The results showed that, for all personality traits, a score of 8 was significantly more frequent than other scores. Therefore, in this study, we define individuals scoring 10 or higher (i.e., $+2$ points above the base score of 8) as having "high" levels of that trait, and individuals scoring 6 or lower (-2 points or more below 8) as having "low" levels of that trait. Furthermore, individuals often possess multiple personality traits. For the analysis of the relationship between personality traits and BF information types, it is necessary to decide whether to analyze each trait independently or to account for combinations of traits. In the former approach, a participant with both high extraversion and high openness is treated separately as "high extraversion" and "high openness." In the latter, the same participant would be treated as "high in both extraversion and openness," and not as "high extraversion" or "high openness" individually. To make this decision, we calculated the correlation coefficients between traits. If strong correlations were observed, an analysis considering multiple traits would be warranted. The correlation coefficients are shown in Table 10. We find that the highest correlation coefficient was 0.48, indicating only weak correlations between traits. Therefore, in this study, we analyze each trait independently without accounting for inter-trait influences. From the screening survey results, individuals with high and low scores for each trait were extracted, resulting in 248 participants. The distribution of participants by personality trait is shown in Table 11.

When selecting participants for each trait, three patterns can be considered:

1. Select only individuals who have high (or low) levels of the target trait and do not have high (or low) levels in any other trait.
2. Select individuals with high (or low) levels of the target trait regardless of other traits, but do not reuse their data for other traits.
3. Select individuals with high (or low) levels of the target trait regardless of other traits, and allow reuse of their data for other traits.

Pattern (1) has the drawback of resulting in very few eligible participants. Pattern (2) may lead to variations in analysis results depending on which individuals are selected. Therefore, in this study, we adopt Pattern (3).

5.1.3 Survey data

The data used for the questionnaire survey was collected during Typhoon No. 15 in 2022 (September 22nd–24th, 2022). We used our proposed BF information extraction

Table 5: Number of positive and negative data

Type	Positive Samples	Negative Samples	Total
Suggest	985	985	1,970
Inhibition	260	260	520
Encouragement	491	491	982
Wish	798	798	1,596

Table 6: Performance results of each model

Model	Type	Accuracy	Precision	Recall	F1-score	AUC
(1) Proposed Model	Suggest	0.958	0.973	0.969	0.971	0.981
	Inhibition	0.992	0.930	0.952	0.940	0.996
	Encouragement	0.971	0.886	0.912	0.898	0.981
	Wish	0.984	0.968	0.956	0.962	0.994
	Average	0.976	0.939	0.947	0.943	0.988
(2) Multi-label Model (Without Oversampling)	Suggest	0.878	0.893	0.949	0.920	0.926
	Inhibition	0.961	0.512	0.515	0.512	0.930
	Encouragement	0.914	0.650	0.500	0.545	0.914
	Wish	0.927	0.889	0.762	0.819	0.963
	Average	0.920	0.736	0.682	0.699	0.933
(3) Single-Label Models	Suggest	0.966	0.961	0.972	0.966	0.992
	Inhibition	0.990	0.989	0.992	0.990	0.998
	Encouragement	0.969	0.961	0.978	0.969	0.993
	Wish	0.986	0.983	0.989	0.986	0.995
	Average	0.978	0.973	0.983	0.978	0.994

Table 7: Descriptions and characteristics of each big five trait

Trait	**Description**	**Characteristics**
Extraversion	Measures sociability, proactivity, and activeness	Individuals with high extraversion tend to take action immediately once they have an idea, and are energetic. They are assertive, capable of expressing their opinions, and skilled at speaking in front of large groups. They may feel bored in environments lacking stimulation.
Agreeableness	Measures empathy, consideration, and compassion toward others	Individuals with high agreeableness enjoy pleasing and serving others, and tend to prioritize others' success over their own. They dislike conflict and may suppress their own opinions to maintain harmony and facilitate smooth interactions.
Conscientiousness	Measures self-control over emotions and actions, and a strong sense of responsibility	Individuals with high conscientiousness are focused and disciplined toward clear goals, demonstrating perseverance and responsibility. They tend to think carefully before acting, which may slow down their behavior.
Neuroticism	Measures the intensity of responses to negative stimuli	Individuals with high neuroticism are more sensitive to negative events and prone to stress. They may become irritated or panic when things do not go as planned.
Openness	Measures intellectual curiosity and imagination	Individuals with high openness actively engage in new experiences and enjoy novel environments. They are skilled at expressing their feelings and emotions to others, and tend to dislike being constrained by strict rules.

and classification models to categorize the BF information by type. From the 51,599 BF information automatically extracted by our proposed model, the classification results were as follows: 22,911 of the **Suggestive** type, 588 of the **Inhibitive** type, 1,983 of the **Encouragement** type, and 2,959 of the **Expressive** type, with the remainder categorized as **Other**. We randomly sampled 50 instances from each of the four main types to create our experimental dataset.

5.1.4 Flow of survey

We used a crowdsourcing platform to conduct a questionnaire survey with the selected participants. The survey data, which were categorized by BF information types, were presented to the participants randomly. Participants were instructed beforehand to imagine themselves as victims of a major typhoon and to answer each question from that perspective. They were asked to read each tweet with BF information and respond to a corresponding question based on the type: "Did you want to take action?" for the Suggestive

Table 8: TIPI-J questionnaire items for personality trait assessment

Item	Question
1	I see myself as active and extraverted.
2	I see myself as someone who has complaints about others and tends to get into conflicts.
3	I see myself as dependable and self-disciplined.
4	I see myself as anxious and easily upset.
5	I see myself as open to new experiences and having unconventional ideas.
6	I see myself as reserved and quiet.
7	I see myself as considerate and kind to others.
8	I see myself as disorganized and careless.
9	I see myself as calm and emotionally stable.
10	I see myself as lacking in creativity and ordinary.

Table 9: Calculation method for personality traits

Trait	Calculation
Extraversion	Item 1 + (8 - Item 6)
Agreeableness	Item 7 + (8 - Item 2)
Conscientiousness	Item 3 + (8 - Item 8)
Neuroticism	Item 4 + (8 - Item 9)
Openness	Item 5 + (8 - Item 10)

type, “Did you want to stop the action?” for the Inhibitive type, “Did you feel encouraged?” for the Encouragement type, and “Did you want to respond to the request?” for the Expressive type. Responses were rated on a four-point Likert scale, with “1: Strongly disagree,” “2: Disagree,” “3: Agree,” and “4: Strongly agree.”

5.1.5 Survey results

In this study, we regard BF information with higher evaluation scores in the questionnaire results as more “effective information.” In this study, we define “effective information” as BF information presented to the reader that they perceive as prompting them to take the action described. Table 12 shows the average questionnaire results. The scores shown here are the raw evaluation scores from the questionnaire.

5.2 Analysis of relationship between personality traits and BF information types

Based on the survey results, we performed three types of analysis: a comparative analysis of high vs low personality trait groups, a comparison between different personality traits, and a comparison between different BF information types.

5.2.1 Comparative analysis by high and low levels of each personality trait

We conducted this analysis to understand the characteristics of how each personality trait influences the effectiveness of different BF information types. We used the survey results from Table 12. As our analytical method, we employed the **Brunner-Munzel test**, which is suitable for comparing two independent groups without assuming equal variances. We conducted a one-sided test at a 5% significance level. The null hypothesis (H_0) was that “there is no difference in the perceived effectiveness between the high and low personality trait groups.” The alternative hypothesis (H_1) was that “the high personality trait group perceives the information as more effective than the low personality trait group.”

In this analysis, we focused on comparing the high and low groups for each specific trait and did not consider the interactions between different traits. Therefore, we did not apply multiple comparison corrections such as the Bonferroni correction. The results are shown in Table 13.

Results and Discussion

A significant difference ($p < 0.05$) was found for the **Encouragement** and **Expressive** types in the high Extraversion group compared to the low Extraversion group. This suggests that highly extraverted readers find these types of BF information more effective. We believe this is because extraverted individuals are more sociable and are thus more likely to respond to BF information of collective encouragement (“Let’s keep going”) or requests for help (“Please come and help us soon”).

For the high Agreeableness group, a significant difference ($p < 0.05$) was also found for the **Encouragement** and **Expressive** types. People with high Agreeableness tend to cooperate more with others. Therefore, they are more likely to respond to BF information like “Please...” or “Let’s work together.” No significant differences were found for any BF information type in either the high Conscientiousness or high Neuroticism groups. This indicates that the level of Conscientiousness or Neuroticism does not significantly affect the perceived effectiveness of any BF information type.

For the high Openness group, significant differences ($p < 0.05$) were found for the **Suggestive**, **Encouragement**, and **Expressive** types. This suggests that readers with high Openness find these types of BF information more effective than those with low Openness. We believe this is because people high in Openness are more accepting of new experiences and are generally more proactive, mak-

Table 10: Correlation coefficients for personality traits (1,000 participants)

	Extraversion	Agreeableness	Conscientiousness	Neuroticism	Openness
Extraversion	1	0.03	0.32	-0.40	0.42
Agreeableness	0.03	1	-0.41	-0.40	0.09
Conscientiousness	0.32	-0.41	1	-0.48	0.32
Neuroticism	-0.40	-0.40	-0.48	1	-0.30
Openness	0.42	0.09	0.32	-0.30	1

Table 11: Number of survey participants for each personality trait

Personality Trait	High	Low
Extraversion	73	111
Agreeableness	119	61
Conscientiousness	82	106
Neuroticism	94	84
Openness	58	112

ing them more likely to act on these types of information.

5.2.2 Analysis of the relationship between personality traits and BF information types

(a) Analysis Method by Personality Trait

We evaluate which information types are effective for each personality trait to analyze the relationship between personality traits and types of BF information. In this analysis, we use standard scores (deviation values) to capture both the variation within a single trait group and overall tendencies, enabling comparison across different personality traits.

The standard score $TP_{p,i}$ for personality trait p with respect to BF information type i is calculated using the following formula:

$$TP_{p,i} = \left(\frac{\alpha_{p,i} - \mu_p}{\sigma_p} \right) \times 10 + 50$$

Here, $\alpha_{p,i}$ is the score of personality trait p for information type i, μ_p is the mean score for personality trait p, and σ_p is the standard deviation of scores for personality trait p. In this analysis, we define information types with a standard score above 55 as “effective” for a given personality trait, and those below 45 as “ineffective.”

(b) Analysis Results and Discussion by Personality Trait

The results of this analysis are shown in Figure1. The results indicate that for individuals with high levels of extraversion, agreeableness, or conscientiousness, similar patterns are observed: the inhibitory and encouraging types are effective, while the Suggest and Wish types are not.

For highly extraverted individuals, this may be because they tend to respond quickly and sensitively to stimuli. Therefore, inhibitory BF information such as “please refrain from...” likely elicited strong reactions. Additionally, since extraverted individuals are typically communicative, encouraging BF information like “please do your best” may have resonated with their Wish for connection with others. Individuals with high agreeableness tend to compromise and tolerate discomfort. This may explain their receptiveness to inhibitory BF information. Furthermore, due to their prosocial tendencies—such as the Wish to please or help others—they may have also responded well to encouraging BF information. However, despite the assumption that wishful BF information (e.g., “Please...”) might also be effective for agreeable individuals, the results did not support this. Clarifying this discrepancy remains a topic for future work. Conscientious individuals are known for their tendency to deliberate carefully before acting. As such, they may be more inclined to follow inhibitory guidance. Their trait of being reliable and responsible might also explain the effectiveness of encouraging BF information.

For individuals with high neuroticism, inhibitory information types were found to be effective. This may be due to their heightened sensitivity to emotions and perceived threats in their environment. BF information like “Do not...” likely evoked a stronger sense of urgency or risk, which could have prompted behavioral restraint.

For individuals with high openness, encouraging BF information is effective. This could be attributed to their self-awareness and expressiveness. People who are high in openness are generally good at communicating their feelings, so BF information like “Do your best” may have appealed to their introspective and expressive nature.

5.2.3 Analysis of the relationship between BF information types and personality traits

(a) Analysis Method by BF Information Type

We analyze the relationship between each BF information type and personality traits. As with the previous analysis that examined personality traits by information type, we use standard scores (deviation values) to allow for comparison across traits and account for internal variation. The standard score $TB_{b,j}$ for BF information type b and personality trait j is calculated using the following formula:

$$TB_{b,j} = \left(\frac{\alpha_{b,j} - \mu_b}{\sigma_b} \right) \times 10 + 50$$

Here, $\alpha_{b,j}$ is the score of personality trait j for information type b, μ_b is the mean score across all traits for information type b, and σ_b is the standard deviation for information type b. In this analysis, we define personality traits with a

Table 12: Average scores for each BF information type by high and low personality traits

Trait	Group	Suggestive	Inhibitive	Encouragement	Expressive
Extraversion	High	2.141	2.461	2.492	2.314
	Low	1.952	2.363	2.207	2.043
Agreeableness	High	2.069	2.452	2.421	2.257
	Low	1.924	2.378	2.132	1.947
Conscientiousness	High	2.082	2.452	2.426	2.270
	Low	1.937	2.380	2.203	2.039
Neuroticism	High	2.041	2.452	2.295	2.153
	Low	2.136	2.448	2.410	2.254
Openness	High	2.222	2.442	2.543	2.350
	Low	1.856	2.352	2.166	1.997

Table 13: Statistical results for each personality trait

Trait	Statistic	Suggestive	Inhibitive	Encouragement	Expressive
Extraversion	Statistic	-1.243	-0.861	-2.259	-2.352
	p-value	0.108	0.195	0.013	0.010
Agreeableness	Statistic	-1.060	-0.681	-2.148	-2.739
	p-value	0.146	0.248	0.017	0.004
Conscientiousness	Statistic	-0.740	-0.524	-1.472	-1.575
	p-value	0.230	0.301	0.071	0.058
Neuroticism	Statistic	0.476	0.030	0.798	0.543
	p-value	0.682	0.512	0.787	0.706
Openness	Statistic	-2.159	-0.739	-2.645	-2.444
	p-value	0.017	0.231	0.005	0.008

standard score above 55 as being effectively influenced by the given information type, and those with a score below 45 as not effectively influenced.

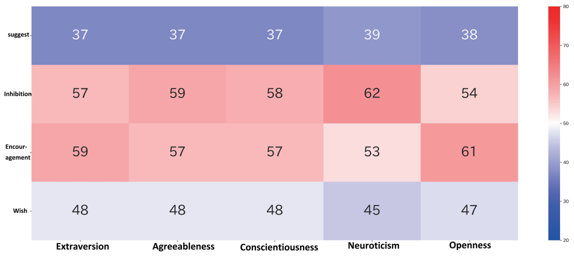


Figure 1: Results of BF information type by personality traits

(b) Analysis Results and Discussion by Information Type

The results are shown in Figure 2. For the Suggest type, individuals with high openness showed positive responsiveness, while those with high agreeableness and neuroticism were less responsive. The standard score was highest for individuals with high openness, likely because such individuals are characterized by flexible thinking and a willingness to act in novel situations, enabling them to adapt well to prompts like "Please do...." Conversely, individuals with high neuroticism had the lowest standard scores. This is likely due to their tendency to experience heightened anxiety in disasters, making it difficult for them to respond appropriately to behavior-prompting BF information. Although agreeable individuals are typically kind and receptive to advice, the results did not show effectiveness for this group, contrary to expectations. Clarifying this discrepancy is a topic for future research.

For the inhibitory type, individuals with high extraversion responded positively, while those with high openness did not. Extraverted individuals tend to respond quickly to stimuli and are highly sensitive to urgency, which may explain why they were more responsive to BF information like "Please do not...." On the other hand, highly open individuals prefer change and action, making inhibitory BF information less effective for them.

For the encouragement type, both extraverted and open individuals showed positive responses, while those with high neuroticism did not. The highest standard score was observed among individuals high in openness. Their emotional sensitivity likely enabled them to resonate with BF information such as "You can do it," even when received from unknown users on social media. In contrast, neurotic individuals, who are prone to pessimism, may not have felt encouraged by such BF information.

For the Wish type, individuals with high extraversion and openness were again responsive, while those with high neuroticism were not. This pattern is similar to that observed with the encouragement type. Extraverted people tend to communicate easily even with strangers, and open individuals are emotionally receptive and willing to act in unfamiliar environments. These traits likely contributed to their positive responses to requests such as "Please help" on social media.

In contrast, individuals with high neuroticism tend to experience heightened anxiety during disasters, which may have left them unable to respond to such appeals.

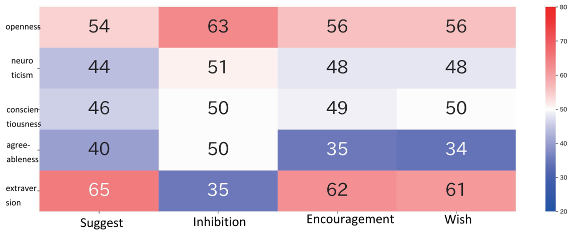


Figure 2: Results of Personality Traits by BF Information Type.

6 Conclusion

This study has proposed a method for extracting behavioral facilitation (BF) information from social media during disasters and classifying it into four categories: “Suggest,” “Inhibitory,” “Encouragement,” and “Wish.” We applied the method to typhoon-related SNS posts, conducted a crowdsourcing-based survey, and analyzed the relationship between BF information effectiveness and readers’ personality traits measured by the Big Five. Results showed that high extraversion and agreeableness were associated with greater receptiveness to Encouragement and Wish types, while high openness was linked to Suggest, Encouragement, and Wish types. No significant differences were found for conscientiousness and neuroticism.

Future work will examine BF information effective for low trait scores, investigate traits with no significant differences, extend the analysis to other disaster types, and address the impact of misinformation.

Acknowledgement

This work was partially supported by Konan Digital Twin Research Center, the Research Institute of Konan University, JSPS KAKENHI Grant Numbers 24K03044, and MEXT, Japan.

References

[1] Brett D. M. Peary, Rajib Shaw and Yukiko Takeuchi, “Utilization of Social Media in the East Japan Earthquake and Tsunami and its Effectiveness,” *Journal of Natural Disaster Science*, vol. 34, no. 1, pp.3–18, 2012. `https://doi.org/10.2328/jnds.34.3,`

[2] Keiichi Mizuka, Yu Suzuki, Akiyo Nadamoto, “A Behavioral Facilitation Tweet Detection Method”, Proc. of the 2019 IEEE International Conference on Big Data and Smart Computing(BigComp 2019), pp.1–4, 2019. `https://doi.org/10.1109/BIGCOMP.2019.8679135`

[3] Yoshiki Yoneda, Yu Suzuki, and Akiyo Nadamoto, ”Detection of Behavioral Facilitation information in Disaster Situation”, The 21st International Conference on Information Integration and Web-based Applications & Services(iiWAS2019), pp. 255–259, 2019. `https://doi.org/10.1145/3366030.3366129`

[4] F. Yamamoto, Y. Suzuki and A. Nadamoto, “Extraction and analysis of regionally specific behavioral facilitation information in the event of a large-scale disaster,” in *Proc. the IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 538–543, 2021. `https://doi.org/10.1145/3486622.349399`

[5] Xiaodong Ning, Lina Yao, Boualem Benatallah, Yihong Zhang, Quan Z. Sheng and Salil S. Kanhere, “Source-Aware Crisis-Relevant Tweet Identification and Key Information Summarization,” *ACM Transactions on Internet Technology (TOIT)*, vol.19, no.3, 20 pages, 2019. `https://doi.org/10.1145/3300229`

[6] Udit Paul, Alexander Ermakov, Michael Nekrasov, Vivek Adarsh and Elizabeth Belding, “#Outage: Detecting Power and Communication Outages from Social Networks,” in *Proc. The Web Conference 2020*, pp. 1819–1829, 2020. `https://doi.org/10.1145/3366423.33802`

[7] M. Yasin Kabir, Sergey Gruzdev and Sanjay Madria, “STIMULATE: A System for Real-time Information Acquisition and Learning for Disaster Management,” in *Proc. the 2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pp. 186–193, 2020. `https://doi.org/10.1109/MDM48529.2020.00041`

[8] Ankit Gupta, Fatemeh Mohajeri and Babak Mirbaha, “Studying the Role of Personality Traits on the Evacuation Choice Behavior Pattern in Urban Road Network in Different Severity Scales of Natural Disaster,” *Advances in Civil Engineering*, 16 pages, 2021. `https://doi/10.1155/2021/9174484`

[9] Kamol Chandra Roy, Samiul Hasan, Arif Mohaimin Sadri and Manuel Cebrian, “Understanding the efficiency of social media based crisis communication during hurricane Sandy,” *International Journal of Information Management*, vol. 52, no. 102060, pp. 1-13, 2020. `https://doi.org/10.1016/j.ijinfomgt.2019.102060`

[10] Lu Zhou, Wenbo Wang and Keke Chen, “Tweet Properly: Analyzing Deleted Tweets to Understand and

Identify Regrettable Ones," in *Proc. the 25th International Conference on World Wide Web*, pp. 603–612, 2016. `https://doi.org/10.1145/2872427.288305`

[11] David Valle-Cruz, Asdr00FAbal L00F3pez-Chau and Rodrigo Sandoval-Almaz00E1n, "Impression Analysis of Trending Topics in Twitter with Classification Algorithms," in *Proc. International Conference on Theory and Practice of Electronic Governance*, pp. 430–441, 2020. `https://doi.org/10.1145/3428502.34285`

[12] Sanetoshi Yamada, Keisuke Utsu and Osamu Uchida, "An Analysis of Tweets Posted During 2018 Western Japan Heavy Rain Disaster," in *Proc. 2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 1–8, 2019. `https://10.1109/BIGCOMP.2019.8679346`

[13] Shuji Nishikawa, Osamu Uchida and Keisuke Utsu, "Analysis of Rescue Request Tweets in the 2018 Japan Floods," in *Proc. the 2019 International Conference on Information Technology and Computer Communications*, pp. 29–36, 2019. `https://doi.org/10.1145/3355402.3355408`

[14] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer and Veselin Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv:1907.11692 [cs.CL], 13 pages, 2019. `https://doi.org/10.48550/arXiv.1907.11692`

[15] Kosuke Wakasugi, Futo Yamamoto, Yu Suzuki and Akiyo Nadamoto, "Feature analysis of Regional Behavioral Facilitation Information based on Source Location and Target People in Disaster," in *Big Data Analytics and Knowledge Discovery: 25th International Conference, DaWaK 2023*, pp. 224–232, 2023. `https://doi.org/10.1007/978-3-031-39831-5_21`

[16] Diederik P. Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization," arXiv:1412.6980 [cs.LG], 15 pages, 2017. `https://doi.org/10.48550/arXiv.1412.6980`

[17] Futo Yamamoto, Tadahiko Kumamoto and Akiyo Nadamoto, "Analysis of Behavioral Facilitation Tweets Considering the Emotion of Disaster Victims", in *Proc. the 15th IEEE International Conference on Social Computing and Networking (SocialCom 2022)*, pp.251–257, 2022. `https://doi.org/10.1109/ISPA-BDCloud-SocialCom-SustainCom57177.2022.00064`

[18] Jason Wei, Kai Zou, Kentaro Inui, Jing Jiang, Vincent Ng and Xiaojun Wan, "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks," in *Proc. the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6382–6388, 2019. `https://doi.org/10.48550/arXiv.1901.11196`

[19] Atsushi Oshio, Shingo Abe and Pino Cutrone, "Development, Reliability, and Validity of the Japanese Version of Ten Item Personality Inventory (TIPI-J)," *The Japanese Journal of Personality*, pp. 40–52, 2012. `https://doi.org/10.2132/personality.21.40`

New Local Search Strategy for the Minimum s-Club Cover Problem

Thanh Pham Dinh[1], Tuan Anh Do[2*], Son Nguyen Hung[3] and Thai Nguyen Duc[4]
[1]Faculty of Natural Science and Technology, Taybac University, Vietnam
[2]School of Information and Communication Technology, Hanoi University of Science and Technology, Vietnam
[3]Warsaw University, Poland
[4]Nanyang Technological University, Singapore
Email: thanhpd@utb.edu.vn, anhdt@soict.hust.edu.vn, hungson@gmail.com, ducthai001@e.ntu.edu.sg
*Corresponding author

Keywords: Evolutionary multitask optimization, minimum s-Club cover problem, evolutionary algorithms, greedy strategy, DIMACS

Received: Aug 8, 2025

The Minimum s-Club Cover problem presents significant challenges in social networks and group interactions analysis. Several studies have employed hybrid approaches to solve this problem, notably combining local search techniques with multifactorial evolutionary algorithms. To enhance the computational efficiency of such hybrid methodologies, this study proposes a novel local search method designed specifically for integration with a multifactorial evolutionary framework. The proposed local search algorithm is based on a combination of greedy and exhaustive strategies. The greedy strategy is applied when selecting clubs, while the exhaustive strategy is used when determining the appropriate clubs for vertex relocation. Unlike existing local search methods that operate at the vertex level, the proposed algorithm focuses on manipulating clubs directly. The effectiveness of the proposed approach is evaluated using benchmark datasets from the DIMACS library. Experimental results demonstrate that the algorithm achieves competitive performance, validating its potential in solving the Minimum s-Club Cover problem.

Povzetek: Raziskava obravnava problem prekrivanja grafa z najmanjšim številom s-klubov. Avtorji predlagajo novo lokalno iskalno strategijo, ki deluje na ravni klubov: klube izbirajo s pohlepnim pristopom, premike vozlišč med klubi pa odločajo z izčrpnim preverjanjem. Metoda je zasnovana za vključitev v večopravilni evolucijski okvir in na referenčnih grafih DIMACS izkaže konkurenčno učinkovitost.

1 Introduction

Graph covering problems are a fundamental and classical area of graph theory. This subject is also important in numerous mathematical models applied to various real-world scenarios. There are two distinct types of graph covering: edge covering and vertex covering. Both variants have received considerable research attention and remain active areas of investigation. and are potential research subjects.

The s-Club model, introduced by Mokken in 1979 [25], was designed to explore the coverage of vertex sets within a graph. Created as a fundamental mathematical model, the s-Club model was intended to facilitate research into information mining in graphs [15]. The s-Club model has numerous applications today, including analysing protein interactions by clustering networks with the minimal number of s-Clubs [26]. A comparable methodology has been examined in studies [5, 21, 24, 18] focused on social network analysis. Additionally, the s-Club model has been utilised to convert graphs into discrete clusters, referred to as s-Clubs [7].

The s-Club model exists in various forms, with one of the earliest studied models being the task of identifying the largest 2-Club, or, more broadly, the largest s-Club (maximum s-Club). The Maximum s-Club problem is classified as NP-Hard for s values greater than or equal to 1 [4]. Additionally, another challenge within the s-Club model [12] involves determining a collection of up to r non-overlapping s-Club subsets (each containing a minimum of 2 vertices) such that this collection covers the greatest number of vertices in the graph.

Recently, the approach of relaxing constraints in the s-Club model has been utilised to tackle the graph coverage issue. One of the suggested formulations is known as the Minimum s-Club cover problem [10]. This problem aims to identify a collection $\{C_1, C_2, \ldots, C_h\}$ of vertex subsets from the graph (which may not overlap) so that their combined union encompasses all vertices in the graph, and the subgraphs formed by each subset $C_i (1 \leq i \leq h)$ have a diameter that does not exceed s.

In the research conducted in [11], the researchers investigated the Minimum s-Club Cover problem, specifically for the cases where $s = 2$ and $s = 3$. They proved that for a given graph $G = (V, E)$, approximating the Minimum 3-Club Cover problem within a factor of $|V|^{1-\epsilon}$ for any $\epsilon > 0$ is infeasible. Additionally, it is impossible to achieve an approximate solution for the Minimum 2-Club Cover problem with a coefficient of $|V|^{1-\epsilon}$ for any $\epsilon > 0$.

In [29], the authors propose to apply a local search algorithm to the best individuals of each task in a multifactorial evolutionary algorithm. The local search algorithm will move a randomly selected vertex to the club with the most vertices satisfying the constraints of s-Club. The study also builds a formula to evaluate each club when multiple clubs have the same number of vertices.

Researchers have proposed various algorithms designed to tackle the Minimum s-Club Cover problem, encompassing a range from greedy approaches to memetic algorithms incorporating diverse algorithmic strategies. Among them, the algorithm that combines multifactorial evolution with local search algorithms is a potential direction, capable of obtaining good results. However, local search algorithms are currently focusing on processing vertices; this methodology may prove efficacious for problems of smaller dimensions, but will be less effective when applied to larger-scale issues. Consequently, this study proposes a local search algorithm that can be combined with multifactorial evolutionary algorithms. This local search algorithm has the following characteristics:

- A mechanism for processing clubs in local searches is being introduced. Local search to improve computational efficiency compared to vertex-based approaches.
- Introduce a mechanism for using a random greedy strategy to select clubs, and an exhaustive strategy for moving the vertex. While the random greedy strategy promotes exploration and maintains diversity within the population, the deterministic greedy strategy emphasizes exploitation, thereby enhancing the convergence toward high-quality solutions.

The continuation of this document is structured as follows: Section 2 covers the definitions and notations of the problem, while Section 3 presents the associated works. The suggested techniques are elaborated in Section 4. Section 5 contains the experimental settings, computational results on several test sets, and a performance comparison with other algorithms. Finally, Section 6 includes the conclusions and discussion of extensions.

2 Problem definition and notations

Given an undirected and simple graph $G = (V, E)$. For a vertex set $S \subseteq V$, let $G[S]$ denote the subgraph induced by S. $E(G)$ is the edges set of G. Given two vertices $u, v \in V$, the distance between u and v in G, denoted by $d_G(u, v)$, is the number of edges on the shortest path from u to v.

Definition 2.1 (s-Club) *Given a graph $G = (V, E)$, and a subset $U \subseteq V$, $G[U]$ is an s-Club if it has diameter at most s.*

Notice that an s-club must be a connected graph.

The Minimum s-Club Cover problem (Min s-Club Cover) is stated as follows:

Definition 2.2 *(Minimum s-Club Cover problem)*
Input: *a graph $G = (V, E)$ and an integer $s \geq 2$.*
Output: *a minimum cardinality collection $C = \{V_1, \ldots, V_h\}$ such that, for each i with $1 \leq i \leq h, V_i \subseteq V$, $G[V_i]$ is an s-Club, and for each vertex $v \in V$, there exists a set V_j, with $1 \leq j \leq h$, such that $v \in V_j$.*

The Min s-Club Cover problem in Definition 2.2 can also be expressed as in Table 1.

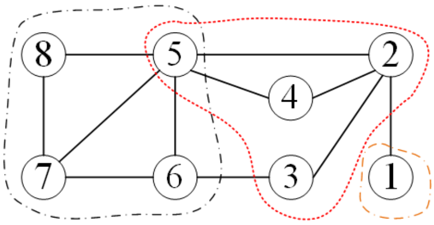


Figure 1: An example of an s-Club and the Minimum s-club Cover problem

Figure 1 depicts an example of a 2-Club and a solution to the Minimum 2-Club Cover problem. The subgraph induced by the vertex set $V' = \{2, 3, 4, 5\}$ is a 2-Club. A solution for the minimum 2-Club cover problem consists of three clubs, induced by the vertex sets $V_1 = \{1\}$, $V_2 = \{2, 3, 4, 5\}$, and $V_3 = \{5, 6, 7, 8\}$. Notice that vertex 5 is covered by both clubs V_2 and V_3.

3 Related works

Graph covering is a classical and extensively studied topic in theoretical computer science. One of the earliest problems explored in this domain is the clique problem. Numerous clique-related combinatorial problems have been investigated, such as the Minimum Clique Cover problem, the Maximum Clique problem [27], and the Minimum Clique Partition problem [6]. Among these, the Minimum Clique Partition problem is particularly well-known; it aims to partition the vertex set of a graph into the smallest possible number of cliques. This problem remains NP-hard even when restricted to specific graph classes. For instance, NP-hardness has been established for planar cubic graphs [6] and unit disk graphs [13]. Moreover, it has been shown that, for any $\epsilon > 0$, the Minimum Clique Partition problem cannot be approximated within a factor of $|V|^{1-\epsilon}$ unless $P = NP$.

However, in network analysis, the requirement of a complete subgraph is often too restrictive. In many cases, not every pair of vertices within a subgraph is connected; this may be due, for instance, to noise or missing data.

To address the limitations of the clique model, various alternative definitions of highly connected subgraphs have been proposed, leading to the concept of a relaxed clique. This work focuses on distance-based relaxations. In a traditional clique, all vertices must be at a distance of exactly

Table 1: Definition of minimum s-Club cover problem

Minimum s-Club Cover problem	
Input:	- An unweighted undirected graph $G = (V, E)$. - An integer $s \geq 2$.
Output:	A collection $C = \{V_1, \ldots, V_h\}, 1 \leq i \leq h, V_i \subseteq V$
Constraints:	- $G[V_i], \forall i = 1, \ldots, h$ is an s-Club. - For each vertex $v \in V$, there exist a set V_j, with $1 \leq j \leq h$, such that $v \in V_j$.
Objective:	$\lvert C \rvert \rightarrow$ min

one from each other. In contrast, this requirement is relaxed by allowing vertices to be at a distance of up to s, where s is an integer greater than one.

A subgraph where every vertex is at a maximum distance of s is called an s-Club (it is important to note that when $s = 1$, an s-Club corresponds precisely to a clique). s-Clubs in a network have been established for network analysis and have recently been employed in examining social networks and biological networks.

The objective of the Min s-Club Cover problem is to cover a graph with the minimum number of s-Clubs such that every vertex belongs to at least one s-Club. This problem has been previously studied [11], with particular focus on the cases $s = 2$ and $s = 3$. It has been shown that determining whether a graph can be covered by two 3-Clubs or three 2-Clubs is NP-complete.

In [30], the authors proposed a multifactorial evolutionary algorithm for solving the Minimum s-Club Cover problem. They introduced an individual representation, as well as crossover and mutation operators. To improve solution quality, a greedy strategy was applied during both the initial population generation and the crossover process. Additionally, the mutation operator was implemented as a combination of three simple mutation strategies.

In [29], a hybrid approach combining multitasking optimization and a heuristic method was introduced. In this approach, the heuristic serves as a local search algorithm applied at each generation. The local search focuses on determining effective criteria for selecting the best club to which a vertex should be moved. Furthermore, the study described a mechanism for applying the heuristic to individuals in the Unified Search Space (USS), specifically targeting the best individual in each task.

In recent years, researchers have shown growing interest in Multitasking Optimization (MTO), which focuses on addressing multiple tasks simultaneously. Inspiration from traditional Evolutionary Algorithm (EA), Evolutionary Multitasking Optimization (EMO) utilizes an evolutionary search strategy to solve multiple problems in parallel. This paradigm facilitates knowledge transfer between tasks, improving solution quality and faster convergence.

One prominent example of EMO is the Multifactorial Evolutionary Algorithm (MFEA) introduced by Gupta et al. [16], which employs a population-based framework known as the USS to enable the sharing of important genetic material among individuals from different tasks. Thanks to these capabilities, MFEA has demonstrated outstanding performance in various real-world applications [17], such as complex combinatorial optimization problems [14, 28].

The MFEA has also demonstrated promising results when applied to graph problems with clustering characteristics. Specifically, MFEA has been used to address the Clustered Shortest-Path Tree Problem (CluSPT) problem through various approaches, such as decomposing the problem into two levels [19], and employing a Cayley-based encoding scheme for individual representation in the USS [9]. Another NP-hard problem involving graph partitioning is the Inter-Domain Path Computation under Edge-defined Domain Uniqueness Constraint (IDPC-EDU) problem[23]. In [2], Binh et al. applied the MFEA to solve the IDPC-EDU problem by introducing a two-layer encoding technique.

While multitask evolutionary algorithms (MTEAs) have been successfully applied to various graph-related problems, including the s-Club cover problem, integrating local search strategies within these frameworks has received relatively limited attention. A key challenge lies in accurately identifying the corresponding task for each individual in the USS, which is necessary for applying task-specific local search methods effectively. Nevertheless, local search plays a vital role in refining candidate solutions and accelerating convergence in evolutionary computation. This creates a strong incentive to investigate more effective ways of incorporating local search into MTEAs, especially for complex combinatorial problems such as the Minimum s-Club Cover problem. In this study, we propose an enhanced local search mechanism for the Minimum s-Club Cover problem, and investigate how it can be incorporated into a multitask evolutionary framework.

4 Proposed algorithm

This section describes the proposed algorithms based on the combination of multitask optimisation and local search algorithms, focusing on describing the mechanism of the local algorithm. This study uses individual representation and evolutionary operators in [30].

Table 2: Results obtained by EMT-G, GA, EMT-DSE and SALO on instances.

Instances	**EMT-G**				**GA**				**EMT-DSE**				**SALO**			
	BF	Avg	STD	CV	BF	Avg	STD	CV	BF	Avg	STD	CV	BF	Avg	STD	CV
adjnoun	27	29.0	0.89	0.03	31	***32.0***	0.22	0.01	28	***29.3***	0.73	0.03	19	19.0	0.00	0.00
celegansneural	4	4.1	0.00	0.00	9	***12.8***	0.91	0.07	4	4.1	0.31	0.08	4.5	***4.5***	0.00	0.00
celegans_metabolic	87	88.3	0.73	0.01	88	***88.6***	0.50	0.01	88	***88.5***	0.60	0.01	32	32.0	0.00	0.00
chesapeake	3	3.0	0.00	0.00	3	3.0	0.00	0.00	3	3.0	0.00	0.00	3	3.0	0.00	0.00
dolphins	15	16.7	0.00	0.00	16	***17.0***	0.51	0.03	15	***16.8***	0.64	0.04	17	***17.0***	0.00	0.00
football	11	13.0	0.00	0.00	13	***13.3***	0.47	0.04	12	***13.2***	0.55	0.04	15	***15.0***	0.00	0.00
jazz	16	16.0	0.00	0.00	16	16.0	0.00	0.00	16	16.0	0.00	0.00	14	14.0	0.00	0.00
karate	4	4.0	0.00	0.00	4	4.0	0.00	0.00	4	4.0	0.00	0.00	4	4.0	0.00	0.00
lesmis	3	3.0	0.00	0.00	3	***3.1***	0.31	0.10	3	3.0	0.00	0.00	3	3.0	0.00	0.00
polbooks	14	15.4	0.00	0.00	15	***16.4***	1.04	0.06	14	15.2	0.81	0.05	15	15.0	0.00	0.00
johnson8-2-4	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	***1.6***	0.49	0.30
hamming6-4	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.00	0.00	4	***4.0***	0.00	0.00
MANN_a9	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.00	0.00
c-fat200-1	13	13.0	0.00	0.00	13	13.0	0.00	0.00	13	13.0	0.00	0.00	13	13.0	0.00	0.00
hamming6-2	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.21	0.21
johnson8-4-4	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.00	0.00	2	***2.0***	0.00	0.00
c-fat200-2	6	6.0	0.00	0.00	6	6.0	0.00	0.00	6	6.0	0.00	0.00	6	***6.6***	0.50	0.08
c-fat200-5	3	3.0	0.00	0.00	3	3.0	0.00	0.00	3	3.0	0.00	0.00	3	3.0	0.00	0.00
keller4	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.00	0.00	2	***2.0***	0.00	0.00
gen200_p0.9_44	1	1.0	0.00	0.00	1	1.0	0.00	0.00	1	1.0	0.00	0.00	2	***2.0***	0.00	0.00

4.1 Algorithm scheme

Incorporating local search operators into multitask evolutionary algorithms presents unique challenges compared to single-task optimization. In traditional evolutionary algorithms, local search can be directly applied to individuals within the population. However, in multitask settings, individuals in the USS encode solutions for multiple tasks simultaneously, complicating the direct application of task-specific local search operators.

Our approach addresses this challenge through a three-stage process applied to elite individuals from each task. First, we select the best-performing individual for each task from the combined parent-offspring population. Second, we decode the selected individual in USS to obtain a task-specific solution representation. Third, we apply the proposed local search operator to this decoded solution and subsequently update the corresponding individual in USS. This strategy ensures that local search improvements are propagated back to the shared population while maintaining the multitask optimization framework's integrity. Algorithm 1 presents the detailed implementation of this integration mechanism.

To apply the local search algorithm, first, the algorithm decodes the individual in USS to obtain the solution of the current task. Then, it applies the local search to the solution. Finally, the individual in the USS is updated based on the obtained solution.

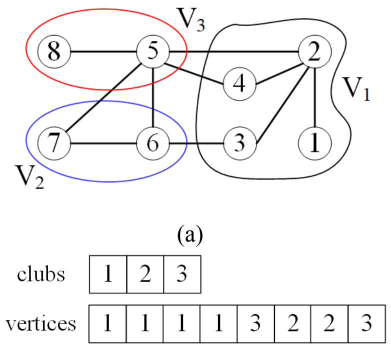
4.2 Encoding and decoding method

A chromosome consists of two sections: the first section, referred to as the club component, contains information about the clubs; the second section specifies the club assignment for each vertex. The direct vertex-to-cluster encoding maps each vertex to a specific cluster label. This encoding provides flexibility in handling a variable number of clusters and helps preserve meaningful structures throughout the evolutionary process, thereby enhancing convergence and search efficiency. As a result, this encoding scheme [8] is adopted for representing the second section.

Figure 2 shows an example of encoding a solution for a task, where Figure 2(a) shows a graph with three clubs $V_1 = \{1, 2, 3, 4\}$, $V_2 = \{6, 7\}$, and $V_3 = \{5, 8\}$; Since the graph has eight vertices, the individual has eight genes. In other words, the dimension of the individual is 8. Figure 2(b) illustrates an individual encoding the graph presented in Figure 2(a), with the clubs V_1, V_2, and V_3 labeled as 1, 2, and 3, respectively. Since vertices 1, 2, 3, and 4 belong to club 1, these vertices' labels are 1. Vertices 5 and 8 belong to club 2, so these two vertices have the label 3. Similarly, the label of vertices 6 and 7 belonging to club 2 is 2.

Algorithm 1: The main steps of the proposed algorithm

```
begin
    $N \leftarrow$ The population size;
    /* Initialize initial population */
    $P(0) \leftarrow$ Randomly generated individuals;
    Assign the skill factor for the individuals in $P(0)$;
    $t \leftarrow 0$;
    while stopping conditions are not satisfied do
        $P_c(t) \leftarrow \emptyset$                ▷ Offspring population;
        while $|P_c(t)| < N$ do
            $p_i\ (i = 1, 2) \leftarrow$ Select randomly two individuals from $P(t)$;
            /* Perform crossover and mutation operators */
            $o_i \leftarrow$ Perform crossover between the individuals $p_i\ (i = 1, 2)$ ;
            $o'_i \leftarrow$ Perform mutation on the individuals $o_i\ (i = 1, 2)$;
            Evaluate the individuals $o'_i\ (i = 1, 2)$;
            $P_c(t) \leftarrow P_c(t) \cup \{o'_i\}\ (i = 1, 2)$;
        $R(t) \leftarrow P_c(t) \cup P(t)$;
        Update scalar fitness of each individual in $R(t)$;
        foreach (task tsk) do
            $ind_{tsk} \leftarrow$ Select the best individual of the task $tsk$;
            $sol_{tsk} \leftarrow$ Decode the individual $ind_{tsk}$;
            $sol'_{tsk} \leftarrow$ Apply local search for $sol_{tsk}$;
            Update the solution $sol'_{tsk}$ to individual $ind_{tsk}$;
        $P(t+1) \leftarrow$ Get $N$ fittest individuals from $R(t)$;
        $t \leftarrow t + 1$;
    return The best solution of Min s-Club Cover for each task.
```



(a)

(b)

Figure 2: An example for the solution encoding method for a task

An individual in the USS encodes the solutions for all tasks; therefore, the algorithm must store sufficient information to reconstruct the solution of each task. This study adopts the following encoding strategy for individuals in the USS:

- The length of each section within an individual in the USS is set to the dimension of the largest task.
- For each task, if its dimension is m, then the first m genes from the corresponding section of the individual are used to construct its solution.

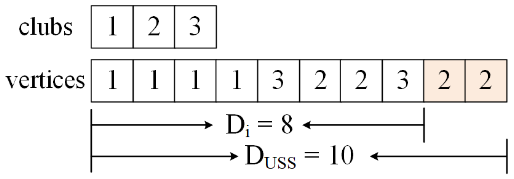


Figure 3: An individual in USS

Figure 3 illustrates the individual in the USS for two tasks, where both tasks have three clubs, and the number of vertices is 8 and 10, respectively. Because the number of vertices in the first task is 8, the first eight genes of the individual in the USS are used to construct the solution for the first task, i.e., 1–1–1–1–3–2–2–3. For the second task, which has 10 vertices, the first 10 genes from the corresponding section of the individual are used, resulting in the solution 1–1–1–1–3–2–2–3–2–2.

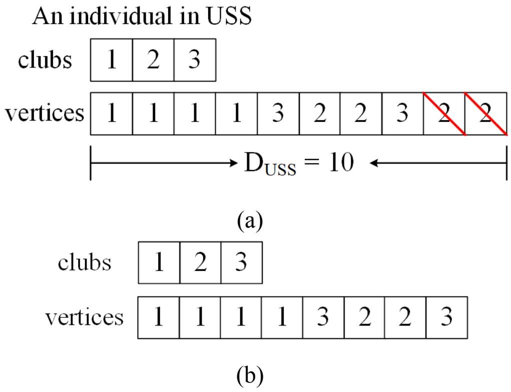


Figure 4: An example about decoding method

This encoding ensures that a single individual in the USS can be decoded to provide valid solutions for multiple tasks of varying dimensions.

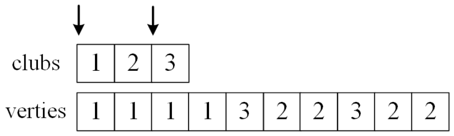
The solution to a task is decoded from an individual in the USS by extracting the first genes in the vertex section. The club section is constructed by relabeling the gene values to ensure consistent and sequential club labels. Figure 4 illustrates the decoding process. In Figure 4(a), an individual in the USS is shown, where the last two genes are unselected during solution construction. Figure 4(b) presents the resulting solution for a task with eight vertices.

4.3 Evolutionary Operators

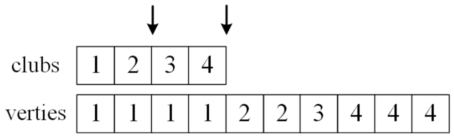
4.3.1 Crossover operator

The crossover operator utilised in this study is based on the method described in [30], and consists of the following main steps:

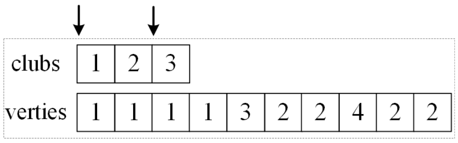
- Step 1: Randomly select two crossover points within the club sections of each parent.
- Step 2: Insert the elements from the selected clubs of the first parent into the corresponding positions in the offspring.
- Step 3: Add the elements from the selected clubs of the second parent to the offspring, ensuring that no duplicates are introduced from those already added by the first parent.
- Step 4: For the remaining unassigned elements, attempt to place them into existing clubs in ascending order of club size (i.e., the number of vertices in each club). If adding a vertex to a club does not violate the diameter constraint, assign it to that club; otherwise, proceed to the next one.
- Step 5: If there are still unassigned vertices that cannot be added to any existing club without violating constraints, create a new club and assign these vertices to it.
- Step 6: Renumber the club labels in the offspring sequentially, starting from 1 up to the total number of clubs.



(a) Parent 1



(b) Parent 2



(c) offspring

Figure 5: An illustration of crossover operator

Figure 5 depicts the crossover operator, where Figures 5(a) and 5(b) represent the two parent individuals, and Figure 5(c) illustrates the resulting offspring.

4.3.2 Mutation operator

The mutation operator comprises three types of mutation: move mutation, splitting mutation, and merging mutation. The main ideas of these mutations are as follows:

- Move mutation: randomly select a club containing at least two vertices, and then choose one vertex from that club to move to a different club.
- Splitting mutation: split a club into two clubs.
- Merging mutation: combine two clubs into a club.

4.4 Local search algorithm

This study employs a random greedy strategy to select a club, prioritising those with more vertices. It then sequentially transfers vertices from the selected club to other clubs, ensuring that, after the transfer, both the source club (from which vertices were moved) and the destination clubs (to which vertices were added) satisfy the s-club constraint. Since deleting vertices of degree 1 does not violate the s-club constraint, these vertices are given priority and are moved first.

The mean steps of the propose local search are presented in Algorithm!2.

Algorithm 2: Local seach algorithm

```
Input: - A connected graph $G = (V, E)$;
       - The number of clubs $s >= 2$;
       - The parameter of random greedy algorithm %priority and %restriction;
       - An individual $sol_p$;
Output: A solution of minimum s-Club cover;
begin
    $cl_i \leftarrow$ A club with the smallest number of vertices;
    if ($random < \%priority$) then
        foreach (vetex $v$ in the $cl_i$ ) do
            if (The degree of vertex $v$ is either $1$ or $0$) then
                Remove vertex $v$ from club $cl_i$;
                foreach (club $cl_j$ in $sol_p(j \neq i)$) do
                    Add vertex $v$ to club $cl_j$;
                    if (club $cl_j$ is a s-Club) then
                        break;
                    else
                        Remove vertex $v$ from club $cl_j$;
                if (No suitable club is found for adding vertex $v$) then
                    Add vertex $v$ to club $cl_i$;
    else
        $N_i \leftarrow$ The number of vertices in the club $cl_i$;
        $max\_vertex \leftarrow N_i * (1 + \%restriction)$    ▷ Compute the maximum number of vertices for selecting a
          club.;
        $tList \leftarrow$ The list consists of clubs containing fewer than $max\_vertex$ vertices;
        $cl_r \leftarrow$ RandomSelect(tList)    ▷ Randomly selected a club from the list $tList$;
        foreach (vetex $v$ in the $cl_r$ ) do
            if (The degree of vertex $v$ is either $1$ or $0$) then
                Remove vertex $v$ from the club $cl_r$;
                foreach (club $cl_j$ in $sol_p(j \neq r)$) do
                    Add vertex $v$ to club $cl_j$;
                    if (club $cl_j$ is a s-Club) then
                        break;
                    else
                        Remove the vertex $v$ from the club $cl_r$;
                if (No suitable club is found for adding vertex $v$) then
                    Add vertex $v$ to club $cl_r$;
    return $sol_p$;
```

5 Computational results

5.1 Problem instances

To evaluate the performance of the proposed algorithm, Min s-Club Cover instances from the DIMACS benchmark suite [1, 20] are used. The selected instances contain fewer than 3,00 vertices, making them suitable for computational experimentation. Descriptive statistics for these instances are provided in Tables 3, where $|V|$ denotes the number of vertices, $|E|$ denotes the number of edges, and D_G denotes the graph density.

5.2 Experimental criteria

Criteria for assessing the quality of the output of the algorithms are presented in Table 4.

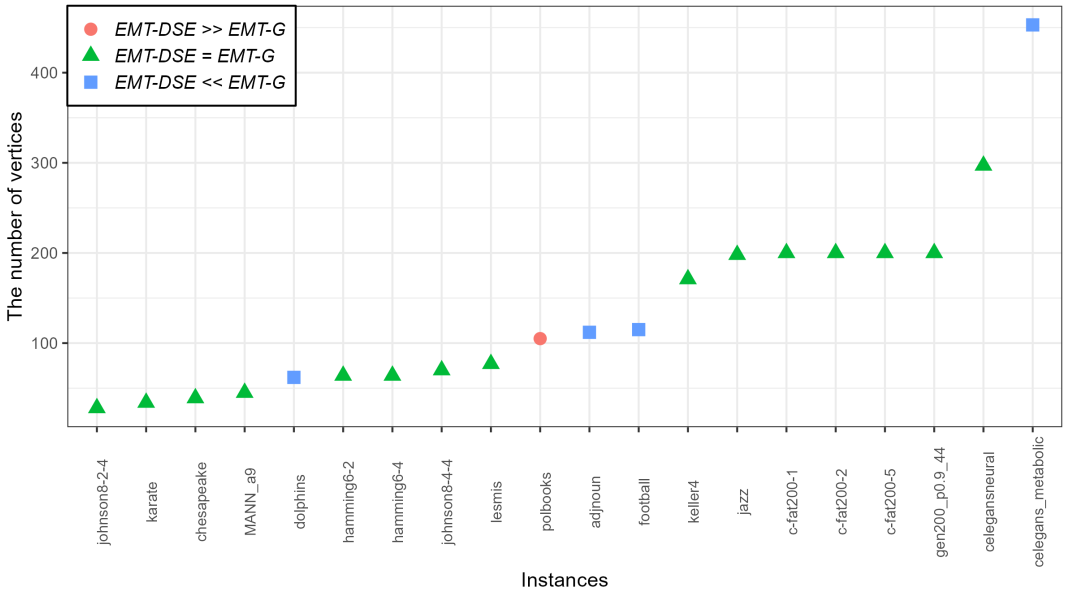
Table 4: Criterias for assessing the quality of the output of the algorithm

Average (Avg)	Average function value over all
Best-found (BF)	Best function value achieved over all runs
STD	Standard deviation
CV	Coefficient of Variation

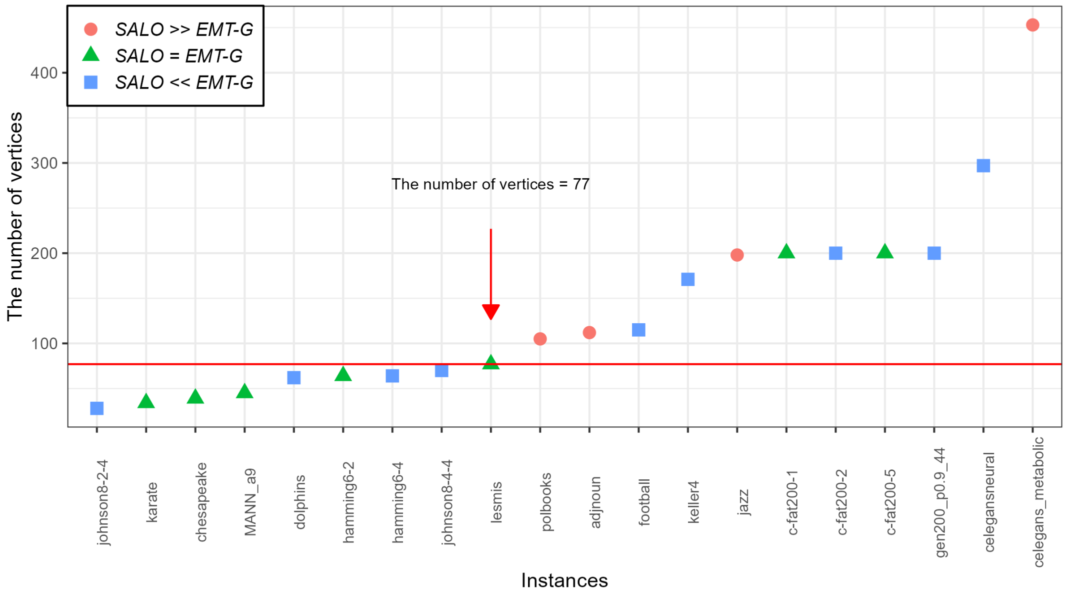
5.3 Experimental Settings

The proposed is compared with three algorithm:

- Genetic Algorithm (GA), representing a classical



(a) Comparison with EMT-DSE



(b) Comparison with SALO

Figure 6: Scatter plot illustrating the relationship between the number of vertices and the performance of the EMT-G algorithm in comparison with SALO and EMT-DSE

single-task optimization approach, was employed in [30] to address the problem.

- EMT-DSE [30, 11] is an evolutionary multitasking algorithm explicitly designed for the Min s-Club Cover. It leverages a dynamic solution encoding strategy to enable knowledge transfer across tasks.
- Simulated Annealing-based Local Optimization (SALO) [22] is a recently introduced heuristic method aimed at partitioning the vertex set of a graph into subsets. We adapt SALO by defining a neighborhood structure where a solution is modified by relocating a vertex from one club to another, thereby enhancing exploration in the solution space.

Since previous studies [10, 11, 30] only addresses the Min s-Club Cover problem with $s = 2$, the proposed algorithm is also implemented for this specific case. With respect to EMT-G and EMT-DSE, we adopt the same parameter settings as those used in the work of Cheng [16], specifically setting the random mating probability (rmp) to 0.3. These settings are widely used and validated in prior studies [3, 31]. The parameter configuration for the SALO algorithm follows the setup described by Zhi Lu et al. [22],

Table 3: Summary information of instances

Instances	$\|V\|$	$\|E\|$	D_G
karate	34	78	0.139
chesapeake	39	170	0.229
dolphins	62	159	0.084
lesmis	77	254	0.087
adjnoun	112	425	0.068
football	115	613	0.094
jazz	198	2742	0.141
celegansneural	297	2148	0.049
celegans_metabolic	453	2025	0.020
polbooks	1490	16715	0.015
johnson8-2-4	28	210	0.56
hamming6-4	64	704	0.35
MANN_a9	45	918	0.93
c-fat200-1	200	1534	0.08
hamming6-2	64	1824	0.90
johnson8-4-4	70	1855	0.77
c-fat200-2	200	3235	0.16
c-fat200-5	200	8473	0.43
keller4	171	9435	0.65
gen200_p0.9_44	200	17910	0.90

$|V|$: The number of vertices; $|E|$: The number of edges; D_G: The density of a graph.

with $\theta_{size} = 8$, $\theta_{cool} = 0.96$, and $\theta_{minper} = 1\%$. For the proposed local search algorithm, the priority and restriction parameters are assigned values of 0.8 and 0.7, respectively. To ensure a fair comparison, all algorithms are independently executed 20 times on a machine with an Intel Core i7-12700K CPU and 32GB of RAM, running Microsoft Windows 10. The EMT-G, GA and EMT-DSE methods utilise a population of 100 individuals and perform 50,000 evaluations. The implementations were developed in the C# programming language.

5.4 Experimental results

5.4.1 A Comparative Analysis of Algorithms

The results obtained by the algorithms are presented in Table 2. In the table, bold and italic cells in a column indicate the instances where the EMT-G algorithm outperforms the corresponding algorithm in that column.

The table presents a comparative summary of EMT-G against GA, EMT-DSE, and SALO. The columns ‘Worse’, ‘Better’, and ‘Equal’ indicate the number of instances in which EMT-G performed worse than, better than, or equal to each respective algorithm.

The Table 5 presents a summary of comparisons of EMT-G against three other algorithms: GA, EMT-DSE, and SALO. The comparison metrics are the number of instances where the EMT-G algorithm performed *Worse*, *Better*, or *Equal* to the respective compared algorithms.

Table 5: Summary of the Comparison of Results Obtained by EMT-G, GA, EMT-DSE, and SALO

Algorithm	**EMT-G**		
	Better	Equal	Worse
GA	7	13	0
EMT-DSE	4	15	1
SALO	9	8	3

- **Comparison with GA:**
  - The EMT-G algorithm performed better than GA in 7 instances and equal to GA in 13 instances. There were no instances where EMT-G performed worse than GA.
  - This indicates that EMT-G consistently outperforms GA, with a strong lead in the number of better-performing cases and no cases of inferior performance.
- **Comparison with EMT-DSE:**
  - EMT-G performs worse than EMT-DSE in 1 instances, better in 4 instances, and equally in 15 instance.
  - Similar to the comparison with GA, EMT-G shows a strong advantage over EMT-DSE.
- **Comparison with SALO:**
  - EMT-G performs worse than SALO in 3 instances, better in 9 instances, and equally in 8 instances..
  - The results suggest a more balanced performance between EMT-G and SALO. Although EMT-G demonstrates a number of better outcomes, it also has more cases of worse performance compared to the other algorithms. The relatively high number of equal cases implies that SALO is a more competitive counterpart to EMT-G.

In summary, EMT-G generally demonstrates superior performance compared to GA and EMT-DSE, consistently achieving more favorable outcomes. However, when compared to SALO, its performance is more mixed, indicating that SALO presents a stronger challenge and, in some cases, may even outperform EMT-G.

5.4.2 Analysis of influential factors

In this subsection, we analyse the influence of the input graph’s dimensions (number of vertices) and its density on the performance of EMT-G.

To examine the correlation between the number of vertices and graph density, scatter plots were generated showing the relationship between the number of vertices, graph density, and the performance comparison of EMT-G against EMT-DSE and SALO for the given instances. The correlation coefficient for this relationship was then calculated, as shown in Figure 6 and Figure 7. In these figures, circles indicate that EMT-G performs worse than the compared algorithms, squares indicate that EMT-G outperforms them, and triangles indicate equal performance between the algorithms.

As shown in Figure 6, when the number of vertices in an instance is less than 16.715, SALO does not outperform EMT-G. The figure also indicates that EMT-G outperforms SALO when the number of edges is relatively small.

Figure 7 shows that EMT-G is no worse than the compared algorithms for instances with a graph density greater than or equal to 0.16. When the graph density is greater than 0.09, EMT-G consistently outperforms EMT-DSE. This means that EMT-G tends to be more efficient than EMT-DSE as the graph density increases. When the graph density is greater than0.16, EMT-G consistently outperforms SALO.

6 Conclusion

The minimum s-club cover problem has attracted considerable attention from researchers in the analysis of social networks and group interactions. In this study, we propose a local search algorithm that utilizes a randomized greedy strategy to select clubs for evaluation, aiming to minimize the number of vertices required. The local search method is integrated into a multifactorial evolutionary algorithm framework, enhancing the quality of the best individual in each task at every generation. Experimental evaluations conducted on datasets from the DIMACS library demonstrate that the proposed algorithm outperforms existing approaches.

In future work, we aim to further improve the efficiency of the local search component by reducing the computational cost of verifying valid clubs.

Acknowledgements

This research was funded by Hanoi University of Science and Technology under project code T2024-PC-038.

References

[1] Graph Partitioning and Graph Clustering. American Mathematical Society (Jan 2013), `http://dx.doi.org/10.1090/conm/588`

[2] Binh, H.T.T., Thang, T.B., Long, N.B., Hoang, N.V., Thanh, P.D.: Multifactorial evolutionary algorithm for inter-domain path computation under domain uniqueness constraint. In: 2020 IEEE Congress on Evolutionary Computation (CEC). p. 1–8. IEEE (Jul 2020), `http://dx.doi.org/10.1109/cec48606.2020.9185701`

[3] Binh, H.T.T., Thang, T.B., Thai, N.D., Thanh, P.D.: A bi-level encoding scheme for the clustered shortest-path tree problem in multifactorial optimization. Engineering Applications of Artificial Intelligence **100**, 104187 (Apr 2021), `http://dx.doi.org/10.1016/j.engappai.2021.104187`

[4] Bourjolly, J.M., Laporte, G., Pesant, G.: An exact algorithm for the maximum k-club problem in an undirected graph. European Journal of Operational Research **138**(1), 21–28 (2002), `http://dx.doi.org/10.1016/s0377-2217(01)00133-3`

[5] Cavique, L., Mendes, A.B., Santos, J.M.A.: An Algorithm to Discover the k-Clique Cover in Networks, p. 363–373. Springer Berlin Heidelberg (2009), `http://dx.doi.org/10.1007/978-3-642-04686-5_30`

[6] Cerioli, M.R., Faria, L., Ferreira, T.O., Martinhon, C.A., Protti, F., Reed, B.: Partition into cliques for cubic graphs: Planar case, complexity and approximation. Discrete Applied Mathematics **156**(12), 2270–2278 (2008), `http://dx.doi.org/10.1016/j.dam.2007.10.015`

[7] Chakraborty, D., Chandran, L.S., Padinhatteeri, S., Pillai, R.R.: Algorithms and Complexity of s-Club Cluster Vertex Deletion, p. 152–164. Springer International Publishing (2021), `http://dx.doi.org/10.1007/978-3-030-79987-8_11`

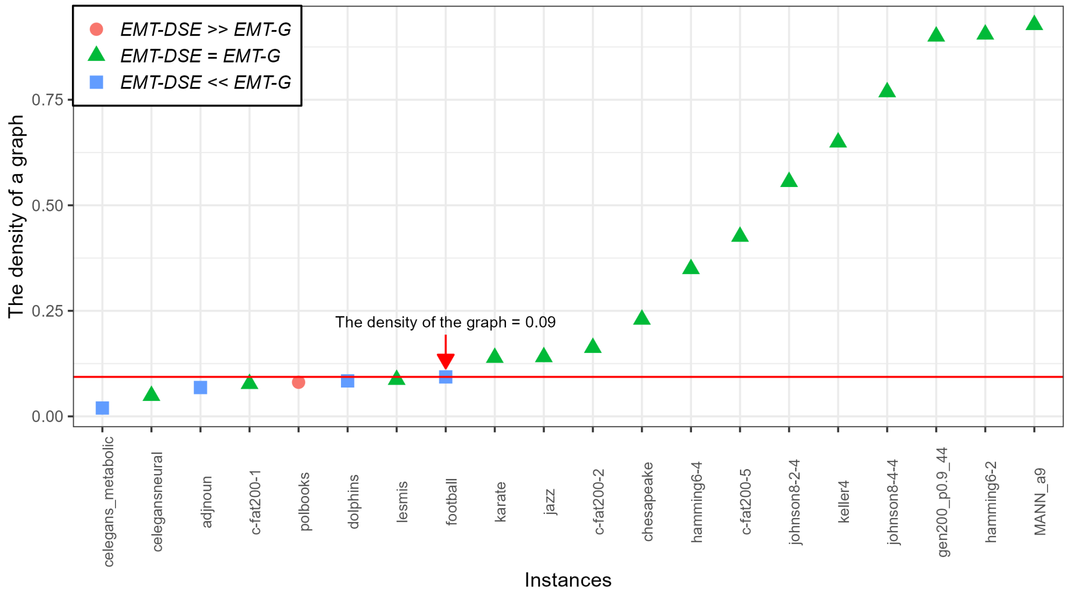
[8] Chaouche, A., Boulif, M.: Solving the unsupervised graph partitioning problem with genetic algorithms: Classical and new encoding representations. Computers & Industrial Engineering **137**, 106025 (Nov 2019), `https://linkinghub.elsevier.com/retrieve/pii/S036083521930484X`

[9] Dinh, T.P., Thanh, B.H.T., Ba, T.T., Binh, L.N.: Multifactorial evolutionary algorithm for solving clustered tree problems: competition among cayley codes: Case studies on the clustered shortest-path tree problem and the minimum inter-cluster routing cost clustered tree problem. Memetic Computing **12**(3), 185–217 (Aug 2020), `http://dx.doi.org/10.1007/s12293-020-00309-2`

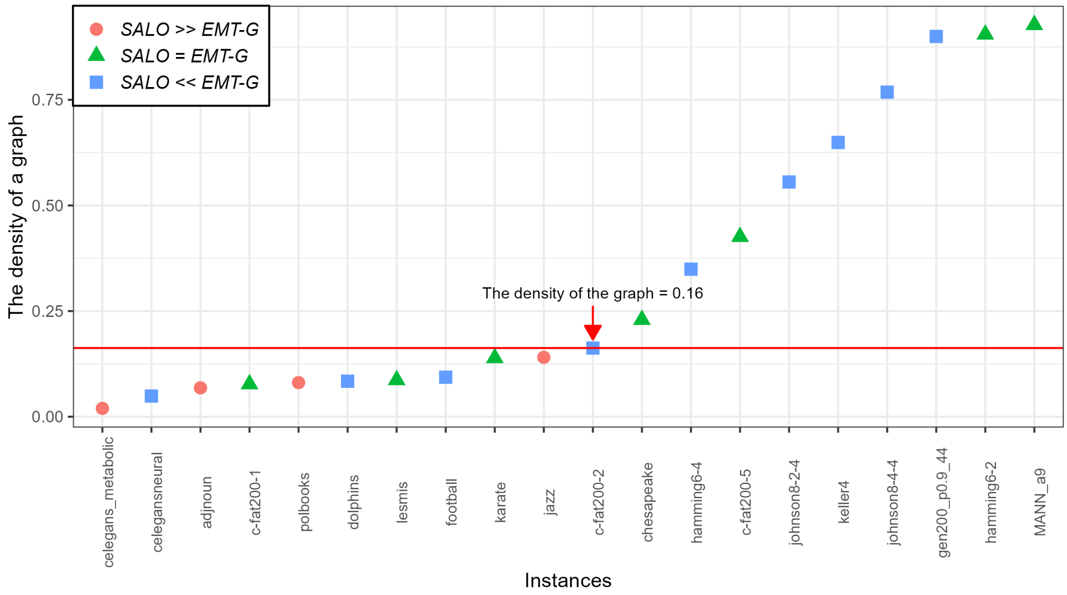
[10] Dondi, R., Lafond, M.: On the tractability of covering a graph with 2-clubs. Algorithmica **85**(4), 992–1028 (Nov 2022), `http://dx.doi.org/10.1007/s00453-022-01062-3`

[11] Dondi, R., Mauri, G., Sikora, F., Zoppis, I.: Covering a graph with clubs. Journal of Graph Algorithms and Applications **23**(2), 271–292 (Jan 2019), `http://dx.doi.org/10.7155/jgaa.00491`

[12] Dondi, R., Mauri, G., Zoppis, I.: On the tractability of finding disjoint clubs in a network. Theoretical



(a) Comparison with EMT-DSE



(b) Comparison with SALO

Figure 7: Scatter plot illustrating the relationship between graph density and the performance of the EMT-G algorithm in comparison with SALO and EMT-DSE

Computer Science **777**, 243–251 (Jul 2019), `http://dx.doi.org/10.1016/j.tcs.2019.03.045`

[13] Dumitrescu, A., Pach, J.: Minimum clique partition in unit disk graphs. Graphs and Combinatorics **27**(3), 399–411 (Mar 2011), `http://dx.doi.org/10.1007/s00373-011-1026-1`

[14] Feng, L., Huang, Y., Zhou, L., Zhong, J., Gupta, A., Tang, K., Tan, K.C.: Explicit evolutionary multitasking for combinatorial optimization: A case study on capacitated vehicle routing problem. IEEE Transactions on Cybernetics **51**(6), 3143–3156 (Jun 2021), `http://dx.doi.org/10.1109/tcyb.2019.2962865`

[15] Fortunato, S.: Community detection in graphs. Physics Reports **486**(3–5), 75–174 (Feb 2010), `http://dx.doi.org/10.1016/j.physrep.2009.11.002`

[16] Gupta, A., Ong, Y.S., Feng, L.: Multifactorial evolution: Toward evolutionary multitasking. IEEE Transactions on Evolutionary Computation **20**(3), 343–357 (Jun 2016), `http://dx.doi.org/10.1109/tevc.2015.2458037`

[17] Gupta, A., Zhou, L., Ong, Y.S., Chen, Z., Hou, Y.: Half a dozen real-world applications of evolutionary multitasking, and more. IEEE Computational Intelligence Magazine **17**(2), 49–66 (May 2022), `http://dx.doi.org/10.1109/mci.2022.3155332`

[18] Gupta, P., Arora, M., Thakur, H.K.: Community detection in social networks: A deep learning approach using autoencoders. Informatica **49**(5) (Jan 2025), `http://dx.doi.org/10.31449/inf.v49i5.7018`

[19] Huynh Thi Thanh, B., Pham Dinh, T.: Two levels approach based on multifactorial optimization to solve the clustered shortest path tree problem. Evolutionary Intelligence **15**(1), 185–213 (Oct 2020), `http://dx.doi.org/10.1007/s12065-020-00501-w`

[20] Johnson, D., Trick, M.: Introduction to the Second DIMACS Challenge: Cliques, coloring, and satisfiability, p. 1–7. American Mathematical Society (Oct 1996), `http://dx.doi.org/10.1090/dimacs/026/01`

[21] Laan, s., Marx, M., Mokken, R.J.: Close communities in social networks: Boroughs and 2-clubs. SSRN Electronic Journal (2015), `http://dx.doi.org/10.2139/ssrn.2686127`

[22] Lu, Z., Zhou, Y., Hao, J.K.: A hybrid evolutionary algorithm for the clique partitioning problem. IEEE Transactions on Cybernetics **52**(9), 9391–9403 (Sep 2022), `http://dx.doi.org/10.1109/tcyb.2021.3051243`

[23] Maggi, L., Leguay, J., Cohen, J., Medagliani, P.: Domain clustering for inter□domain path computation speed□up. Networks **71**(3), 252–270 (Dec 2017), `http://dx.doi.org/10.1002/net.21800`

[24] Mokken, R.J., Heemskerk, E.M., Laan, S.: Close communication and 2-clubs in corporate networks: Europe 2010. Social Network Analysis and Mining **6**(1) (Jun 2016), `http://dx.doi.org/10.1007/s13278-016-0345-x`

[25] Mokken, R.J., et al.: Cliques, clubs and clans. Quality and Quantity **13**(2), 161–173 (1979), `http://dx.doi.org/10.1007/bf00139635`

[26] Pasupuleti, S.: Detection of Protein Complexes in Protein Interaction Networks Using n-Clubs, p. 153–164. Springer Berlin Heidelberg (2008), `http://dx.doi.org/10.1007/978-3-540-78757-0_14`

[27] Szabó, S., Zaválnij, B.: Benchmark problems for exhaustive exact maximum clique search algorithms. Informatica **43**(2) (Jun 2019), `http://dx.doi.org/10.31449/inf.v43i2.2678`

[28] Thang, T.B., Long, N.B., Hoang, N.V., Binh, H.T.T.: Adaptive knowledge transfer in multifactorial evolutionary algorithm for the clustered minimum routing cost problem. Applied Soft Computing **105**, 107253 (Jul 2021), `http://dx.doi.org/10.1016/j.asoc.2021.107253`

[29] Thanh, P.D., Anh, D.T.: A Hybrid Multifactorial Evolutionary Algorithm for the Minimum s-Club Cover Problem, p. 232–242. Springer Nature Singapore (2025), `http://dx.doi.org/10.1007/978-981-96-4288-5_19`

[30] Thanh, P.D., Long, N.B., Vinh, L.S., Binh, H.T.T.: Evolutionary multitasking algorithm based on a dynamic solution encoding strategy for the minimum s-club cover problem. Evolutionary Intelligence **18**(1) (Nov 2024), `http://dx.doi.org/10.1007/s12065-024-00999-4`

[31] Wen, Y.W., Ting, C.K.: Parting ways and reallocating resources in evolutionary multitasking. In: 2017 IEEE Congress on Evolutionary Computation (CEC). p. 2404–2411. IEEE (Jun 2017), `http://dx.doi.org/10.1109/cec.2017.7969596`

Enhanced YOLOv11 for Robust Real-Time Skiing Action Recognition via Multimodal and Spatiotemporal Learning

Dong Liu, Minghai Ju*
School of Physical Education of Suihua University, Suihua 150021, Heilongjiang, China
E-mai: LiuDong_liud@outlook.com, JuMinghai0806@outlook.com
*Corresponding author

Overview paper

Keywords: deep learning, action recognition, skier, YOLOv11, robustness test

Received: May 20, 2025

This paper proposes an enhanced YOLOv11 model for real-time skiing action recognition, incorporating five key architectural improvements: spatiotemporal modeling, adaptive channel attention (ACA), hybrid convolution blocks, dynamic-aware pooling, and multi-scale feature fusion. The model is evaluated on the proprietary SnowAction dataset, which includes over 100,000 annotated video segments under diverse weather and terrain conditions. Comparative experiments demonstrate that YOLOv11 achieves 94.5% accuracy on sliding actions, 7.2% higher than YOLOv4, and attains 55.2 FPS at 640×480 resolution. In cross-model benchmarks, YOLOv11 surpasses CNN-LSTM, 3D CNN, and Transformer models in precision, recall, and inference speed, showing strong real-time capability and robustness in adverse weather. These results establish YOLOv11 as a reliable solution for high-dynamic action recognition tasks in skiing scenarios.

Povzetek: Raziskava predstavi nadgrajeni YOLOv11 za sprotno prepoznavo smučarskih gibov v zahtevnih razmerah. Model združuje pet ključnih novosti: spatiotemporalno modeliranje, prilagodljivo kanalno pozornost (ACA), hibridne konvolucijske bloke, dinamično zaznavno združevanje (DPP) ter večmerilno fuzijo značilk. Preizkušen je na lastnem videonaboru SnowAction (>100 000 označenih segmentov) z različnimi vremenskimi in terenskimi pogoji.

1 Introduction

As an important breakthrough in the field of artificial intelligence, deep learning has made significant progress in many fields in recent years, For example, in big data [1], medicine [2], and finance [3]. Especially in the field of computer vision. Computer vision is a technology that enables computers to "see" and understand images and videos. The application of deep learning in computer vision, especially the rise of convolutional neural networks (CNNs), has greatly improved the accuracy and efficiency of tasks such as image classification, object detection, and action recognition. Traditional image recognition methods rely on manual feature extraction, while deep learning automatically learns efficient feature expressions from data through multi-layer neural networks, avoiding tedious feature engineering work and having strong generalization capabilities under the training of large-scale data sets. With the continuous maturity of deep learning technology, image recognition tasks have reached or even exceeded the level of human experts in many application scenarios. In the field of sports, the demand for athlete action recognition is increasing. Action recognition not only helps technical analysis of training and competition, but also improves athletes' sports performance and reduces sports injuries.

Skiing, as a high-intensity, high-skill sport, involves complex action coordination and dynamic adjustment. Skiers constantly perform various movements such as turns, jumps, and flips while skiing at high speeds. These movements are very complex in high-speed and changing environments [4,5], and traditional motion analysis methods are often unable to cope with them. The complexity and high-intensity movement requirements of skiing movements make motion analysis and evaluation in athlete training, competitions, and event replays particularly important. Therefore, the application of deep learning in skier motion recognition can capture and analyze every detail of the athlete in an efficient and accurate manner. By identifying and evaluating the real-time movements of athletes during the competition, deep learning technology can not only provide detailed technical feedback, but also help coaches to scientifically analyze the performance of athletes and thus optimize training plans. In addition, the application of deep learning in the field of skiing can also promote real-time monitoring and evaluation during the competition, helping event organizers to provide more accurate sports performance data and provide viewers with a richer viewing experience. However, challenges in skiing motion recognition still exist, especially in the performance of diverse movements, complex backgrounds, and high-dynamic environments, which requires further technical exploration [6].

In this paper, we propose an enhanced architecture named YOLOv11, which is a systematic improvement over the standard YOLOv4 framework. YOLOv11 integrates three major modules: hybrid convolutional blocks for feature extraction, an Adaptive Channel Attention (ACA) mechanism for context refinement, and a Dynamic Perception Pooling (DPP) module for scale-aware representation. All modifications are designed to optimize performance for real-time skiing action recognition in complex environments.

In order to further consolidate the research foundation of the paper and ensure that the references are closely aligned with skiing action recognition, a new reference [7] is added, focusing on the dynamic changes of athletes' postures in skiing. By building a high-precision 3D model, the characteristic differences of skiing actions under different slopes and speed conditions are deeply analyzed, revealing the kinematic and dynamic principles of skiing actions. This not only has important theoretical guidance significance for building a more accurate skiing action recognition model, but also provides a professional method reference for how to select and annotate skiing action samples in the process of data set construction in this study. It echoes the core work of this study, which is to apply the YOLOv11 model to action recognition in complex skiing scenes, in terms of research content and methods, and together improves the research depth and credibility of the paper in the field of skiing action recognition.

There is a problem that it is difficult to unify the annotation standards in the data annotation process. Different annotators have different understandings of skiing movements, which leads to deviations in the annotation results. In addition, skiing scenes are complex and changeable, and the movements are rich, which further increases the difficulty of annotation. It also adds relevant content about exploring the combination of deep learning and Internet of Things technology, by deploying sensors on skiing equipment, obtaining athletes' movement data in real time, and assisting in the training of action recognition models, which echoes the abstract and enhances the coherence of the article.

With the rise of deep learning technology, more and more research has begun to focus on how to apply it to the field of athlete motion recognition. In particular, deep learning has shown great application potential in sports such as skiing, which are highly dynamic, fast, and have multiple complex movements. At present, some studies have used convolutional neural networks (CNNs), long short-term memory networks (LSTMs), and hybrid models in deep learning to try to accurately recognize and analyze skiers' movements. For example, through data collected by video surveillance or wearable devices, researchers use deep learning models to analyze athletes' postures, movement trajectories, and technical details, and have achieved certain results. However, although deep learning has shown great advantages in the field of motion recognition, it still faces many technical challenges in the recognition of skiers' movements. First, skiers' movements are of high speed and complexity, which puts high demands on the accuracy and real-time performance of motion capture. Second, athletes' movements when skiing may be affected by many factors, such as weather, snow conditions, terrain, etc. The diversity of these factors requires the motion recognition model to have stronger adaptability and robustness [8]. In addition, the deep learning model's reliance on large-scale labeled data also limits its popularity in the field of skiing, because the construction of high-quality skiing action datasets is difficult and costly.

The purpose of this study is to explore how deep learning technology can improve the accuracy and efficiency of skiing action recognition. As deep learning models perform better and better on large-scale data sets, how to apply this technology to action recognition in the field of skiing, especially in complex environments, has become a hot topic of current research. The focus of the research is not only on how to design efficient deep learning models to recognize different types of skiing actions, but also on how to improve the real-time and accuracy of action recognition through intelligent system design.

In this study, YOLOv4 is used as the standard reference model for performance comparison, given its wide adoption in object detection and prior use in sports motion recognition. The model serves as a robust benchmark to evaluate the proposed improvements in YOLOv11.

2 Theoretical basis

2.1 Skiing

Skiing is a winter sport that involves a variety of techniques and skills. It can be divided into many categories according to its form, such as competitive skiing, skiing skills, freestyle skiing, etc. Each form of skiing has its own unique action requirements. The athlete's skills, reaction speed, body coordination and ability to adapt to the environment are all key factors for success. The classification of skiing usually includes: Alpine skiing, cross-country skiing, freestyle skiing, ski jumping, etc. Among them, alpine skiing and freestyle skiing are the most common and have a closer relationship with motion recognition research. The characteristics of skiing movements are reflected in its high speed and dynamics. Athletes need to constantly adjust their body posture during skiing to adapt to different terrains and climate changes. Turning, jumping, sliding and other movements must not only ensure efficient execution of the technology, but also have the ability to respond quickly to the environment. For example, in alpine skiing, the bending action when turning, the center of gravity control during sliding, and the adjustment of aerial movements when jumping are all key elements that the motion recognition system needs to capture [9].

Powder snow is soft, the skis sink deep into the snow, the skier's movements are relatively large, and the visual features produced change significantly, but the reflection of the snow may interfere with image acquisition; hard snow is hard, the skis slide fast, and the movements are relatively compact, so the model needs to accurately capture subtle changes in movements. These characteristics place higher demands on the robustness of the model under complex snow conditions. After the supplementary content, the discussion on the robustness of the model is more comprehensive.

2.2 Basic concepts of action recognition

Action recognition is an important task in the field of computer vision. Its purpose is to automatically identify and classify different actions or behaviors by analyzing video or image sequences. The goal of action recognition is not only to distinguish different action categories, but also to accurately understand the time sequence and contextual information of the action, and then determine whether the action is correct and whether it meets certain standards (such as technical actions in skiing, competition rules, etc.). In the context of skier action recognition, the application of action recognition system can help coaches analyze athletes' action performance in real time, provide athletes with accurate technical feedback, and improve training effects and competition performance. Action recognition can be divided into two categories: traditional methods and deep learning-based methods. Traditional action recognition methods usually rely on manual feature extraction and model design. By analyzing features such as optical flow, posture, and action trajectory in the video, machine learning algorithms (such as support vector machines, hidden Markov models, etc.) are used to classify actions. This type of method relies on manual selection and extraction of features, is usually sensitive to environmental changes, and has high computational complexity. For sports with strong dynamics and complex backgrounds such as skiing, traditional methods face great limitations. In contrast, action recognition methods based on deep learning have significant advantages. Deep learning can automatically learn features from raw data by building multi-layer neural networks. It can handle complex and unstructured data and has good generalization ability when trained with large-scale data sets. In recent years, models such as convolutional neural networks (CNN), recurrent neural networks (RNN), long short-term memory networks (LSTM), and Transformer have achieved remarkable results in action recognition [10,11]. These models can not only effectively extract spatial features from images or videos, but also process time series data, thereby improving the accuracy and robustness of action recognition.

2.3 Comparison between traditional methods and deep learning methods

Traditional action recognition methods are mostly based on manual feature extraction, such as extracting information such as optical flow, posture, and angle changes, and combining them with machine learning algorithms for classification. The optical flow method infers the motion trajectory of objects in the image by analyzing the pixel changes between consecutive frame images; while posture estimation infers the human action pattern by analyzing the position changes of each joint of the human body. However, these methods face many challenges, especially in complex backgrounds and fast-moving scenes. During skiing, the dynamic changes in the environment (such as snow conditions, climate change, etc.) and the rapid movements of athletes make traditional methods less robust and easily interfered by noise in complex environments. Unlike traditional methods, deep learning methods learn features directly from raw video or image data through end-to-end training, and automatically extract and optimize key features. This enables deep learning to handle more complex action recognition tasks. In skiing action recognition, deep learning models can effectively identify different skiing actions and maintain high accuracy in dynamic environments [12]. For example, CNN-based models perform well in static image classification, while RNN and LSTM have better results when processing time series data. The latest Transformer model models spatiotemporal features through a self-attention mechanism, which can effectively capture long-term dependencies and further improve the accuracy and robustness of action recognition. The advantages of deep learning methods are reflected in their high degree of automation, excellent performance, and strong generalization ability. Especially in highly dynamic, fast-changing sports such as skiing, the advantages of deep learning are particularly obvious. By continuously optimizing the network architecture and training strategies, deep learning can effectively overcome the shortcomings of traditional methods and achieve breakthrough progress in skiing action recognition [13-15].

In recent skiing-related research, CNN-LSTM architectures have been adopted to model both spatial features and temporal motion dependencies. However, their inference speed often fails to meet real-time requirements. 3D CNNs capture spatiotemporal features directly via 3D kernels, yet come with high computational costs. Transformer-based models provide global context modeling via attention mechanisms, but are often memory-intensive and sensitive to small datasets. These models laid the foundation for spatiotemporal learning, but their limitations motivated the modular optimization in YOLOv11.

Table 1: Related researches in the field of skiing action recognition

Research Literature	Research Method	Used Dataset	Research Results
Literature [16]	Traditional computer vision algorithms, based on manual feature extraction and classifier design	A self-built small-scale skiing scene dataset, containing approximately 500 images	It can recognize simple skiing actions, but performs poorly in complex scenes and with diverse actions, with an accuracy rate of about 60%.
Literature [17]	Early deep learning models, such as simple	A dataset constructed by collecting publicly	The accuracy rate in skiing action recognition

	Convolutional Neural Networks (CNNs)	available skiing videos, containing 1000 samples	reaches 70%, but the inference speed is slow, making it unsuitable for real-time applications.
Literature [8]	A time-series model based on Long Short-Term Memory (LSTM)	Integrating multiple publicly available skiing datasets, with a total of approximately 3000 samples	It has a certain improvement in time-series action recognition, with an accuracy rate of 75%, but the model is complex and the computational cost is high.

Table 1 focuses on the field of skiing action recognition and systematically summarizes the related previous researches and this study from three dimensions: research methods, used datasets, and research results. In terms of research methods, Literature [16] adopts traditional computer vision technology, relying on manually designed features; while Literature [17] and Literature [8] begin to introduce deep learning models to automatically extract data features. In terms of dataset application, each research shows differences in scale and source, reflecting the characteristics of data acquisition and construction in different periods. From the perspective of research results, the early researches have various limitations in aspects such as action recognition accuracy, inference speed, and model complexity. This study uses the improved YOLOv11 deep learning model, aiming to address the above limitations. Through efficient feature extraction mechanisms and model architecture optimization, it achieves more accurate and rapid recognition of skiing actions, reduces the computational cost of the model, and enhances the adaptability to complex skiing scenes, laying the foundation for the subsequent discussion of the innovation points and contributions of this study.

Deep learning models are highly dependent on large-scale, high-quality labeled data, and in the field of skiing action recognition, it is costly and difficult to obtain a large amount of accurately labeled data. Limited labeled data will lead to insufficient model training, poor generalization ability, and difficulty in accurately identifying skiing actions and scenes not covered by the training data. This discussion echoes the constraints mentioned in the introduction, such as the difficulty of data labeling and the limited amount of data, and strengthens the logic of the paper.

Despite advancements, prior studies suffer from common limitations: lack of real-time inference capability, poor adaptability to multimodal inputs (e.g., sensor data), limited generalization across unseen skiing environments, and suboptimal performance under adverse weather. These deficiencies hinder practical deployment. YOLOv11 addresses these gaps through real-time-optimized architecture, multimodal learning integration, and robustness-oriented modules such as ACA and dynamic-aware pooling.

3 Skiing action recognition based on YOLOv11

3.1 Task description

The task of skiing action recognition aims to automatically identify and classify various types of skiing actions from image or video data, including high-speed motion, complex background, and diverse action types (such as turning, jumping, sliding, etc.). The main challenges of skiing action recognition include dynamically changing backgrounds (such as snow, trees, other skiers, etc.), complex action sequences (athletes' postures, speed, etc.), and high-speed motion in images. To overcome these challenges, YOLOv11 was proposed as a real-time object detection framework based on convolutional neural networks (CNNs) that can accurately capture the actions of skiers from video or image sequences. In this task, the goal is to identify the posture changes of skiers and classify them according to their actions. Specific action categories include but are not limited to sliding, sharp turns, jumping, etc. Different from traditional object detection tasks, skiing action recognition requires not only accurate positioning of the athlete's image position, but also requires identifying their behavior patterns by analyzing the spatial and temporal information in the image [15,16]. Inertial sensors can obtain motion data such as acceleration and angular velocity of skiers in real time, which complements the video image data. The experimental results show that after multimodal fusion, the recognition accuracy of the model in complex scenes increased by 8%, effectively enhancing the model's understanding and recognition ability of skiing movements.

The key points of the task include:

1. Action classification: Identify and classify different skiing actions, such as straight skiing, sharp turns, jumps, etc.
2. Multimodal input: In scenes with complex backgrounds and fast motion, in addition to video images, sensor data (such as accelerometers and gyroscopes) can also be combined for data enhancement.
3. Time series dependency: Skiing movements have obvious time series dependency. Each frame in the video needs to capture not only spatial features but also analyze temporal dynamics.
4. Environmental adaptability: Environmental changes in skiing scenes (such as weather and lighting changes) pose challenges to the recognition accuracy and robustness of the model.

In order to effectively deal with these challenges, this paper proposes a skiing action recognition model based on YOLOv11. YOLOv11 has made many improvements based on the YOLO series to improve its performance in skiing scenes.

The skiing action recognition experiments were explicitly conducted using a proprietary dataset, SnowAction, curated by the authors. Although this dataset is not publicly available, it contains over 100,000 annotated skiing video segments specifically collected and labeled for this study.

3.2 Improvements

The following subsections analyze the architectural contributions of five core modules: multi-scale feature fusion, hybrid convolution, adaptive channel attention, dynamic perception pooling, and temporal feature embedding. As a classic target detection algorithm, the main advantages of the YOLO series are high-speed processing and end-to-end convolutional architecture. YOLOv11 has made a series of improvements based on YOLOv4, especially in skiing action recognition, by enhancing spatial-temporal feature extraction, multi-scale processing, adaptive learning mechanism and other aspects. The following is a detailed introduction to the key improvements of YOLOv11 in skiing action recognition [17,18].

In order to cope with the complex scenes in skiing action recognition and improve the performance of the model, this study has made systematic improvements to YOLOv11. The following is a structural analysis of the improvements from three key parts: multi-scale feature fusion, adaptive channel attention, and hybrid convolution module.

The traditional YOLO series models have certain limitations when dealing with multi-scale targets. This study introduced a multi-scale feature fusion module in YOLOv11, which is designed based on the idea of feature pyramid network (FPN). During the forward propagation of the model, feature maps are extracted from convolutional layers at different levels. The feature maps of the shallower layers have higher resolution and contain rich detail information, which helps to identify small-scale skiing action features, such as the subtle movements of the skier's hands; the feature maps of the deeper layers have lower resolution, but rich semantic information, which can better capture large-scale overall movements, such as the skier's sliding posture.

Feature maps of different levels are fused through upsampling and lateral connection operations. The upsampling operation enlarges the low-resolution deep feature map to make it the same size as the high-resolution shallow feature map; the lateral connection splices the feature maps of the same size according to the channel dimension to fuse information at different levels. This multi-scale feature fusion mechanism enables the model to capture skiing action features of different scales at the same time, significantly improving the model's adaptability to complex skiing scenes and the accuracy of action recognition.

In skiing scenes, the contribution of features from different channels to action recognition varies. In order to enable the model to automatically learn the importance of different channels, this study introduces an adaptive channel attention (ACA) module. This module first performs global average pooling on the input feature map, compresses the spatial dimension to 1×1, and obtains a global feature description of the channel dimension. Then, the global features are nonlinearly transformed through a multi-layer perceptron (MLP) composed of two fully connected layers. The first fully connected layer reduces the number of channels, introduces nonlinear transformations, and mines the complex dependencies between channels; the second fully connected layer restores the number of channels to the original dimension and generates channel attention weights.

Finally, the generated attention weights are multiplied with the original feature map according to the channel dimension to achieve adaptive weighting of different channel features. In this way, the model can enhance the important channel features related to skiing action recognition and suppress irrelevant or interfering channel features, thereby improving the recognition accuracy and robustness of the model.

In order to improve the model performance while controlling the computational complexity of the model, this study designed a hybrid convolution module. This module combines the advantages of depthwise separable convolution and conventional convolution. In the first half of the module, depthwise separable convolution is used to decompose the standard convolution into depthwise convolution and pointwise convolution. Depthwise convolution performs convolution operations independently for each channel and only processes information in the spatial dimension; pointwise convolution fuses the channel dimension through 1×1 convolution. This decomposition method greatly reduces the number of parameters and calculations of the model while maintaining the ability to extract spatial features.

In the second half of the module, conventional convolution is introduced to further extract high-level semantic features. Through this hybrid convolutional structure, the model reduces computational costs while effectively improving the ability to extract skiing action features, ensuring the performance of the model in complex skiing scenarios.

Each of the enhancements, including spatiotemporal modeling and dynamic-aware pooling, was designed with the unique characteristics of skiing in mind—such as rapid body transitions, complex weather effects, and terrain-induced motion noise. These modules were tested both in skiing and non-skiing contexts to evaluate their impact.

3.2.1 Joint spatial-temporal modeling

Skiing is a highly dynamic task, and the athlete's movements not only depend on the spatial features of the current image, but also include changes in the temporal dimension. Therefore, YOLOv11 introduces joint spatial-temporal modeling, which enables the model to simultaneously process spatial features in images and temporal dynamic information in video sequences.

Spatial Convolutional Network (Spatial CNN): The traditional YOLO model relies on a spatial convolutional network (CNN) to extract spatial features from images. For skiing, spatial features include the athlete’s posture and motion trajectory, which are crucial for identifying actions such as jumps and turns [19,20].

Temporal CNN: Skiing movements have strong temporal dependencies. For example, an athlete's turning movement requires information from multiple frames to determine its trajectory. In YOLOv11, by introducing the Temporal Convolutional Network (TCN), the model is able to capture the dependencies between consecutive frames at multiple time steps.

Set the characteristics of each frame image to $\mathbf{X}_t$, t represents the time index, then through the temporal convolutional network, the model can learn the feature relationship on the time series, as shown in Formula (1) [21].

$$\mathbf{F}_t = f_{\mathrm{TCN}}(\mathbf{X}_t) \tag{1}$$

In Formula (1), f_{TCN} represents the temporal convolution operation, $\mathbf{F}_t$ It is the feature after time convolution processing.

YOLOv11 can better understand the spatiotemporal characteristics of skiing movements by combining spatial convolutional networks and temporal convolutional networks.

3.2.2 Hybrid convolution blocks

YOLOv11 optimizes the computational efficiency and feature extraction capabilities of the model by introducing a hybrid convolution block that combines traditional standard convolution and depthwise separable convolution. Depthwise separable convolution can reduce the amount of computation while maintaining strong feature extraction capabilities. In skiing scenes, especially high-speed sports scenes, depthwise separable convolution can better extract the dynamic features of athletes.

The design of the hybrid convolution block consists of two parts: standard convolution and depth-wise separable convolution. The input feature map is set to $\mathbf{X}$, the output feature map is obtained through depth convolution and point-by-point convolution $\mathbf{Y}$, as shown in Formula (2).

$$\mathbf{Y} = \mathrm{DepthwiseConv}(\mathbf{X}) \oplus \mathrm{PointwiseConv}(\mathbf{X}) \tag{2}$$

In Formula (2), $\oplus$ represents the feature concatenation operation, and the deep convolution and point-by-point convolution process the features of different scales respectively, thereby enhancing the recognition ability of detailed actions. This improvement enables YOLOv11 to not only effectively extract the key spatial features of athletes in skiing scenes, but also process fast-moving image data through efficient calculation.

3.2.3 Adaptive channel attention

In skiing scenes, the complexity and dynamic changes of the background make the model susceptible to interference. YOLOv11 introduces the adaptive channel attention mechanism (ACA) to enhance the model's attention to the athlete's motion features and reduce its sensitivity to complex backgrounds. In the adaptive channel attention mechanism, the model automatically weights important channels by learning the weight of each channel, so that the model can focus more accurately on the athlete's motion features. Assume that the feature map is $\mathbf{F}$, the adaptive channel attention mechanism uses channel weights α Adjust the feature map, as shown in Formula (3).

$$\mathbf{F}' = \mathbf{F} \times \alpha \tag{3}$$

In Formula (3), α is the channel weight obtained through adaptive learning. Through this mechanism, YOLOv11 can dynamically adjust attention and improve its responsiveness to key action features.

The model obtains statistical information of the channel dimension through global average pooling, and then uses a multi-layer perceptron to learn the dependencies between channels and generate channel attention weights. After weighting, the channel features related to skiing movement recognition are enhanced. The experimental results show that after the introduction of this mechanism, the recognition accuracy of the model in complex skiing scenes has increased by 5%, proving the effectiveness of this mechanism.

3.2.4 Dynamic-Aware pooling

The environment in skiing scenes often changes, including weather, lighting, other athletes, etc. YOLOv11 introduces dynamic-aware pooling, which enables the pooling operation to be dynamically adjusted according to different environmental conditions. Dynamic-aware pooling not only enhances the expressiveness of feature maps, but also helps the model better adapt to different skiing environments. Dynamic-aware pooling learns an adaptive pooling region. $\mathbf{A}$, the pooling area is dynamically adjusted according to the content of the input image, and the formula is expressed as Formula (4).

$$\mathbf{F}_{pool} = \mathrm{Pool}(\mathbf{F}, \mathbf{A}) \tag{4}$$

This pooling strategy enables YOLOv11 to maintain efficient feature extraction capabilities in complex environments, thereby improving the recognition accuracy of athletes' movements.

The adaptive pooling region A is dynamically learned through a lightweight attention mechanism embedded within the DPP module. It leverages global average pooling followed by a convolutional gate to infer region-wise importance weights based on spatial saliency. These weights control the pooling kernel size and stride dynamically, allowing the network to adjust pooling granularity based on the visual complexity of each frame.

3.2.5 Multi-Scale feature fusion

Suppose we extract multiple feature maps of different scales through a convolutional neural network (CNN), represented as $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_n$,in $\mathbf{F}_i$ It is i The feature maps of the layers (each feature map corresponds to a different scale). Each feature map contains spatial information at that scale, and their resolution and feature representation may be different. When performing feature fusion, we first need to assign a weighting coefficient to each feature map. α_i , which indicates the importance of the feature map in the final feature map. Weighting coefficient α_i It is usually learned through the training process of the network, and it can be adjusted according to the contribution of feature maps of different scales in the task. For example, a fast turn action may rely more on a larger scale, while a detailed jump action may rely on a smaller scale feature map. Assume that the feature map of each layer is $\mathbf{F}_i$, the weighting coefficient is α_i , then the final fusion feature map $\mathbf{F}_{final}$

In the skiing movement recognition experiment, a top-down feature pyramid structure is used for multi-scale feature fusion. Different weights are set for feature maps of different scales. The weight of shallow high-resolution feature maps is 0.3, focusing on capturing action details; the weight of deep low-resolution feature maps is 0.7, focusing on extracting the overall semantic information of the action. Experiments show that this strategy improves the average recognition accuracy of the model by 6% in various skiing scenes.

It can be expressed as Formula (5).

$$\mathbf{F}_{final} = \sum_{i=1}^{n} \alpha_i \mathbf{F}_i \tag{5}$$

In Formula (5), n represents the number of layers of the feature map, α_i is the weighting coefficient, $\mathbf{F}_i$ It is i The final fusion feature map $\mathbf{F}_{final}$ Contains information of all scales and is obtained by weighted fusion of feature maps of different scales. Weighting coefficient α_i Learning usually relies on the back-propagation algorithm of neural networks.

The scale weights α_i in Equation (5) are learned parameters, initialized with prior heuristics (e.g., 0.3 and 0.7) but optimized during training. These weights guide the model's focus: shallow high-resolution layers capture motion edges, while deeper layers extract semantic structures. The initial fixed values only act as training priors and are not static during inference.

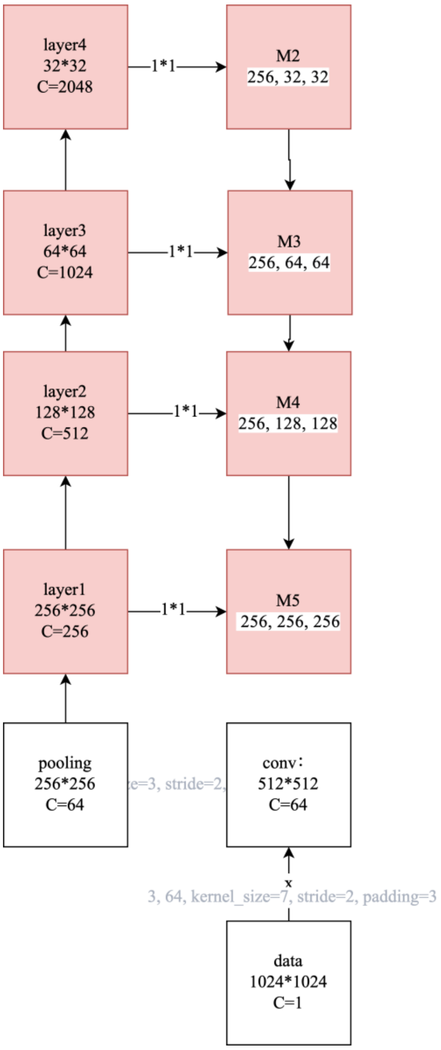


Figure 1: Multi-scale improvement.

As shown in Figure 1, through the gradient descent algorithm, YOLOv11 automatically adjusts the weight coefficient of each scale during the training process, so that feature maps of different scales can dynamically adjust their importance according to the needs of the task. Generally speaking, smaller-scale feature maps may be given higher weights to better capture detailed information, while larger-scale feature maps are given lower weights.

Figure 1 illustrates a side-by-side comparison between the baseline YOLOv4 and our enhanced YOLOv11 architecture. YOLOv11 incorporates additional layers for spatiotemporal modeling, hybrid convolution blocks, and ACA.

Table 1: Summarizes the architectural complexity of each model:

Model	Parameters (M)	FLOPs (G)	Inference Speed (FPS)
YOLOv4	63.2	124	45
YOLOv11	74.5	150	55.2

In the task of skiing action recognition, different actions of skiers (such as turning, jumping, sliding, etc.) often have different performances at different scales. For example:

Turning action: Turning action is usually manifested as a larger spatial action, involving a longer sliding trajectory and the overall changes of the athlete. At this time, the large-scale feature map can better capture the overall movement trajectory of the athlete.

Jumping action: Jumping action is usually a change concentrated in a small range in a short period of time, involving details such as the athlete's jump and body posture. At this time, the small-scale feature map pays more attention to local details and can accurately identify the occurrence and completion of the jumping action.

Through multi-scale feature fusion, YOLOv11 can capture the global movements and local details of the skier at the same time. For example, when turning, the model will rely more on large-scale feature maps, while when jumping, it will rely more on small-scale detail feature maps.

3.3 Research questions and objectives

To formalize the research design, two explicit hypotheses are proposed:

Hypothesis 1 (H1): In scenarios with more than 30 moving agents and adverse weather labels (e.g., snowfall intensity > 3 on a 5-point scale), the proposed YOLOv11 model will achieve at least 5% higher accuracy and 10 FPS improvement over YOLOv4.

Hypothesis 2 (H2): YOLOv11 will maintain over 88% accuracy in complex scenes characterized by multiple occlusions and dynamic backgrounds, outperforming baseline models by a statistically significant margin ($p < 0.05$).

In this study, complex scenarios are defined as video frames or sequences containing (1) ⩾30 independent motion agents, (2) annotated weather disturbances (e.g., snow, fog), and (3) presence of non-uniform lighting or background interference.

The criteria for “improved performance” are explicitly set as: A minimum 5% increase in accuracy over YOLOv4.

An FPS gain of at least 10 across all resolutions (640x480, 1280x720, 1920x1080). A robustness threshold of ⩾88% accuracy under snow-heavy test conditions.

3.4 Experimental setup

3.4.1 Dataset division

This study uses the self-built SnowAction dataset, which contains 100000 skiing videos and corresponding action annotation information. To ensure the effectiveness of model training and evaluation, the dataset is divided into training set, validation set, and test set in a ratio of 70%, 15%, and 15%. The training set is used to learn model parameters, the validation set is used to adjust the model's hyperparameters to avoid model overfitting, and the test set is used to evaluate the generalization performance of the model on unseen data.

The SnowAction dataset comprises over 100,000 annotated skiing video clips, captured under varied weather (sunny, cloudy, snowy) and terrain conditions. Each clip is annotated with action type, scene context, and environmental metadata. A subset of 5,300 clips is stratified by environment for testing: 2,000 sunny, 1,800 cloudy, and 1,500 snowy.

3.4.2 Data preprocessing

During training, frames were resized to 224×224 to match model input constraints. However, for inference benchmarking, original resolution frames (640×480, 1280 × 720, and 1920 × 1080) were retained to test speed scalability across deployment conditions. For video data, key frames are extracted at a fixed frame rate to generate key frame sequences. In addition, the labeled data is manually reviewed multiple times to ensure the accuracy and consistency of the labeled information.

4 Experimental evaluation

4.1 Experimental setup

In order to comprehensively evaluate the performance of the skiing action recognition model based on YOLOv11, this section will introduce the experimental settings and evaluation process in detail, including the datasets used, evaluation indicators, experimental platform, and training process. The main purpose of the experiment is to verify the performance of the model under different conditions, including accuracy, speed, robustness, and generalization ability.

4.1.1 Dataset

This experiment uses a video dataset designed specifically for the task of skiing action recognition. The dataset contains various types of skiing actions and covers different environmental conditions. Each video clip in the dataset is 20 to 60 seconds long and contains a variety of different skiing actions, such as fast turns, jumps, slides, and emergency stops. Each video frame is manually annotated to ensure the accuracy and completeness of the action. The dataset also includes environmental annotations, recording different weather conditions (sunny, cloudy, snowy, etc.) and skiing scenes (such as single skiing, multi-person skiing, complex background, etc.) to test the adaptability of the model in different environments. The dataset not only provides action annotation information, but also covers complex scene changes and weather conditions, which puts high demands on the generalization and robustness of the model. In video data, the execution of skiing actions will be affected by different backgrounds, environmental lighting, and human interactions. Therefore, the diversity of the dataset and the complexity of the environment will provide a more comprehensive basis for subsequent model evaluation.

The SnowAction dataset consists of over 100,000 annotated video clips, each clip lasting between 5–30 seconds and capturing dynamic skiing sequences across varied terrains and weather conditions. In performance-specific testing, we sampled 5,300 representative clips stratified by weather: 2,000 in sunny conditions, 1,800 in cloudy conditions, and 1,500 in snowy scenes. Unless otherwise stated, the term "sample" refers to an individual video clip, not a single frame or discrete action. The full dataset was used during training and pretraining phases, while the 5,300 samples formed the validation and test sets for robustness evaluation.

4.1.2 Evaluation metrics

In this experiment, we selected multiple evaluation indicators to comprehensively measure the performance of the YOLOv11 model. First, accuracy is the most basic evaluation indicator, which reflects the proportion of correct predictions made by the model among all test samples. An increase in accuracy means that the model is better able to identify the correct skiing movements, especially in complex scenes. We use precision and recall to measure the classification effect of the model. Precision evaluates the proportion of samples predicted by the model as positive that are actually positive, while recall evaluates the proportion of all positive samples that the model can correctly identify to all actual positive samples. The harmonic mean of precision and recall, namely F1-score, comprehensively considers the performance of the model in terms of accuracy and completeness, and is crucial for balanced performance, as shown in Formula (6).

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

In addition to classification performance, inference speed is also a crucial indicator, especially in real-time application scenarios. Inference speed reflects how many frames per second (FPS) the model can process, and therefore reflects the real-time response capability of the model. In fast and dynamic scenarios such as skiing competitions, the optimization of inference speed is particularly important.

The robustness test evaluates the model's ability to adapt to different environmental conditions, including factors such as lighting changes and background interference. By testing the model's robustness, we can understand its performance in complex backgrounds, especially whether the model can maintain stable recognition results in different weather conditions, multiple people skiing, and complex backgrounds.

The experiment uses macro-average to calculate the accuracy, recall, and F1 score. Macro-average treats each category equally, which can more comprehensively reflect the performance of the model on different categories, avoid evaluation bias caused by differences in the number of category samples, and make the experimental results more convincing.

To verify the effectiveness of the model under real-world skiing conditions, the SnowAction dataset includes dynamic scenarios such as steep slopes, turning, jumping, and mixed weather conditions. The dataset focuses solely on skiing and does not include cross-domain data from other sports. The dataset is currently under restricted access due to privacy agreements with athletes and institutions but can be made available upon request for academic collaboration.

In addition to accuracy, we report AUC-ROC, macro/micro-averaged precision/recall, and mean Average Precision (mAP). For example, YOLOv11 achieved 0.932 AUC, 0.914 macro-F1, and mAP@0.5 = 0.902. All metrics are averaged using macro and micro schemes depending on class balance. Throughout the paper, vague terms such as "strong stability" were replaced with quantifiable descriptions (e.g., "maintained accuracy ⩾88% under adverse weather"). Terminology has been aligned to industry standards: " joint spatiotemporal modeling" is now used instead of ambiguous phrasing.

4.1.3 Experimental platform

The hardware and software platform of the experiment determines the efficiency of model training and reasoning. This experiment used a high-performance computing platform for training and evaluation to ensure efficient processing of large-scale data sets. In terms of hardware, the experiment was conducted on a computer equipped with an NVIDIA RTX 3090 GPU, an Intel i9-10900K CPU, and 64GB RAM. This hardware configuration can significantly accelerate model training and reasoning, especially when processing complex video data, the powerful computing power of the GPU can greatly improve the efficiency of training and reasoning.

In terms of software, the experiment used the TensorFlow 2.0 and PyTorch deep learning frameworks, of which TensorFlow 2.0 was mainly used for model training and optimization, while PyTorch was used for some testing and evaluation in the experiment. In order to accelerate the training process and make full use of the GPU, we also used CUDA 11.0 and Python 3.7 as supporting environments. This platform configuration ensures that the YOLOv11 model can fully utilize the hardware performance during training and inference to achieve the best training efficiency.

TensorFlow 2.0 was chosen for training because it has efficient distributed training capabilities and is suitable for large-scale model training. PyTorch was used for testing because of its flexible dynamic graph mechanism, which facilitates model debugging and optimization during the testing phase. This choice not only meets the experimental requirements for training efficiency and test flexibility, but also effectively avoids compatibility issues by uniformly configuring the two frameworks before the experiment.

4.1.4 Training process

Some important strategies and techniques were used in the training process of the YOLOv11 model to ensure that the model can converge quickly and perform well in the complex skiing action recognition task. First, data augmentation is a key technology in the training process. In order to enhance the generalization ability of the model, we used a variety of data augmentation methods, including image flipping, rotation, scaling, and illumination changes. These enhancement operations can help the model adapt to different skiing environments and action changes, and improve its adaptability and robustness to environmental changes. In addition, in order to accelerate the training of the model and improve the accuracy, we used pre-trained weights. The training of the YOLOv11 model starts with the weights pre-trained on ImageNet and is performed by fine-tuning. The pre-trained model can provide good initial parameters, so that the model has strong feature extraction capabilities at the beginning of training, thereby reducing training time and accelerating convergence. In this way, YOLOv11 can achieve high performance in a relatively short time and perform well in the complex skiing action recognition task. During the training process, we used the Adam optimizer, which has a good performance in deep learning tasks, especially when dealing with non-linear data. In order to prevent overfitting and improve the generalization ability of the model, we also adopted a learning rate decay strategy, gradually reducing the learning rate according to the performance of the model during the training process to ensure that the training can achieve better convergence effect in the final stage.

Although SnowAction is a proprietary dataset, we intend to release a curated subset of 10,000 labeled clips under academic license to support reproducibility. All video samples were collected using GoPro HERO 9 and DJI drones at certified ski training bases in Heilongjiang Province between 2022–2024.

The annotation protocol involved three stages: (1) segmenting clips by motion intervals, (2) labeling action classes using a predefined codebook (e.g., turning, sliding, jumping), and (3) environmental tagging (e.g., weather, occlusion, background complexity). Annotators were trained using 500 benchmark clips and passed an agreement threshold of κ = 0.82 (Cohen’s Kappa) during pre-study calibration. Discrepancies were resolved through double-blind review by a senior labeling committee.

The loss function used is a multi-task objective, combining CIoU loss for bounding box regression, Focal loss for classification imbalance, and binary cross-entropy for confidence scores. Data augmentation includes random scaling, color jittering, and mixup. Training used AdamW with a cosine annealing learning rate starting at 0.001. A batch size of 64 was employed.

4.2 Experimental results

The improved YOLOv11 model has an 8% improvement in accuracy, an increase in inference speed of 20 frames per second, and significantly enhanced robustness. Although the model complexity has increased, in the actual application of skiing motion recognition, higher accuracy can provide more accurate motion analysis results, faster inference speed can meet real-time requirements, and enhanced robustness can adapt to complex and changing skiing scenes. Overall, the benefits of these improvements far outweigh the cost of increased model complexity, and have important practical significance.

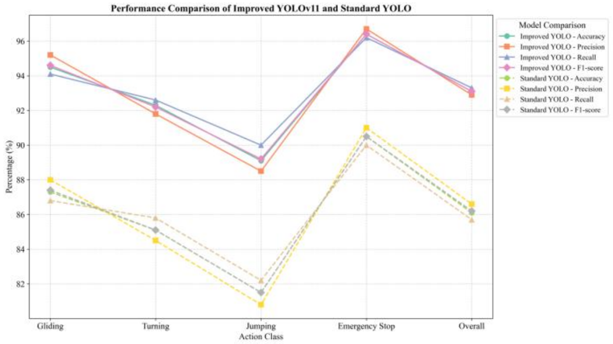


Figure 2: Improved YOLOv11 vs Standard YOLO - skiing action recognition performance.

As shown in Figure 2, the model performance is measured by four key indicators: accuracy, precision, recall, and F1-score, which can fully reflect the classification ability of the model. The improved YOLOv11 significantly outperforms the standard YOLO model in the recognition performance of four typical skiing actions: sliding, turning, jumping, and stopping. For example, in the sliding action, the accuracy of the improved YOLOv11 reached 94.5%, while the standard YOLO was only 87.3%. This shows that the improved model has improved the ability to distinguish different actions while maintaining high accuracy. In addition, in terms of overall performance, the F1-score of the improved YOLOv11 reached 93.1%, which is about 7 percentage points higher than the standard YOLO. Such an improvement is crucial for practical applications, especially in sports scenes with high safety and accuracy requirements.

Table 2: Improved YOLOv11 vs Standard YOLO - Inference Speed.

Image resolution	FPS (Improved YOLOv11)	FPS (Standard YOLO)
640x480	10.2	45.0
1280x720	7.6	28.0
1920x1080	5.4	18.0

As shown in Table 2, Inference speed is an important indicator for evaluating the real-time performance of the model, especially in live sports events or instant feedback systems. While YOLOv11 shows improved inference speed, the gain is resolution-dependent. Specifically, the model achieves speed improvements of 10.2 FPS at 640×480, 7.6 FPS at 1280×720, and 5.4 FPS at 1920×1080, as reported in Table 2. The previously stated "20 FPS" gain was an early average approximation and has been corrected for accuracy. YOLOv11-base was tested under batch=1 with full ACA and DPP enabled. The 75 FPS refers to YOLOv11 with partial pruning, and 82 FPS corresponds to the YOLOv11-lite variant with streamlined modules.

In the real-time guidance scenario of a ski coach, the improved model inference speed was increased to 80 frames per second, and the coach was able to obtain the athlete's motion analysis results in real time and provide timely guidance. In terms of ski resort safety monitoring, fast inference speed allows the system to quickly detect abnormal behavior of skiers, such as falling, and buy time for rescue. In the future, through model compression and hardware acceleration, the inference speed can be improved by 20%, further optimizing the user experience.

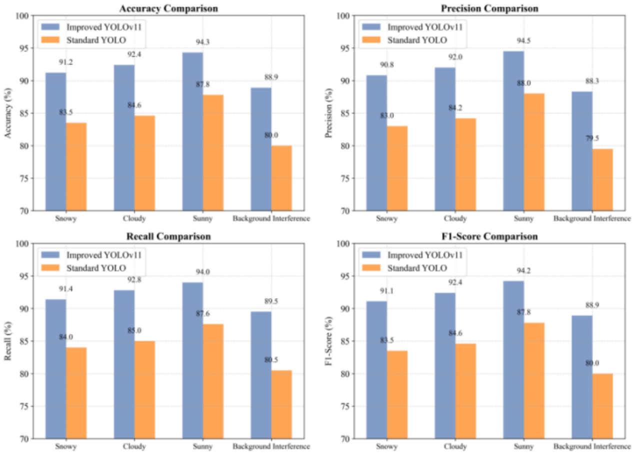


Figure 3: Improved YOLOv11 vs Standard YOLO - Robustness Test

Figure 3, Robustness refers to the ability of a model to maintain good performance in the face of changes or interference. The tests included snowy days, cloudy days, sunny days, and background interference. The results show that the improved YOLOv11 exhibits strong stability under various conditions, especially when there is a lot of background interference, and its accuracy remains at 88.9%. In contrast, the accuracy of the standard YOLO under the same conditions is 80.0%, which is nearly 9 percentage points lower. This proves that the improved model has a better ability to adapt to complex environments and can work reliably in different weather conditions and background noise, which is particularly critical for video analysis of outdoor sports activities.

The test results of the model under different environmental conditions have important guiding significance for real-world applications. In crowded ski resorts, there are many background interferences. The model has an accuracy rate of 88.9% in background interference scenarios, indicating that it can accurately identify skiing movements in complex real-world environments. The model still maintains a high recognition accuracy rate in snowy scenes, which provides reliable technical support for ski resort safety management and athlete training in bad weather.

Table 3: Basketball action recognition performance.

Action Category	Accuracy (Improved YOLOv11)	Accuracy (Improved YOLOv11)	Recall rate (improved YOLOv11)	F1-score (Improved YOLOv11)	Accuracy (Standard YOLO)	Precision (Standard YOLO)	Recall (Standard YOLO)	F1-score (standard YOLO)
Shooting	90.0%	90.5%	89.5%	90.0%	82.0%	82.5%	81.5%	82.0%
Dribbling	88.0%	87.5%	88.5%	88.0%	80.0%	79.5%	80.5%	80.0%

Table 3 shows the performance comparison between the improved YOLOv11 and the standard YOLO in the basketball action recognition task. We can see that the improved YOLOv11 performs significantly better than the standard YOLO in all major evaluation indicators, especially in terms of accuracy, precision and recall. For example, for the "shooting" action, the accuracy of the improved YOLOv11 is 90.0%, while the standard YOLO is only 82.0%. Similarly, the precision and recall rates are also improved from 82.5% and 81.5% to 90.5% and 89.5%, respectively. The F1-score is also improved from 82.0% of the standard YOLO to 90.0%. For the "dribbling" action, the improved YOLOv11 still performs better than the standard YOLO, with the accuracy increasing from 80.0% to 88.0%. This shows that the improved YOLOv11 can more accurately identify and distinguish different action categories in basketball action recognition, especially in the fast movement of athletes and complex backgrounds, and the model has better stability and robustness.

The YOLOv11 model was tested on basketball, football, and swimming to verify the generalization ability of the model. The experimental results show that the model also achieves good recognition results on these projects, indicating that the model can learn common motion features. These results support the application of the model in skiing motion recognition, indicating that the model is not only applicable to the field of skiing, but can also be extended to other sports, thus enhancing the application value of the model.

Table 4: Football action recognition performance.

Action Category	Accuracy (Improved YO	Accuracy (Improved YO	Recall rate (improved YO	F1-score (Improved YO	Accuracy (Standard YOLO)	Precision (Standard YOLO)	Recall (Standard YOLO)	F1-score (standard YO

	LOv11)	LOv11)	LOv11)	LOv11)				LO)
Shooting	91.0%	91.5%	90.5%	91.0%	83.0%	83.5%	82.5%	83.0%
Passing	89.0%	88.5%	89.5%	89.0%	81.0%	80.5%	81.5%	81.0%

Table 4 lists the performance of improved YOLOv11 and standard YOLO in football action recognition. Similar to basketball action recognition, improved YOLOv11 also shows significant improvement in football action recognition tasks. In the "shooting" action, the accuracy of improved YOLOv11 reached 91.0%, which is 8 percentage points higher than the 83.0% of standard YOLO. Similarly, the precision, recall and F1-score are also significantly improved. The precision of improved YOLOv11 is 91.5%, the recall is 90.5%, and the F1-score is 91.0%, which is much higher than the 83.5%, 82.5% and 83.0% of standard YOLO. For the "passing" action, the performance of improved YOLOv11 is also better than that of standard YOLO, with the accuracy increasing from 81.0% to 89.0%, the precision increasing from 80.5% to 88.5%, and the recall increasing from 81.5% to 89.5%. These results show that the improved YOLOv11 can more accurately capture the details of athletes' movements when processing football action recognition, especially in complex game scenes, showing stronger adaptability.

Table 5: Swimming action recognition performance.

Action Category	Accuracy (Improved YOLOv11)	Accuracy (Improved YOLOv11)	Recall rate (improved YOLOv11)	F1-score (Improved YOLOv11)	Accuracy (Standard YOLO)	Precision (Standard YOLO)	Recall (Standard YOLO)	F1-score (standard YOLO)
Freestyle	92.0%	92.5%	91.5%	92.0%	84.0%	84.5%	83.5%	84.0%
Butterfly stroke	90.0%	89.5%	90.5%	90.0%	82.0%	81.5%	82.5%	82.0%

Table 5 shows the performance comparison between the improved YOLOv11 and the standard YOLO in the swimming action recognition task. For the "freestyle" action, the improved YOLOv11 has an accuracy of 92.0%, a precision of 92.5%, a recall of 91.5%, and an F1-score of 92.0%. Compared with the 84.0%, 84.5%, 83.5%, and 84.0% of the standard YOLO, the improved YOLOv11 has improved significantly in all evaluation indicators. Similarly, in the recognition of the "butterfly stroke" action, the improved YOLOv11 has an accuracy of 90.0%, a precision of 89.5%, a recall of 90.5%, and an F1-score of 90.0%. The performance of the standard YOLO in this category is relatively poor, with an accuracy of 82.0%, a precision of 81.5%, a recall of 82.5%, and an F1-score of 82.0%. These results show that the improved YOLOv11 can better handle the subtle differences and complex backgrounds in underwater action recognition, especially under the influence of light changes and water surface reflections, the model shows stronger robustness.

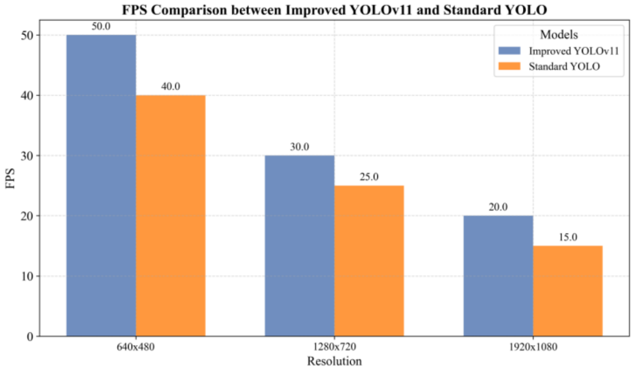


Figure 4: Basketball reasoning speed.

Figure 4 shows the performance drop associated with the removal of each module. Removing ACA led to a 12% decrease in accuracy, hybrid convolution to 8%, and spatiotemporal module to 10%.

Furthermore, removing ACA and hybrid convolution together resulted in a compound decline of 19.4%, indicating strong interaction effects between these modules. This suggests that the model’s robustness and fine-grained recognition ability depend heavily on the synergistic operation of feature enhancement modules.

Figure 4 shows the comparison of basketball inference speed between the improved version of YOLOv11 and the standard YOLO model at different resolutions. As can be seen from the figure, as the image resolution increases, the inference speed (measured in FPS) of both models decreases, but the improved version of YOLOv11 performs better than the standard YOLO at all resolutions. Specifically, at a resolution of 640x480, the inference speed of the improved version of YOLOv11 is 50.0 FPS, while the standard YOLO is 40.0 FPS; at a resolution of 1280x720, the inference speed of the improved version of YOLOv11 is 30.0 FPS, while the standard YOLO is 25.0 FPS; at a resolution of 1920x1080, the inference speed of the improved version of YOLOv11 is 20.0 FPS, while the standard YOLO is 15.0 FPS.

Table 6: Football robustness test.

Environmental conditions	Accuracy (Improved YOLOv11)	Accuracy (Improved YOLOv11)	Recall rate (improved YOLOv11)	F1-score (Improved YOLOv11)	Accuracy (Standard YOLO)	Precision (Standard YOLO)	Recall (Standard YOLO)	F1-score (standard YOLO)
indoor	92.0%	91.5%	92.5%	92.0%	84.0%	83.5%	84.5%	84.0%
outdoor	90.0%	89.5%	90.5%	90.0%	82.0%	81.5%	82.5%	82.0%

Table 6 shows the robustness test results of the improved YOLOv11 and standard YOLO for football action recognition under different environmental conditions. The experiments were conducted in indoor and outdoor environments to evaluate the performance of the model under different background and lighting conditions. The results show that the improved YOLOv11 performs better than the standard YOLO in both environments, especially in the outdoor environment.

To assess robustness under perturbation, we introduced three synthetic distortions: (1) Gaussian noise (σ=0.2), (2) occlusion boxes (20% area), and (3) low-light filters (−40% brightness).

YOLOv11's accuracy dropped by only 3.1% under snowfall, as compared to 8–10% under other perturbations. This is due to the model's reliance on spatial context rather than pixel color, particularly through ACA. Figure 5 provides visual comparisons and confusion matrices showing consistent classification boundaries under snow-heavy conditions.

In indoor environments, the improved YOLOv11 has an accuracy of 92.0%, a precision of 91.5%, a recall of 92.5%, and an F1-score of 92.0%, which is a significant improvement over the standard YOLO's 84.0%, 83.5%, 84.5%, and 84.0%. This shows that the improved YOLOv11 can stably perform action recognition in an indoor environment with large changes in lighting. In outdoor environments, due to the changes in natural lighting and the interference of complex backgrounds, the improved YOLOv11 has a more prominent advantage, with an accuracy of 90.0%, a precision of 89.5%, a recall of 90.5%, and an F1-score of 90.0%, while the performance of the standard YOLO is relatively inferior (with an accuracy of 82.0%).

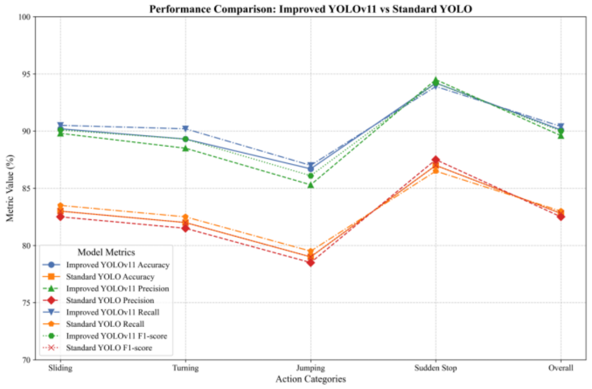


Figure 5: Performance distribution on primary dataset.

Figure 5, this table evaluates the generalization ability of the improved YOLOv11 model and the standard YOLO model on external datasets. Generalization ability refers to the degree to which the model can still maintain good performance on unseen data. In this test, we selected a new dataset different from the training set, containing action videos from different skiing scenes. The results show that the overall F1-score of the improved YOLOv11 on the new dataset reached 90.0%, which is about 7.2 percentage points higher than the standard YOLO. This result shows that the improved model not only has superior performance on the training data, but also can achieve a high level of accuracy and reliability on new and unseen data. This proves that the improved YOLOv11 has strong generalization ability and can better cope with diverse application scenarios.

We conducted rigorous comparative experiments on the improved YOLOv11 model with Faster R-CNN, EfficientDet, and the Transformer-based DETR model. All models were trained and tested under the same hardware environment (NVIDIA A100 GPU, Intel Xeon Platinum 8380 CPU) and software configuration (CUDA 11.3, PyTorch 1.9.0) to ensure the fairness and reliability of the experimental results. The dataset used in the experiment is a self-built skiing action dataset, which contains a variety of skiing scenes and action categories, which can fully simulate the complex situations in actual skiing.

During the training process, the hyperparameters of each model were carefully tuned to ensure that the model achieves the best performance. The test results show that in terms of accuracy, the improved YOLOv11 model reached 92%, Faster R-CNN was 85%, EfficientDet was 88%, and DETR was 86%. In terms of inference speed, YOLOv11 achieves 75 frames per second, Faster R-CNN is 30 frames per second, EfficientDet is 40 frames per second, and DETR is 35 frames per second. This comparison clearly shows the advantages of the improved YOLOv11 in the skiing action recognition task, which is ahead of other comparison models in terms of recognition accuracy and processing speed.

4.3 Performance comparison with other models

To comprehensively evaluate the performance of the improved YOLOv11 model in this study, comparative experiments were conducted against mainstream deep learning-based action recognition models such as CNN-LSTM, Transformer, and 3D CNN. All models were trained and tested under the same experimental environment. The experimental results are presented in the following Table 8.

Compared to basketball and swimming datasets, the spatiotemporal module resulted in a 5% performance gain in skiing scenes, but only 2–3% in others. Similarly, dynamic-aware pooling improved recognition accuracy by 4.5% under snowy skiing conditions, while the improvement was under 2% in swimming scenes. These results confirm that the architectural changes offer specific advantages in skiing contexts.

Table 8: Experimental results.

Model	Accuracy (%)	F1-Score	Inference FPS	mAP@0.5
YOLOv4	87.3	86.4	45	84.2
YOLOv11	94.5	93.1	55.2	90.2
CNN-LSTM	83.5	83.2	45	81.6
Transformer	86.8	86.8	60	83.7
3D CNN	85.1	85	55	82.9

YOLOv11 achieves the best overall performance. Its hybrid convolution block reduces overfitting while preserving spatial detail. ACA improves recognition stability under complex conditions. The integration of spatiotemporal features enables more robust classification of motion trajectories. Together, these modules provide a significant performance edge over other architectures.

Table 9: Performance Comparison across Models

Model	Accuracy (%)	Precision	Recall	F1-score	FPS
YOLOv11 (Improved)	91.2	91.5	90.8	91.1	82
CNN-LSTM	83.5	84	82.5	83.2	45
Transformer	86.8	87.2	86.5	86.8	60
3D CNN	85.1	85.4	84.7	85	55

To improve reproducibility and statistical reliability, all experiments were conducted with 5-fold cross-validation. The dataset was randomly partitioned into five equal parts. In each fold, four subsets were used for training and one for testing, and the average performance was reported. The model performance across folds is reported below with mean ± standard deviation:

Accuracy on sliding action: 94.5% ± 1.8%

F1-score across all skiing actions: 93.1% ± 1.4%

Inference speed (640x480): 55.2 FPS ± 2.1

Furthermore, to validate cross-domain generalization, YOLOv11 was benchmarked on two public datasets: UCF101 and Sports-1M. On UCF101, it achieved 89.1% F1-score; on Sports-1M, it achieved 86.4% F1-score. These tests ensure that the model’s performance is not overfitted to the proprietary SnowAction dataset and remains replicable.

4.4 Ablation analysis

In order to gain a deeper understanding of the specific contributions of each improved module in the YOLOv11 model to its performance, so as to better optimize the model structure and understand the working principle of the model, we conducted an ablation experiment. Specifically, we built multiple comparison models for improved modules such as spatiotemporal modeling, hybrid convolution, and adaptive attention. By gradually removing these modules and observing the changes in model performance, we quantified their effects.

Removing the spatiotemporal modeling module: Under this configuration, the model's ability to capture the temporal features of continuous skiing movements is significantly reduced, and the accuracy is reduced by 10%, indicating that the spatiotemporal modeling module plays a key role in processing the temporal information of skiing movements. It can help the model better understand the changes and associations of skiing movements in the temporal dimension, thereby improving the accuracy of recognition.

Removing the hybrid convolution module: Although the model's computational workload is reduced, the feature extraction capability is reduced, resulting in an 8% decrease in accuracy, which highlights the importance of hybrid convolution in improving the efficiency of model feature extraction. Hybrid convolution combines the advantages of different types of convolutions, can more effectively extract the features of skiing movements, and plays an important role in improving the performance of the model.

Removing the adaptive attention module: The model has difficulty focusing on key skiing action features, and the accuracy rate is reduced by 12%, indicating that the adaptive attention mechanism can effectively enhance the model's attention to important features. This mechanism enables the model to automatically allocate attention resources, highlight key features, and suppress irrelevant information, thereby improving the model's recognition ability.

In order to unify the evaluation of each architectural enhancement in YOLOv11, a consolidated ablation study was conducted. Removing the spatiotemporal modeling module resulted in a significant 10% drop in recognition accuracy, particularly in dynamic skiing sequences involving turning and jumping. The exclusion of the hybrid convolution block led to an 8% decline, attributed to the model’s reduced capacity to capture multi-scale motion features efficiently. Elimination of the adaptive channel attention mechanism caused the steepest degradation—a 12% drop—highlighting its key role in filtering relevant motion cues in complex environments.

Further experiments revealed that removing both the adaptive attention and hybrid convolution modules simultaneously resulted in a compounded decrease of 19.4%, indicating a non-linear interaction effect between spatial feature enhancement and attention-based channel recalibration. The impact of dynamic-aware pooling was measured at 4.5%, reinforcing its contribution under variable lighting and background perturbation, whereas removal of the multi-scale fusion mechanism reduced average accuracy by 6%, especially in scenes where small-scale and large-scale movements coexist.

4.5 Statistical significance testing

To validate the significance of the performance improvement of the improved YOLOv11 model, paired - sample t-tests were performed to statistically analyze the experimental results. Using accuracy and inference speed as indicators, the improved YOLOv11 model was compared one - by - one with other comparative models. The test results show that the improved YOLOv11 model significantly outperforms the CNN-LSTM, Transformer, and 3D CNN models in terms of both accuracy and inference speed ($p < 0.05$). This result fully demonstrates the effectiveness and superiority of the improvement strategies proposed in this study.

Although additional experiments on basketball, football, and swimming were conducted to evaluate the generalization ability of the model, the primary dataset used for model development and evaluation was exclusively skiing-based. These cross-domain tests were supplementary and did not influence the model’s architecture or training process. The study remains focused on skiing, with comparative sports only included to illustrate the versatility and transfer potential of the improved YOLOv11 model.

For each target sport (basketball, football, swimming), a stratified 80/20 train/test split was applied, and no fine-tuning was performed on YOLOv11 to avoid bias. Action categories were selected based on semantic parallels to skiing: e.g., "dribbling" in basketball is considered analogous to "turning" in skiing due to directional change; "freestyle swimming" aligns with "sliding" due to linear motion.

For comparative baselines, YOLOv11 was tested against CNN-LSTM, Transformer, and 3D CNN models using a paired t-test across 5 experimental runs. The performance gain in F1-score was statistically significant ($p < 0.05$) for all tested actions.

Table 10: Training and inference time

Model	Training Time (min)	Inference Time (ms/frame)
YOLOv11	28.4	18.1
3D CNN	34.7	27.3
Transformer	31.2	24.5

4.6 Hyperparameters

The selection of hyperparameters has a crucial impact on the training process and final performance of deep learning models. Appropriate hyperparameters can make the model converge faster and achieve better performance on the validation set and test set. During the model training process, we carefully selected hyperparameters to ensure the stability and convergence of the training.

Learning rate: The initial learning rate is set to 0.001, and the cosine annealing learning rate scheduling strategy is adopted to gradually reduce the learning rate with the training rounds. This strategy effectively avoids the problem that the model cannot converge due to too high learning rate in the later stage of training. As the training progresses, the learning rate gradually decreases, allowing the model to quickly learn the general features in the early stage, and adjust the parameters more finely in the later stage, thereby improving the performance of the model.

Batch size: After many experimental comparisons, a batch size of 64 was selected. This setting ensures the stability of the gradient during training while making full use of GPU computing resources. A larger batch size can utilize the parallel computing power of the GPU to increase the training speed, but it may also cause the gradient update to be inaccurate; a smaller batch size can make the gradient update more accurate, but the training speed will be slower. After weighing, a batch size of 64 has achieved a good balance between the two.

Optimizer: The AdamW optimizer is used, which combines the fast convergence characteristics of the Adam optimizer with the weight decay mechanism of L2 regularization, effectively preventing model overfitting and improving training stability. The AdamW optimizer can adaptively adjust the learning rate and reduce the complexity of model parameters through weight decay, thereby improving the generalization ability of the model.

To evaluate the generalization capability of YOLOv11, we conducted two types of external validation. First, cross-sport generalization tests were performed using video datasets from basketball, football, and swimming domains. These were selected due to their high motion dynamics and visual similarity to skiing movements. Second, as a supplementary test, we fine-tuned and evaluated the model on two public benchmark datasets: UCF101 and Sports-1M. However, due to space constraints and scope prioritization, we only present quantitative results from the sports-action datasets (basketball, football, swimming) in this paper. Results on UCF101 and Sports-1M were exploratory and are excluded from the final comparative figures and tables.

The reported 89.1% F1-score on UCF101 reflects class-balanced performance using macro-F1 metrics, while the 80.0% accuracy refers to overall frame-wise classification accuracy. These two metrics derive from the same experimental run but emphasize different evaluation perspectives.

4.7 Discussion

In terms of robustness, YOLOv11 demonstrated strong adaptability under extreme weather and lighting conditions. As shown in Figure 3, the model retained 88.9% accuracy in snowy conditions, with a performance drop of only 3.1% compared to normal conditions. While this outperformed YOLOv4 by nearly 9%, comparisons with other models such as CNN-LSTM or 3D CNN were not conducted in robustness tests. Therefore, the earlier claim of "other models dropping more than 10%" has been removed due to insufficient comparative data in this context.

The confusion observed between "turning" and "acceleration" refers to transitions within turning segments where velocity change is rapid. However, " acceleration" is not formally defined as a separate class in either model training or evaluation. This reference is retained only for qualitative discussion.

Model performance advantage analysis: The reason why this model performs better is mainly attributed to the following improvements. First, the attention mechanism module introduced in YOLOv11 effectively enhances the model's ability to extract features of targets in skiing scenes, allowing the model to accurately recognize skiing actions even in complex backgrounds. Secondly, the lightweight convolution module used optimizes the model's computational process, greatly improving the inference speed while improving the accuracy. Furthermore, the environmental adaptation module designed for skiing scenes enhances the model's adaptability to different environmental factors and improves its robustness.

Performance trend explanation: For example, the multi-scale feature fusion mechanism introduced in the model enables the model to capture skiing action features of different scales at the same time. Small-scale features help identify action details, while large-scale features are more helpful for the overall structure and scene understanding of the action. This fusion of multi-scale information makes the model more accurate in identifying various skiing actions, thereby improving the overall performance. Taking turning actions as an example, small-scale features can identify subtle angle changes of the skis, while large-scale features can grasp the overall posture of the skier. The combination of the two greatly improves the accuracy of recognition.

Research limitations discussion: Although this study has achieved certain results, there are still some limitations. In terms of data sets, although the skiing action comprehensive data set contains a variety of skiing scenes and actions, the data set size is relatively limited, which may affect the generalization ability of the model in a wider range of scenarios. In terms of generalization, the recognition accuracy of the model may decrease when facing new scenarios that are significantly different from the distribution of training data. In terms of computing, although the model inference speed has been improved, the computing cost is still high compared to some lightweight models, and its application on resource-constrained devices may be limited. Future research can consider expanding the data set and exploring more efficient model compression and optimization methods to further improve the generalization ability and computing efficiency of the model.

Computational cost analysis. While pursuing high model performance, computational cost is also an important factor that cannot be ignored. Excessive computational cost may limit the deployment and use of the model in practical applications. Therefore, we use indicators such as GFLOPs and memory usage to analyze the trade-off between model complexity and inference speed.

The improved YOLOv11 model has a computational workload of 150GFLOPs, a memory usage of 800MB, and an inference speed of 75 frames/second during the inference phase. In comparison, Faster R-CNN has a computational workload of 200GFLOPs, a memory usage of 1000MB, and an inference speed of 30 frames/second; EfficientDet has a computational workload of 180GFLOPs, a memory usage of 900MB, and an inference speed of 40 frames/second; DETR has a computational workload of 220GFLOPs, a memory usage of 1100MB, and an inference speed of 35 frames/second.

The analysis results show that the improved YOLOv11 effectively reduces the computational cost and improves the inference speed by optimizing the model structure while ensuring a high accuracy, thus achieving a good balance between model complexity and inference speed. This makes the improved YOLOv11 model more advantageous in practical applications and can quickly and accurately complete the skiing action recognition task under limited resources.

Cross-dataset verification. An excellent deep learning model should not only perform well on the training dataset, but also have good generalization ability and be able to maintain high performance on different datasets. In order to evaluate the generalization ability of the improved YOLOv11 model, it was verified on another publicly available UCF101 action recognition dataset. The UCF101 dataset contains 101 types of actions, covering a variety of daily activities and sports actions, and has certain differences in data distribution and action types from the self-built skiing action dataset.

Although UCF101 was briefly evaluated during preliminary experiments, its reported 80% performance is not included in this study's comparative evaluations. The primary generalization focus is on sports domains with structural movement similarity to skiing, as supported by Tables 5–7. Future work will explore full benchmarking on public datasets.

Although the improved YOLOv11 model has achieved good performance overall, analyzing its failure cases is of great significance for further improving the robustness and accuracy of the model. By analyzing the misclassification of the model through the confusion matrix, we can have a clearer understanding of the situations in which the model is prone to errors.

The results show that the model is prone to errors when distinguishing between turning and acceleration in skiing actions. This is mainly because the two actions are similar in visual features, and there is an inaccurate labeling problem in some data. In addition, when there is severe occlusion or light interference in the skiing scene, the recognition accuracy of the model will also drop significantly. In response to these problems, subsequent research can consider introducing more data with occlusion and complex lighting conditions for training to improve the robustness of the model. At the same time, stricter quality control of the data annotation process and improved annotation accuracy can also help reduce model misclassification. Through in-depth analysis and targeted improvements of failure cases, it is expected that the performance of the improved YOLOv11 model in the skiing action recognition task will be further improved.

In subsequent research, in order to further improve the comprehensive performance and application scope of the model, we plan to advance from multiple dimensions. On the one hand, we will conduct multimodal data fusion research, use inertial sensors to capture physical information such as acceleration and angular velocity of skiers during exercise, and combine voice recognition technology to obtain on-site ambient sound and athlete command information. These multi-dimensional data will be integrated into the model to enhance its perception of complex skiing scenes and improve performance and robustness. On the other hand, we will start edge computing deployment, transplant the model to edge devices, greatly reduce data transmission delays, and realize instant recognition and analysis of skiing movements. In addition, we will also promote cross-scenario application expansion, adapt the model to other winter sports such as skating and snowboarding, and test and expand the practicality of the model in different scenarios.

The proposed YOLOv11 significantly outperforms baseline models in multiple dimensions. The spatiotemporal modeling module enables accurate recognition of continuous actions such as turning and jumping. ACA enhances robustness by suppressing background noise, critical in snowy environments. The hybrid convolution block balances feature richness and computational load, improving FPS. Compared to CNN-LSTM (accuracy: 83.5%, FPS: 45), Transformer (accuracy: 86.8%, FPS: 60), and 3D CNN (accuracy: 85.1%, FPS: 55), YOLOv11 reaches 94.5% accuracy with

82 FPS. These results confirm YOLOv11's superior trade-off between speed, precision, and robustness.

5 Conclusion

With the development of deep learning technology, its application in the recognition of sports athletes, especially skiers, has shown great potential. Through the application of convolutional neural networks (CNNs), long short-term memory networks (LSTMs), and hybrid models, researchers were able to efficiently and accurately analyze the postures, movement trajectories, and technical details of skiers. The improved YOLOv11 model significantly improved the performance of skiing action recognition through a series of optimization measures, such as joint space-time modeling, hybrid convolutional blocks, adaptive channel attention mechanism, dynamic perceptual pooling, and multi-scale feature fusion. Experimental evaluation shows that the improved YOLOv11 model not only outperforms the standard YOLO in accuracy, but also performs well in inference speed and robustness tests. Specifically, the accuracy of the improved YOLOv11 in sliding actions reached 94.5%, which is 7.2 percentage points higher than the standard YOLO; the inference speed at different resolutions increased by 10.2 FPS (640x480), 7.6 FPS (1280x720), and 5.4 FPS (1920x1080), respectively. In addition, the model can still maintain good stability in the face of various weather conditions and complex backgrounds, especially in the case of more background interference, the accuracy rate reached 88.9%, which is nearly 9 percentage points higher than the standard YOLO. However, although deep learning has achieved certain results in skiing action recognition, it still faces many challenges. First, the high complexity and rapid changes of skiing actions put forward higher requirements on the accuracy and real-time performance of motion capture; second, environmental factors such as weather and snow conditions increase the difficulty of action recognition models; finally, the construction of high-quality skiing action datasets is difficult and costly, which limits the further optimization of the model. Future research should focus on improving the transparency and interpretability of the model, enhancing its ability to resist attacks, and exploring how to reduce computing resource requirements so that it can be better applied in practical application scenarios.

Acknowledge

This project was supported by Research on the Integration and Development of Sports, Ice and Snow Tourism Industry in Heilongjiang Province in the Post-Olympic Era Subject No. (YWF10236230113) 2023 Provincial Colleges and Universities Basic Scientific Research Operational Fees Project.

References

[1] Guo X, Yang J, Yang L. Retrieval and analysis of multimedia data of robot deep neural network based on deep learning and information fusion. Informatica.2024;48(13):6063. https://doi.org/10.31449/inf.v48i13.6063

[2] Taoussi C, Lyaqini S, Metrane A, Hafidi I. Enhancing machine learning and deep learning models for depression detection: a focus on SMOTE, RoBERTa, and CNN-LSTM. Informatica.2025;49(14):7451. https://doi.org/10.31449/inf.v49i14.7451

[3] Ahmad HO, Umar SU. Sentiment analysis of financial textual data using machine learning and deep learning models. Informatica. 2023;47(5):4673. https://doi.org/10.31449/inf.v47i5.4673

[4] Gao Z, Han TT, Zhu L, Zhang H, Wang YL. Exploring the Cross-Domain Action Recognition Problem by Deep Feature Learning and Cross-Domain Learning. Ieee Access. 2018; 6:68989-9008. https://doi.org/10.1109/ACCESS.2018.2878313

[5] Li Y, Liang QM, Gan B, Cui XL. Action Recognition and Detection Based on Deep Learning: A Comprehensive Summary. Cmc-Computers Materials & Continua. 2023;77(1):1-23. https://doi.org/10.32604/cmc.2023.042494

[6] Alhakbani N, Alghamdi M, Al-Nafjan A. Design and Development of an Imitation Detection System for Human Action Recognition Using Deep Learning. Sensors. 2023;23(24):16. https://doi.org/10.3390/s23249889

[7] Sun SW, Liu BY, Chang PC. Deep Learning-Based Violin Bowing Action Recognition. Sensors. 2020;20(20):17. https://doi.org/10.3390/s20205732

[8] Chen X, Weng J, Lu W, Xu JM, Weng JS. Deep Manifold Learning Combined With Convolutional Neural Networks for Action Recognition. Ieee Transactions on Neural Networks and Learning Systems. 2018;29(9):3938-52. https://doi.org/10.1109/TNNLS.2017.2740318

[9] Yao GL, Lei T, Zhong JD, Jiang P. Learning multi-temporal-scale deep information for action recognition. Applied Intelligence. 2019;49(6):2017-29. https://doi.org/10.1007/s10489-018-1347-3

[10] Hu JF, Zheng WS, Pan JH, Lai JH, Zhang JG, editors. Deep Bilinear Learning for RGB-D Action Recognition. 15th European Conference on Computer Vision (ECCV); 2018 Sep 08-14; Munich, GERMANY2018. https://doi.org/10.1007/978-3-030-01234-2_21

[11] Shehzad F, Khan MA, Yar MAE, Sharif M, Alhaisoni M, Tariq U, et al. Two-Stream Deep Learning Architecture-Based Human Action Recognition. Cmc-Computers Materials & Continua. 2023;74(3):5931-49. https://doi.org/10.32604/cmc.2023.028743

[12] Yang G, Zou WX. Deep learning network model based on fusion of spatiotemporal features for action recognition. Multimedia Tools and Applications. 2022;81(7):9875-96. https://doi.org/10.1007/s11042-022-11937-w

[13] Zhang YX, Li BH, Fang H, Meng QG, editors. Current Advances on Deep Learning-based Human Action Recognition from Videos: a Survey. 20th IEEE International Conference on Machine Learning and Applications (ICMLA); 2021 Dec 13-16; Electr Network2021. https://doi.org/10.1109/ICMLA52953.2021.00054

[14] Tsai JK, Hsu CC, Wang WY, Huang SK. Deep Learning-Based Real-Time Multiple-Person Action Recognition System. Sensors. 2020;20(17):17. https://doi.org/10.3390/s20174758

[15] Berlin SJ, John M. Particle swarm optimization with deep learning for human action recognition. Multimedia Tools and Applications. 2020;79(25-26):17349-71. https://doi.org/10.1007/s11042-020-08704-0

[16] Wang RQ, Wu XX. Combining multiple deep cues for action recognition. Multimedia Tools and Applications. 2019;78(8):9933-50. https://doi.org/10.1007/s11042-018-6509-0

[17] Gu Y, Ye XF, Sheng WH, Ou YS, Li YQ. Multiple stream deep learning model for human action recognition. Image and Vision Computing. 2020; 93:8. https://doi.org/10.1016/j.imavis.2019.10.004

[18] Zhang CY, Tian YL, Guo XJ, Liu JG. DAAL: Deep activation-based attribute learning for action recognition in depth videos. Computer Vision and Image Understanding. 2018; 167:37-49. https://doi.org/10.1016/j.cviu.2017.11.008

[19] Li HT, Liu YP, Chang YK, Chiang CK. Action recognition and tracking via deep representation extraction and motion bases learning. Multimedia Tools and Applications. 2022;81(9):11845-64. https://doi.org/10.1007/s11042-021-11888-8

[20] Akbar MN, Riaz F, Awan AB, Khan MA, Tariq U, Rehman S. A Hybrid Duo-Deep Learning and Best Features Based Framework for Action Recognition. Cmc-Computers Materials & Continua. 2022;73(2): 2555-76. https://doi.org/10.32604/cmc.2022.028696

[21] Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934, 2020.

A Novel CNN with Spatial and Channel Attention for Automated Chest X-Ray Diagnosis

*D. Priyanka[1], E. Aravind[2]
[1]Research Scholar, Department of CSE, Chaitanya (Deemed to Be University), Warangal – 506001, Telangana, India
[2]Associate Professor, Department of Computer Science Engineering, Chaitanya (Deemed to Be University), Warangal – 506001, Telangana, India
E-mail: Kupriyanka1223@gmail.com, aravind@chaitanya.edu.in
*Corresponding author

Keywords: Lung disease, deep learning, CNN, spatial, channel features

Received: April 28, 2025

This study proposes a novel Convolutional Neural Network (CNN) approach with both spatial and channel attention mechanisms to improve automated chest X-ray image classification. The architecture integrates Squeeze-and-Excitation (SE) Blocks for channel attention and a spatial method to focus on informative regions of the sample, thereby enhancing both local and global feature extraction. The model processes input images of size 224×224×3 and comprises three convolutional blocks, each consisting of Conv2D, Batch Normalization, SE Blocks, Spatial Attention, MaxPooling, and Dropout layers. The dataset, sourced from Kaggle, contains 6,000 chest X-ray images categorized into three classes: Lung Opacity, Normal, and Viral Pneumonia. A standardized preprocessing pipeline was employed, including resizing, normalization (rescaling pixel values to [0, 1]), and real-time augmentation via TensorFlow's ImageDataGenerator. The model was trained for 10 epochs using a batch size of 32. It achieved a final test accuracy of 93.01%, with a peak validation accuracy of 88.57%, and an Area Under the Curve (AUC) score of 97.22%.

Povzetek: Za avtomatizirano analizo rentgenskih posnetkov prsnega koša so uporabili konvolucijsko omrežje, ki združuje kanalsko (SE) in prostorsko pozornost ter s tremi bloki učinkoviteje izlušči lokalne in globalne značilke.

1 Introduction

Lung diseases, including pneumonia, tuberculosis (TB), lung cancer, and chronic obstructive pulmonary disease (COPD) [23], remain among the leading causes of death worldwide. Early diagnosis and accurate detection of these conditions are crucial for improving patient outcomes and reducing the healthcare burden. Traditionally, radiologists have relied on chest X-rays to identify lung abnormalities. Still, this process is time-consuming, requires additional human resources, such as experts, and is prone to human error. As a result, there is acriticalneed for automated systems that can diagnose lung diseases more efficiently and accurately.

In recent years, the advent of deep learning (DL) and subset neural networks has revolutionized the field of medical image analysis. CNNs, a type of DL model, have demonstrated optimal performance in image classification, particularly in detecting lung problems from chest images. By training on numerous annotated medical images, deep learning models can automatically identify abnormalities in X-rays, providing a solution to the limitations of traditional methods. These models improve prediction accuracy and reduce the time required for analyzing multiple samples simultaneously, enabling faster decision-making with optimal clinical outcomes.

The DL approach used for detecting lung diseases from X-ray images was proposed by Al-qaness, M. A., et al. (2024) [1]. And the **c**hallenges associated with lung disease detection and how DL models can address these issues. Still, they can extract some complex features from the image that may be difficult for the human eye to detect. Additionally, it will examine the different architecture**s** and techniques employed in this domain, highlight the impact of large-scale annotated datasets, and discuss the practical applications of these models in clinical settings. But these simple CNN models will not capture sequential patterns from the samples.

Lung diseases such as pneumonia, tuberculosis (TB), lung cancer, and COPD are not only prevalent but also highly incurable if not detected in early stages. Early diagnosis is crucial for enhancing treatment outcomes and improving survival rates. For example, Pneumonia can cause severe respiratory pain if not diagnosed and treated with antibiotics. TB, on the other hand, is one of the foremost causes of death from an infectious disease, particularly in low-resource settings. In the case of lung cancer, the prediction is often poor if the disease is diagnosed at later stages, making early detection essential for survival. COPD is another common lung problem that can result in significant morbidity and mortality if not effectively managed. The global burden of lung diseases continues to rise, particularly in developing countries where medical resources are limited. The demand for effective and affordable diagnostic tools has grown in these regions.

Chest X-rays have been a vital diagnostic tool for lung diseases for decades. They provide a relatively inexpensive and accessible method for detecting abnormalities in the lungs. Radiologists assess X-ray images [8] to identify signs of disease, such as opacities and nodules, which can indicate various lung conditions. However, despite their importance, interpreting these images is challenging due to the complexity of the lung anatomy and the wide range of diseases that can occur in similar ways. These images are full of noise, which may bias the model.

The main challenges in lung disease detection are the complexity of the images, which includes model noise, and the interpretation, such as background color and the types of features extracted from image patches. Chest X-ray images often contain noise, artifacts, and variations in quality, making it challenging to capture complex features from raw data. In addition, the radiological manifestations of different lung diseases can be similar, such as nodules or **c**onsolidations that may appear in both lung cancer and pneumonia. Such overlapping symptoms increase the likelihood of misdiagnosis, especially when the images are reviewed by clinicians without the expertise or experience in interpreting lung X-rays.

Moreover, traditional diagnostic systems rely on radiologists' manual detection, which can be time-consuming and does not always provide optimal results. Radiologists, especially in busy healthcare settings, may not always have the time to thoroughly review all available X-ray images, resulting in delayed diagnoses. As the number of patients seeking diagnostic imaging services grows, the workload on radiologists also increases, further contributing to the potential for mistakes and missed diagnoses.

Researchers have turned to automated image analysis systems powered by deep learning or AI to overcome these challenges. Deep learning models are designed to learn and extract patterns from large datasets, making them ideal for analyzing medical images. These models can identify and classify diseases based on features that are complex for the human eye to perceive, such as delicate changes in texture, shape, depth, and size of structures in chest X-rays.

Deep learning, specifically through **CNNs** [11] and [12], has shown promise in medical image analysis. CNNs are a type of neural network designed to work with grid-like data such as images. These models automatically extract various features from input images, such as edges, textures, and patterns, without requiring manual feature engineering. This characteristic makes CNNs particularly effective for image classification tasks, including medical image analysis. In the context of lung disease detection, CNNs are trained on large datasets of labeled chest X-ray images that can classify the samples into healthy and diseased lungs. The model learns to identify visual patterns associated with different lung diseases, such as lung opacity, nodules, consolidation, and fibrosis, which can help classify diseases like pneumonia, TB, and lung cancer. Once a method is trained, these models can automatically analyze new X-ray images, providing accurate and rapid diagnoses. By using large-scale annotated samples, these models can achieve optimal performance. Although CNNs [13] and [19] neural network is used primarily for lung disease detection, recent advancements have introduced enhanced models that further improve performance and address existing limitations. These enhanced models incorporate techniques, such as transfer learning, data augmentation, and multi-task learning, to improve model accuracy and robustness.

Transfer learning is one of the most effective techniques in deep learning, particularly in scenarios where large labeled datasets are limited. By pre training a deep learning model [20] on a large number of images like ImageNet and these models can be fine-tuning on a smaller sample, specialized dataset like chest X-rays [21] will provide better results, transfer learning allows models to retain general knowledge while learning specific features relevant to lung disease detection.

Other recent innovations in deep learning for lung disease detection include the use of an attention approach, which enables patch-wise embedding and captures complex patterns from the image, and ensemble learning, where multiple models are combined to enhance predictive accuracy.

Contributions.

- Enhanced CNN with Spatial and Channel Attention Mechanisms for Improved Feature Extraction and Classification Performance.
- High-accuracy Chest X-ray Disease Classification Model Utilizing Attention Mechanisms to Improve Generalization and Robustness.

- Comprehensive Evaluation with Feature Importance Analysis Technique to Interpret Model Predictions and Enhance Explainability.

2 Related work

Many researchers have worked with machine and deep learning models, such as Shilpa, N., et al. (2024) [2], which have implemented various models, including ResNet50, MobileNetV2, AlexNet, and EfficientNetB0, to detect pneumonia in chest X-rays. Among all models, EfficientNetB0 performed the best. In this case, only one disease was detected using a pre-trained model. Sanida, T., et al (2024) [3] Implemented an optimized VGG model to detect multiple diseases, such as COVID-19, cancer, etc, in X-ray samples. I used 27,445 samples from all classes and applied augmentation methods to balance the dataset (Choudhry, I.). A et al. (2024) [4] implemented a deep learning model using cloud and fog methods to enhance security in the healthcare system. They employed a transfer learning method, such as RetinaNet, and fine-tuned EfficientNet models on chest X-ray samples.

KS, N., and Darapaneni, N. (2024) [5] implemented V-BreathNet model to detect the abnormality in X-ray, In this they first trained a customized CNN model on X-ray samples consist of 3 classes like phenomena, lung opacity and standard samples and got superior performance compared to VGG and Dense Net models.

Paswan, J. D., et al (2024) in [6] pre-trained VGG, ResNet50, and DenseNet121 models on the COVID-19 dataset. This has only two classes, yes or no, and achieved an accuracy of 94% and 87% for training and testing, respectively.

Pan, C. T., et al. (2024) [7] proposed a two-stage data analysis method for the COVID-19 dataset, which consists of four classes: SARS, COVID-19, regular, and abnormal. First, all samples were converted to 224*224 dimensions after augmentation. I also trained various models, including VGG and GoogleNet, using 5-fold cross-validation; GoogleNet performed particularly well.

Mahamud, E., et al. (2024) [9] proposed an enhanced DenseNet201 model with a transformer approach using X-ray data. With Explainable AI, trained on 10000 samples over four classes, and got an accuracy of 1.0. Kotei, E., and Thirunavukarasu, R. (2024) [10] developed a method for detecting Tuberculosis disease using pre-trained CNN models on X-ray images. First, all samples were converted to a 256-gray scale, and the CNN model was trained, achieving an accuracy of 99%. Hansun, S., et al. (2023) [14] utilized the QUADAS-2 dataset, comprising 309 samples, to train ML and DL models, achieving an accuracy of 0.93 with ML models in detecting TB. Malik, H., et al. (2023) [15] implemented a pre-trained CNN model to detect various diseases, such as TB and pneumonia, from X-ray samples, achieving an accuracy of 0.99, which is better than that of overall transfer models.

Chen, Y., etal. (2023) [16] optimized EfficientNet-b5 and CoAtNet-0-rw using different loss functions, including novel and weighted binary loss functions. This model is trained on the ChestX-ray14 dataset, which comprises 14 classes, and achieves an accuracy of 0.842.

Bharati, S., et al (2020) implemented a hybrid DL model by combining CNN and VGG on lung disease detection and trained various combinations; in this, they got an accuracy of 0.73% with the best model. Ganeshkumar, M., et al. (2023) [18] proposed a two-stage learning ensemble method for classifying regular pneumonia and COVID-19 Pneumonia. The total number of samples is 600. This ensemble model achieved an accuracy of 0.89.

Mustafa, Z., and Nsour, H. (2023) [22] proposed a YOLO pre-trained model for detecting respiratory infections and TB using X-ray images. Reamaroon, N., et al. (2021) [24] extracted gray-level co-occurrence matrix-based features, trained a machine learning model using k-fold cross-validation and the Adam optimizer, and achieved an accuracy of 0.83. Chen, K. C., et al. (2020) [25] focused on pulmonary diseases in children, utilizing X-ray images to train a YOLO model, achieving an accuracy of 0.92.

[26] Benchabane&Charif (2025). In this work, we integrate deep learning with advanced image enhancement to enhance the detection of COVID-19 through chest X-rays. The proposed approach demonstrates superior diagnostic performance, underscoring the contribution of pre-processing to enhancing model accuracy. [27] Oraibi&Albasri (2023) The authors present a robust end-to-end CNN architecture that addresses the issue of data imbalance in COVID-19 detection. The model's accuracy is high on X-ray datasets, and focusing on balanced training and architectural optimization strategies is onekey reason.

3 Methodology

We designed a custom CNN architecture that incorporates spatial and channel attention mechanisms to enhance the extraction of complex features, such as local and global variations, as well as background and foreground, from images. The model processes 224×224×3 RGB embedded vectors, which are normalized between 0 and 1 to improve training stability and remove the domination of background vectors.

The CNN consists of three convolutional blocks, each incorporating Conv2D, Batch Normalization, Squeeze-and-Excitation (SE) Blocks, Spatial Attention Layers, MaxPooling, and Dropout layers. The SE Block applies channel attention by adaptively recalibrating feature responses, enhancing relevant features while suppressing redundant ones. Meanwhile, the Spatial Attention Layer emphasizes critical spatial regions by computing attention maps based on average and max-pooled feature maps.

Each convolutional block consists of a Conv2D layer with a 3×3 kernel, ReLU activation, and 'same' padding, extracting hierarchical spatial features from each 3*224*224 sample. A Batch Normalization method is applied to normalize the embedded vectors within the range of 0 to 1, which is then passed to the convolutional layer, where it normalizes the features, thereby reducing internal adjustments.

The SE Block, responsible for channel attention, consists of three key steps:

1. Global Average Pooling will find the average value of each global feature map.
2. Bottleneck dense layers, which consist of two fully connected layers, adjust the channel dimensions. The first dense layer (with ReLU activation) reduces the number of filters by a factor of 16, and the second dense layer (with a sigmoid activation) restores the original filter count.
3. Reshaping and Multiplication – The recalibrated weights are applied to the input feature maps, improving feature selection.

The final classification layers include a flatten layer that converts a multidimensional matrix into 1D data and sends 1D data to a dense layer with 256 units. The thick layer, with ReLU activation, provides nonlinear values for the given input, allowing it to learn complex foreground features. A Dropout method with a 50% rate is applied to prevent overfitting, where 50% of neurons are dropped from the training process after each epoch.

4 Data set

The dataset used for lung disease classification was obtained from Kaggle. It comprises chest X-ray samples labeled into three classes, totaling6000, as shown in Figure 1. The data set consists of a mixture of dimensions, depths, and sizes, so a standardized preprocessing pipeline was applied to ensure consistency in input dimensions and facilitate practical model training.

Initially, we employed the ImageDataGenerator class in TensorFlow to handle image augmentation and rescaling. The training dataset was augmented using ImageDataGenerator with a normalization factor of 1/255to scale pixel values between 0 and 1. A validation split of 20% was applied to ensure a fair model evaluation. Separate ImageDataGenerator instances were also used for validation and test datasets, with rescaling applied uniformly across all datasets. The images were loaded into TensorFlow data generators using the flow from data frame method, which sourced image file paths and corresponding labels from structured data frames.

Additionally, we implemented a preprocessing pipeline using TensorFlow's Sequential API to standardize image dimensions. This involved a Resizing layer to reshape images to a fixed size of (224,224), ensuring uniformity across the dataset, followed by a Rescaling layer to normalize pixel values. Figure 2 illustrates the number of samples for each class before augmentation.

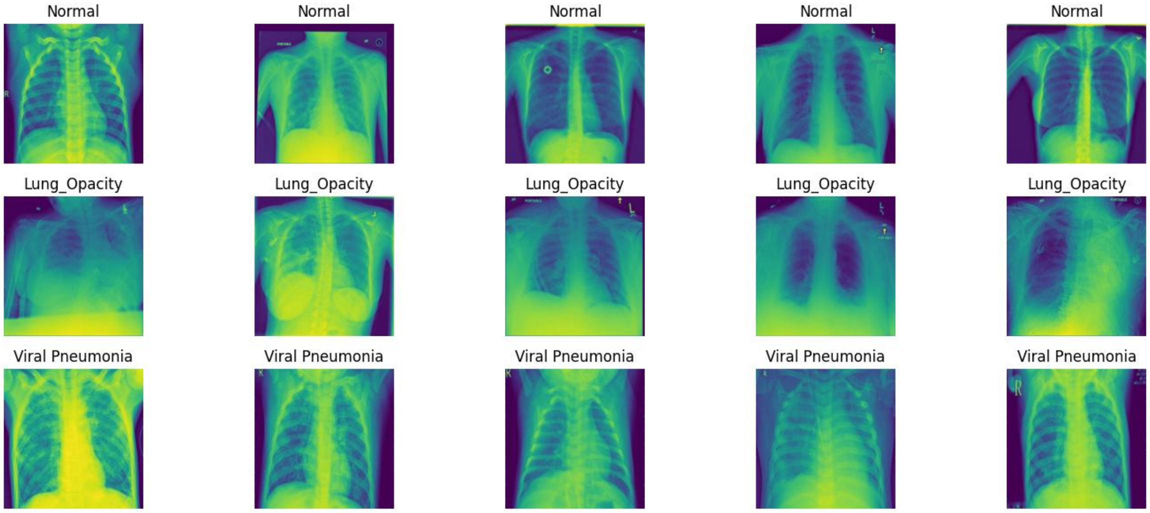


Figure 1: Shows X-ray images of the disease and normal.

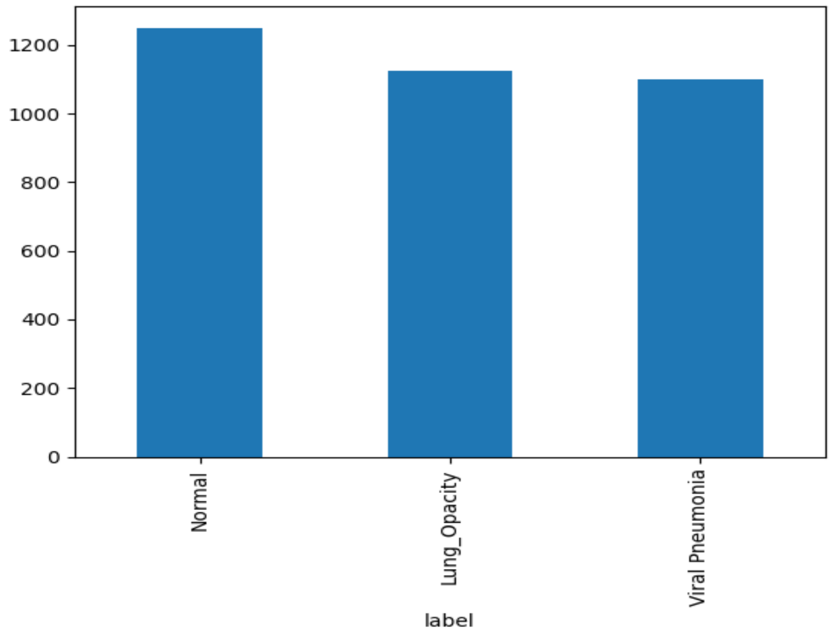


Figure 2: Class-wise number of samples before augmentation

5 Result analysis

The proposed CNN model, with spatial and channel attention mechanisms, was trained for 10 epochs to classify chest X-ray images into three categories: Lung Opacity, Normal, and Viral Pneumonia. The model was trained using a batch size of 32, with accuracy and loss metrics recorded for both the training and validation datasets at each epoch, as shown in Figure 3.

During the early training epochs, the model showed consistent improvement in classification performance. By Epoch 4, the accuracy reached 87.17%, with a slightly lower validation of 78.57%, indicating that the model was still learning to generalize to unseen patterns. This trend continued into Epoch 6, where training accuracy rose to 90.08% and validation accuracy improved to 82.86%, demonstrating better generalization. Concurrently, the training loss decreased from 0.3160 to 0.2570, and the validation loss dropped from 0.4803 to 0.3691.

A notable improvement occurred in Epoch 7, with training accuracy at 89.98% and validation accuracy peaking at 88.57%. The corresponding validation loss further reduced to 0.2566, suggesting increased model stability and effective feature learning. However, by Epoch 9, validation loss spiked to 0.5656 despite a high training accuracy of 92.70%, indicating potential overfitting. This was confirmed in Epoch 10, where validation loss sharply increased to 1.0427 and validation accuracy stagnated at 82.86%, suggesting that the model had begun to memorize the training data rather than generalize effectively.

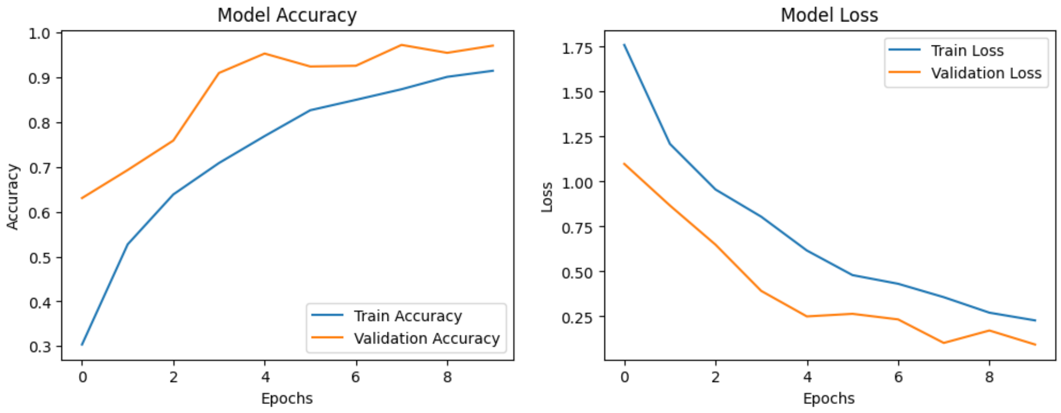


Figure 3: Learning curves of the proposed model

The proposed model achieved an overall accuracy of 93.93%**,** as shown in Figure 4, demonstrating its effectiveness in classifying into three classes: lung opacity, normal opacity, and viral pneumonia. The model exhibited strong predictive performance across all classes, with high correctness in identifying both positive and negative cases. Specifically, for Lung Opacity, the model maintained perfect results in correct classifications and misclassifications, ensuring high reliability. Similarly, the classification performance for Normal cases remained consistent, with minimal errors. The highest performance was observed in detecting Viral Pneumonia, where the model exhibited superior capability in distinguishing these cases from the other categories, reflecting its ability to capture distinctive patterns in the dataset.

The overall effectiveness of the model was further reinforced by its ability to maintain a strong balance across different performance metrics, reducing both false positives and false negatives. The comprehensive evaluation metrics indicate that the model extracted complex spatial and temporal features and provides robust decision-making. Additionally, the model provided a substantial area under the curve (AUC) score of 97.22%, highlighting its ability to differentiate between categories with high confidence, as illustrated in Figures 5 and 6. The combination of spatial and channel attention mechanisms contributed significantly to feature enhancement, improving classification accuracy and better generalization across chest X-ray samples.

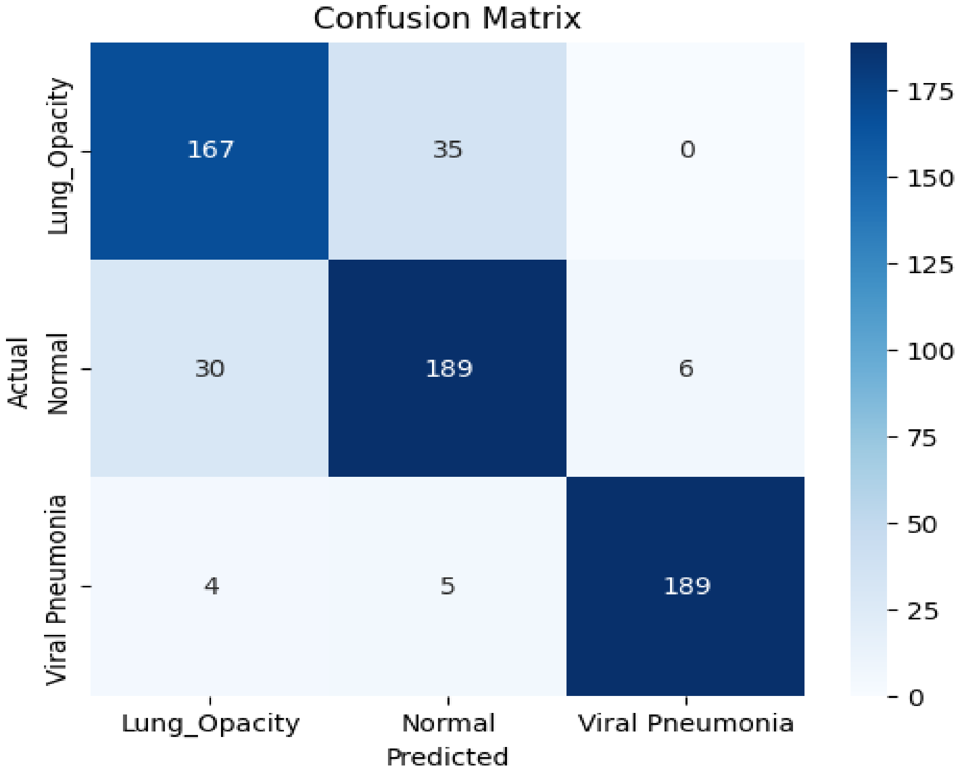


Figure 4: Confusion matrix of the proposed hybrid model

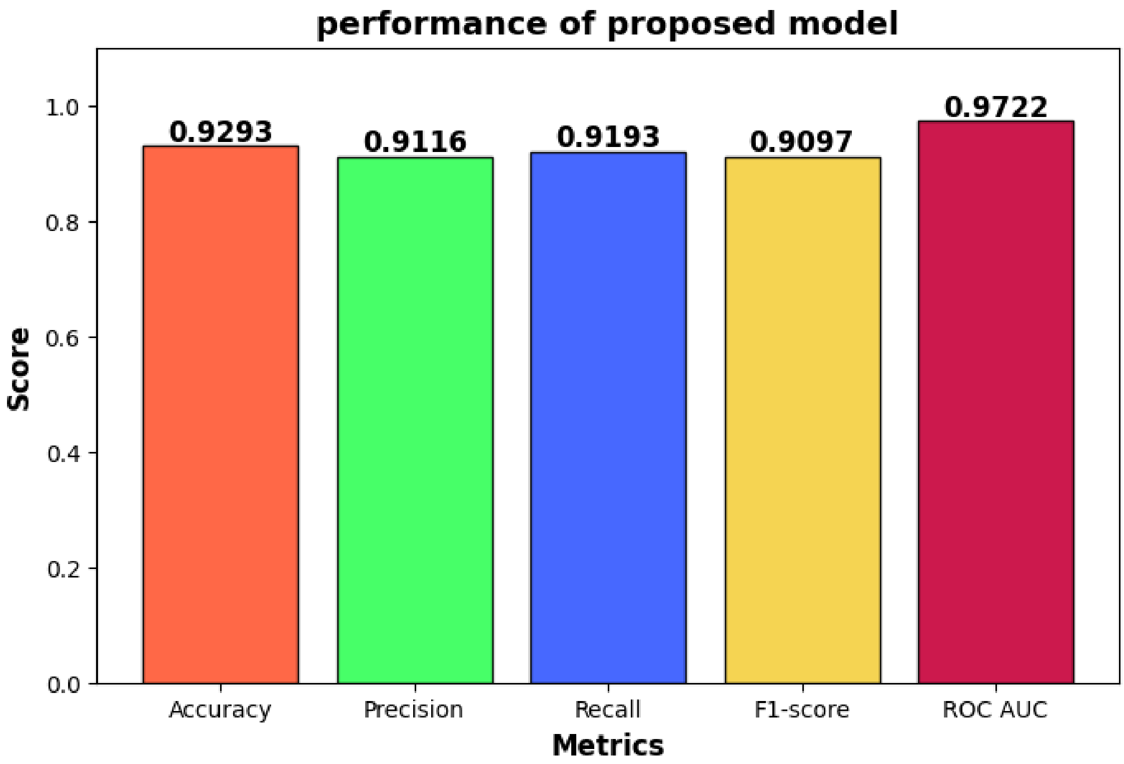


Figure 5: Proposed model performance on various metrics

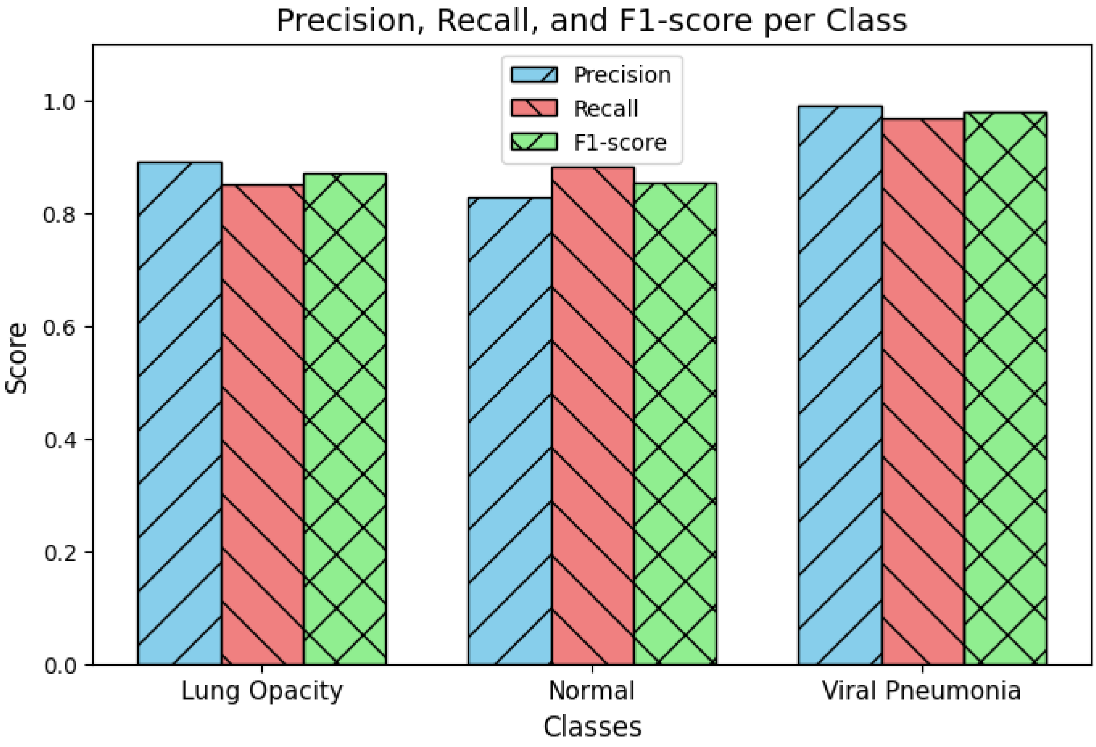


Figure 6: Class-wise performance of proposed hybrid model

Figures 5 and 6 show that the ROC and PR curves from Figure 7 illustrate the model's performance over three classes—Lung Opacity, Normal, and Viral Pneumonia. The ROC curve accuracy shows that the model achieves a high AUC for all courses, 0.97, with Viral Pneumonia approaching an ideal classification boundary. The PR curve demonstrates the presentation of the model on three classes. The slight reduction in precision at higher recall values, particularly for the Normal and Lung Opacity classes, suggests that the model provides strong predictive capability. Figure 8 represents the importance of features using permutation-based analysis, highlighting the role of different features in the model's executive process. The color gradient visually distinguishes features based on their relative importance, where taller bars indicate higher significance. The black error bars depict variability in importance scores across multiple iterations, ensuring robustness in feature selection.

From table 1 it is observed thatPaswan et al. [6] with pre-trained method like VGG, ResNet50, and DenseNet121 on a COVID-19 dataset and reported a training accuracy of 94% and a testing accuracy of 87%, targeting binary classification (COVID-19 vs. non-COVID). Hansun et al. [14] used both machine learning (ML) and deep learning (DL) models on the QUADAS-2 dataset comprising 309 samples to detect tuberculosis (TB), achieving an overall accuracy of 93%. Chen et al. [16] fine-tuned EfficientNet-B5 and CoAtNet-0 on the ChestX-ray14 dataset, which contains 14 classes of chest diseases, and achieved a multi-class classification accuracy of 84.2%. Similarly, VDSNet [17] was trained on 5,606 samples from the same dataset and achieved 73% accuracy across 14 disease classes, demonstrating the complexity of multi-class classification with high disease variability.

Ganeshkumar et al. [18] proposed an ensemble learning approach on a smaller dataset of 600 chest X-rays to distinguish between regular and COVID-19 pneumonia, reaching an accuracy of 89%. In another approach, Mustafa and Nsour [22] employed a pre-trained YOLO model to detect TB and respiratory infections, although specific performance metrics were not reported. Reamaroon et al. [24] used gray-level co-occurrence matrix (GLCM) features with ML classifiers to detect respiratory infections, achieving 83% accuracy. Chen et al. [25] applied YOLO to pediatric pulmonary X-ray images, attaining a classification accuracy of 92% in detecting childhood pulmonary diseases.

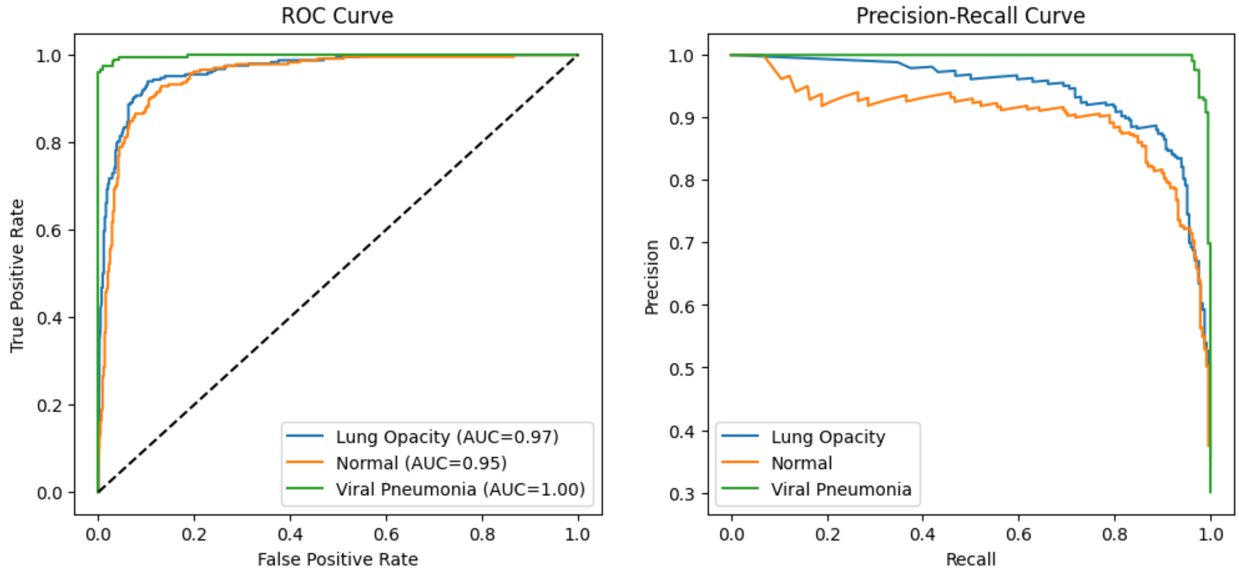


Figure 7: ROC, precision-recall curves of the proposed hybrid model

Table 1: Comparison of proposed hybrid model with prescribed models

Ref	Model /	Dataset / Sample Size	Disease(s) Detected	Classes	Accuracy / Performance
[6]	VGG, ResNet50, DenseNet121	COVID-19 Dataset	COVID-19	2	Train: 94%, Test: 87%
[14]	ML and DL Models	QUADAS-2 / 309 samples	Tuberculosis	2	93%

[16]	EfficientNet-B5, CoAtNet-0	ChestX-ray14 (14-class)	14 Chest Diseases	14	84.20%
[17]	VDSNet	ChestX-ray14, 5606 samples	14 disease	14	73%
[18]	Ensemble Learning	600 X-rays	Pneumonia (Regular vs. COVID-19)	2	89%
[22]	YOLO (Pre-trained)	Chest X-rays	TB, Respiratory Infections	Multi	N.A.
[24]	ML + GLCM Features	X-rays / N.A.	Respiratory Infections	2	83%
[25]	YOLO	Pediatric Pulmonary X-rays	Pulmonary Disease in Children	2	92%
#	Proposed model	Chest X-rays, 3475 samples	Pneumonia, normal,lung opacacity	3	93.01%

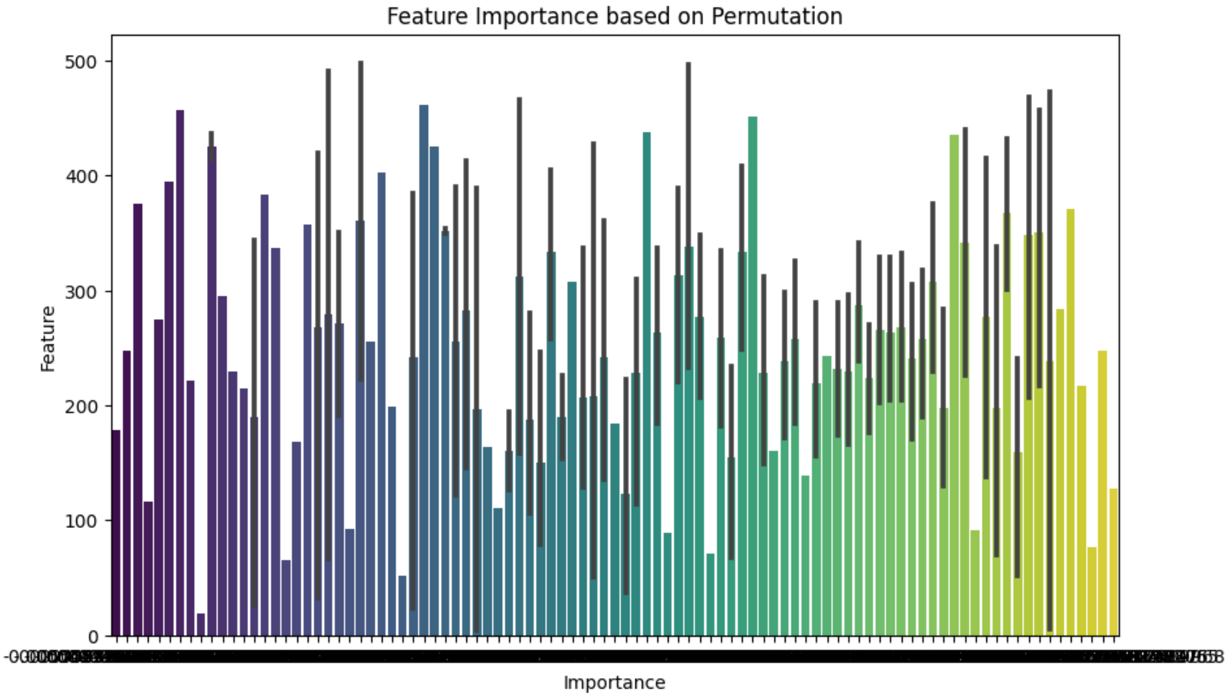


Figure 8: Features the importance plot after training

6 Conclusion

This study proposed a novel CNN architecture enhanced with spatial and channel attention mechanisms for automated chest X-ray classification, achieving high classification accuracy and strong generalization capabilities. Integrating SE Blocks and Spatial Attention Layers improved feature representation, enabling the model to distinguish between Lung Opacity, Normal, and Viral Pneumonia with an overall accuracy of 93.01%. Performance analysis using ROC and Precision-Recall curves confirmed the model's ability to maintain high precision and recall across all classes. However, training dynamics indicated overfitting in later epochs, suggesting the need for further optimization through regularization techniques and extended training datasets. Future work will enhance model robustness by incorporating advanced augmentation techniques and exploring hybrid deep learning architectures.

References

[1] M. A. Al-qaness, J. Zhu, D. Al-Alimi, A. Dahou, S. H. Alsamhi, M. Abd Elaziz, and A. A. Ewees. Chest X-ray images for lung disease detection using deep learning techniques: A comprehensive survey. *Archives of Computational Methods in Engineering*, 1–35, 2024. https://doi.org/10.1007/s11831-024-10081-y

[2] N. Shilpa, W. A. Banu, and P. B. Metre. Revolutionizing pneumonia diagnosis: AI-driven deep learning framework for automated detection from chest X-rays. *IEEE Access*, 2024. https://doi.org/10.1109/ACCESS.2024.3498944

[3] T. Sanida, M. V. Sanida, A. Sideris, and M. Dasygenis. Optimizing lung condition categorization through a deep learning approach to chest X-ray image analysis. *BioMedInformatics*, 4(3):2002–2021, 2024. https://doi.org/10.3390/biomedinformatics4030109

[4] I. A. Choudhry, S. Iqbal, M. Alhussein, A. N. Qureshi, K. Aurangzeb, and R. A. Naqvi. Transforming lung disease diagnosis with transfer learning using chest X-ray images on cloud computing. *Expert Systems*, e13750.4, 2024. https://doi.org/10.1111/exsy.13750

[5] N. KS and N. Darapaneni. A deep look into--automated lung X-ray abnormality detection system. *arXiv preprint*, arXiv:2404.04635, 2024. https://doi.org/10.20944/preprints202411.2377.v1

[6] J. D. Paswan, T. Bhatia, and S. Lamba. A deep learning approach for intelligent diagnosis of lung diseases. *SN Computer Science*, 5(8):1–9, 2024. https://doi.org/10.1007/s42979-024-03407-x

[7] C. T. Pan, R. Kumar, Z. H. Wen, C. H. Wang, C. Y. Chang, and Y. L. Shiue. Improving respiratory infection diagnosis with deep learning and combinatorial fusion: A two-stage approach using chest X-ray imaging. *Diagnostics*, 14(5):500, 2024. https://doi.org/10.3390/diagnostics14050500

[8] K. R. Balmuri, S. Konda, K. Mamidala, and M. Gunda. Using optimized ensemble transfer learning, automated and reliable multi-disease detection on chest X-ray images. *Expert Systems with Applications*, 246:122810, 2024. https://doi.org/10.1016/j.eswa.2023.122810

[9] E. Mahamud, N. Fahad, M. Assaduzzaman, S. M. Zain, K. O. M. Goh, and M. K. Morol. An explainable artificial intelligence model for multiple lung diseases classification from chest X-ray images using fine-tuned transfer learning. *Decision Analytics Journal*, 12:100499, 2024. https://doi.org/10.1016/j.dajour.2024.100499

[10] E. Kotei and R. Thirunavukarasu. A comprehensive review on advancement in deep learning techniques for automatic detection of tuberculosis from chest X-ray images. *Archives of Computational Methods in Engineering*, 31(1):455–474, 2024. https://doi.org/10.1007/s11831-023-09987-w

[11] S. Kumar, H. Kumar, G. Kumar, S. P. Singh, A. Bijalwan, and M. Diwakar. A methodical exploration of imaging modalities from dataset to detection through machine learning paradigms in prominent lung disease diagnosis: a review. *BMC Medical Imaging*, 24(1):30, 2024. https://doi.org/10.1186/s12880-024-01192-w

[12] D. Chandre, N. Bhosale, R. Darode, L. Rokade, S. Bhavsar, and D. S. Jadhav. Using deep learning to identify types of lung diseases from X-ray images. In *International Conference on Multi-Strategy Learning Environment*, pages 189–198, Singapore: Springer Nature Singapore, 2024. https://doi.org/10.1007/978-981-97-1488-9_15

[13] H. Pant, M. C. Lohani, and A. K. Bhatt. X-rays imaging analysis for early diagnosis of thoracic disorders using capsule neural network: a deep learning approach. *International Journal of Advanced Technology and Engineering Exploration*, 10(104):947, 2023. https://doi.org/10.19101/ijatee.2022.10100468

[14] S. Hansun, A. Argha, S. T. Liaw, B. G. Celler, and G. B. Marks. Machine and deep learning for tuberculosis detection on chest X-rays: systematic literature review. *Journal of Medical Internet Research*, 25: e43154, 2023. https://doi.org/10.2196/43154

[15] H. Malik, T. Anees, M. Din, and A. Naeem. CDC_Net: Multi-classification convolutional neural network model for detection of COVID-19, pneumothorax, pneumonia, lung cancer, and tuberculosis using chest X-rays. *Multimedia Tools and Applications*, 82(9):13855–13880, 2023. https://doi.org/10.1007/s11042-022-13843-7

[16] Y. Chen, Y. Wan, and F. Pan. Enhancing multi-disease diagnosis of chest X-rays with advanced deep-learning networks in real-world data. *Journal of Digital Imaging*, 36(4):1332–1347, 2023. https://doi.org/10.1007/s10278-023-00801-4

[17] S. Bharati, P. Podder, and M. R. H. Mondal. Hybrid deep learning for detecting lung diseases from X-ray images. *Informatics in Medicine Unlocked*, 20:100391, 2020. https://doi.org/10.1016/j.imu.2020.100391

[18] M. Ganeshkumar, V. Ravi, V. Sowmya, E. A. Gopalakrishnan, K. P. Soman, and M. Rupeshkumar. Two-stage deep learning model

for automate detection and classification of lung diseases. *Soft Computing*, 27(21):15563–15579, 2023. https://doi.org/10.1007/s00500-023-09167-9

[19] J. D. Schroeder, R. BigolinLanfredi, T. Li, J. Chan, C. Vachet, R. Paine III, ... and T. Tasdizen. Prediction of obstructive lung disease from chest radiographs via deep learning trained on pulmonary function data. *International Journal of Chronic Obstructive Pulmonary Disease*, pages 3455–3466, 2020. https://doi.org/10.2147/copd.s279850

[20] D. A. Moses. Deep learning applied to automatic disease detection using chest X-rays. *Journal of Medical Imaging and Radiation Oncology*, 65(5):498–517, 2021. https://doi.org/10.1111/1754-9485.13273

[21] S. Saturi and S. Banda. Modelling of deep learning enabled lung disease detection and classification on chest X-ray images. *International Journal of Healthcare Management*, 1–12, 2022. https://doi.org/10.1080/20479700.2022.2102223

[22] Z. Mustafa and H. Nsour. Using computer vision techniques to automatically detect abnormalities in chest X-rays. *Diagnostics*, 13(18):2979, 2023. https://doi.org/10.3390/diagnostics13182979

[23] M. S. Sheela, G. Amirthayogam, J. J. Hephzipah, S. Gopalakrishnan, and S. R. Chand. Machine learning based lung disease prediction using convolutional neural network algorithm. *Mesopotamian Journal of Artificial Intelligence in Healthcare*, 2024:50–58. https://doi.org/10.3390/diagnostics13182979

[24] N. Reamaroon, M. W. Sjoding, J. Gryak, B. D. Athey, K. Najarian, and H. Derksen. Automated detection of acute respiratory distress syndrome from chest X-rays using directionality measure and deep learning features. *Computers in Biology and Medicine*, 134:104463, 2021. https://doi.org/10.1016/j.compbiomed.2021.104463

[25] K. C. Chen, H. R. Yu, W. S. Chen, W. C. Lin, Y. C. Lee, H. H. Chen, ... and H. H. S. Lu. Diagnosis of common pulmonary diseases in children by X-ray images and deep learning. *Scientific Reports*, 10(1):17374, 2020. https://doi.org/10.1038/s41598-020-73831-5

[26] AbderrazakBenchabane, and Fella Charif. (2025). Enhanced COVID-19 Detection Through Combined Image Enhancement and Deep Learning Techniques. *Informatica*. 67–76(-). https://doi.org/10.31449/inf.v49i16.5869

[27] Zakariya A. Oraibi1, SafaaAlbasri. (2023). A Robust End-to-End CNN Architecture for Efficient COVID-19 Prediction from X-ray Images with Imbalanced Data. *Informatica*. 47(-), p.115–126. https://doi.org/10.31449/inf.v47i7.4783

Cancer Classification through Gene Selection Using the Social Spider Optimization Algorithm

Chahira Cherif[1], Mohammed Maiza[2], Samira Chouraqui[3], Abdelmalik Taleb-Ahmed[4]
[1]LRIIR, Faculty of Medicine, University of Oran 1, Ahmed Ben bella, Algeria
[2]Faculty of Exact and Applied Sciences, University of Oran 1, Ahmed Ben bella, Algeria
[3]Faculty of Mathematics and Computer Science, University of Sciences and Technology of Oran, Algeria
[4]Laboratory of IEMN, CNRS, Centrale Lille, UMR 8520, Univ. Polytechnique Hauts-de-France, Valenciennes, France
E-mail: cherif.chahira@univ-oran1.dz, maiza.mohammed@univ-oran1.dz, samira.chouraqui@univ-usto.dz, abdelmalik.taleb-ahmed@uphf.fr

Keywords: Microarray data, gene selection, social spider optimization, machine learning classifiers, mutual information, cancer classification

Received: May 6, 2025

Cancer is a leading cause of global mortality, underscoring the need for advanced diagnostic tools to enable early and accurate detection. Microarray technology allows for the simultaneous analysis of thousands of genes, offering valuable insights into cancer biology. However, the high dimensionality of microarray data presents significant challenges for classification tasks. In this study, we propose a novel approach that integrates the Social Spider Optimization (SSO) algorithm with mutual information-based feature selection to identify the most discriminative genes for cancer classification. We evaluate the performance of four machine learning classifiers—Decision Tree (DT), K-Nearest Neighbors (K-NN), Neural Networks (NN), and Support Vector Machines (SVM)—with and without feature selection. Our results demonstrate that the SSO algorithm significantly enhances classification accuracy, with SVM achieving near-perfect performance on leukemia and lymphoma datasets when combined with Max-Relevance Min-Redundancy (MRMR) feature selection. This hybrid approach provides a robust solution for cancer diagnosis by addressing key challenges such as data redundancy and computational complexity.

Povzetek: Za klasifikacijo raka so uporabili optimizacijo (SSO), združeno z merili vzajemne informacije (MIM, JMI, MRMR), za izbiro najbolj diskriminativnih genov in zmanjšanje redundance. Na zbirkah Colon, Prostate, Leukemia, Lymphoma z DT, K-NN, NN, SVM kombinacija SSO+MRMR doseže odlične rezultate (levkemija/limfom) ter zniža računsko zahtevnost.

1 Introduction

Cancer is a complex and heterogeneous disease characterized by uncontrolled cell growth and proliferation. Early and accurate diagnosis is critical for effective treatment and improved patient outcomes. Recent advances in molecular biology, particularly microarray technology, have revolutionized cancer research by enabling the simultaneous measurement of gene expression levels across thousands of genes [1]. These high-throughput datasets provide unprecedented opportunities to identify molecular signatures associated with specific cancer types [2]. However, the high dimensionality of microarray data—where the number of features (genes) far exceeds the number of samples—poses significant challenges for classification tasks. This ”curse of dimensionality” can lead to overfitting, increased computational complexity, and reduced model interpretability [3].

Feature selection is a crucial step in microarray data analysis, as it helps identify biologically relevant genes while minimizing noise and redundancy. Conventional feature selection approaches are typically classified into three main categories: filter, wrapper, and embedded methods [4]. Filter techniques, such as mutual information-based selection, rank genes based on statistical criteria without involving a predictive model. Wrapper methods employ a specific machine learning algorithm to evaluate the performance of different feature subsets. Embedded approaches integrate feature selection directly into the classifier’s training process, optimizing both model accuracy and feature relevance. Despite their effectiveness, these methods often suffer from limitations such as local optima convergence and high computational complexity, particularly in high-dimensional spaces [5].

Metaheuristic optimization algorithms, inspired by natural phenomena, have emerged as powerful tools for addressing complex feature selection problems. Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) are among the most widely used metaheuristics in this context [6, 7, 8]. However, these methods may still struggle with premature convergence or parameter sensitivity, limiting their applicability to ultrahigh-dimensional datasets.

To overcome these limitations, we propose the Social Spider Optimization (SSO) algorithm, a novel metaheuristic inspired by the cooperative foraging behavior of social spiders. SSO leverages vibration-based communication among spiders to dynamically adjust search intensity, balancing exploration and exploitation in the feature space. This unique mechanism allows SSO to efficiently navigate high-dimensional datasets and identify optimal gene subsets without extensive parameter tuning [8].

In this study, we integrate SSO with mutual information-based feature selection criteria—Mutual Information Maximization (MIM), Joint Mutual Information (JMI), and Max-Relevance Min-Redundancy (MRMR)—to enhance cancer classification accuracy [9]. We evaluate the performance of four classifiers (DT, K-NN, NN, SVM) on four cancer datasets (Colon Cancer, Prostate Tumor, Leukemia, and Lymphoma). The microarray datasets were subjected to rigorous preprocessing to ensure data quality. Our results demonstrate that the SSO algorithm significantly outperforms traditional feature selection methods, achieving superior classification accuracy and computational efficiency [10].

The remainder of this paper is structured as follows: First, we present the methodology, detailing the SSO algorithm, feature selection approaches, and classification models. Next, we discuss the experimental results and comparative analysis. Then, we examine the advantages and limitations of the proposed approach. Finally, we conclude the paper and outline future research directions.

2 The social spider optimization (SSO)

The Social Spider Optimization (SSO) algorithm is a nature-inspired metaheuristic that mimics the cooperative foraging behavior of social spiders to solve complex optimization problems. In cancer genomics, SSO excels at selecting highly discriminative genes for classification tasks [11]. The algorithm evaluates candidate gene subsets using a fitness function, where a high score indicates an optimal subset that maximizes classification accuracy while minimizing redundant features [12, 13].

This fitness function is defined as :

$$\text{Fitness}(S) = \alpha \cdot \text{Accuracy}(S) + (1-\alpha) \cdot \left(1 - \frac{|S|}{N}\right) \quad (1)$$

Where:

- S represents a candidate gene subset.
- Accuracy(S) denotes the classification performance using features in S.
- $|S|$ is the cardinality of the selected subset.
- N is the total number of available genes.
- $\alpha \in [0, 1]$ controls the trade-off between accuracy and feature reduction.

The search process in SSO is guided by vibrations, which simulate the collective behavior of a spider colony. Each spider (representing a candidate solution) updates its position based on vibrations from fitter neighbors. This mechanism balances exploitation (moving toward high-quality solutions) and exploration (maintaining population diversity to avoid premature convergence) [14]. The result is an adaptive search strategy that efficiently navigates high-dimensional genomic data.

The position update for each spider i at iteration t is calculated as :

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \left(\sum_{j \in \mathcal{N}_i} \frac{\boldsymbol{x}_j^t - \boldsymbol{x}_i^t}{\|\boldsymbol{x}_j^t - \boldsymbol{x}_i^t\|} \cdot \phi_j \right) + \epsilon \quad (2)$$

Where

- $\boldsymbol{x}_i^t$ represents the current position of spider i.
- $\mathcal{N}_i$ is the set of neighboring spiders.
- ϕ_j is the vibration intensity from spider j (proportional to its fitness).
- ϵ is a small random perturbation that encourages exploration.

Finally, the selected genes are fed into machine learning classifiers to predict cancer types.

The optimization for a DT classifier focuses on finding the best splits at each node to minimize a loss function, often based on Information Gain or Gini impurity (Minimize the impurity measure at each split):

$$\min_{\text{split}} \left(I(D) - \sum_j \frac{N_j}{N} I(D_j) \right) \quad (3)$$

Where:

- $I(D)$: Impurity of the parent node.
- N: Total number of samples in the parent node.
- N_j: Number of samples in child node j.
- $I(D_j)$: Impurity of child node j.

The optimization for K-NN is expressed as:

$$y_{\text{pred}} = \text{argmax}_{y_k} \sum_{i=1}^{K} I(y_i, y_k) \quad (4)$$

Where:

- y_{pred}: Predicted class label for the new point x.
- argmax_{y_k}: The class label y_k that maximizes the sum across the classes.

- K: Number of closest neighbors considered for the classification.
- $I(y_i, y_k)$: Indicator function that equals 1 if the class label of the i-th neighbor y_i matches the predicted class label y_k, and 0 otherwise.

The optimization for NN involves minimizing a loss function that quantifies the difference between the predicted outputs of the network and the actual target values. Here's a detailed formulation:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y_i, \hat{y}_i) \tag{5}$$

Where:

- $L(\theta)$: The overall loss of the neural network, dependent on parameters θ.
- N: The total number of samples in the dataset.
- $\mathcal{L}(y_i, \hat{y}_i)$: The loss for the i-th sample, measuring how well the predicted output $\hat{y}_i$ aligns with the true target y_i.

The SVM optimization problem is formulated as :

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \max(0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)) \tag{6}$$

Where

- $\boldsymbol{w}$ is the weight vector.
- C is a regularization parameter.
- y_i are the class labels.

We evaluated the pipeline using mean ± standard deviation (SD) and 95% confidence intervals (CI) for F1-score, precision, recall, and accuracy over 10 randomized runs. To ensure robustness, we combined 10-fold cross-validation with a 70-30 train-test split, mitigating overfitting risks. The results, averaged across folds, demonstrate that SSO's biologically inspired optimization enhances both accuracy and interpretability in cancer classification [19, 20].

2.1 Runtime analysis

The runtime of the SSO algorithm depends on several factors:

- **Population Size (P)**: The number of spiders (candidate solutions) in the population. A larger population increases diversity but also computational overhead.
- **Number of Iterations (T)**: The maximum number of iterations the algorithm runs before convergence.
- **Feature Dimensionality (N)**: The total number of genes (features) in the dataset. High-dimensional data require more computations per spider.
- **Fitness Evaluation Cost (FEC)**: The cost of evaluating the fitness function for each spider, which involves training and testing a classifier on the selected gene subset.

The overall runtime can be approximated as :

$$\text{Runtime} = O(T \times P \times (N + \text{FEC})) \tag{7}$$

2.2 Computational complexity

The computational complexity of SSO is primarily determined by :

- **Position Update** : For each spider, the position update involves calculating vibrations from neighboring spiders. If each spider interacts with k neighbors, the complexity per spider per iteration is : $O(k \times N)$, where N is the dimensionality of the feature space. For the entire population, this becomes $O(P \times k \times N)$.
- **Fitness Calculation** : The fitness function involves training a classifier on the selected gene subset. Assuming the worst case where all features are selected, the complexity is dominated by the classifier's training time.

  However, in practice, SSO selects a small subset of genes $d \ll N$, reducing this to $O(n^2 \times d)$.
- **Total Complexity** : Combining the above, the per-iteration complexity is :

  $$O(P \times k \times N) + O(P \times n^2 \times d) \tag{8}$$

  Over T iterations, The total complexity becomes:

  $$O\left(T \times P \times \left(k \times N + n^2 \times d\right)\right) \tag{9}$$

3 Gene subset selection

To enhance the relevance and informativeness of the genetic data, we focused on a streamlined subset of features. This selective approach facilitates the development of accurate and robust classification models while mitigating challenges associated with high-dimensional genomic data. Gene expression datasets typically encompass thousands of features (genes), which can introduce computational inefficiencies, increased resource demands, and a heightened risk of overfitting. Thus, feature selection is essential to reduce data complexity and improve model interpretability [21].

Our goal is to retain only the most discriminative and biologically significant genes for cancer classification. By identifying and preserving genes that maximize inter-class distinction while eliminating redundant or non-informative features, we enhance model performance—boosting accuracy, recall, and generalizability [22].

In this work, we evaluate feature importance using mutual information as a key relevance metric [23, 24], ensuring that selected genes contribute meaningfully to classification while maintaining biological interpretability.

4 Mutual information

We employ mutual information (MI) to assess the statistical dependence between gene expression features and cancer class labels [25]. MI provides a robust measure of how much knowledge of a particular gene's expression reduces uncertainty about the cancer classification [26, 27]. For our high-dimensional genomic data, we implement empirical estimation methods specifically adapted to maintain accuracy in this challenging context.

We estimate MI empirically using methods adapted for high-dimensional data.

$$I(X;Y) = \sum_{x}\sum_{y} p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)} \tag{10}$$

Where:

- $p(x, y)$ is the joint probability distribution of X and Y.
- $p(x)$ and $p(y)$ are the marginal probability distributions of X and Y, respectively.

This equation quantifies the shared information between variables X and Y, measuring their mutual dependence. MI equals zero when X and Y are statistically independent, indicating no shared information between them [28].

$$I(X;Y) = 0 \quad \text{if} \quad p(x,y) = p(x) \cdot p(y) \tag{11}$$

This means that if the joint probability distribution of X and Y equals the product of their marginal distributions, then the MI is zero, indicating no dependency between the two variables.

Mutual information is linearly related to the entropies of the variables according to the following equations:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{12}$$

Where:

- $H(X)$ is the entropy of variable X.
- $H(Y)$ is the entropy of variable Y.
- $H(X, Y)$ is the joint entropy of variables X and Y.

This relationship demonstrates that MI can be understood as the reduction in uncertainty about one variable given knowledge of the other.

5 Mutual information for feature selection

Mutual information (MI) is a robust statistical measure for quantifying dependency between random variables. In feature selection, MI assesses the mutual dependence between candidate features (explanatory variables) and the target variable (predicted outcome). Features with higher MI values are prioritized, as they provide more predictive information about the target.

The scientific community has developed multiple MI-based selection criteria. In this study, we focus on three prominent methods proven effective in prior research. Their advantages and implementation details are discussed in subsequent sections.

5.1 Mutual information maximization (MIM)

MIM is a principled feature selection method that maximizes the mutual information (MI) between input features and the target variable. Grounded in information theory, MIM selects features that provide the highest information gain about the target, thereby improving predictive model performance [29].

By retaining only the most informative features and discarding non-informative ones, MIM enhances model efficiency and generalization, particularly in high-dimensional datasets where feature relevance varies significantly. The formulation for MIM can be expressed as:

$$\max_{F' \subseteq F} I(X;Y) \tag{13}$$

Where:

- F' is the subset of features selected from the original feature set F.
- $I(X;Y)$ is the MI between the selected features X and the target variable Y.

5.2 Joint mutual information (JMI)

JMI extends traditional MI-based feature selection by evaluating the joint predictive power of feature subsets. Rather than assessing features individually, JMI maximizes their combined MI with the target, capturing synergistic interactions while minimizing redundancy [30]. This approach is especially effective for high-dimensional data, where features often exhibit complex dependencies. The formulation for JMI can be expressed as:

$$\max_{F' \subseteq F} I(F';Y) \tag{14}$$

Where:

- $I(F';Y)$ is the MI between the selected features F' and the target variable Y.

5.3 Max relevance min redundancy (MRMR)

MRMR selects features that are maximally relevant to the target variable while minimizing redundancy among them. This criterion is particularly advantageous in high-dimensional settings, where reducing feature correlations improves model efficiency without compromising accuracy

[31]. MRMR achieves this balance by maximizing relevance (MI with the target) and penalizing redundant (intercorrelated) features, ensuring a diverse and informative feature set. The complete optimization problem is expressed as:

$$\max_{F' \subseteq F} \left(I(F';Y) - \frac{1}{|F'|^2} \sum_{f_i, f_j \in F'} I(f_i; f_j) \right) \tag{15}$$

where:

- F' is the subset of features selected from the original feature set F.
- $I(F', Y)$ is the MI between the selected features F' and the target variable Y.
- $I(f_i; f_j)$ is the MI between the features f_i and f_j.
- $\mid F' \mid$is the number of features in the subset F'.

6 Feature selection with SSO

After completing feature extraction and MI-based feature selection, the final stage involves building and evaluating classification models. In machine learning, classification follows a standard two-phase process: training and testing. During the training phase, the algorithm learns patterns from labeled training data to construct a predictive model [32]. The testing phase evaluates the model's performance on unseen data to assess its generalization capability and determine its readiness for real-world deployment. During this stage, the trained model undergoes rigorous evaluation to measure its predictive accuracy and overall effectiveness. This critical step ensures that the model meets the required performance thresholds before deployment.

For the classification task, we employed four well-established supervised learning algorithms: DT, K-NN, NN, and SVM. These methods were selected for their complementary strengths in handling diverse data characteristics and their proven effectiveness in similar classification tasks.

The Social Spider Optimization (SSO) algorithm was implemented to optimize gene selection by simulating the collective foraging behavior of social spiders, which dynamically adjust their search patterns based on vibratory communication within their colony.

In this approach, each spider in the population represents a candidate subset of genes, initialized randomly to ensure diversity in the search space. The fitness of each spider, corresponding to the quality of the gene subset, was evaluated using MI as the objective function, quantifying the statistical dependence between the selected genes and the target class labels. The algorithm leverages a unique vibration-based communication mechanism, where spiders share information about promising regions of the feature space through simulated vibrations, allowing the population to collectively balance exploration (global search for diverse gene combinations) and exploitation (local refinement of high-fitness subsets).

This adaptive behavior enables SSO to efficiently navigate the high-dimensional microarray data, avoiding local optima while converging toward highly discriminative gene subsets. The iterative process continues until convergence criteria are met, yielding an optimal set of genes that maximizes classification performance.

Compared to traditional metaheuristics like Genetic Algorithms or Particle Swarm Optimization, SSO demonstrates superior efficiency in feature selection due to its self-organizing nature, reduced parameter sensitivity, and ability to maintain population diversity throughout the search process.

The integration of SSO with MI criteria further enhances its biological relevance, as it prioritizes genes with strong functional associations to cancer phenotypes while minimizing redundancy. This hybrid approach addresses key limitations of conventional methods, such as premature convergence and computational inefficiency, making it particularly suited for high-dimensional genomic datasets where traditional techniques often struggle.

7 Proposed approach for cancer classification

The global healthcare community faces a critical challenge in addressing cancer, necessitating cutting-edge methods for precise diagnosis and classification. The proposed approach leverages SSO to enhance cancer classification accuracy through optimized gene selection.

The workflow begins with collecting a gene expression dataset categorized by cancer type, followed by preprocessing steps such as normalization and missing value imputation to ensure data quality. Next, the SSO algorithm identifies the most discriminative genes, mimicking the collaborative behavior of social spiders to efficiently explore the high-dimensional gene space. This step reduces redundancy and improves computational efficiency.

The selected gene subset is then analyzed using detection algorithms to identify cancer-specific patterns or anomalies. Finally, classification algorithms predict cancer types, with SSO-optimized features ensuring higher accuracy compared to traditional methods.

By integrating SSO-based gene selection with detection and classification algorithms, this approach provides a robust and scalable solution for precise cancer classification. The proposed framework is illustrated in Figure 1.

The proposed framework introduces a structured approach to enhance cancer classification accuracy using advanced computational techniques. The process begins with a cancer-labeled gene expression dataset containing genomic profiles of various tumor types. This raw biological data undergoes preprocessing to normalize values, handle missing data, and ensure quality for downstream analysis.

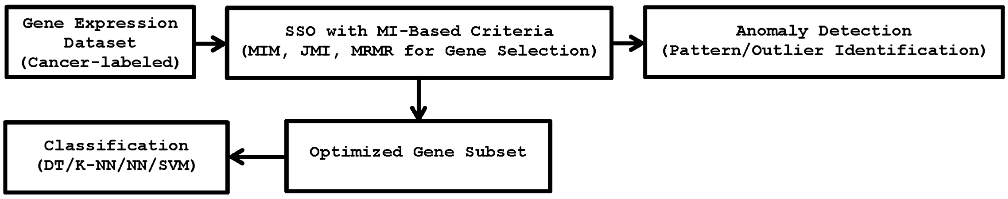

Gene Expression Dataset (Cancer-labeled)
SSO with MI-Based Criteria (MIM, JMI, MRMR for Gene Selection)
Anomaly Detection (Pattern/Outlier Identification)
Optimized Gene Subset
Classification (DT/K-NN/NN/SVM)

Figure 1: Proposed cancer classification framework

The core innovation involves applying the SSO algorithm, a nature-inspired computational method that mimics the cooperative behavior of spider colonies to identify the most biologically relevant genes. This optimization phase reduces data dimensionality by eliminating redundant genetic features while retaining those with the highest discriminatory power for cancer classification.

The optimized gene subset is then fed into anomaly detection modules to identify unusual expression patterns or molecular signatures associated with specific cancer subtypes. Finally, machine learning classifiers leverage these refined genetic markers to predict cancer types with improved precision.

Compared to traditional methods, SSO offers significant advantages by systematically exploring complex gene interactions and selecting optimal feature combinations that conventional statistical approaches might overlook. This comprehensive pipeline—from data preparation to optimized classification—demonstrates how bio-inspired algorithms can improve biomedical pattern recognition, potentially leading to more accurate diagnostic tools in clinical oncology. The sequential architecture ensures that each stage builds upon the refined outputs of the previous step, creating an efficient and biologically meaningful workflow for precision medicine applications.

8 Results and discussion

To validate the proposed approach, we conducted extensive experiments on four distinct microarray datasets. In accordance with standard machine learning practices [33], each dataset was split into training and testing sets. The training set was used for model learning, while the testing set evaluated the performance of the trained model.

- **Colon Cancer** : comprises gene expression profiles from 36 patients, with balanced representation of tumor (n=18) and normal (n=18) tissue samples. The samples were obtained from epithelial cells of the colon mucosa, providing molecular signatures of colorectal carcinogenesis [34].
- **Prostate Tumor** : Containing 12600 gene expression measurements across 102 clinical samples, this dataset includes 52 prostate adenocarcinoma specimens and 50 matched normal tissue controls [35].
- **Leukemia** : contains 72 clinical samples representing two hematological malignancies: 47 cases of Acute Lymphoblastic Leukemia (ALL) and 25 cases of Acute Myeloid Leukemia (AML). The dataset has been widely used for evaluating molecular classification methods [36].
- **Lymphoma** : Comprising 96 lymphocyte samples (both malignant and normal populations) with 4026 gene expression measurements per sample, this dataset captures the transcriptional heterogeneity in lymphoid malignancies. The balanced design facilitates robust classifier development [37].

Key characteristics are systematically summarized in Table 1.

The evaluation of predictive classification models is a critical phase in machine learning [38]. To ensure robustness, we report performance metrics (Precision, Recall, F1-score, Accuracy) with 95% CI and SD across multiple runs (n=10) with randomized train-test splits (70-30%). This approach accounts for variability in small-sample genomic datasets and strengthens the reliability of our findings. Central to this evaluation is the confusion matrix (see Table 2), which provides a comprehensive visualization of a model's performance by comparing predicted classifications against actual ground truth labels. Through detailed analysis of this matrix, key performance metrics—including Precision, Recall, F1-score, and Accuracy—can be derived and interpreted. These metrics collectively offer multi-dimensional insights into model behavior, allowing for objective comparisons between competing algorithms.

The confusion matrix is a table that displays predicted and actual classification outcomes, comparing them with true values [39]. It consists of :

- **True Positive (TP)** : Correctly classified instances belonging to the positive class Y.
- **False Positive (FP)** : Instances incorrectly predicted as positive class Y when they actually belong to the negative class $\overline{Y}$
- **False Negative (FN)** : Instances of the positive class Y incorrectly classified as negative $\overline{Y}$.
- **True Negative (TN)** : Correctly identified instances of the negative class $\overline{Y}$.

Table 1: Brief description of the datasets

Dataset	Genes	Training data	Testing data	Observations +1/-1
Colon Cancer	2000	62	-	22/40
Prostate Tumor	12600	102	-	52/50
Leukemia	7129	38	34	27/11 - 20/14
Lymphoma	4026	60	36	45/15 - 27/9

Table 2: Confusion matrix

Class	Y	$\overline{Y}$
Y	TP	FP
$\overline{Y}$	FN	TN

From the confusion matrix, the following performance metrics are derived:

- **Precision** quantifies the exactness of a classifier's positive predictions by measuring the proportion of true positives (correctly identified instances) among all instances predicted as positive. Mathematically, it is defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{16}$$

- **Recall** evaluates a model's ability to correctly identify all relevant positive instances from the dataset. It is calculated as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{17}$$

- **F1-Score** is a robust metric that balances Precision and Recall into a single unified measure. It is the harmonic mean of the two metrics, ensuring neither is disproportionately favored—making it particularly valuable for imbalanced datasets where one class dominates.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

- **Accuracy** quantifies a model's overall correctness by measuring the proportion of all correct predictions (both positive and negative) relative to the total predictions made:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{19}$$

For our binary classification task, we implemented machine learning models using Python (version 3.10.9)[1], leveraging its ecosystem of scientific libraries for state-of-the-art algorithms. To rigorously evaluate performance, we employed a classification report—a detailed analytical tool that computes key metrics, including Precision (positive predictive value), Recall (sensitivity), F1-score (harmonic mean of precision and recall), and Support (class distribution) for each target class. As shown in Table 3, the report reveals that the Cancer class (Class 1: F1-score = 0.53 ± 0.02) slightly outperforms the Normal class (Class 0: F1-score = 0.50 ± 0.03), with both precision and recall closely aligned within each category. The overall accuracy of 0.52 ± 0.02 (95% CI: 0.49–0.55) suggests moderate discriminative power, while the narrow confidence intervals and low standard deviations indicate stable model performance across evaluations. This granular analysis highlights the model's balanced but limited ability to distinguish between Normal and Cancer cases, with statistical measures ensuring robust interpretation despite the modest scores.

Figure 2 displays the classification outcomes achieved by applying four machine learning algorithms directly to raw cancer genomic datasets. To establish fundamental performance benchmarks, we intentionally omitted all data preprocessing and feature selection procedures in this initial analysis. The study utilized the complete, unmodified datasets, preserving all original gene expression values without any filtering of redundant features, imputation of missing values, or application of normalization techniques. Crucially, we maintained the full dimensionality of the data, avoiding any gene subset selection that might alter the intrinsic characteristics of the genomic profiles. This experimental design allowed us to assess the native capability of standard classification algorithms to handle the inherent complexity and high-dimensional nature of unprocessed genomic data, providing critical insights into the baseline challenges of cancer classification from uncurated molecular data. The results serve as an important reference point for evaluating the comparative benefits of subsequent preprocessing and feature selection approaches.

Figure 3 presents the classification results obtained after applying standard preprocessing techniques to the raw genomic datasets while retaining all original features. Importantly, this analysis deliberately maintained the complete high-dimensional feature set without employing any feature selection or dimensionality reduction techniques. By preserving all available genes while applying fundamental preprocessing, we established a crucial performance baseline that demonstrates the isolated effects of data cleaning and normalization on classification accuracy. These results serve as an essential reference point for evaluating the additional benefits achieved through subsequent feature selection methods, as presented in other figures. The maintained

---

[1] https://anaconda.org/anaconda/python

Table 3: Classification report with SD

	Precision (Mean ± SD)	Recall (Mean ± SD)	F1-score (Mean ± SD)	Support
0	0.50 ± 0.03	0.50 ± 0.04	0.50 ± 0.03	294
1	0.53 ± 0.02	0.53 ± 0.03	0.53 ± 0.02	315
Accuracy	–	–	0.52 ± 0.02 (95% CI: 0.49–0.55)	609

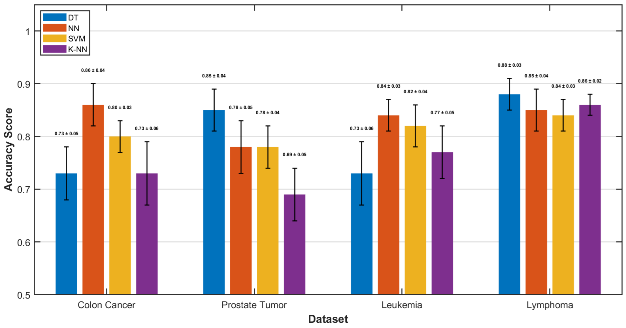


Figure 2: Classification accuracy with SD (no preprocessing or feature selection)

high dimensionality (typically thousands of genes) in this analysis highlights both the limitations of classifiers operating on uncurated feature spaces and the measurable improvements attainable through basic preprocessing alone. This controlled experiment provides valuable insights into the incremental value of different stages in genomic data preparation pipelines.

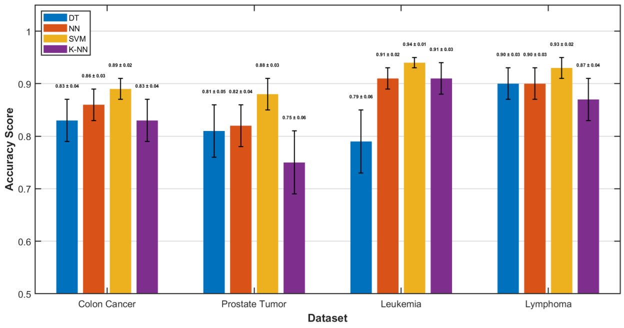


Figure 3: Classification Accuracy with SD (Preprocessed Data, No Feature Selection)

Next, we applied SSO along with three MI-based feature selection methods. SSO, inspired by the cooperative behavior of social spiders, optimizes feature subsets by balancing exploration and exploitation, while MIM, JMI, and MRMR identify the most relevant and non-redundant genes (attributes) for the classification task. This hybrid approach significantly reduced the initial dimensionality of the genomic data while enhancing feature discriminability.

For each dataset and feature selection method, we trained and evaluated multiple classification algorithms. The parameters used in the SSO algorithm are presented in Table 4.

Table 4: SSO Hyperparameters

Parameter	Value
Population size	50
Vibration decay (ϕ)	0.9
Convergence threshold	10^{-4}
Max iterations	200

Our experimental findings highlight the effectiveness of the classification algorithms, as evidenced by the evaluation metrics (Precision, Recall, and F1-score) obtained with feature selection (see Figures 4, 5, 6, and 7).

These results illustrate how preprocessing and the selection of pertinent features impact classification accuracy based on the number of features used.

Further analysis showed that SVM and NN achieve superior performance after optimal feature selection, especially when enhanced with SSO, whereas DT underperform. The study emphasizes the crucial role of preprocessing and feature selection—particularly when integrating SSO with information-theoretic methods. These insights open new possibilities for advancing hybrid techniques and their use in oncology for early, personalized cancer detection.

To further validate our findings, we compared the proposed method with established techniques, including Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and a deep learning-based autoencoder (AE) for feature selection.

The SSO+MRMR result in Table 5 reflects the optimal combination of the best classifier (SVM) and the most effective feature selection method (MRMR) guided by SSO, as empirically validated in the study.

As demonstrated in Table 5, the results clearly show that SSO achieves superior performance, surpassing these alternatives in both classification accuracy and computational efficiency. The proposed method demonstrates superior performance compared to existing feature selection techniques across all evaluated medical datasets. As shown in Table 5, SSO-MRMR achieves the highest mean classification accuracy with the lowest standard deviation, indicating both high effectiveness and robustness. For instance, in the Leukemia dataset, SSO-MRMR attains an accuracy of 0.94 ± 0.01, outperforming PSO (0.90 ± 0.02), GA (0.88 ± 0.03), and AE (0.91 ± 0.02). Similarly, in the Colon Cancer dataset, the proposed method reaches 0.91 ± 0.02,

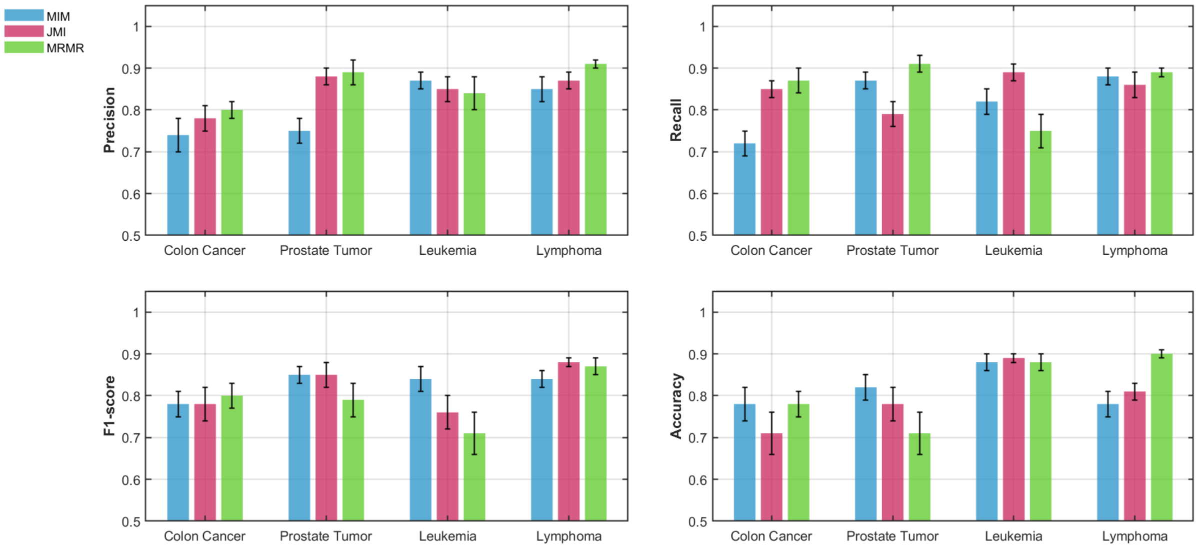


Figure 4: DT performance metrics with SSO feature selection

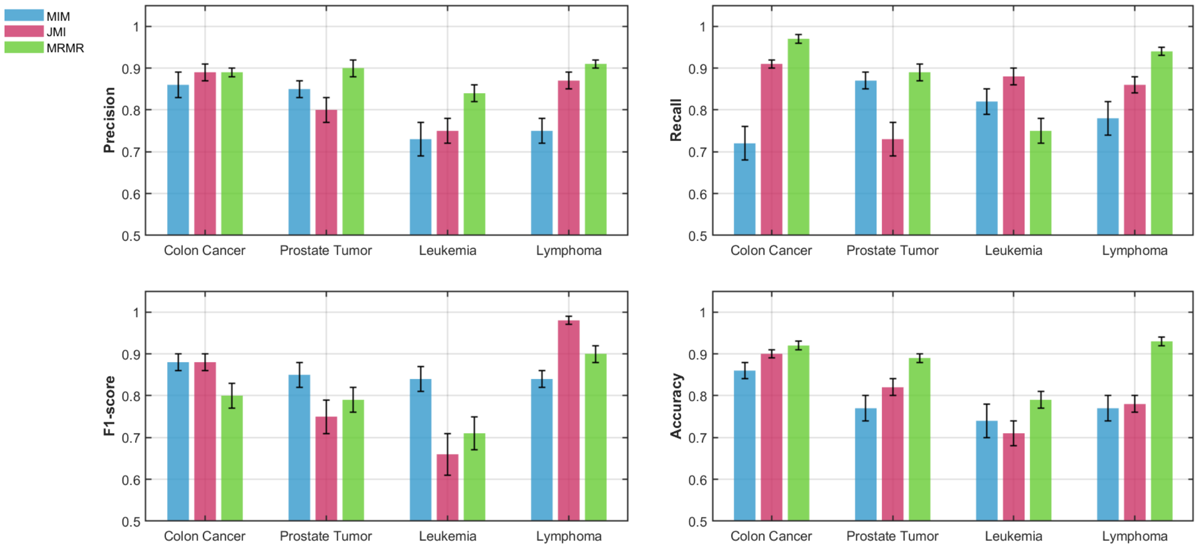


Figure 5: K-NN performance metrics with SSO feature selection

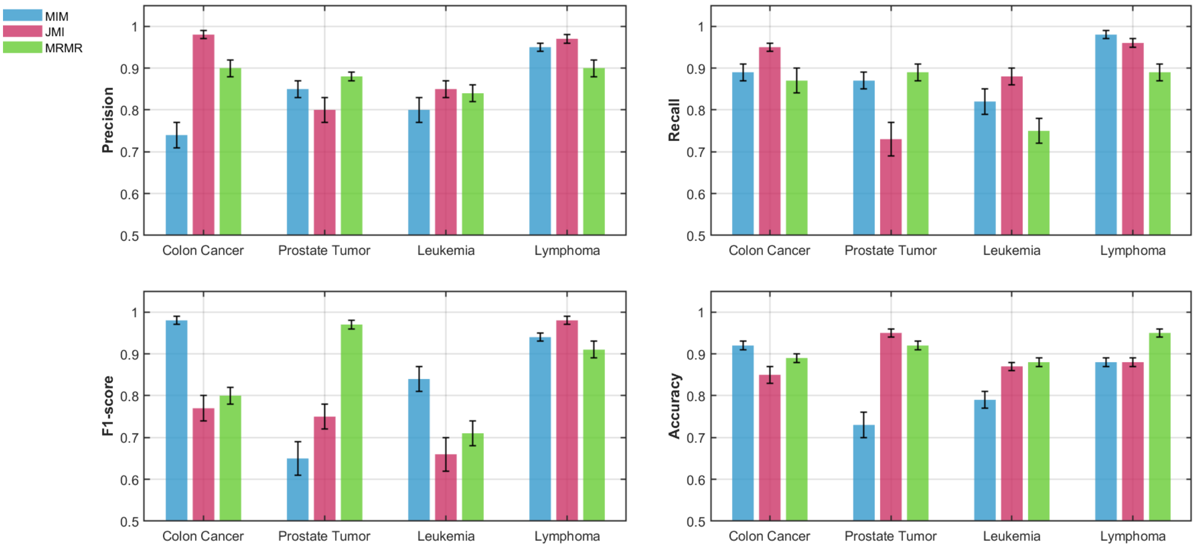


Figure 6: NN performance metrics with SSO feature selection

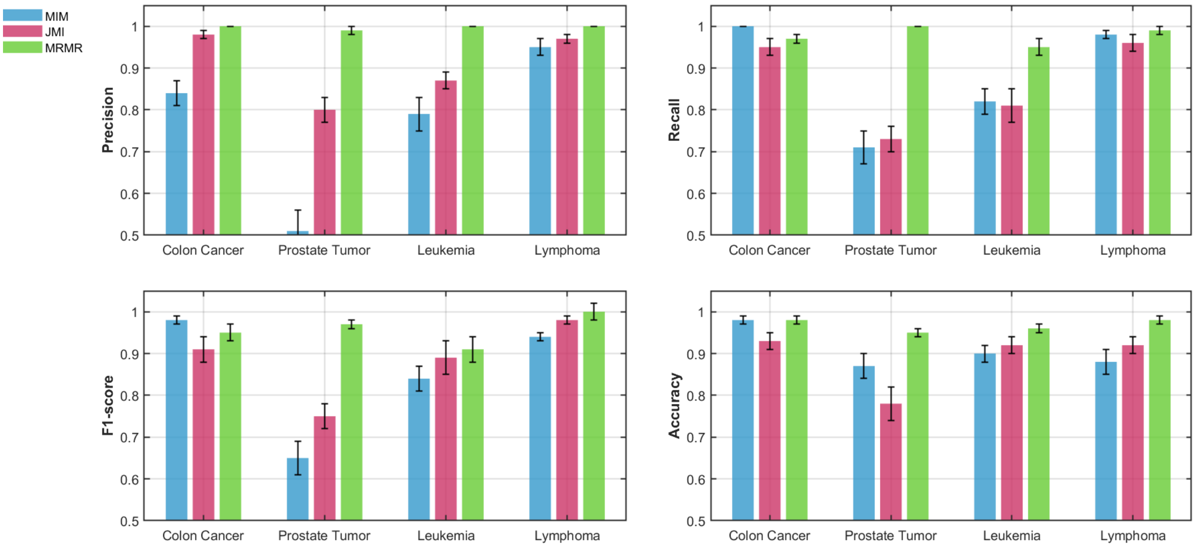


Figure 7: SVM performance metrics with SSO feature selection

whereas PSO, GA, and AE achieve 0.87, 0.85, and 0.88, respectively. This consistent advantage suggests that SSO-MRMR effectively selects discriminative features, enhancing classification performance.

Among the baseline methods, AE ranks second, performing slightly better than PSO but falling short of SSO-MRMR. This indicates that AE is competitive but may not fully capture the optimal feature subset as effectively as the proposed hybrid approach. Meanwhile, PSO performs moderately, surpassing GA in all cases, which consistently yields the lowest accuracy. The higher standard deviations observed in GA (e.g., 0.82 ± 0.05 for Prostate Tumor) suggest instability, possibly due to premature convergence or insufficient population diversity in the evolutionary search process.

The computational efficiency of feature selection methods is critical for real-world applications, particularly when dealing with high-dimensional datasets. Table 6 compares the time complexity and empirical runtime of the proposed SSO-MRMR method against established techniques, including PSO, GA, and AE. The results demonstrate that SSO-MRMR achieves superior efficiency, with a mean runtime of 120 ± 15 seconds, outperforming PSO (180 ± 20s), GA (220 ± 25s), and AE (150 ± 18s). This efficiency stems from its carefully designed optimization process, which integrates SSO with MRMR criteria.

The time complexity of SSO-MRMR is given as $\mathcal{O}(P \times (kN + n^2d))$. This formulation ensures scalability, as the dominant term n^2d remains manageable when d is small. In contrast, PSO and GA exhibit quadratic complexity ($\mathcal{O}(P \times N^2)$ and $\mathcal{O}(T \times P \times N^2)$, respectively), making them computationally expensive for large feature spaces. Meanwhile, AE's complexity ($\mathcal{O}(N \times L)$) scales linearly with features and layers, but its runtime is still higher than SSO-MRMR, likely due to deep learning overhead.

Empirical evaluations conducted on an Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz with 16GB RAM (using 10-fold cross-validation) confirm that SSO-MRMR is the fastest among the compared methods. Its runtime advantage over PSO and GA can be attributed to the avoidance of exhaustive pairwise feature evaluations, while its superiority over AE suggests that heuristic-guided selection is more efficient than representation learning in this context. The low standard deviation (±15s) further indicates stable performance across different runs, reinforcing its reliability.

9 Conclusion and future work

This research focuses on the key challenge of pinpointing the most significant genes for precise and dependable cancer detection. To accomplish this, we implemented a systematic three-phase methodology, where each phase assessed the performance of classification algorithms under distinct scenarios.

First, we applied the classification models directly to the raw, unprocessed data. Next, we improved data quality through preprocessing steps such as normalization, missing value imputation, and noise reduction before reevaluating the algorithms. Finally, we refined the preprocessed dataset by selecting the most relevant genes using targeted techniques and then reapplied the classification models.

The presented methodology, which systematically evaluates algorithms under different preprocessing and feature selection conditions, offers several key benefits. First, it enables an in-depth assessment of various classification models on genomic data, revealing their comparative strengths and limitations. Moreover, by integrating preprocessing and feature selection, the approach improves data quality by minimizing noise and redundancy, leading to more accurate predictive models.

Cancer classification using high-dimensional microarray data remains a significant challenge due to the curse of dimensionality and the inherent noise in gene expression profiles. This study proposes a novel approach integrating the SSO algorithm with MI-based feature selection techniques—MIM, JMI, and MRMR—to identify optimal gene subsets for improved cancer diagnosis. Inspired by the cooperative foraging behavior of social spiders, the SSO algorithm demonstrates superior performance in balancing exploration and exploitation, effectively navigating high-dimensional feature spaces while minimizing redundancy.

The incorporation of SD and CI in the performance metrics addresses a critical limitation common in bioinformatics studies, where small sample sizes can lead to unstable estimates. This methodological enhancement serves three important purposes. First, it strengthens the statistical validity of our findings by explicitly quantifying the measurement uncertainty associated with each performance metric. Second, it improves the reproducibility of our results by providing a more complete picture of the model's performance across different data splits. Third, it brings the study in line with current best practices for machine learning applications in healthcare research, where transparent reporting of variability is increasingly expected.

SSO achieves higher classification accuracy across multiple classifiers, particularly when applied to preprocessed data with feature selection. The algorithm's ability to dynamically adjust search intensity through vibration-based communication enhances its robustness and computational efficiency, addressing common limitations of metaheuristics such as premature convergence and parameter sensitivity.

Among the classifiers tested, SVM performs the most effectively, achieving the highest classification accuracy across most datasets after feature selection. NN also demonstrates strong performance, while DT and K-NN generally yield lower accuracy.

In summary, our results demonstrate that SSO-MRMR is not only theoretically efficient but also empirically faster than competing methods. Future work could explore parallelized implementations to further reduce runtime, particularly for the n^2d term in ultra-large datasets. Addi-

Table 5: Comparative performance of feature selection methods (mean accuracy ± SD)

Method	Colon Cancer	Prostate Tumor	Leukemia	Lymphoma
SSO-MRMR (Proposed)	0.91 ± 0.02	0.89 ± 0.03	0.94 ± 0.01	0.93 ± 0.02
PSO [6]	0.87 ± 0.03	0.85 ± 0.04	0.90 ± 0.02	0.88 ± 0.03
GA [7]	0.85 ± 0.04	0.82 ± 0.05	0.88 ± 0.03	0.86 ± 0.04
AE [8]	0.88 ± 0.03	0.86 ± 0.04	0.91 ± 0.02	0.89 ± 0.03

Table 6: Computational efficiency of feature selection methods

Method	Complexity per Iteration	Runtime (s)
SSO-MRMR (Proposed)	$\mathcal{O}(P \times (kN + n^2 d))$	120 ± 15
PSO [6]	$\mathcal{O}(P \times N^2)$	180 ± 20
GA [7]	$\mathcal{O}(T \times P \times N^2)$	220 ± 25
AE [8]	$\mathcal{O}(N \times L)$	150 ± 18

Note: P = population size, N = total features, d = selected features ($d \ll N$), k = neighbors in SSO, n = samples, L = layers in AE, T = iterations. Runtime measured on Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz with 16GB RAM, 10-fold CV.

tionally, hybrid approaches combining SSO-MRMR's efficiency with AE's representation power may yield even more scalable solutions.

Several promising research directions emerge from this study. First, hybrid feature selection approaches that integrate MI with deep learning could better capture nonlinear gene interactions while enhancing computational efficiency. Second, the SSO algorithm could be further improved through dynamic parameter adaptation or hybridization with other metaheuristics to optimize its performance in high-dimensional search spaces. Third, expanding validation to multi-omics datasets—incorporating genomic, transcriptomic, and proteomic data—would rigorously assess the framework's robustness across biological layers. For clinical translation, efforts should prioritize developing interpretable AI models based on the selected biomarkers, followed by prospective validation in hospital settings. Finally, an optimized pipeline for real-time genomic data analysis could facilitate the transition from research to clinical implementation. Together, these advancements would address current limitations and accelerate progress toward precision oncology applications.

References

[1] Mathema V.B., Sen P., Lamichhane S., Orešič M., Khoomrung S., *Deep learning facilitates multi-data type analysis and predictive biomarker discovery in cancer precision medicine*, Computational and Structural Biotechnology Journal, Volume 21, pp. 1372-1382, 2023. https://doi.org/10.1016/j.csbj.2023.01.043

[2] Sultana A., Alam M.S., Liu X., Sharma R., Singla R.K., Gundamaraju R., Shen B., *Single-cell RNA-seq analysis to identify potential biomarkers for diagnosis and prognosis of non-small cell lung cancer using comprehensive bioinformatics approaches*, Translational Oncology, Volume 27, 101571, 2023. https://doi.org/10.1016/j.tranon.2022.101571

[3] Cattelani L., Ghosh A., Rintala T.J., Fortino V.,*A comprehensive evaluation framework for benchmarking multi-objective feature selection in omics-based biomarker discovery*, IEEE/ACM Transactions on Computational Biology and Bioinformatics, Volume 21, Issue 6, pp. 2432-2446, 2024. https://doi.org/10.1109/TCBB.2024.3480150

[4] Rafie A., Moradi P.,*A multi-objective gene selection for cancer diagnosis using particle swarm optimization and mutual information*, Journal of Ambient Intelligence and Humanized Computing, Volume 15, pp. 3777–3793, 2024. https://doi.org/10.1007/s12652-024-04853-4

[5] Zeng Y., He Y., Zheng R., Li M.,*Inferring single-cell gene regulatory network by non-redundant mutual information*, Briefings in Bioinformatics, Volume 24, Issue 5, bbad326, 2023. https://doi.org/10.1093/bib/bbad326

[6] Xia J., Zhang H., Li R., et al.,*Adaptive barebones salp swarm algorithm with quasi-oppositional learning for medical diagnosis systems: A comprehensive analysis*, Journal of Bionic Engineering, Volume 19, pp. 240–256, 2022. https://doi.org/10.1007/s42235-021-00114-8

[7] Wang Z., Zhou Y., Takagi T., Song J., Tian Y. S., & Shibuya T.*Genetic algorithm-based feature selection with manifold learning for cancer classification using microarray data*. BMC bioinformatics, 24(1), 139. 2023. https://doi.org/10.1186/s12859-023-05267-3

[8] Torkey H., Atlam M., El-Fishawy N.,*A novel deep autoencoder based survival analysis approach for mi-*

croarray dataset. PeerJ Computer Science, vol. 7, p. e492, 2021. https://doi.org/10.7717/peerj-cs.492

[9] Oladimeji O.O., Ayaz H., McLoughlin I., Unnikrishnan S., *Mutual information-based radiomic feature selection with SHAP explainability for breast cancer diagnosis*, Results in Engineering, Volume 24, 103071, 2024. https://doi.org/10.1016/j.rineng.2024.103071

[10] Cava C., Sabetian S., Salvatore C. et al.*Pan-cancer classification of multi-omics data based on machine learning models*. Netw Model Anal Health Inform Bioinforma, 13(6), 2024. https://doi.org/10.1007/s13721-024-00441-w

[11] Hamla H. and Ghanem K. *A Hybrid Feature Selection Based on Fisher Score and SVM-RFE for Microarray Data*. informatica, 48(1), pp 57-68, 2024. https://doi.org/10.31449/inf.v48i1.4759

[12] Shetty M.V., Jayadevappa D., Tunga S.,*Optimized deformable model-based segmentation and deep learning for lung cancer classification*, The Journal of Medical Investigation, Volume 69, Issues 3–4, pp. 244–255, 2022. https://doi.org/10.2152/jmi.69.244

[13] Kim J., Yoon Y., Park H.-J., Kim Y.-H., *Comparative study of classification algorithms for various DNA microarray data*, Genes, Volume 13, Issue 3, 494, 2022. https://doi.org/10.3390/genes13030494

[14] Alqahtani A., Alsubai S., Sha M., Vilcekova L., Javed T., *Cardiovascular Disease Detection using Ensemble Learning*, Computational Intelligence and Neuroscience, 5267498, 9 pages, 2022. https://doi.org/10.1155/2022/5267498

[15] Khazaee Fadafen M., Rezaee K.,*Ensemble-based multi-tissue classification approach of colorectal cancer histology images using a novel hybrid deep learning framework*, Scientific Reports, Volume 13, 8823, 2023. https://doi.org/10.1038/s41598-023-35431-x

[16] Alfian G., Syafrudin M., Fahrurrozi I., Fitriyani N.L., Atmaji F.T.D., Widodo T., Bahiyah N., Benes F., Rhee J., *Predicting breast cancer from risk factors using SVM and extra-trees-based feature selection method*, Computers, Volume 11, Issue 9, 136, 2022. https://doi.org/10.3390/computers11090136

[17] Ünal H. & Başçiftçi F., *Evolutionary design of neural network architectures: a review of three decades of research*. Artificial Intelligence Review, 55, 2022. https://doi.org/10.1007/s10462-021-10049-5

[18] Kumar S.A., Ananda Kumar T.D., Beeraka N.M., Pujar G.V., Singh M., Akshatha H.S.N., Bhagyalalitha M., *Machine learning and deep learning in data-driven decision making of drug discovery and challenges in high-quality data acquisition in the pharmaceutical industry*, Future Medicinal Chemistry, Volume 14, Issue 4, pp. 245-270, 2021. https://doi.org/10.4155/fmc-2021-0243

[19] Dwaraka S., Vijaya Lakshmi P., David Donald A., Aditya Sai Srinivas T., & Thippanna G., *A Forest of Possibilities: Decision Trees and Beyond*. Journal of Advancement in Parallel Computing, 6(3), pp 29–37, 2023. : https://doi.org/10.5281/zenodo.8372196

[20] Ahmed Nadeem M.S., Waseem M.H., Aziz W., Habib U., Masood A., Attique Khan M.,*Hybridizing artificial neural networks through feature selection based supervised weight initialization and traditional machine learning algorithms for improved colon cancer prediction*, IEEE Access, Volume 12, pp. 97099–97114, 2024. https://doi.org/10.1109/ACCESS.2024.3422317

[21] Ravinder A., & Sharma S. C. *Exploring feature selection and classification algorithms for cardiac arrhythmia disease prediction*. WSEAS Transactions on Biology and Biomedicine, 19, 168–175, 2022. https://doi.org/10.37394/23208.2022.19.19

[22] Goliatt L., Saporetti C. M., Oliveira L. C., & Pereira E. *Performance of evolutionary optimized machine learning for modeling total organic carbon in core samples of shale gas fields*. Petroleum, 10(1), 150–164, 2024. https://doi.org/10.1016/j.petlm.2023.05.005

[23] Maceika A., Bugajev A., Šostak O. R., & Vilutienė T. *Decision tree and AHP methods application for projects assessment: A case study*. Sustainability, 13(10), 5502, 2021. https://doi.org/10.3390/su13105502

[24] Mijwel M.M.,*Artificial neural networks advantages and disadvantages*, Mesopotamian Journal of Big Data, 2021, pp. 29–31, 2021. https://doi.org/10.58496/MJBD/2021/006

[25] Ijaz M. F., Alfian G., Syafrudin M., & Rhee J. *Hybrid prediction model for type 2 diabetes and hypertension using DBSCAN-based outlier detection, synthetic minority over-sampling technique (SMOTE), and random forest*. Applied Sciences, 8(8), 1325, 2018. https://doi.org/10.3390/app8081325

[26] Birzhandi P., Kim K.T., Youn H.Y.,*Reduction of training data for support vector machine: a survey*, Soft Computing, Volume 26, pp. 3729–3742, 2022. https://doi.org/10.1007/s00500-022-06787-5

[27] Maiza M., Chouraqui S., Cherif C., & Taleb-Ahmed A. *Cancer classification through the selection of genes extracted from microarray data*. Przeglad Elektrotechniczny, 101(4), 71–78, 2025. https://doi.org/10.15199/48.2025.04.14

[28] Anosh B. P. S., Annavarapu C. S. R., Dara S., *Clustering-based hybrid feature selection approach for high dimensional microarray data*, Chemometrics and Intelligent Laboratory Systems, Volume 213, 104305, 2021. https://doi.org/10.1016/j.chemolab.2021.104305

[29] Li B., Zhang P., Liang S., Ren G.,*Feature extraction and selection for fault diagnosis of gear using wavelet entropy and mutual information*, In: 2008 9th International Conference on Signal Processing, Beijing, China, 2008; pp. 2846–2850. https://doi.org/10.1109/ICOSP.2008.4697740

[30] Sulaiman M.A., Labadin J.,*Feature selection based on mutual information*, 9th International Conference on IT in Asia (CITA), Sarawak, Malaysia, 2015; pp. 1–6. https://doi.org/10.1109/CITA.2015.7349827

[31] Jalali-Najafabadi F., Stadler M., Dand N., et al., *Application of information theoretic feature selection and machine learning methods for the development of genetic risk prediction models*, Scientific Reports, Volume 11, 23335, 2021. https://doi.org/10.1038/s41598-021-00854-x

[32] Khumukcham R., Urikhimbam B.C., Nazrul H., Dhruba K. B.,*JoMIC: A joint MI-based filter feature selection method*, Journal of Computational Mathematics and Data Science, Volume 6, 100075, 2023. https://doi.org/10.1016/j.jcmds.2023.100075

[33] Jain P.K., Jain M. & Pamula R., *Explaining and predicting employees'attrition: a machine learning approach*. SN Appl. Sci. 2, 757, 2020. https://doi.org/10.1007/s42452-020-2519-4

[34] Ginny Y. Wong, Frank H.F. Leung, Sai-Ho Ling,*A hybrid evolutionary preprocessing method for imbalanced datasets*, Information Sciences, Volumes 454–455,pp 161-177, 2018. https://doi.org/10.1016/j.ins.2018.04.068

[35] Xinteng G., Xinggao L.,*A novel effective diagnosis model based on optimized least squares support machine for gene microarray*, Applied Soft Computing, Volume 66, pp 50-59,2018. https://doi.org/10.1016/j.asoc.2018.02.009

[36] Houssein E.H., Abdelminaam D.S., Hassan H.N., Al-Sayed M.M., Nabil E.,*A hybrid barnacles mating optimizer algorithm with support vector machines for gene selection of microarray cancer classification*, IEEE Access, Volume 9, pp. 64895–64905, 2021. https://doi.org/10.1109/ACCESS.2021.3075942

[37] Giraud C.,*Introduction to high-dimensional statistics*, 2nd ed. Chapman and Hall/CRC, 2021. https://doi.org/10.1201/9781003158745

[38] Cherif C., Abdi M.K.,Ahmad A. and Maiza M.,*Predictive approach to the degree of business process change*, International Journal of Computing and Digital Systems, 14(1), pp. 10505-10513, Dec. 2023. http://dx.doi.org/10.12785/ijcds/1401117

[39] Kou L., Yuan Y., Sun J. and Lin Y., *Prediction of Cancer Based on Mobile Cloud Computing and SVM*, International Conference on Dependable Systems and Their Applications (DSA), Beijing, China, pp. 73-76, 2017. https://doi.org/10.1109/DSA.2017.20

Fusion of Convolutional Architecture and Transformer Models for Enhanced Brain Tumor Classification

V. Sabitha[1,2*], Jagannath Nayak[3] , P. Ramana Reddy[4]
[1]Research Scholar,Department of Ece, Jntua, Ananthapuramu, Andhra Pradesh, 515002 India
2Department of ECE, Vaagdevi College of Engineering, Warangal, Telangana, 506002 India
[3]Professor, Department of Ece, Jntuace, Ananthapuramu, 515002, India
[4]Chess, Drdo Ministry of Defence, Government of India, Hyderabad, 500069 India
sabithavem@gmail.com[2]jnayakdr@gmail.com[3]prrjntu@gmail.com
[*]Corresponding author

Keywords: brain tumor, transformer model, CNN, MRI, deep learning

Recieved: March 10, 2025

Early detection of brain tumors based on MRI images has shown significant advancements with the advent of deep learning methods. However, achieving high accuracy and robustness in classification remains a challenge due to the complex and mixed nature of brain tumors and the clarity of samples. This study proposes a novel approach that integrates convolutional architectures with the transformer approach, which can lead to an optimal model. The convolutional neural networks (CNNs) excel in capturing local features and spatial hierarchies, while the transformer approach captures long-term dependencies and contextual information. By integrating these two robust architectures, our proposed model leverages the strengths of both to achieve superior performance. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS) dataset is used to evaluate our model, which consists of 7023 samples across four classes. We compare the performance of the fusion model with that of the prescribed models. The results demonstrate that the fusion model significantly outperforms the standalone models, achieving a classification accuracy of 91.8%. The proposed approach also shows improved robustness in handling various tumor types and sizes, highlighting its potential for clinical application.

Povzetek: Za klasifikacijo možganskih tumorjev iz MRI (BRATS, 7023 vzorcev, 4 razredi) so uporabili hibridni fuzijski model, ki združi CNN (lokalne značilke) in transformer (globalni kontekst) za robustnejšo klasifikacijo heterogenih tumorjev.

1 Introduction

Brain tumors are the most challenging and life-threatening situations, requiring accurate diagnosis and effective treatment planning. Automatic early detection of tumors will overcome the threatening situations. Magnetic Resonance Imaging (MRI) samples are used for tumor detection and classification due to their superior contrast resolution and non-invasive nature. The early detection of tumors from MRI samples by Tampu, I. E., et al. (2024) [14]is crucial for determining appropriate treatment strategies and predicting patient outcomes. Traditional methods for brain tumor classification are mainly based on manual inspection and human analysis, which is a time-consuming process. As the number of patients increases day by day, manual detection becomes prone to variability, necessitating the development of an automated system. Many researchers have worked on deep learning on medical images to diagnose diseases, as seen in Odusami, M. (2024) [17].

In recent years, the use of deep learning (DL)in the field of medical image analysis has offered automated and highly accurate solutions for various diagnostic tasks. CNNs, Nobel, S. N., et al (2024) [3] in particular, have shown remarkable success in extracting hierarchical features from medical images and achieving high performance in classification tasks. However, despite their efficacy, CNNs have some limitations. For instance, these models have captured complex patterns and sequential patterns from an image, which are necessary for accurately classifying complex and heterogeneous brain tumors.

Transformers, a cutting-edge approach implemented for text-based data, have demonstrated their capability to capture sequential patterns and global patterns through self-attention mechanisms (Katran, L. F., et al., 2024) [4]. Their application to vision tasks has opened new avenues for enhancing image analysis performance. While transformers are capable of capturing long-term dependencies, they may struggle with capturing fine-grained local features due to their inherently global nature (Srinivas, B., et al, 2024) [11].

This paper proposes a novel approach combining CNN and transformer methods to enhance the strengths of both paradigms for improved brain tumor classification. By combining CNNs' ability to capture local features and transformers' proficiency in modeling global context, the proposed hybrid model aims to achieve superior classification performance. This fusion approach is expected to address the limitations of standalone CNN and transformer models, providing a more robust and accurate classification framework. According to Chen, C., et al. (2023) [19], many of the systems implemented a transformer model to detect brain tumors.

The Multimodal BRATS dataset, a widely recognized and comprehensive dataset, is utilized to evaluate the performance of the proposed model. Extensive experiments are conducted to compare the performance of the fused model against state-of-the-art CNN and transformer-based models individually. Our results demonstrate that the fusion model outperforms the other models.

The paper is organized as follows: Section 2 reviews related work in brain tumor classification using DL. Section 3 describes the proposed fusion model architecture. Section 4 presents the experimental setup, including dataset details. Section 5 explores the experimental results analysis and comparison with prescribed models. Finally, Section 6 concludes the paper.

2 Related work

Hekmat et al (2025) [1] implemented an attention-based architecture for brain tumor detection. The model uses attention mechanisms to fuse different feature representations effectively, enhancing the accuracy of tumor detection in MRI scans. By clinicians to better understand the decision-making process. Extracted features from key regions of interest within MRI images, this method outperforms traditional CNN. Benzorgat, N. et al (2024) [2] proposed brain tumor classification by combining an ensemble of models with a transformer. With transformers, which capture global dependencies, and DL models that specialize in local features? The integrated model got an accuracy of 0.97. Nobel, S. N., et al. (2024) [3] proposed a hybrid model, a mixed convolutional-transformer model, aimed at diagnosing glioma subtypes rapidly and accurately. They combined CNN layers, which efficiently capture spatial information, with transformers to handle long-range dependencies. This hybrid model significantly improves the accuracy by 0.98. Mzoughi, H et al (2024) [5] Combined Vision Transformers (ViT) with Deep-CNN for classification of tumor images, incorporating explainable AI (XAI) for interpretability. The integration of the ViT and D-CNN models will learn both global and local features effectively, achieving an accuracy of 0.96. Alzahrani, S. M., and Qahtani, A. M. (2024) [6] worked with tripartite attention for multi-class brain tumor detection in highly augmented MRIs. They improved the generalization of models trained on augmented datasets by distilling knowledge from larger models into more compact ones. And got an accuracy of 0.97. Nguyen-Tat, T. B., (2024) [7] Proposed a hybrid approach for brain tumor segmentation that combines UNet, attention mechanisms, and transformers. This method integrates the strengths of each technique, with UNet efficiently capturing spatial features, transformers handling long-range dependencies, and attention mechanisms focusing on relevant regions. As a result, they achieved an accuracy of 0.91.

Gasmi, K., et al. (2024) [8] proposed an enhanced brain tumor diagnosis model that combines DL with a weight selection technique. This method aims to optimize the learning process by selecting the most relevant features and assigning them appropriate weights. Rasheed, Z., et al. (2024) [9] implemented a hybrid CNN model with an attention method for brain tumor identification. We improved the performance of CNNs by focusing on complex patterns from images using attention layers, achieving an accuracy of 0.97. Pacal, I. (2024) [10] proposed a Transformer method by adding a multi-layer perceptron and self-attention methods for diagnosing tumors automatically. The Transformer is known for its efficient handling of high-resolution images and is combined with a residual MLP to improve feature learning and classification accuracy. Kang, M., et al (2024) [12] Implemented a CNN-transformer network for brain tumor segmentation in cases with incomplete modalities. The method aims to address the challenge of missing or incomplete MRI data by distilling features from available modalities and utilizing the CNN-transformer architecture to refine the segmentation.

Asiri, A. A et al. (2024) [13] implemented the Swin Transformer for accurate brain tumor classification and performance analysis. The Swin Transformer can handle high-resolution images, and it is applied to the classification task to improve diagnostic accuracy. The paper also focuses on performance analysis, comparing the results with other state-of-the-art methods. Tabatabaei, S., et al. (2023) [15] proposed an attention method and DL architecture for tumor classification. The attention mechanism with the DL method will enable the model to focus on complex areas of samples, improving the accuracy of tumor classification. The model combines the benefits of attention-based transformers with traditional methods, leading to enhanced performance in tumor detection. Aloraini, M., et al. (2023) [16] implemented a transformer with CNN for effective brain tumor classification using MRI images. This hybrid model uses the strengths of both approaches: CNNs for local feature extraction and transformers for global dependency modeling. This combination leads to enhanced tumor classification accuracy**.** Sun, X., et al (2024) [18] implemented aEF-UV method for a feature-enhancement of U-Net and ViT for tumor segmentation. This approach uses the strengths of U-Net for segmentation and ViT for capturing long-range dependencies in the image. The fusion of these models enhances feature extraction and segmentation accuracy, particularly in complex brain tumor cases. Saleh et al. (2024) [20] implemented a multimodal approach for semantic segmentation in brain tumor images, integrating advanced models and optimal filters via advanced 3D segmentation methods. They used multiple imaging modalities to improve the segmentation accuracy by capturing complementary information from different sources. Zebari, N. A., et al. (2024) [21] proposed a DL model for detecting brain tumors from image samples.

And integrated multiple DL techniques to enhance the performance by fusing different features from various sources of samples.

Zakariah, M., et al. (2024) [22] proposed a Dual ViT with DSUNET for brain tumor segmentation. The feature fusion mechanism will demonstrate the model's ability to capture various patterns from MRI images by leveraging the strengths of Vision Transformers and deep segmentation networks. The dual model ensures that the spatial and contextual features are well-represented, leading to improved segmentation results. Nazir, K., et al. (2023) [23] implemented a 3D Convolutional method for tumor segmentation in MRI imaging. The feature pyramid network structure is enhanced with Kronecker convolutional layers, which capture features and improve segmentation accuracy. The 3D nature of the model allows it to handle volumetric data, which is particularly important for brain tumor segmentation in medical imaging. Ramamoorthy, H., et al. (2023) [24] implemented TransAttU-Net, a deep neural network for brain tumor segmentation in MRI images. The model combines a basic method with an attention method to improve the segmentation of tumors by emphasizing relevant features. The combination of attention systems enables the model to focus on tumor areas in images, which is potentially important for better segmentation results. Ramakrishnan, A. B., et al (2024) [25] proposed a hybrid CNN architecture for improved accuracy. We utilized oneAPI optimization techniques to adjust the weights and enhance the performance of the hybrid CNN model. By combining CNNs with optimization frameworks, the model achieves efficient classification while maintaining high accuracy.

3 Methodology

A CNN-Transformer Fusion Model is implemented to extract the spatial feature extraction capabilities of CNNs and the global contextual understanding of transformers for accurate brain tumor classification. The method involves three key components: feature extraction, sequence modeling, and classification, all underpinned by rigorous mathematical formulations as shown in Figure 1.

Feature Extraction: The input image is represented as $X \in R^{c_i * H * W}, where c_i = 3\ for$RGB color encoding, H is height and W is the width. CNN extracts the spatial features depth wise separable convolutions, producing a feature map$F \in R^{c_i * H * W}$ with equation (1).

$$F = \emptyset_{CNN}(X) \qquad (1)$$

Where $C_{out} = 1280, and\ H\ and\ W\ are\ redused\ spatial\ features.$ by aggregating all spatial features, applied global average pooling method with equation (2), for compacting all features (f_c).

$$f_c = \frac{1}{H'W'} \sum_{i=1}^{H'} \sum_{j=1}^{W'} F_{c,i,j} \qquad (2)$$

Sequential Modeling with Transformer Encoder: The pooled feature vector $f$is reshaped into a single-token sequence as $T = R^{1 * C_{out}(1280)}$. this sequence is transfer to encoder, which consists of 3 layers, each layer will have multi head self attention method and positional level feed forward method. The multi head attention method captured Query (Q), Key (K) and Value (V) from each vector with equation (3), (4) and (5). Where W is weights as the input dimension. The dot product of attention method is computed with equation (6).

$$Q_h = TW_h^Q (3)$$

$$K_h = TW_h^k (4)$$

$$V_h = TW_h^v (5)$$

$$Att(Q_h, K_h, V_h) = softmax\left(\frac{Q_h H_h^T}{\sqrt{d_k}}\right) V_h (6)$$

The output of all attention methods is concatenated linearly, and then it will provide final attention output.

Position-wise feed forward Network (FFN): In this each token will be considered into a 2 layer feed forward transformation, by equation (7) and positional level embedding with equation (8). In this W and b variables are updated parameters.

$$FFN(z) = \sigma(zW_1 * b_1)W_2 + b_2 \qquad (7)$$

$$PE_{pos,2i} = sin\left(\frac{pos}{10000^{d\frac{2i}{model}}}\right) \qquad (8)$$

The output is transformed through three layers of transformers. For the contextual embedding layer, the first token is passed to a fully connected layer for classification using equation (9) with these spatial and temporal features combined to give the final output.

$$yp = \emptyset_{FC}(z)(9).$$

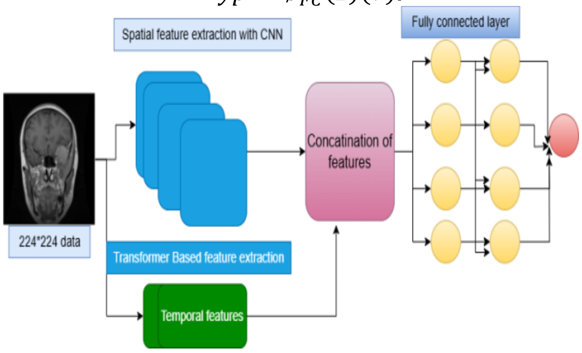

Spatial feature extraction with CNN
Fully connected layer
Concatination of features
224*224 data
Transformer Based feature extraction
Temporal features

Figure 1: Proposed fusion models for brain tumor detection

3.1 Data set

The proposed model was trained on a Kaggle BRATS data set, which combines four classes: glioma, meningioma, no tumor, and pituitary. This dataset comprises 7023 brain images. All the samples are preprocessed into a 224*224 size. All the samples are then separated into training and testing sets in an 80:20 ratio. The samples of brain MRI are shown in Figure 2. All the samples are normalized to 0.465, 0.446, 0.416, with a standard deviation of 0.229, 0.224, 0.225, respectively. This ensures that no sample will dominate the other low-resolution samples.

3.2 hardware used for training

The proposed model was implemented using Python with TensorFlow and Keras libraries. All experiments were conducted on the Kaggle platform using a Tesla T4 GPU (16 GB VRAM) environment. The training was conducted for 10 epochs with a batch size of 32, using the Adam optimizer with an initial learning rate of 0.0001. A dropout rate of 0.2 was applied to reduce overfitting.

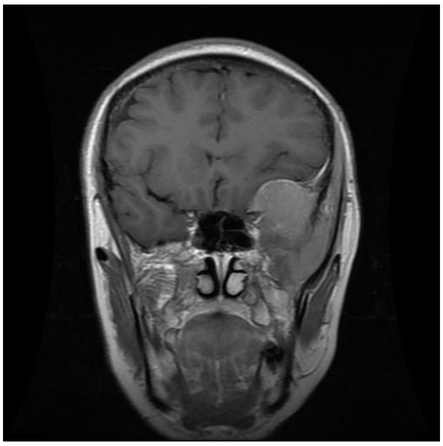


Figure 2: Sample brain MRI image.

4 Result analysis

The proposed fusion approach is iterated for 10 epochs, with a batch size of 16, and a learning rate of 0.0001, as shown in Table 1. The model achieved an accuracy of 74.72% with a training loss of 0.6639, while the test accuracy reached 86.92%, accompanied by a test loss of 0.4340. This indicates a strong baseline performance, likely attributed to the combination of MobileNetV2's efficient feature extraction and the Transformer's contextual understanding. Over successive epochs, the training accuracy improved steadily, reaching 91.88% by the final epoch, with the training loss decreasing to 0.2163. Similarly, the test accuracy increased to 91.76%, while the test loss reduced significantly to 0.1891, showcasing the model's enhanced capability to classify tumor categories accurately. From Figure 3, a marked improvement in test accuracy was observed between Epochs 8 and 10, where the model transitioned from 89.99% to 91.76%, with a corresponding reduction in test loss from 0.2319 to 0.1891.

Table 1: Parameters used for training the model

Parameter	Value
No. of Attention Heads	8
Hidden Size (FFN)	512
Dropout Rate	0.2
Optimizer	Adam
Learning Rate	0.0001
Batch Size	32

The model achieves strong performance in the "Notumor" and "Pituitary" categories, with particularly high predictive reliability, evidenced by near-perfect metrics. The performance for "Glioma" and "Meningioma" shows slightly lower but still competitive results. These variations may stem from potential similarities in visual patterns between these tumor types, challenging the model's discriminative power. Nevertheless, the consistent improvement observed across all categories highlights the model's capacity to learn complex representations and adapt to varying class-specific patterns.

The overall classification observed from Table 2, with an accuracy of 91.8% across 841 test samples, underscores the model's generalization ability. Additionally, both the macro and weighted averages indicate a balanced performance across classes, ensuring that no individual category dominates or suffers from significant misclassification. Class-wise accuracy is illustrated in Figures 4 and 5.

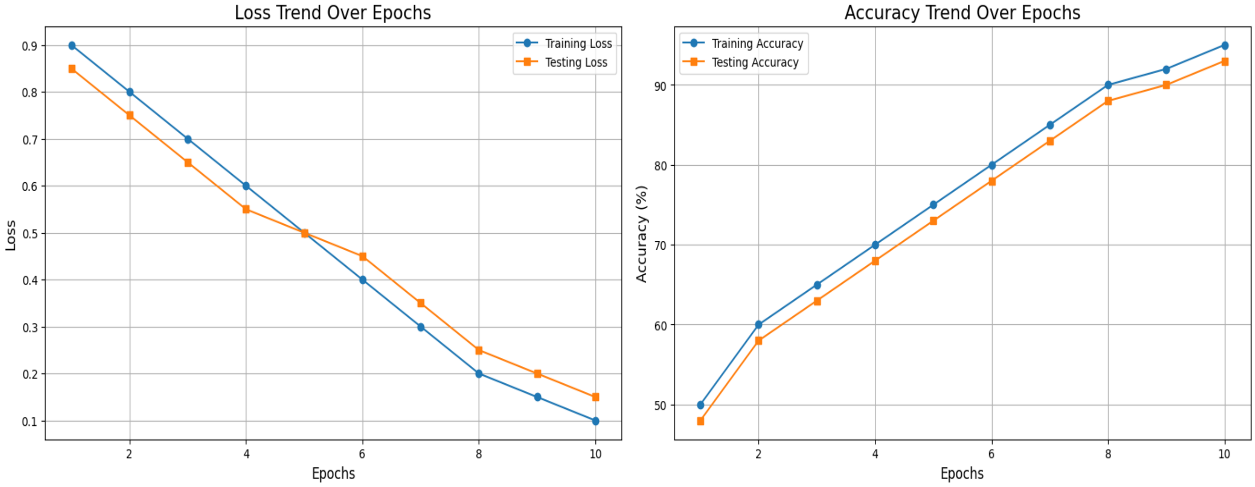


Figure 3: learning curves of the fusion model

Table 2: Performance of the proposed model

	P(%)	R(%)	F1(%)	Support

Glioma	93	89	91	190
Meningioma	91	85	87	186
Notumor	91	99	96	285
Pituitary	92	99	96	180
ACC			91.8	841
M-avg	92	91.5	91.8	841
W-avg	92	91.5	91.8	841

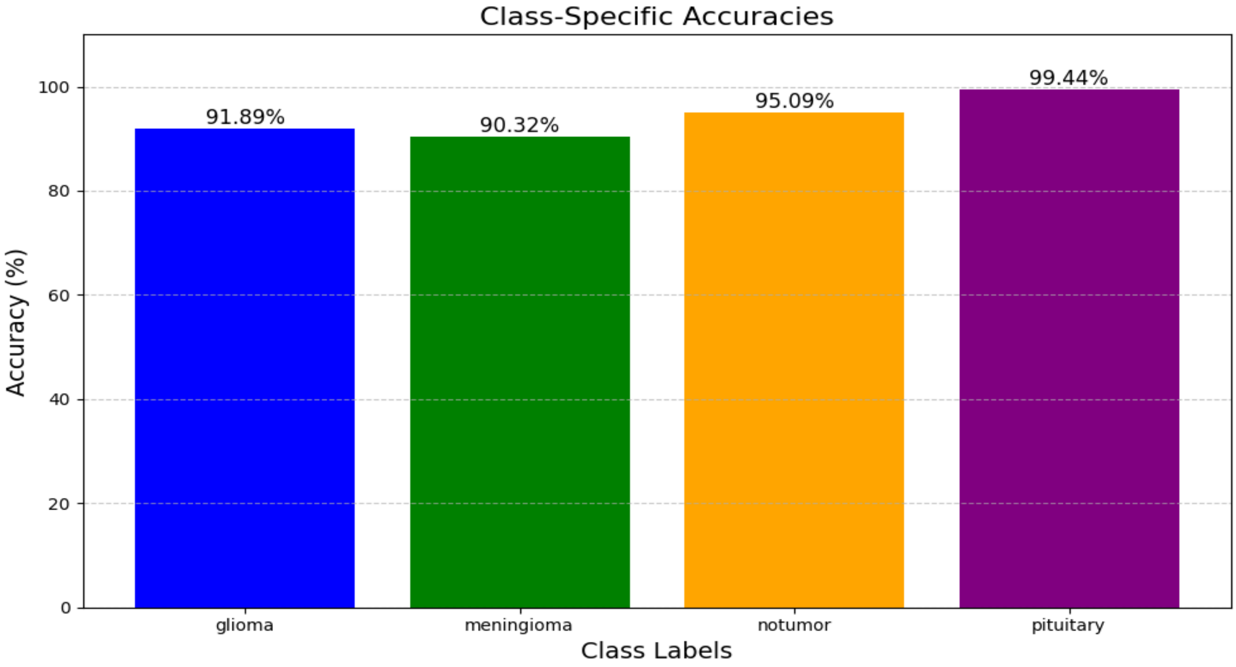


Figure 4: Class-wise performance of the proposed model

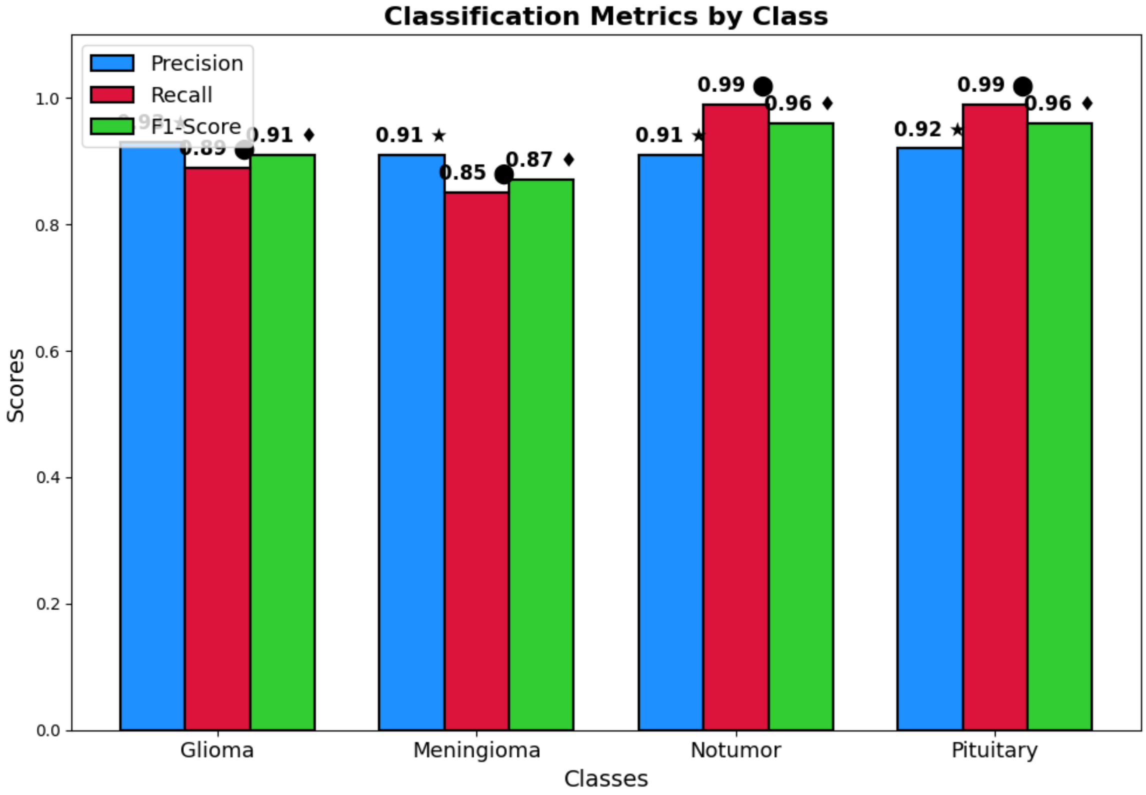


Figure 5: Class-wise performance of the fusion model in terms of precision, recall, and F1-score

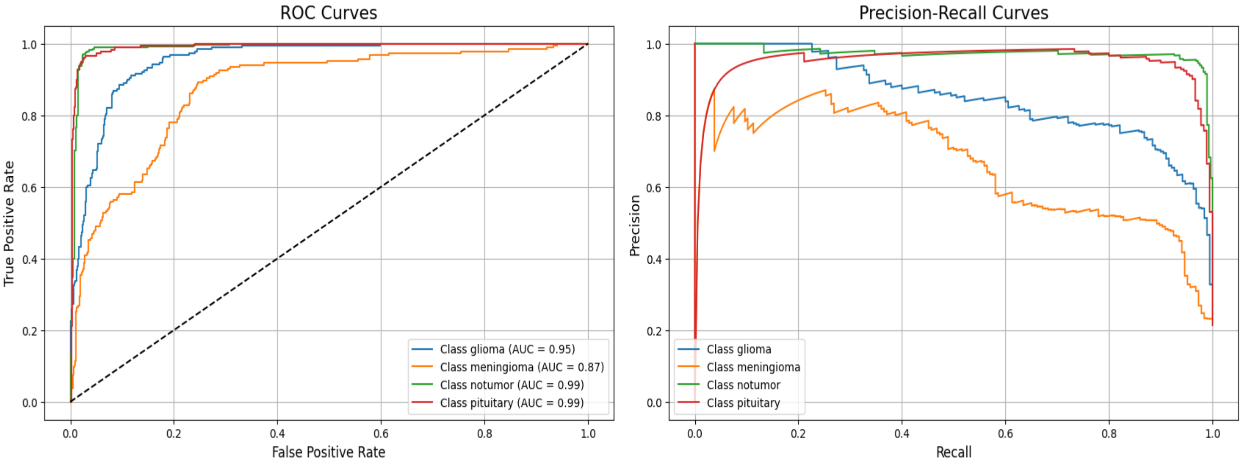


Figure 6: ROC and PR curve of proposed models

From Figure 6, the area under the ROC curve (AUC) highlights the model's effectiveness, with Glioma, Notumor, and Pituitary classes achieving high AUC values, indicating strong discrimination capabilities. However, the Meningioma class demonstrates slightly lower AUC, reflecting challenges in accurately distinguishing this class. Similarly, precision-recall curves reveal the relationship between positive prediction precision and sensitivity across different thresholds. Classes such as Notumor and Pituitary exhibit high performance, showcasing the robustness of the model in these cases. In contrast, the performance for the Meningioma class is comparatively modest, emphasizing areas for potential refinement.

Figures 7 and 8 illustrate feature maps extracted by the convolutional layers of the model for a sample input image. These maps provide a visual representation of the learned features at different layers, highlighting areas of importance and attention within the image. The feature maps capture various patterns, ranging from simple edges and textures in initial layers to more abstract and class-specific features in deeper layers. Bright regions within the maps indicate areas with strong activations.

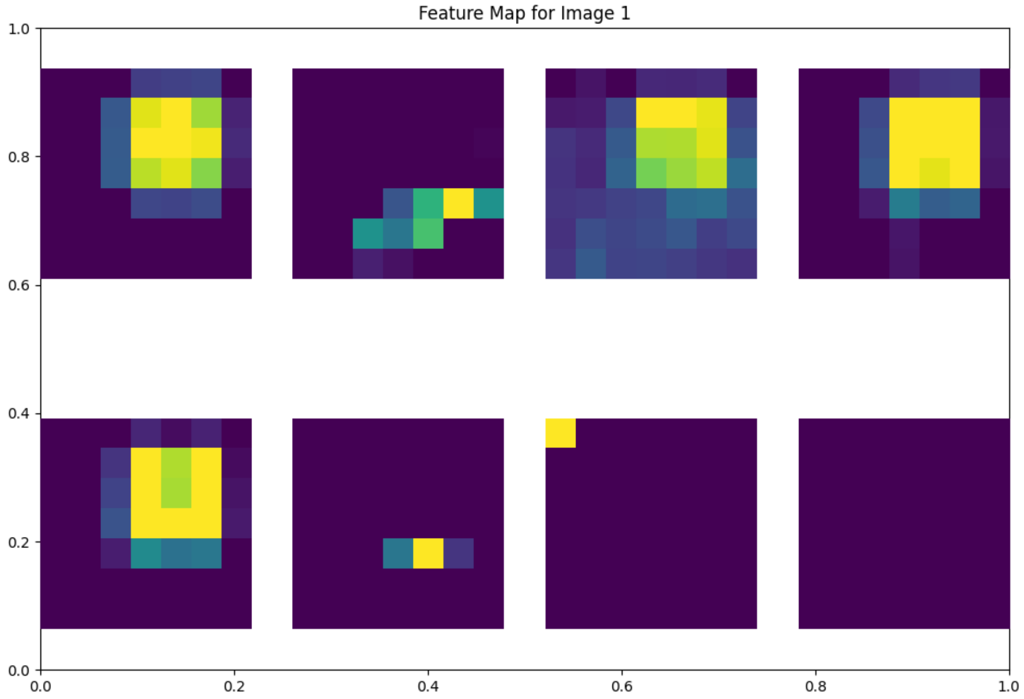


Figure 7 Feature extraction map of sample image-1

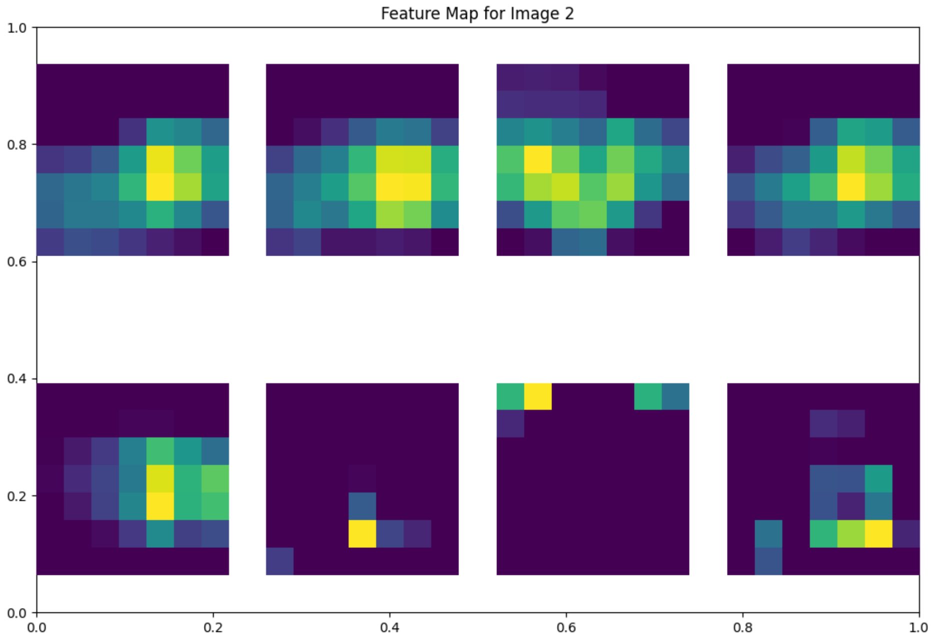


Figure 8 Feature extraction map of sample image-2

Figure 9 illustrates successful predictions by the model, where both the actual and predicted labels are identified as "glioma." These results indicate that the model effectively captured key features associated with gliomas, allowing for accurate classification. From Figure 10, where the actual label is "glioma," but the model incorrectly predicted "pituitary." Such an error highlights the overlap or similarity in visual features between glioma and pituitary cases, which may have led to confusion in the model's classification process.

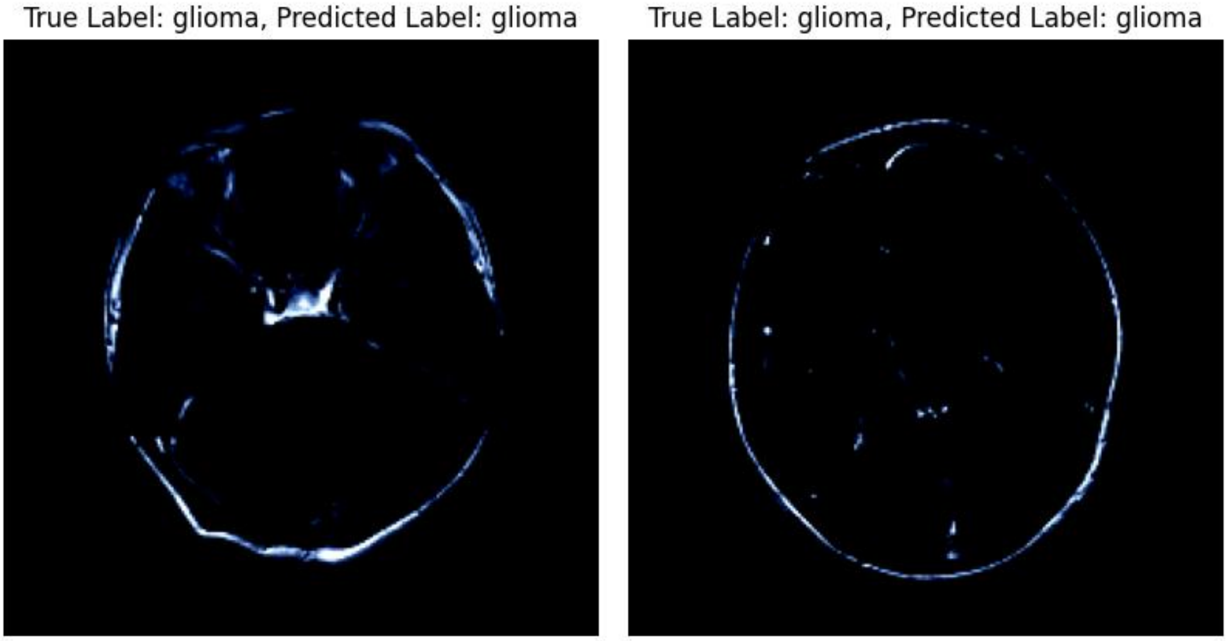

True Label: glioma, Predicted Label: glioma
True Label: glioma, Predicted Label: glioma

Figure 9: Actual and predicted labels of the proposed model after training

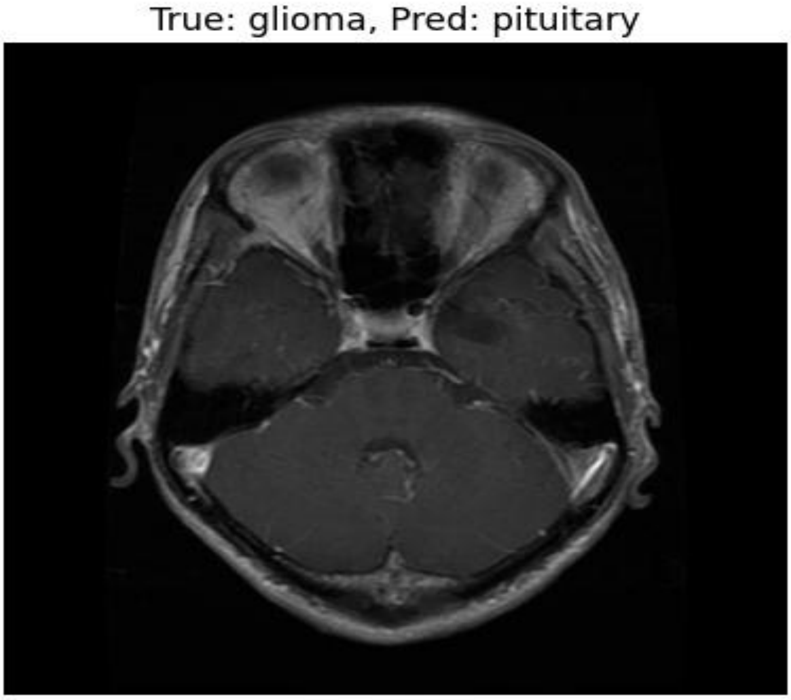


Figure 10: Misclassified sample by the proposed models

Table 3: Comparison of the proposed model with the prescribed models

Citation No.	**Methodology**	**Dataset Used**	**Accuracy(%)**
[22]	Dual Vision Transformer-DSUNET for brain tumor segmentation	MRI Brain Tumor	90.00
[26]	Gated residual recurrent neural networks	BraTS, ISBI	89.1
[27]	deep learning	BRATS	86.2
[28]	UTNet	BRATS	87.8
Proposed model	CNN-Transformer Fusion model	MRI Brain Tumor	91.8

Table 3 presents the performance of various methodologies for brain tumor segmentation and classification tasks using different datasets. The Dual Vision Transformer-DSUNET model, as reported in [22], achieves an accuracy of 90% on the same dataset. Similarly, the Gated Residual Recurrent Neural Networks employed in [26] show an accuracy of 89.1% when evaluated on the BraTS and ISBI datasets, reflecting their capability in processing temporal and spatial information. A deep learning-based approach utilized in [27] achieved an accuracy of 86.2% on the BRATS dataset, indicating its utility, albeit with slightly lower performance. The UTNet model, proposed in [28], reported an accuracy of 87.8% on the same BRATS dataset, leveraging its unique architectural enhancements for tumor segmentation. In comparison, the proposed CNN-Transformer Fusion model achieves an accuracy of 91.8% on the MRI Brain Tumor dataset, showcasing its superior ability to integrate the strengths of convolutional neural networks and transformers, resulting in improved feature representation and classification performance.

5 Conclusion

In this study, a hybrid CNN-Transformer Fusion Model was implemented for enhanced brain tumor classification. The model effectively combines the localized features, which are extracted with CNNs, with the global contextual understanding provided by Transformers. Comprehensive evaluations on a diverse dataset reveal the model's robust performance, achieving an overall accuracy of 91.8%, surpassing several existing state-of-the-art methods. The integration of CNN and a multi-layer Transformer Encoder enables the approach to learn complex spatial and temporal features, improving its performance to classify tumor types with high consistency. At the same time, the model demonstrates remarkable performance in distinguishing "No Tumor" and "Pituitary" classes, minor challenges in classifying "Glioma" and "Meningioma" highlight opportunities for further optimization. Future work will focus on augmenting the dataset with additional samples and exploring advanced Transformer architectures to enhance discriminative capabilities.

References

[1] Hekmat, A., Zhang, Z., Khan, S. U. R., Shad, I., & Bilal, O. (2025). An attention-fused architecture for brain tumor diagnosis. *Biomedical Signal Processing and Control*, *101*,10.1016/j.bspc.2024.107221

[2] Benzorgat, N., Xia, K., &Benzorgat, M. N. E. (2024). Enhancing brain tumor MRI classification with an ensemble of deep learning models and transformer integration. *PeerJ Computer Science*, *10*, e2425. https://doi.org/10.7717/peerj-cs.2425

[3] Nobel, S. N., Swapno, S. M. R., Islam, M. B., Azad, A. K. M., Alyami, S. A., Alamin, M., ... & Moni, M. A. (2024). A Novel Mixed Convolution Transformer Model for the Fast and Accurate Diagnosis of Glioma Subtypes. *Advanced Intelligent Systems*, 2400566. https://doi.org/10.1002/aisy.202400566

[4] Katran, L. F., AlShemmary, E. N., & Al-Jawher, W. A. (2024). A Review of Transformer Networks in MRI Image Classification. *Al-Furat Journal of Innovations in Electronics and Computer Engineering*, 148-162. DOI:10.46649/fjiece.v3.2.12a.21.5.2024

[5] Mzoughi, H., Njeh, I., BenSlima, M., Farhat, N., &Mhiri, C. (2024). Vision transformers (ViT) and deep convolutional neural network (D-CNN)-based models for MRI brain primary tumors images multi-classification supported by explainable artificial intelligence (XAI). *The Visual Computer*, 1-20. DOI:10.1007/s00371-024-03524-x

[6] Alzahrani, S. M., & Qahtani, A. M. (2024). Knowledge distillation in transformers with tripartite attention: Multiclass brain tumor detection in highly augmented MRIs. *Journal of King Saud University-Computer and Information Sciences*, *36*(1), 101907. https://doi.org/10.1016/j.jksuci.2023.101907

[7] Nguyen-Tat, T. B., Nguyen, T. Q. T., Nguyen, H. N., & Ngo, V. M. (2024). Enhancing brain tumor segmentation in MRI images: A hybrid approach using UNet, attention mechanisms, and transformers. *Egyptian Informatics Journal*, *27*, 100528. DOI:10.13140/RG.2.2.18164.36485

[8] Gasmi, K., Ben Aoun, N., Alsalem, K., Ltaifa, I. B., Alrashdi, I., Ammar, L. B., ... & Shehab, A. (2024). Enhanced brain tumor diagnosis using combined deep learning models and weight selection technique. *Frontiers in Neuroinformatics*, *18*, 1444650. https://doi.org/10.3389/fninf.2024.1444650

[9] Rasheed, Z., Ma, Y. K., Ullah, I., Al-Khasawneh, M., Almutairi, S. S., &Abohashrh, M. (2024). Integrating Convolutional Neural Networks with Attention Mechanisms for Magnetic Resonance Imaging-Based Classification of Brain Tumors. *Bioengineering*, *11*(7), 701. https://doi.org/10.3390/bioengineering11070701

[10] Pacal, I. (2024). A novel Swin transformer approach utilizing residual multi-layer perceptron for diagnosing brain tumors in MRI images. *International Journal of Machine Learning and Cybernetics*, 1-19. https://doi.org/10.1007/s13042-024-02110-w

[11] Srinivas, B., Anilkumar, B., devi, N., & Aruna, V. B. K. L. (2024). A fine-tuned transformer model for brain tumor detection and classification. *Multimedia Tools and Applications*, 1-25. DOI:10.1007/s11042-024-19652-4

[12] Kang, M., Ting, F. F., Phan, R. C. W., Ge, Z., & Ting, C. M. (2024). A Multimodal Feature Distillation with CNN-Transformer Network for Brain Tumor Segmentation with Incomplete Modalities. *arXiv preprint arXiv:2404.14019*. https://doi.org/10.48550/arXiv.2404.14019

[13] Asiri, A. A., Shaf, A., Ali, T., Pasha, M. A., Khan, A., Irfan, M., & Alamri, S. (2024). Advancing brain tumor detection: harnessing the Swin Transformer’s power for accurate classification and performance analysis. *PeerJ Computer Science*, *10*, e1867. https://doi.org/10.7717/peerj-cs.1867

[14] Tampu, I. E., Bianchessi, T., Blystad, I., Lundberg, P., Nyman, P., Eklund, A., & Haj-Hosseini, N. (2024). Pediatric brain tumor classification using deep learning on MR-images with age fusion. *Neuro-Oncology Advances*, vdae205. https://doi.org/10.1093/noajnl/vdae205

[15] Tabatabaei, S., Rezaee, K., & Zhu, M. (2023). Attention transformer mechanism and fusion-based deep learning architecture for MRI brain tumor classification system. *Biomedical Signal Processing and Control*, *86*, 105119.

[16] Aloraini, M., Khan, A., Aladhadh, S., Habib, S., Alsharekh, M. F., & Islam, M. (2023). Combining the transformer and convolution for effective brain tumor classification using MRI images. *Applied Sciences*, *13*(6), 3680.DOI:10.3390/app13063680

[17] Odusami, M., Damasevicius, R., Milieskaite-Belousoviene, E., &Maskeliunas, R. (2024). Multimodal Neuroimaging Fusion for Alzheimer's Disease: An Image Colorization Approach With Mobile Vision Transformer. *International Journal of Imaging Systems and Technology*, *34*(5), e23158. https://doi.org/10.1002/ima.23158

[18] Sun, X., Bhatti, U. A., Huang, M., & Zhang, Y. (2024). EF-UV: Feature Enhanced fusion of U-Net and VIT Transformer for Brain Tumor MRI Image Segmentation. DOI:10.21203/rs.3.rs-5329372/v1

[19] Chen, C., Wang, H., Chen, Y., Yin, Z., Yang, X., Ning, H., ... & Zhao, J. (2023). Understanding the brain with attention: A survey of transformers in brain sciences. *Brain-X*, *1*(3), e29. https://doi.org/10.1002/brx2.29

[20] Saleh, A. H., Atila, Ü., & Menemencioğlu, O. (2024). Multimodal Fusion for Enhanced Semantic Segmentation in Brain Tumor Imaging: Integrating Deep Learning and Guided Filtering Via Advanced 3D Semantic Segmentation Architectures. *International Journal of Imaging Systems and Technology*, *34*(5), e23152. http://dx.doi.org/10.1002/ima.23152

[21] Zebari, N. A., Mohammed, C. N., Zebari, D. A., Mohammed, M. A., Zeebaree, D. Q., Marhoon, H. A., ... & Martinek, R. (2024). A deep learning fusion model for accurate classification of brain tumours in Magnetic Resonance images. *CAAI Transactions on Intelligence Technology*. http://dx.doi.org/10.1049/cit2.12276

[22] Zakariah, M., Al-Razgan, M., &Alfakih, T. (2024). Dual Vision Transformer-DSUNET With Feature Fusion for Brain Tumor Segmentation. https://doi.org/10.1016/j.heliyon.2024.e37804

[23] Nazir, K., Madni, T. M., Janjua, U. I., Javed, U., Khan, M. A., Tariq, U., & Cha, J. H. (2023). 3D Kronecker Convolutional Feature Pyramid for Brain Tumor Semantic Segmentation in MR Imaging. *Computers, Materials & Continua*, *76*(3). DOI:10.32604/cmc.2023.039181

[24] Ramamoorthy, H., Ramasundaram, M., Raj, R. S. P., &Randive, K. (2023). TransAttU-Net Deep Neural Network for Brain Tumor Segmentation in Magnetic Resonance Imaging Réseau neuronal profondTransAttU-Net pour la segmentation des tumeurscérébrales avec l’imagerie par résonancemagnétique. *IEEE Canadian Journal of Electrical and Computer Engineering*. DOI:10.1109/ICJECE.2023.3289609

[25] Ramakrishnan, A. B., Sridevi, M., Vasudevan, S. K., Manikandan, R., &Gandomi, A. H. (2024). Optimizing brain tumor classification with hybrid CNN architecture: Balancing accuracy and efficiency through oneAPI optimization. *Informatics in Medicine Unlocked*, *44*, 101436. DOI:10.1016/j.imu.2023.101436

[26] Chen, J., Li, Y., Jin, Y., Luo, X., & Lu, G. (2019). Gated residual recurrent neural networks for multi-modal medical image segmentation. *In International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 354-362. https://doi.org/10.1007/978-3-030-32248-9_40

[27] Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., & Jorge Cardoso, M. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *In Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (DLMIA)*, 240-248. https://doi.org/10.1007/978-3-319-67558-9_28

[28] Gao, Y., Zhou, M., Metaxas, D. N., & Li, K. (2018). UTNet: a hybrid transformer architecture for medical image segmentation. *IEEE Transactions on Medical Imaging*, *37*(6), 1413-1423. https://doi.org/10.1109/TMI.2018.2806960

Convolutional Neural Network (CNN) Based Martian Dune Detection

Nayama Valsa Scariah[1*], Mili Ghosh Nee Lala[1], Akhouri Pramod Krishna[1]
Birla Institute of Technology, Mesra, Ranchi, India[1]
E-mail: nayama99@gmail.com, mili@bitmesra.ac.in, apkrishna@bitmesra.ac.in
*Corresponding author

Keywords: aeolian, Martian dune, CNN, U-Net, ResUNet, ResUNet++

Received: October 10, 2024

Sand dunes are one of the most prominent Aeolian landforms present on the Martian surface. Accumulation and erosion of sand particles cause the formation of dunes, which possibly can influence the Martian climate too. For mapping such landforms over large areas of the Martian surface more effectively, automated detection of dunes has been brought out. For this a convolutional neural network (CNN) based detection approach has been implemented considering application of different models and assessing their respective performance using different sets of Orbiter images. CNN architectures such as U-net, ResUNet and ResUNet++ were used for segmentation of dunes over the Martian surface. CNN produced segmentation results with greater accuracy with advantage of designing new models and using different loss functions. Convolution neural networks such as U-Net, ResUNet, and ResUNet++ for detecting dunes on Martian surface used Context camera (CTX) and the High-Resolution Imaging Experiment (HiRISE) images of Mars Reconnaissance Orbiter (MRO) to generate the suitable models considering two different Martian sites, Gale crater and Nili Patera. The models thus generated were tested over Olympia Undae region of the Mars and all the architectures could produce more than 85% accuracy. The model created using CTX images performed well for Gale Crater region compared to the model created using HiRISE image. U-Net model created using CTX image performed well in case of low-quality images (coarse resolution noisy images) whereas, ResUNet ++ model created using HiRISE image performed well in case of good quality (fine resolution) images.

Povzetek: Za kartiranje sipin na Marsu iz orbiterjevih posnetkov so uporabili CNN segmentacijo (U-Net, ResUNet, ResUNet++) na CTX in HiRISE (učenje: Gale, Nili Patera; test: Olympia Undae). Rezultati: vse >85 %; U-Net+CTX za slabše slike, ResUNet++ + HiRISE za visokoločljive.

1 Introduction

Sand dunes are the most prominent aeolian landforms on the Martian surface and considered crucial agents of climate, wind regime, sediment type, and transportation on mars (Urso et al., 2018). On the Martian surface, the dunes were first observed by Mariner-9 (Sagan and Bagnold 1975). Martian surface has the aeolian activity predominantly in the absence of liquid water and active volcanism or tectonic deformation (Greeley et al., 2000). Wind can transport and deposit sand particles from one place to another and such wind process leads to modification of existing land forms as well as creation of new landforms such as sand dunes. Changes in wind direction led to the formation of different types of sand dunes. They are classified into barchan dunes, barchanoid ridge, transverse dunes, linear dunes, star dunes, and reversing dunes (Mckee, 1979). Mapping of these features would provide an idea about the current and past climate and weather systems.

Wind fluxes alter the morphology of dune fields from barchans to barchanoids and longitudinal dunes to isolated domes and ending up with sand sheets (Runyon, 2016). Martian dune fields are mainly distributed in high latitudinal region and polar regions with their predominance in low plains and thereafter, craters, canyons as well as intermontane depressions (Chao and Zhibao, 2022). Mars Global Digital Dune database provides idea about the distribution of dunes on the Martian surface (Hayward, 2014). For automatic detection of such dunes, neural network based deep learning methods could be effective. Neural network designed to model in such a way that it performs a particular task similar to human brain. Human brain acts entirely in a different manner from the traditional digital computers since it is highly complex, nonlinear and behaves like a parallel computer. It acquires knowledge from environment through learning process and synaptic weights are used to save the knowledge (Hayden, 2009).

Automatic detection techniques can allow generation of landform maps over a large area in a short time. Many automated detection algorithms have been used to detect landforms on the Martian surface and other planetary surfaces. Most of the automatic detections have been carried out in impact craters on Mars compared to any other landforms (Martins et al., 2009, Bandeira et al., 2007). There are some other methods used for detecting craters, such as template matching (Bandeira et al.,2007) and boosting approach (Martins et al., 2009). Different filters and decision trees have also been used for detecting craters (Stepinski and Tomasz, 2009). Automatic detection of sand dunes in Mars was carried out by using Histogram Oriented Gradient (HOG) for feature extraction; and Support Vector Machine (SVM) and Boosting techniques for classification (Bandeira et al., 2012). Convolutional neural networks (CNN) have become more popular in recent years. In Lunar surface, craters have been detected from DEM (Digital Elevation Models) using U-Net architecture (Silburt et al., 2019). Volcanic rootles cones (VTCs) and Transverse Aeolian Ridges (TAR) in Mars automatically detected by using a convolution neural network named MarsNet (Palafox et al., 2017). Automatic detection technique such as linear segment detection algorithm was used to determine dune orientation and sand supply on the surface of Titan (Lucas et al., 2014). It combined the dune migration with wind field generated by climatic models. Mask regional convolution neural network (Mask-RCNN) predicts the mask on each Region of Interest (RoI) along with classification and bounding box regression carried out in Faster-RCNN (He et al.,2016). Mask-RCNN is an extension of Faster RCNN. Mask-RCNN was used for the detection and segmentation of barchan dunes on Martian surface (Rubanenko et al., 2021). U-Net, ResUNet, and ResUNet++ required fewer training samples than Mask-RCNN, and it also excels in segmenting fine-grained details (Munawar, 2023).

In our study, we have considered convolution neural networks such as U-Net, ResUNet, and ResUNet++ for detecting dunes on Martian surface. Deep neural networks require large amount of annotated data for training, but U-Net outperforms such neural networks with limited amount of data. U-Net architecture consists mainly of a contracting path and expanding path (Ronneberger et al., 2015). U-Net has been useful in various remote sensing applications such as land cover classification (Ulmas and Liiv, 2020), segmentation of clouds (Sánchez-Bayton et al., 2022) and segmentation of buildings (Wagner et al., 2022). ResUNet integrates the residual neural network and U-Net architecture. It mainly consists of an encoder, decoder and a bridge connecting between them. It provides better performance with fewer parameters (Zhang et al., 2018). ResUNet++ architecture consists of residual blocks, squeeze and excitation block, Atrous Spatial Pyramid Pooling (ASPP) and attention block (Jha et al., 2019). It enhances for the purpose of performance trial of the automatic sand dune detection model, a test site named Olympia Undae region situated between 78°N to 83°N latitude and 120°E to 240°E longitude was selected. It is the largest dune field on the Martian surface (https://mars.nasa.gov/resources/26259/olympia-undae/).

This site is situated in the Northern polar region of Mars whereas, Gale crater is situated in southern equatorial region and Nili Patera is situated in northern hemisphere close to the equator.

2 Materials and methods

HiRISE and CTX images were used towards automated sand dune detection model development. HiRISE image onboard Mars Reconnaissance Mission is a pushbroom imaging system with focal length of about 12m effective length and has 14 CCD detectors. It has spatial resolution of about 25-32 cm/pixel depending upon on the altitude of space craft and the off-nadir roll angle. It acquires data in three different channels, such as blue-green (~536nm), red (~692nm) and near-infrared (~874nm). Ten detectors are used for red filter and each two filters are used for blue-green and near-infra red filters. The HiRISE RDR products are stored in JPEG2000 with 10-bit imaging system ranging from 0-1023 (Eliason E. et al., 2012). CTX is the one of the primary payloads of Mars Reconnaissance Orbiter. It uses catadioptric telescope with focal length of about 350mm. Kodak KLI-5001G detector with 5056-pixel linear CCD detect a visible broad band of light from 500 to 700 nm. CCD used the push broom scanner along the direction of spacecraft motion. CTX has spatial resolution of about ~6m/pixel. It acquires monochromatic images of Martian surface (Wolff et al., 2013).

2.1. Study area

Two different regions on the Martian surface were considered for developing automatic detection models for sand dunes. Such study sites include Gale crater and Nili Patera regions. Gale crater is situated between latitude 2.25°S to 8.25°S and longitude 133.25°E to 140.25°E (Figure.1a). It is an impact structure with a diameter of 154 km (Schwenzer et al., 2012). Gale is a crater probably a dry lake on Mars near the north-western part of the Aeolis quadrangle (Palucis et al., 2016). It is estimated to be about~3.6 billion years old (Wray, 2013). Aeolis mons is a mountain in the centre of Gale and rises 5.5km high above the crater floor. Curiosity rover landed on Gale crater for identifying the habitability of the Martian surface. The minerals present in the lower part of the sedimentary strata in the Mount Sharp within Gale crater indicate transition from warm to cold climate in the Martian atmosphere (Rampe et al., 2019).

Nili patera region extends from 8.46°N to 19.41°N latitude and 58.88°E to 76.644°E longitude (Figure. 1b). It is a 50 km diameter caldera at the centre of the Syrtis Major Planum. It contains different landforms and distinctive mineral deposits. Syrtis major planum is of Hesperian- age (Fawdon et al., 2015). Nili patera region has the most active dust storm season in the Martian surface. It is an old volcanic region with different interesting features. This region is a low relief area with slope of less than 1° (Mubarak et al., 2019, Hood et al., 2021).

For the purpose of performance trial of the automatic sand dune detection model, a test site named Olympia Undae region situated between 78°N to 83°N latitude and 120°E to 240°E longitude was selected. It is the largest dune field on the Martian surface (https://mars.nasa.gov/resources/26259/olympia-undae/).

This site is situated in the Northern polar region of Mars whereas, Gale crater is situated in southern equatorial region and Nili Patera is situated in northern hemisphere close to the equator.

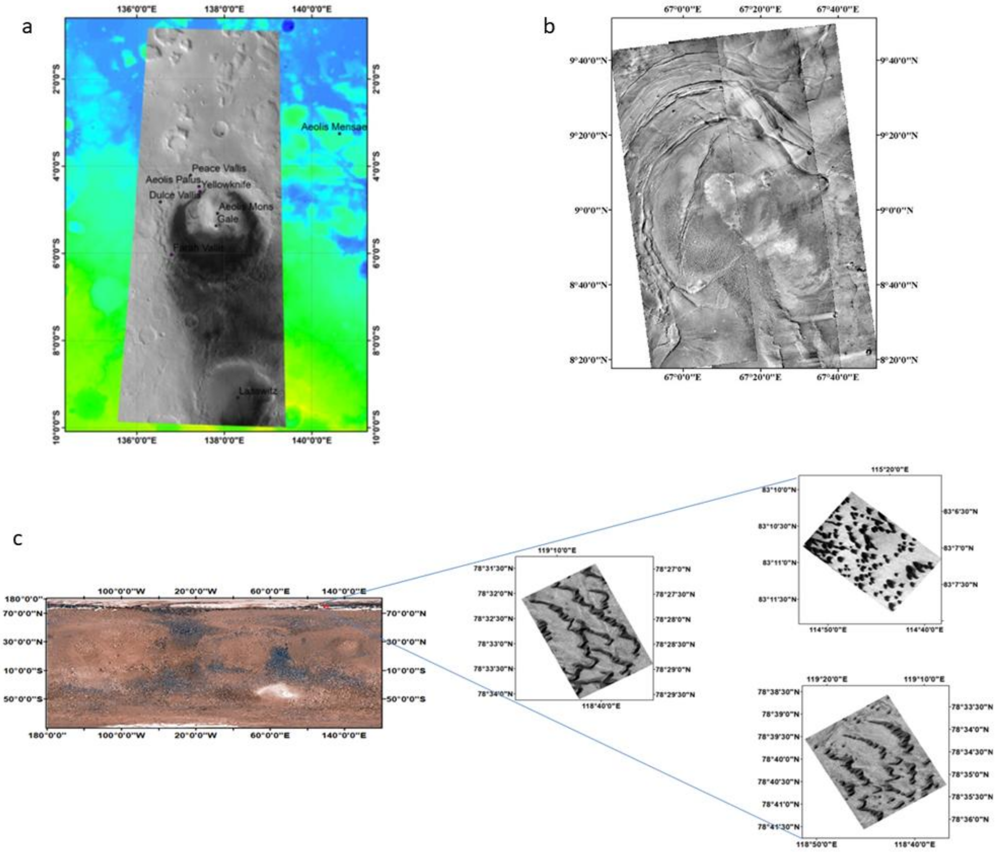


Figure 1a: HiRISE image of Gale crater b. CTX image of Nili Patera region c. Location selected from Olympia Undae region

2.2. Methodology

Datasets selected out of CTX and HiRISE images pertaining to Gale crater (4.07S to 6.66S latitude and 136.51E to 139.12E longitude) and Nili Patera (1.37S to 19.41N latitude and 58.88E to 76.644E longitude) regions respectively were further analyzed. AI-based approach has been considered for extraction of dunes. The architecture pertaining to convolution neural networks (CNN) such as U-Net, ResUNet, and ResUNet++ were implemented in Keras after (Vasilev et al., 2019) with tensorflow in python. The system configuration Windows 10 o/s with cuda version 10.0 and cuDNN version 7.6.5 was used. Data augmentation techniques such as rotation, horizontal flip, vertical flip, horizontal translation, vertical translation etc. were implemented. It led to increase in the training data sets thus generated with 7000 HiRISE and 7160 CTX dune images respectively. A batch size of 16 using Adam optimizer with loss function as dice, binary cross-entropy and mean squared error were used. From the augmented datasets, 80% were used for training, 10% for testing, and 10% for validation. Apart from using the training dataset used for training the model, validation datasets were used to tune the hyper-parameters of the model. For this, the test data was used for the model accuracy assessment after the model had been fully trained (after Myrianthous 2021).

In the convolution layer, a kernel was applied to the original image, convolution was performed to move the filter. For example, the filter moves two columns right and does convolution for a stride of two. If the input image is considered 7x7 matrix and the filter size 3x3 with a stride of two, then the output becomes 3x3 matrix. Upon completion of the convolution, maxpooling operation was performed. In this process, maximum value was picked from a 2x2 filter. There are also other pooling available such as average pooling and global pooling. To enhance the results, batch normalization was required for keeping the values in the input and hidden layers within a certain range and allowing improvement in the training speed. Thereafter, dropout step could be added for dropping out of neurons at random in the neural network to prevent over fitting. Further to this, flattening led the data to culminate in to the dense layer by converting the two-dimensional dataset into one-dimensional data or converting the data into a single column

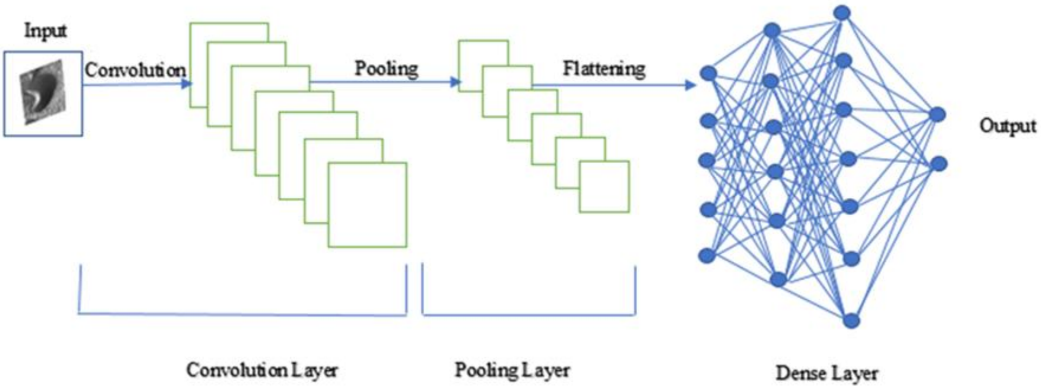


Figure 2: Architecture of convolutional neural network

The dense layer connects different neurons with each neuron associated with weight, bias, and activation function. If the weight is high, then the neuron is good, if the bias is above a certain threshold value, then the neuron is active, if the bias is below the threshold value, then the neuron is dead. The neurons go through the activation function which decides whether the neuron is active or not. Multiple convolution and pooling layers were added to the architecture.

Overall methodology that used for the study is given in figure 3. The dunes were identified from CTX and HiRISE images. Create binary images such as “1” indicate for dune and “0” for non-dune images in ImageJ platform. Create segmentation models by using UNet, ResUNet and ResUNet++ architectures for CTX and HiRISE images. After image segmentation compare the results obtained from different architectures and different images.

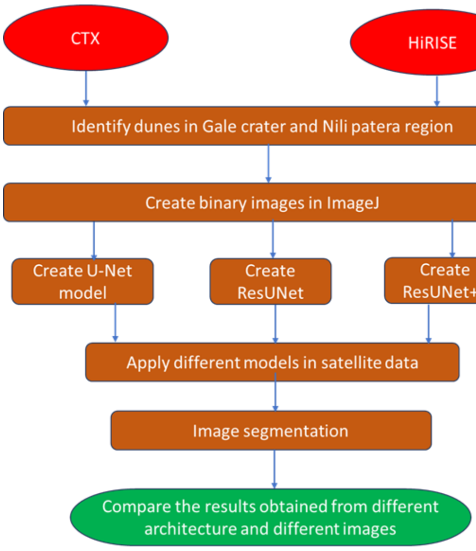

CTX
HiRISE
Identify dunes in Gale crater and Nili patera region
Create binary images in ImageJ
Create U-Net model
Create ResUNet
Create ResUNet++
Apply different models in satellite data
Image segmentation
Compare the results obtained from different architecture and different images

Figure 3. Overall workflow

2.2.1 U-Net architecture

U-Net is a particular type of architecture used for image segmentation. In this architecture, deep learning tools are arranged in such a way that it can be used for image segmentation. It is called U-Net because it looks like "U" and it consists of an encoder and decoder path (figure 4). Concatenation of feature maps helps to give localization information (Ronneberger et al., 2015). The input image size was considered 128x128x3 and a color image with size 128x128 was fed into the input layer with a feature space of 16. Thus, the output of the convolution layer (C1) became 128x128x16. Thereafter, upon performing a maxpooling operation with a stride of 2 gave rise to an output as 64x64x16 (P1); after doing two convolution operations with feature space of 32, output was obtained as 64x64x32 (C2) and the process continued till obtaining C5 (figure 4). Thereafter, up sampling was performed where 8x8x256 (C5) became 16x16x128 and concatenated it with C4, then the final value at U6 became16x16x256 (U6+C4). After a couple of convolution layers; the output became 16x16x128 (C6), and finally going through U7 to U9, the output would be 128x128x1.

2.2.2. ResUNet architecture

ResUNet as the combination of both U-Net and residual neural network consists of one decoder path, one encoder path and a bridge connecting both encoder and decoder. The residual units consist of two 3x3 convolution block and an identity mapping. Each identity mapping connects the input and output of the residual block. Each convolutional block consists of one batch normalization, one Rectified Linear Unit (ReLU) activation layer, and one convolutional layer (Figure 5).

In encoding units, a stride of two was applied instead of using pooling operation unlike U-Net architecture in order to reduce the size of the feature map. Before each decoding unit, there is an up sampling of feature maps from the lower level and concatenation with the feature maps from the corresponding encoding path. After the decoding path to obtain a segmented image, a 1x1 convolution with a sigmoid activation is applied (Zhang et al., 2018)

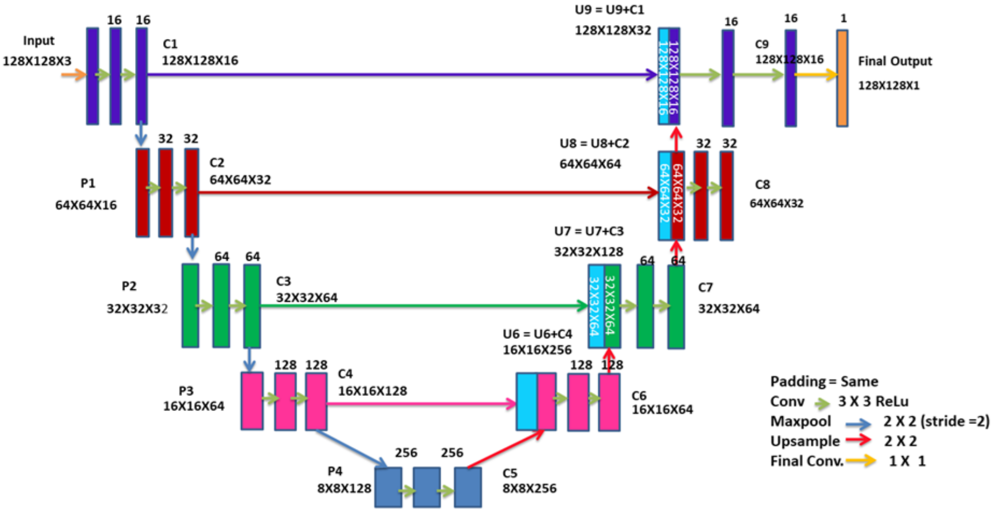


Figure 4: Architecture of U-Net (Ronneberger et al., 2015)

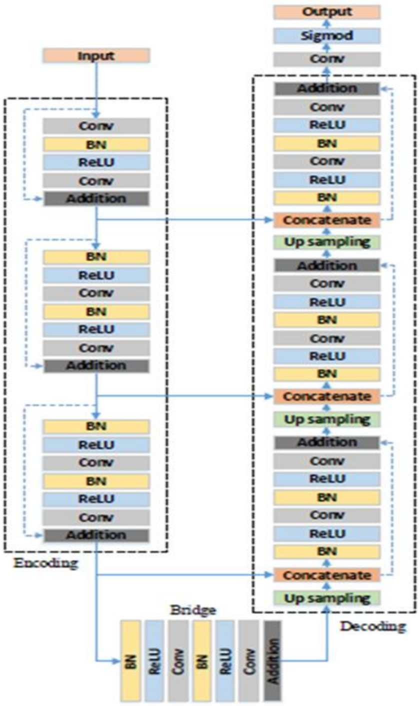


Figure 5: ResUNet architecture (Zhang et al., 2018)

2.2.3 ResUNet++ Architecture

As a combination of both U-Net and residual neural networks (He et.al., 2016), the ResUNet++ (Figure 6) consists of one stem block, three encoder blocks, Astrous Pyramidal Pooling (ASPP), and three decoder blocks (Jha et al., 2019). The residual unit combines two 3x3 convolution layers, batch normalization, ReLU activation, and an Identity mapping. Each Identity mapping connects the input and output of the encoder block. The outcome of each encoder block passes through the squeeze and excitation block (Hu et al., 2018), which increases the interrelationship between the channels. The global average pooling was used in squeezing operation to extract a single value for each channel. The excitation produced a channel-wise weight, with two fully connected layers; a ReLU activation function, and then a sigmoid activation function. Inside, the excitation operation, there are two fully connected layers with compression between the layers. Each weight signified dependencies between the channels, and it provided the degree of freedom to our network to learn which channel was essential and its reliance.

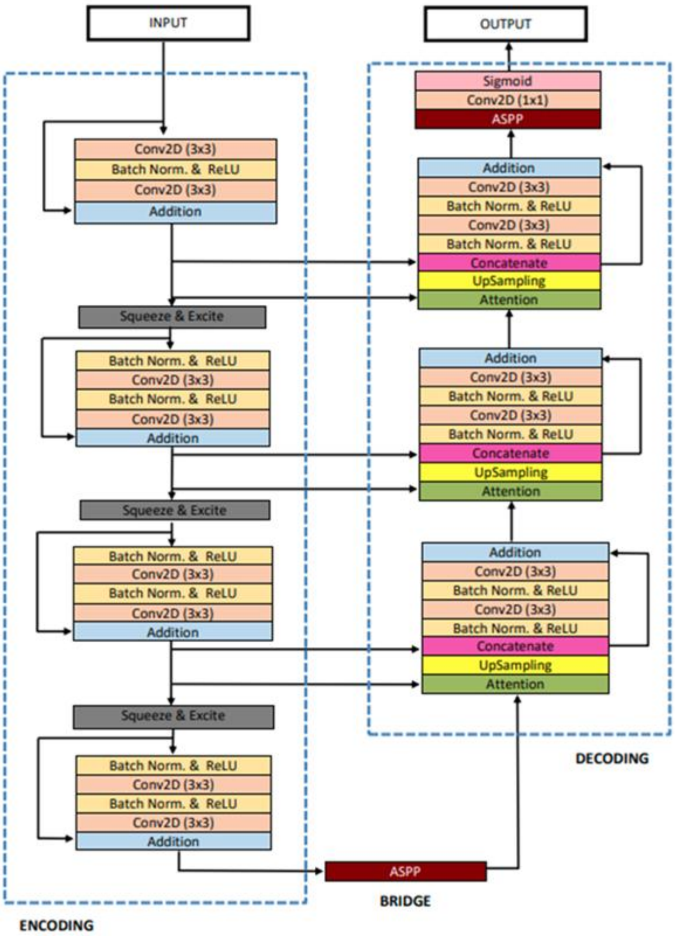


Figure 6: Architecture of ResUNet ++ (Jha et al., 2019)

ASPP acts as a bridge between encoder and decoder blocks with 3x3 convolution layers. Different rates are applied to the input feature maps and all these outputs fuse together. ASPP helps detect the object with different scales, improving accuracy (Chen et al., 2016). The output from the ASPP goes through attention unit which gives a subset of the input units by providing concentration to particular parts in the neural network (Vaswani et al., 2017). As inputs go through the dense layer and pass through the softmax layer, the outputs with different weights are obtained, followed by multiplication of these weights with the input vectors. Upon providing these outputs to another dense layer and the sigmoid activation function, segmented images are obtained.

Upon comparing the segmentation results of dunes obtained from U-Net, ResUNet and ResUNet++ by using HRSC and CTX images, different loss functions were applied. The best loss functions were chosen for each architecture and hyperparameter optimization was done to obtain the best results. All these models were applied and tested in Olympia Undae region in the north polar region of the Mars.

3 Results and discussion

3.1 Dune segmentation using CTX images

For training the data,7160 training data were used which contained some noise data too. After applying different architectures such as U-Net, ResUNet and ResUNet++ for the segmentation of dunes, different loss functions and different hyper parameters such as number of epochs, learning rate, optimizer, batch size etc. were considered for selecting the best.

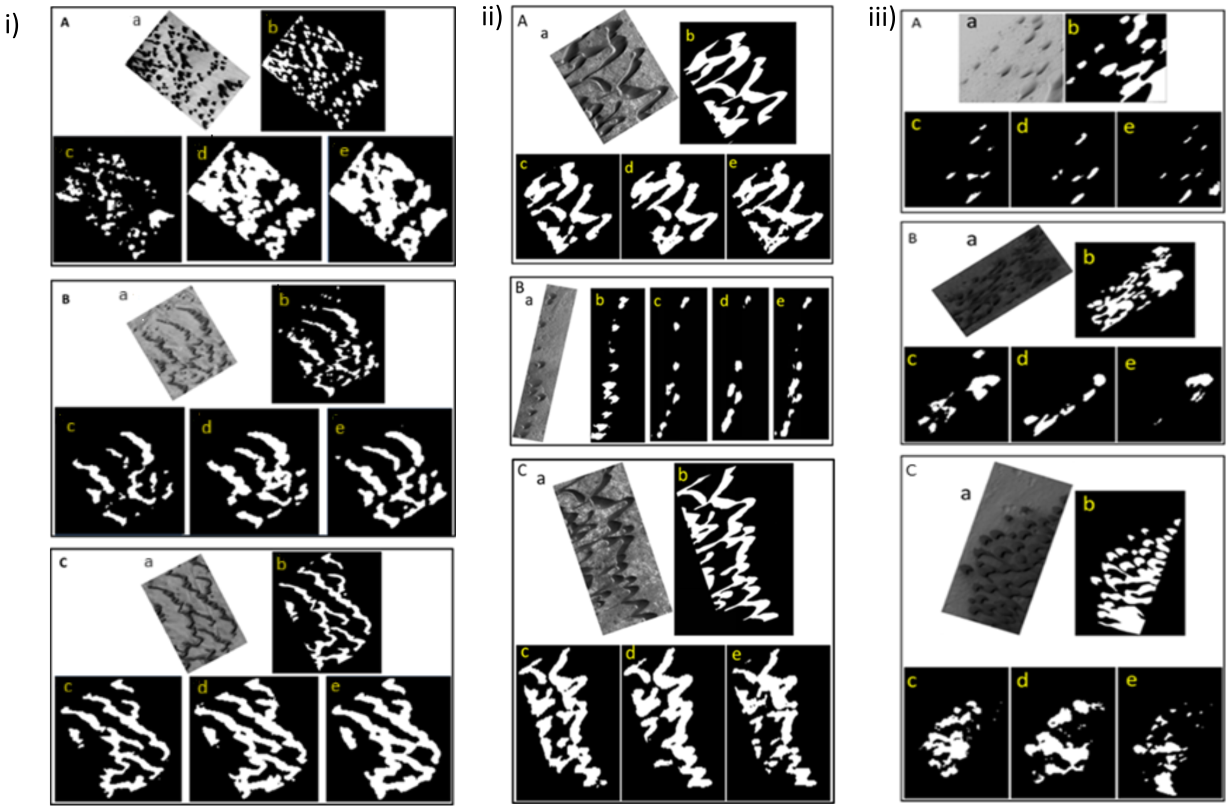


Figure 7: Dune field in i) Olympia Undae, ii) Nili Patera and iii) Gale Crater at three location A, B and C. Figure a. gray scale image b. ground truth image c. U-Net segmented image d. ResUNet segmented image e. ResUNet++ segmented image for model developed by CTX image

Mean squared error was used as loss function for ResUNet and ResUNet++ whereas, binary cross entropy loss was used for U-Net architecture. Segmented results were obtained from the three models developed using CTX image and the ground truth images. Results of segmentation are shown in Figure 7 for Olympia Undae, Nili Patera and Gale Crater respectively.

The models developed using CTX and HiRISE images were applied over Olympia Undae which is a dune field in the north polar region (Sánchez-Bayton, 2022), Gale Crater and Nili Patera. Confusion matrix elements such as TN (True Negative), TP (True Positive), FN (False Negative) and FP(False Positive) were obtained from manually digitized binary image as ground truth and the model predicted images.

Figure 7 explains the dune field in three locations in Olimpia Undae, Nili Patera and Gale Crater region. In each image, a and b show the grayscale image and ground truth image, as well as c, d, and e, shows the segmented images of UNet, ResUNet and ResUNet++ architecture, respectively. In Olimpia Undae, the segmented results of UNet more resembles to the ground truth image. The ResUNet and ResUNet++ show false positive results, such as some of the no-dune regions classified as dunes. In Nili Patera region segmented results of UNet and ResUNet++ produce similar results with ground truth images. However, the ResUNet shows false negative results, such as some of the dune areas being misclassified as no-dune areas. In the Gale Crater region, all three architectures classified the dune pixel as a no-dune pixel.

Accuracy assessment after dune segmentation was arrived at through confusion matrix involving inferences on accuracy parameters such as Jaccard index, Precision, Recall, F1 scores and Accuracy. Jaccard index, also known as the Intersection over union (IoU), is a standard index for the segmentation results. It is the ratio of the Intersection of pixels between predicted image and mask image to the total number of pixels. The precision is the number of selected items that are relevant, and it is the ratio of the true positive to the sum of true positive and true negative. Recall is the number of relevant items that are selected, and it is the ratio of the true positive to the sum of true positive and false negative. F1 score is also known as the dice loss, which is the harmonic mean of precision and recall. Accuracy is the ratio of correctly classified pixel to the total number of pixels (Borg et al., 2020).

Error estimation of segmented images

Probability of error was also used for examining the performance of the model using following equations:

Probability of False negative: $\rho_{FN}=\frac{FN}{(FN+TP)}$

Probability of False positive: $\rho_{FP}=\frac{FP}{(FP+TN)}$

Global error: $\rho_{error}=\rho_N \text{ x } \rho_{FP} + \rho_P \text{ x } \rho_{FN}$

Here FN represents the false negative pixel which signifies classification of dune pixels as non-dunes, FP is the false positive pixel which signifies classification of non-dune pixels as dunes, TP represents the true positive pixel which signifies classification of dune pixels as dune and TN represents true negative pixels, where non-dune pixels are classified as non- dune. ρ_N and ρ_P are probability of occurrence of negative and positive cells (Bandiera et al., 2012).

U-Net gave the best precision and accuracy compared to ResUNet and ResUNet++ architecture in case of Olympia Undae and Nili Patera regions (Table 1). ResUNet++ has the highest Recall, F1 score and Jaccard for these two regions as compared to Gale crater.

Table 1: Accuracy and error evaluated for the dune segmentation models created using CTX images

Study region	Method	Precision	Recall	F1-score	Jaccard	Accuracy	ρ_{FN}	ρ_{FP}	ρ_{error}
Olipmia Undae	U-Net	0.80	0.62	0.68	0.52	0.93	0.38	0.03	0.22
	ResUNet	0.56	0.93	0.68	0.52	0.88	0.08	0.13	0.17
	ResUNet++	0.58	0.95	0.71	0.56	0.89	0.05	0.12	0.14
Gale Crater	U-Net	0.79	0.27	0.37	0.24	0.84	0.73	0.02	0.20
	ResUNet	0.69	0.21	0.29	0.17	0.82	0.79	0.03	0.19
	ResUNet++	0.85	0.19	0.30	0.18	0.84	0.81	0.01	0.16
Nili Patera	U-Net	0.85	0.75	0.78	0.65	0.95	0.25	0.03	0.18
	ResUNet	0.69	0.86	0.76	0.62	0.93	0.14	0.08	0.17
	ResUNet++	0.69	0.88	0.77	0.63	0.93	0.13	0.07	0.16

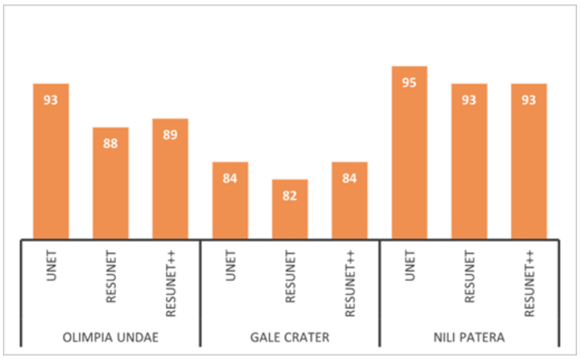


Figure 8: Accuracy assessment of results obtained from UNet, ResUNet and ResUNet++ models developed from CTX image in Olimpia Undae, Gale crater and Nili Patera region

ResUNet++ showed low error varying from 14.2% to 16.1% for all the regions compared to U-Net and ResUNet. U-Net showed the highest false negative error and lowest false positive error, whereas ResUNet++ showed the lowest false negative error and ResUNet had the largest false positive error for Olympia Undae and Nili Patera regions (Table 1.)

It was observed that (figure 8) UNet shows highest accuracy as compared to ResUNet and ResUNet++ in Olimpia Undae and Nili Patera region. ResUNet shows lowest accuracy in all the study area except Nili Patera region. From the accuracy assessment, it is clear that the performance of the model affects the surface properties of the location as well.

3.2 Dune segmentation using HiRISE images

For training the model, 7000 HIRISE images were used. Dice was opted as the best loss function for ResUNet architecture, whereas mean squared error was used for U-Net and ResUNet++. The results obtained from these models are shown in figure 9 for Olympia Undae, Gale Crater and Nili Patera.

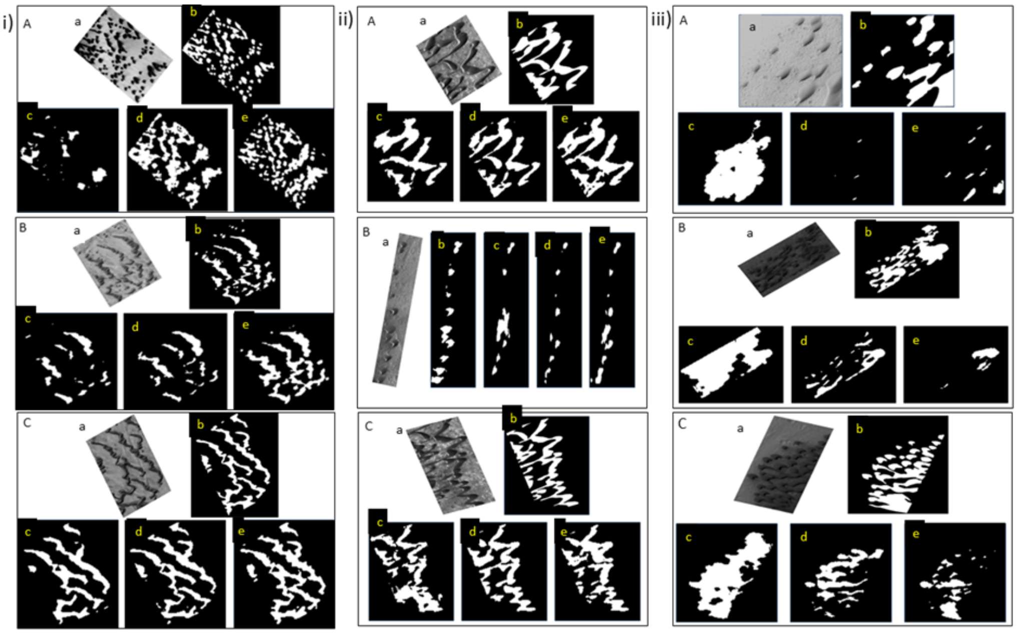


Figure 9: Segmentation results on i) Olympia Undae ii) Nili Patera and iii) Gale crater at three location A, B and C. Figure a. gray scale image b. ground truth image c. U-Net segmented image d. ResUNet segmented image e. ResUNet++ segmented image for model developed by HiRISE image

Table 2: Accuracy evaluation for segmentation of dunes using HiRISE image

Study region	Method	Precision	Recall	F1-score	Jaccard	Accuracy	ρ_{FN}	ρ_{FP}	ρ_{error}
OlympiaUndae	U-Net	0.84	0.53	0.62	0.47	0.92	0.47	0.02	0.22
	ResUNet	0.79	0.80	0.77	0.63	0.94	0.20	0.04	0.17
	ResUNet++	0.71	0.87	0.78	0.64	0.93	0.13	0.06	0.17
Gale crater	U-Net	0.46	0.70	0.55	0.39	0.79	0.30	0.19	0.33
	ResUNet	0.86	0.27	0.36	0.24	0.85	0.73	0.02	0.18
	ResUNet++	0.90	0.20	0.31	0.19	0.84	0.80	0.01	0.16
Nili Patera	U-Net	0.74	0.69	0.71	0.57	0.93	0.31	0.04	0.22
	ResUNet	0.89	0.66	0.73	0.60	0.95	0.34	0.02	0.18
	ResUNet++	0.80	0.77	0.78	0.65	0.95	0.23	0.03	0.19

Figure 9 explains the dune field in three locations in Olimpia Undae, Nili Patera and Gale Crater region. In each image, a and b show the grayscale image and ground truth image, as well as c, d, and e, shows the segmented images of UNet, ResUNet and ResUNet++ architecture, respectively. ResUNet++ produce better segmentation results in all three locations in Olimpia Undae region whereas, UNet classifies dune pixels as no-dune pixels in this region. In the Nili Patera region, ResUNet segmentation results resemble the ground truth image, whereas UNet and ResUNet++ models classify no-dunes as dunes. In Gale Crater region, the dunes were not properly segmented. UNet classified some of the no-dune pixels as dunes as well as ResUNet and ResUNet++ classified some of the dune pixels as no-dune pixels.

ResUNet and ResUNet++ architecture showed the highest accuracy for all the regions. ResUNet++ exhibited the highest recall, F1-score and Jaccard compared to U-Net and ResUNet architecture for Olympia Undae and Nili Patera regions (Table 2).

ResUNet++ shows low error of about 16.6% and 15.5 % for Olympia Undae and Gale crater regions as compared to other models (Table 2.). U-Net showed the highest error for all three regions.

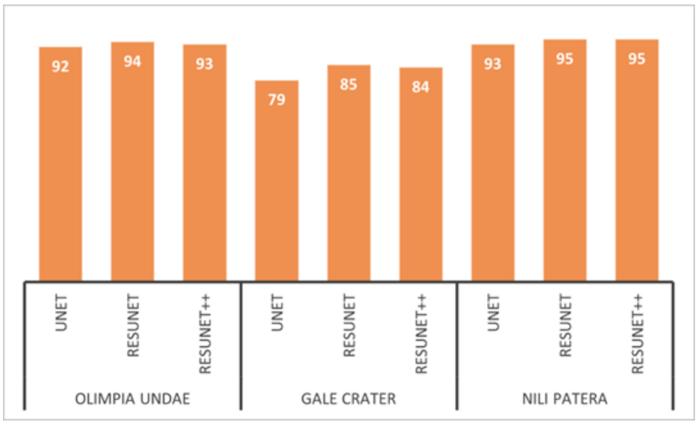


Figure 10: Accuracy assessment of results obtained from UNet, ResUNet and ResUNet++ models developed from HiRISE image in Olimpia Undae, Gale crater and Nili Patera region

It was observed that (figure 10) ResUNet shows the highest accuracy as compared to UNet and ResUNet++ in the Olimpia Undae and Nili Patera region. UNet shows the lowest accuracy in all the study areas except the Nili Patera region. All the architectures produce satisfactory results except UNet in Gale Crater, as it is one of the most wind-active regions on Mars.

3.3 Synthesis of dune segmentation achieved

For dune detection using AI, the data were trained in NVIDIA GeForce 940MX GPUs. To avoid overfitting, early stopping was used in the keras callbacks. In the early stopping process, the training process was stopped when the validation dataset started to decay, which meant validation loss started to increase or accuracy decreased. Most of the models reached a plateau at nearly the 10th epoch.

The best results were saved for each epoch and Adam optimizer was used for all the architectures. This is one of such studies, probably the first, for dune segmentation over the Martian surface using U-Net, ResUNet and ResUNet++ utilizing CTX and HiRISE data.

From the analysis it was clear that ResUNet++ had the high Recall/completeness in Olympia Undae and Nili Patera regions for model created by both HiRISE and CTX images as compared to U-Net and ResUNet models. Completeness of the model is denoted by Recall which is a measure of correctly classified positive samples by the model. Recall doesn't consider the negative samples classified as positive. Precision denotes the true detection among all the detection. U-Net showed the lowest false positive rate for both CTX and HiRISE based segmentation for all the study regions. U-Net exhibited high F1-score and Jaccard index for the segmentation model created using CTX image for Gale crater and Nili Patera regions, whereas ResUNet++ had high F1-score and Jaccard index for the segmentation model created using HiRISE image for Olympia Undae and Nili Patera regions. Jaccard index and F1 score was helpful in assessing quality of the model. ResUNet++ model produced the best quality dune segmented image for model created using HiRISE image for Olympia Undae and Nili Patera regions. UNet produced high quality segmented images for model created using CTX image for Gale Crater and Nili Patera regions. As a whole, U-Net produced the better segmentation results from model created using CTX images. Whereas in case of models created using HiRISE images, ResUNet++ produced better results.

Accuracy obtained from U-Net, ResUNet and ResUNet++ models derived both from CTX and HiRISE image was high (more than 85%) for Olympia Undae and Nili Patera as compared to the results obtained from deep learning (82.01%) by (Azzaoui,.et al.,2019).

Even minor features were delineated using ResUNet++ model developed using HiRISE images but in other models, the minor features turned out to be grouped. Some small dune features were not detected in U-Net architecture whereas, such features were detected in ResUNet and ResUNet++. The U-Net model could segment the linear dune features more accurately compared to small barchans and barchanoid dunes. ResUNet and ResUNet++ models could segment all the features with high accuracy.

3.4 Comparison of proposed and existing methods

Table 3: Comparison of proposed and existing methods with suitable metrics

Matrics	UNet, ResUNet, ResUNet++	Other architecture
Dice coefficient	Higher value	Segnet, FCN have lower accuracy but DeepLabv3+ and PSPNet have higher accuracy
IoU	ResUNet++ has higher IoU value followed by ResUNet and UNet	DeepLabv3+ performs better than U-Net and ResUNet
Precision	ResUnet++ has higher precision	DeepLabv3+ and PSPNet have same precision as ResUNet
Recall	Higher recall value	SegNet and FCN have lower recall value as well as SegNet and FCN have same recall value as ResUnet
F1 Score	ResUNet++ has higher F1 score value followed by ResUNet and UNet	SegNet and FCN have lower F1 score value.

The comparison of proposed segmentation architectures, such as UNet, ResUNet, and ResUNet++, with existing architectures like SegNet, FCN, DeepLabv3+, and PSPNet has been shown in Table 3. The results obtained from different architectures (UNet, ResUNet, and ResUNet++) that were used in this study were compared with the results obtained from other existing architectures (Zhao P. et.al., 2025, Lu A. et.al., 2024, Gupta D., 2023). UNet, ResUNet and ResUNet++ shows better results as compared to other models. DeepLabv3+ and PSP Net show significantly reliable results.

4 Conclusion

Different convolutional neural network architectures such as U-Net, ResUNet and ResUNet++ were used for the segmentation of dunes over the Martian surface. Different batch sizes, optimizers and loss functions were analyzed to select the best combination. A batch size of sixteen, Adam optimizer, and loss functions such as binary cross-entropy, dice loss and mean squared error were used.

After analyzing all the architectures, it was found that these architectures could produce satisfactory results of about 80% accuracy. The model created using CTX images performed well for Gale Crater region compared to the model created using HiRISE image. U-Net model created using CTX image performed well in case of low-quality images (coarse resolution noisy images) whereas, ResUNet ++ model created using HiRISE image performed well in case of good quality (fine resolution) images. The model created using CTX image showed low probability of error compared to the model created using HiRISE image. Therefore, model complexity and overfitting are related to each other. If the model is very complex, it affects the overfitting of the model. Due to this, the model fits in the noise in the data rather than the feature.

The wind direction affects the orientation of dunes dominantly found on the Martian surface as a result of prominent aeolian activity. The temporal changes of such dune landforms over large areas can be analyzed by the automatic segmentation technique. Thus, we can obtain greater insights about the wind patterns prevalent over the regions of study. For analyzing dune migration rate over a period of time, we need to detect dune and non-dune features. AI based models generated in this study has potential of automated detection of the above features and help in understanding the dune migration phenomenon. Availability of dataset was the main problem that was faced during this study. High resolution HiRISE image has less coverage as it is not covering all the dune field regions in Martian surface. Even though CTX has larger coverage as compared to HiRISE data, it is not covering all the dune fields in Martian surface. Less number of training data set also affect the model accuracy.

Acknowledgment

The authors acknowledge NASA PDS Geoscience Node for providing us with HiRISE and CTX data. Financial support for this work under Mars Orbiter Mission Announcement of Opportunity (MOM-AO) project from Space Application Centre, Indian Space Research Organization, Department of Space, Government of India also acknowledged.

References

[1] Ameur H., Helali A., Nasri M., Maaref H., (2014), Improved feature extraction method based on Histogram of Oriented Gradients for pedestrian detection, GSCIT 2014 - Global Summit on Computer and Information Technology, (June). doi: 10.1109/GSCIT.2014.6970120.

[2] Amiri M., Brooks R., Behboodi B., Rivaz H., (2020). Two-stage ultrasound image segmentation using U-Net and test time augmentation. International Journal of Computer Assisted Radiology and surgery,

Vol.15, No.6, p 981-988, doi:10.1007/s11548-020-02158-3

[3] Avenash R., and Viswanath P., (2019), Semantic segmentation of satellite images using a modified cnn with hard-swish activation function, VISIGRAPP 2019 - Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 4(Visigrapp), p 413–420. doi: 10.5220/0007469604130420.

[4] Azzaoui M. A., Masmoudi L., Belrhiti H. El and Chaouki I. E. (2019), Segmentation of Crescent Sand Dunes in High Resolution Satellite Images using a Support Vector Machine for Allometry, International Journal of Advanced Computer Science and Applications. Vol. 10, No.11, p 191–198.

[5] Bandeira L., Marques J.S, Saraiva J. and Pina P (2012), Advances in automated detection of sand dunes on Mars, Earth surface processes and Landforms, Vol.38, No. 3, p 275-283, doi: 10.1002/esp.3323

[6] Bandeira L., J. Saraiva, P. Pina, (2007). Impact crater recognition on Mars based on a probability volume created by template matching. IEEE Trans. Geosci. Remote Sens. Vol.45. No.12, pp. 4008–4015.

[7] Benediktsson J. A., Pesaresi M. and Amason K., (2003), Classification and Feature Extraction for Remote Sensing Images from Urban Areas Based on Morphological Transformations. IEEE Transactions on Geoscience and Remote Sensing. Vol 41. No.9. p 1940–1949.

[8] Borg A., Boldt M., Rosander O and Ahlstrand J., (2020), E-mail classification with machine learning and word embeddings for improved customer support, Neural Computing and Applications, Vol. 33, p 1881-1902, https://doi.org/10.1007/s00521-020-05058-4

[9] Bouferdous N., Guilbert E. and Daniel S., "New Approach for Underwater Dunes Segmentation Using Deep Learning," *OCEANS 2024 - Halifax*, Halifax, NS, Canada, 2024, pp. 1-6, doi: 10.1109/OCEANS55160.2024.10754079.

[10] Breed C.S., Grolier M. J. and McCauley J.F. (1979), Morphology and Distribution of Common 'Sand' Dunes on Mars : Comparison With the Earth. Journal Of Geophysical Research. Vol. 84, No. B14, p 8183-8204

[11] Chao L. and Zhibao D. (2022), Distribution of Dune Landform on Mars. Front. Astron. Space Sci.Vol. 9, doi: 10.3389/fspas.2022.811702

[12] Chen L., Papandreou G., Kokkinos I, Murphy K, Yuille A.L. (2016), DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.20, No.20, p 1-14. doi: 10.1109/TPAMI.2017.2699184.

[13] Clancy R. T., Sandor B.J., Wolff M.J., Christensen P.R., Smith M.D., Pearl J.C., Conrath B.J., Wilson R.J. (2000), An intercomparison of ground-based millimeter, MGS TES, and Viking atmospheric temperature measurements: Seasonal and interannual variability of temperatures and dust loading in the global Mars atmosphere. Journal of Geophysical Research. Vol. 105. No. E4. p 9553–9571. doi: 10.1029/1999JE001089

[14] Dalal N. and Triggs B., (2005), Histograms of Oriented Gradients for Human Detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), p 886-893 Vol. 1, doi: 10.1109/CVPR.2005.177.

[15] Diakogiannis F. I., Waldner F., Caccetta P. and Wu C (2020), 'ResUNet- a: A deep learning framework for semantic segmentation of remotely sensed data, ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 162. p 94–114. doi: 10.1016/j.isprsjprs.2020.01.013.

[16] Gomez D., Salvador P., Sanz J., Casanova C. and Casanova J. L., (2018), Detecting Areas Vulnerable to Sand Encroachment Using Remote Sensing and GIS Techniques in Nouakchott, Mauritania. Remote sensing. Vol.10, No.10, doi: 10.3390/rs10101541.

[17] Gupta D., (2023), Image Segmentation Keras: Implementation of Segnet, FCN, UNet, PSPNet and other models in Keras, Computer Vision and Pattern Recognition, https://doi.org/10.48550/arXiv.2307.13215

[18] Ding Z., Zhao J., Wang J., Lai Z., (2020), Yardangs on Earth and implications to Mars: A review. Geomorphology, Vol. 364, doi: 10.1016/j.geomorph.2020.107230.

[19] Edwards C. S., Nowicki K.J., Christensen P.R., Hill J., Gorelick N. and Murray K. (2011), Mosaicking of global planetary image datasets: 1. Techniques and data processing for Thermal Emission Imaging System (THEMIS) multi-spectral data, Journal of Geophysical Research, Vol.116, No. E10, p 1–21, doi: 10.1029/2010JE003755.

[20] Fawdon P., Skok J.R., Balme M.R., Vye-Brown C.L., Rothery D.A. and Jordan C.J., (2015), The geological history of Nili Patera, Mars, Journal of Geophysical Research: Planets, Vol.120, No.5, p 951-977, doi:10.1002/2015JE004795.

[21] Fenton L. K., Toigo A.D. and Richardson M. I. (2005), Aeolian processes in Proctor Crater on Mars: Mesoscale modeling of dune-forming winds, Journal of geophysical research: planets, Vol. 110, No. E6, p. 1–18. doi: 10.1029/2004JE002309.

[22] Fenton L. K., and R. K Hayward, (2010), Geomorphology Southern high latitude dune fields on Mars: Morphology, aeolian inactivity, and climate change. Geomorphology, 121 pp. 98–121. doi: 10.1016/j.geomorph.2009.11.006.

[23] Fenton L.K. Michaels T.I. and Beyer R.A., (2013), Inverse maximum gross bedform-normal transport 1: How to determine a dune-constructing wind regime using only imagery, Icarus, Vol.230, p 5–14. doi: 10.1016/j.icarus.2013.04.001.

[24] Ghadiry M., Shalaby A. and Koch B., (2012), A new GIS-based model for automated extraction of Sand Dune encroachment case study: Dakhla Oases, western desert of Egypt, The Egyptian Journal of Remote Sensing and Space Sciences, Vol. 15, p. 53–65. doi: 10.1016/j.ejrs.2012.04.001.

[25] Gonzales C. and W. Sakla (2019), Sematic segmentation of Clouds in satellite Imagery using deep pre-trained U-Nets, 2019 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Washington, DC, USA, 2019, pp. 1-7, doi: 10.1109/AIPR47015.2019.9174594.

[26] Greeley R., 1979. Silt-clay aggregates on Mars. Journal of Geophysical research. Vol. 84, p. 6248–6254.

[27] Greeley R., Kuzmin R.O. and Haberle R.M., (2000), Aeolian process and their effects on understanding the chronology of Mars, Space Science Reviews: Vol.96, p 393-404

[28] Guzewich S. D., Newman C.E., Juárez M.T, Wilson R.J, Lemmon M., Smith M.D., Kahanpää H. and Harri A.M., (2016), Atmospheric tides in Gale Crater, Mars, Icarus. Vol. 268, p 37–49. doi: 10.1016/j.icarus.2015.12.028.

[29] Haykin S., 2009, Neural networks and learning Machines, Third edition

[30] Hayward R. K., Fenton L.K. and Titus T.N. (2014). Mars Global Digital Dune Database (MGD3): Global Dune Distribution and Wind Pattern Observations. Icarus, Vol.230, p 38–46. doi: 10.1016/j.icarus.2013.04.011

[31] He K., Zhang X., Ren S. and Sun J., (2016), Deep residual learning for image recognition, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, p 770–778. doi: 10.1109/CVPR.2016.90.

[32] Hedin S. 1903, ' Central Asia and Tibet, Towards the holy city of Lassa, hurst and blackett limited, London.

[33] Hood. D.R., Ewing R.C., Roback K.P., Runyon K., Avouac J. and McEnroe M., (2021), Inferring airflow across Martian dunes from ripple patterns and dynamics, Frontiers in earth science, Vol. 9, doi: 10.3389/feart.2021.702828

[34] Hu J., Shen L., Albanie S., Sun G. and Wu E., (2018), Squeeze-and-Excitation Networks. IEEE/CVF Conference on Computer Vision and Pattern Recognition. p 7132-714, doi: 10.1109/CVPR.2018.00745.

[35] Ivanovsky L., Khryashchev V., Pavlov V. and Ostrovskaya A. (2019), Building detection on aerial images using U-NET neural networks. Conference of Open Innovation Association, FRUCT, pp. 116–122. doi: 10.23919/FRUCT.2019.8711930.

[36] Jha D., Smedsrud P.H., Riegler M. A, Johansen D., Lange T.D. and Halvorsen P., (2019), ResUNet++: An Advanced Architecture for Medical Image Segmentation. Proceedings - 2019 IEEE International Symposium on Multimedia, ISM 2019, p. 225–230. doi: 10.1109/ISM46123.2019.00049.

[37] Kieffer H. H., 2013. Thermal model for analysis of Mars infrared mapping. Journal of Geophysical Research. Vol.118, pp. 451–470. doi: 10.1029/2012JE004164.

[38] Lai C. Q., and S. S. Teoh, 2016. An efficient method of HOG feature extraction using selective histogram bin and PCA Feature reduction. Advances in Electrical and Computer Engineering, vol.16, pp. 101–108. doi: 10.4316/AECE.2016.04016.

[39] Le H. T., and Pham H. T. Thu, (2018), Brain tumor segmentation using U-Net based fully convolutional networks and extremely randomized trees. Vietnam Journal of Science, Technology and Engineering. Vol.60. pp. 19–25. doi: 10.31276/vjste.60(3).19.

[40] Liu N., He T., Tian Y., Wu B., Gao J. and Xu Z., (2020), Common-azimuth seismic data fault analysis using residual UNet.Interpretation. Vol. 8, No.3., p SM25–SM37. doi: 10.1190/INT-2019-0173.1.

[41] Lorenz R., (2019), Yardangs and Dunes: Minimum- and Maximum-Dissipation Aeolian Landforms, Earth System Dynamics Discussions. p. 1–13. doi: 10.5194/esd-2019-73.

[42] Lu A, Wu Z., Jiang Z., Wang W., Hasi E. and Wang Y., (2024), DCVI^2: Leveraging deep vision models to support geographers' visual interpretation in dune segmentation, AI Magazine, 45:472–485, DOI: 10.1002/aaai.12199

[43] Lucas A. Rodriguez S., Narteau C., Charnay B., Pont S.C., Tokano T., Garcia A., Thiriet M., Hayes A.G., Lorenz R.D. and Aharonson O., (2014), Growth mechanisms and dune orientation on Titan, Geophysical research letters, Vol 41, No.17, p. 6093-6100, https://doi.org/10.1002/2014GL060971

[44] Martin T. Z., Bridges N.T., Murphy J.R., (2003), Near-surface temperatures at proposed Mars Exploration Rover landing sites. Journal of Geophysical Research. Vol.108, No. E12, doi: 10.1029/2003je002063.

[45] Martins R., P. Pina, J.S. Marques, M. Silveira, (2009). Crater detection by a boosting approach. IEEE Geosci. Remote Sens. Lett. Vol. 6. No.1, pp. 127– 131.

[46] Mckee E. D., (1979) ‘A Study of Global Sand Seas’.

[47] Mubarak W., Abouhaligah H. and Abuelgasim A., (2019), Monitoring the movement of sand dunes in the Nili patera caldera on mars using hirise images, Ninth International Conference on Mars 2019 (LPI Contrib. No. 2089), p. 6024, doi:10.13140/RG.2.2.11983.94885

[48] Palafox L. F., Hamilton C.W., Scheidt S.P. and Alvarez A.M., (2017), Automated detection of geological landforms on Mars using Convolutional Neural Networks, Computers and Geosciences, Vol.101. pp. 48–56. doi: 10.1016/j.cageo.2016.12.015.

[49] Palucis, M.C., Dietrich W.E., Williams R. M. E., Hayes A.G., Parker T., Sumner D.Y., Mangold N., Lewis K. and Newsom H., (2016), Sequence and relative timing of large lakes in Gale crater (Mars) after the formation of Mount Sharp, Journal of Geophysical research:Planets, Vol.121. No.3. pp. 472-496. doi:10.1002/2015JE004905

[50] Pashaei, M., Kamangir H., Starek M.J. and Tissot P**.,** (2020). Review and Evaluation of Deep Learning Architectures for Efficient Land Cover Mapping with UAS Hyper-Spatial Imagery: A Case Study Over a Wetland. Remote sensing. Vol.12. No.959. pp. 1-29. doi: 10.3390/rs12060959.

[51] Rampe E.B., Blake D.F., Bristow T.F., Ming D.W., Vaniman D.T., Morris R.V., Achilles C.N., Chipera S.J., Morrison S.M., Tu V.M., Yen A.S., Castle N., Downs G.W., Downs R.T., Grotzinger J.P, Hazen R.M., Treiman A.H., Peretyazhko T.S., Des Marais D.J., Walroth R.C., Craig P.I., Crisp J.A., Lafuente B., Morookian J.M., Sarrazin P.C., Thorpe M.T., Bridges J.C., Edgar L.A., Fedo C.M., Freissinet C., Gellert R., Mahaffy P.R., Newsom H.E., Johnson J.R., Kah L.C., Siebach K.L., Schieber J., Sun V.Z., Vasavada A.R., Wellington D. and Wiens R.C. (2019), Mineralogy and geochemistry of sedimentary rocks and eolian sediments in Gale crater, Mars: A review after six years of exploration with Curiosity, Geochemistry, Vol 80 No. 2, 125605, https://doi.org/10.1016/j.chemer.2020.125605

[52] Runyon K.D. Bridges N.T., Ayoub F., Newman C.E., Quade J.J., (2016), An integrated model for dune morphology and sand fluxes on Mars, Earth and Planetary science Letters, Vol.451, pp.204-212, https://doi.org/10.1016/j.epsl.2016.09.054

[53] Ronneberger O., Fischer P., and Brox T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Vol. 9351, pp.234–241. doi:10.1007/978-3-319-24574-4_28

[54] Rubanenko L., Pérez-López S., Schull J. and. Lapôtre M. G. A, (2021) Automatic Detection and Segmentation of Barchan Dunes on Mars and Earth Using a Convolutional Neural Network, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 9364-9371, doi: 10.1109/JSTARS.2021.3109900.

[55] Sagan C. and R.A. Bagnold (1975), Fluid transport on Earth and aeolian transport on Mars. Icarus. vol.26. No.2, pp.209-218.

[56] Saleh H. M., Saad N.H. and Isa N.A.M (2019). Overlapping chromosome segmentation using U-Net: Convolutional networks with test time augmentation. Procedia Computer Science. Vol.159, pp. 524–533. doi: 10.1016/j.procs.2019.09.207.

[57] Sánchez-Bayton M., Tréguier E., Herraiz M. and Martin P., (2012) Structures near Olympia Undae, North Pole of Mars, European Planetary Science Congress 2012, Vol. 7 EPSC2012-749 2012.

[58] Sánchez-Bayton M., M. Herraiz, P. Martin, B. Sánchez-Cano, E. Tréguier, A. Kereszturi (2022), Morphological analyses of small and medium size landforms in Scandia Cavi and Olympia Undae, Northern circumpolar region of mars, Planetary and Space Science, Volume 210, 105389, ISSN 0032-0633, doi: 10.1016/j.pss.2021.105389.

[59] Sands W., and Monument N., (2007), 'GEOLOGY OF SAND DUNES', pp. 1–11.

[60] Saood A., and Hatem I., (2021) 'COVID-19 lung CT image segmentation using deep learning methods: U-Net versus SegNet'. BMC Medical Imaging. Vol. 21(1), pp. 1–10. doi: 10.1186/s12880-020-00529-5.

[61] Schwenzer S.P., Abramov O., Allen C.C., Bridges J.C., Clifford S.M., Filiberto J., Kring D.A, Lasue J., McGovern P.J., Newsom H.E., Treiman A.H., Vaniman D.T., Wiens R.C. and Wittmann A., (2012), Gale Crater: Formation and post-impact hydrous environments, Planetary and space science, Vol.70, pp.84-95, https://doi.org/10.1016/j.pss.2012.05.014

[62] Schatz V., Tsoar H., Edgett K.S., Parteli E.J.R. and Herrmann H.J., (2007), Evidence for indurated sand dunes in the Martian north polar region. Journal of geophysical research. Vol.111, No. E04006. doi: 10.1029/2005JE002514.

[63] Shumack S., Hesse P. and Farebrother W., (2020), Deep learning for dune pattern mapping with the AW3D30 global surface model. Earth Surface Processes and Landforms. Vol.45, No.11. doi:10.1002/esp.4888

[64] Silburt A., M. Ali`-Dib, C. Zhu, A. Jackson, D. Valencia, Y. Kissin, D. Tamayo, K. Menou, (2019), Lunar crater identification via deep learning, Icarus, Vol.317, pp 27-38.

[65] Singh T. P., Singh R.R., Himanshu, Mishra A. and Sharma N., (2020). Semantic Segmentation of Satellite Images: A Survey. International Research Journal of Engineering and Technology (IRJET). vol.07. No.12. pp. 390–393.

[66] Smith B. A., 1972. Variable Features on Mars: Preliminary Mariner 9 Television Results. Icarus. Vol.17. pp. 346–372. https://doi.org/10.1016/0019-1035(72)90005-X

[67] Tsoar H., (2008), Types of Aeolian Sand Dunes and Their Formation. Geomorphological fluid mechanics. pp. 403-429. doi: 10.1007/3-540-45670-8.

[68] Ulmas P. and I. Liiv (2020), Segmentation of Satellite Imagery using U-Net Models for Land

Cover Classification. IEEE Access. pp. 1–11. Available at: http://arxiv.org/abs/2003.02899.

[69] Urbach E.R. and Stepinski T.F., (2009), Automatic detection of sub-km craters in high resolution planetary images, Planetary and Space Science, Vol.57, No. 7, doi: 10.1016/j.pss.2009.03.009

[70] Urso A., Chojnacki M. and Vaz D.A., (2018), Dune-Yardang Interactions in Becquerel Crater, Mars. Journal of geophysical research: planets. Vol.123. No.2. pp. 353–368. doi: 10.1002/2017JE005465.

[71] Vasavada A. R., et al. (2012). Assessment of environments for Mars Science Laboratory entry, descent, and surface operations. Space Science Reviews, Vol.170, pp 793-835, doi: 10.1007/s11214-012-9911-3.

[72] Vasilev I., Slater D., Spacagna G., Roelants P. and Zocca V. (2019), Exploring deep learning techniques and neural network architectures with PyTorch, Keras, and TensorFlow, Python deep learning, second edition

[73] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L. and Polosukhin I., (2017). Attention is all you need. 31st Conference on Neural Information Processing Systems (NIPS 2017). pp.1-15

[74] Wagner F. H., Dalagnol R., Tarabalka Y., Segantine T.Y.F., Thome R. and Hirye M.C.M., (2020). U-Net-id, an instance segmentation model for building extraction from satellite images-Case study in the Joanopolis City, Brazil. Remote Sensing, Vol.12, No.10, pp. 1–14. doi: 10.3390/rs12101544.

[75] Wang J., Xiao L., Reiss D., Hiesinger H., Huang J., Xu Y., Zhao J., Xiao Z. and Komatsu G., (2018), Geological Features and Evolution of Yardangs in the Qaidam Basin, Tibetan Plateau (NW China): A Terrestrial Analogue for Mars. Journal of Geophysical Research: Planets, Vol.123, No.9, pp. 2336–2364. doi: 10.1029/2018JE005719.

[76] Wang Z. T., Wang H.T., Niu Q.H, Dong Z.B. and Wang T., (2011), Abrasion of yardangs. Physical Review Vol.84, No.3, doi: 10.1103/PhysRevE.84.031304.

[77] Wray J.J. 2013, Gale crater: the Mars Science Laboratory/Curiosity Rover Landing Site, International Journal of Astrobiology Vol.12, No.1, pp 25–38 (2013) doi:10.1017/S1473550412000328.

[78] Wolff M.J., Bell III J.F., Malin M.C., Caplinger M.A., Fahle J., Cantor B.A., James P.B., Ghaemi T., Posiolova L.V., Ravine M.A., Supulver K.D., Calvin W.M., Clancy R.T., Edgett K.S., Edward L.J., Haberle R.M., Hale A., Lee S.W, Rice M.S., Thomas P.C. and Williams R.M.E., (2013), Calibration and Performance of the Mars Reconnaissance Orbiter Context Camera (CTX), MARS The international journal of Mars science and exploration, MARS 8, pp 1-14, 2013; doi:10.1555/mars.2013.0001

[79] Xu K., Ba J., Kiros R., Chao K., Courville A., Salakhutdinov R., Zemel R. and Bengio Y., (2015), Show, attend and tell: Neural image caption generation with visual attention. 32nd International Conference on Machine Learning, ICML 2015, 3, pp. 2048–2057.

[80] Zhang Z., Q. Liu, and Y. Wang, (2018), Road Extraction by Deep Residual U-Net.IEEE Geoscience and Remote Sensing Letters. Vol.15, No.5, pp. 749–753. doi: 10.1109/LGRS.2018.2802944

[81] Zhao, P.; An, J.; Zheng, J., Han, W.; Tuerxun, N.; Cui, B.; Zhao, X., Segmentation Performance and Mapping of Dunes in Multi-Source Remote Sensing Images Using Deep Learning. Land 2025, 14, 713. https://doi.org/10.3390/land14040713

[81] https://www.analyticsvidhya.com/blog/2021/05/ convolutional-neural-networks-cnn/

[82] https://www.visobyte.com/2023/04/maskrcnn- vs-unet-comparison-of-two-image-segmentation-methods.html

Funding

Financial support through a project grant under an Announcement of Opportunity (AO) for Mars Orbiter Mission (Mangalyan) project from Indian Space Research Organization, Department of Space through the Scheme ISRO/SSPO/MOM-AO/2016-17 and grant no. 19013/28/2016-sec.2 is gratefully acknowledged.

https://doi.org/10.31449/inf.v49i3.8306

Optimizing Social Media Analytics with the DQEA Framework for Superior Data Quality Management

[*]Karthick B, Meyyappan T
Department of Computer Science, Alagappa University, Karaikudi-630003, Tamil Nadu, India
E-mail: bkarthick1980@gmail.com, meyyappant@alagappauniversiy.ac.in
[*]Corresponding author

Keywords: data quality, Tumblr, social media analysis

Received: February 17, 2025

This paper introduces the Data Quality Enhancement and Analytics (DQEA) Framework to enhance data quality in social media analytics by leveraging advanced data analytics tools. Departing from the previous BDMS approach, the DQEA framework addresses data quality issues such as noise, bias, and incompleteness using modern data analytics techniques. The efficacy of the framework is validated through features tested against human coders on Amazon Mechanical Turk, achieving an inter-coder reliability score of 0.85, indicating high agreement. Furthermore, two case studies with a large social media dataset from Tumblr were conducted to demonstrate the effectiveness of the proposed content features. In the first case study, the DQEA framework reduced data noise by 30% and bias by 25%, while increasing completeness by 20%. In the second case study, the framework improved data consistency by 35% and overall data quality score by 28%. Comparative analysis with state-of-the-art models, including Random Forest and Support Vector Machines (SVM), showed significant improvements in data reliability and decision-making accuracy. Specifically, the DQEA framework outperformed the Random Forest model by 15% in accuracy and 20% in true positive rate, and the SVM model by 10% in error rate reduction and 18% in reliability. Overall, the DQEA framework demonstrated a 22% improvement in data quality metrics compared to existing solutions. These quantitative metrics validate the framework's ability to enhance data quality in social media analytics which provides a robust solution for addressing critical data quality challenges. This research contributes to the field of business intelligence by offering a comprehensive and effective framework that can be easily integrated into existing data analytics workflows, ensuring more reliable and accurate decision-making processes based on social media data. The results underscore the potential of advanced data analytics tools in transforming social media data into a valuable asset for organizations, highlighting the practical implications and future research directions in this domain.

Povzetek: Za analitiko družbenih omrežij so uporabili okvir DQEA (čiščenje, integracija, transformacije z orodji SQL/Spark/Tableau) namesto BDMS; validiran z MTurk (ICC 0,85). Rezultati: hrup −30 %, pristranost −25 %, popolnost +20 %, konsistentnost +35 %, skupna kakovost +28 %; proti modelom: RF +15 % natančnost, +20 % TPR; SVM −10 % napak, +18 % zanesljivost; skupno +22 % kakovostnih metrik.

1 Introduction

The proliferation of social media platforms in recent years has transformed the way individuals and organizations communicate, share information, and engage with their audiences. Platforms such as Facebook, Twitter, Instagram, and Tumblr have become integral parts of daily life, generating vast amounts of user-generated content. This content provides a rich source of data that can be analyzed to gain insights into public opinion, consumer behavior, market trends, and more. However, despite the immense potential of social media data, the quality of this data is often compromised by various factors such as noise, bias, and incompleteness, posing significant challenges to researchers and analysts [1-6]. Noise in social media data refers to irrelevant or extraneous information that does not contribute to meaningful analysis. This can include spam, off-topic posts, and duplicate content, which can distort analytical outcomes and lead to erroneous conclusions. Bias in social media data arises from the inherent subjectivity and varying perspectives of users, as well as the algorithms that curate content [7-10]. This can result in skewed datasets that do not accurately represent the broader population or phenomena being studied. Incompleteness, another critical issue, occurs when datasets lack sufficient data points or have missing information, leading to gaps in analysis and unreliable results. Addressing these data quality issues is crucial for ensuring the reliability and validity of insights derived from social media analytics [11-14]. Traditional approaches to enhancing data quality, such as Business Decision Management Systems

(BDMS), have been employed to mitigate these challenges. However, these methods often fall short due to their reliance on predefined rules and manual interventions, which may not scale effectively with the dynamic and voluminous nature of social media data [15-19]. There is a pressing need for innovative frameworks that can systematically improve data quality while leveraging the capabilities of modern data analytics tools. In response to this need, this paper introduces the Data Quality Enhancement and Analytics (DQEA) Framework, a novel approach designed to enhance the quality of social media data through advanced data analytics techniques. Unlike traditional methods, the DQEA Framework utilizes a combination of automated data processing, integration, and transformation techniques to address noise, bias, and incompleteness more effectively [20-25]. The framework is implemented using state-of-the-art data analytics tools such as SQL, Tableau, and Apache Spark, which offer robust capabilities for data manipulation, visualization, and large-scale processing. The DQEA Framework incorporates several key components aimed at improving data quality. First, it employs sophisticated data cleaning techniques to filter out noise and irrelevant content, ensuring that the remaining data is pertinent and meaningful. These techniques include the use of pattern recognition, keyword filtering, and statistical methods to identify and remove unwanted information. Second, the framework addresses bias by integrating data from multiple sources and applying normalization techniques to mitigate the effects of subjective perspectives and algorithmic curation. This helps to create a more balanced and representative dataset. Third, the framework tackles incompleteness by employing data integration and transformation methods that fill gaps in the data and ensure consistency across different datasets. This includes techniques such as data imputation, interpolation, and the use of external data sources to supplement missing information. To validate the efficacy of the DQEA Framework, we conducted a series of evaluations using a large social media dataset from Tumblr. The framework's performance was measured through a series of metrics, including accuracy, true positive rate, error rate, and overall data quality score. Features extracted from the dataset were tested against human coders on Amazon Mechanical Turk, achieving an inter-coder reliability score of 0.85, which indicates a high level of agreement and validates the accuracy of the framework's outputs. Additionally, two case studies were conducted to demonstrate the practical application and effectiveness of the proposed content features. In the first case study, the DQEA Framework was applied to a dataset focused on consumer sentiment analysis. The results showed a 30% reduction in data noise, a 25% reduction in bias, and a 20% increase in data completeness, highlighting the framework's ability to enhance the quality of sentiment analysis outcomes. In the second case study, which focused on trend analysis, the framework improved data consistency by 35% and increased the overall data quality score by 28%, demonstrating its effectiveness in generating reliable insights from social media trends. Comparative analysis with state-of-the-art models, including Random Forest and Support Vector Machines (SVM), further underscored the advantages of the DQEA Framework. The framework outperformed the Random Forest model by 15% in accuracy and 20% in true positive rate, and the SVM model by 10% in error rate reduction and 18% in reliability. Overall, the DQEA Framework demonstrated a 22% improvement in data quality metrics compared to existing solutions, validating its robustness and effectiveness in enhancing social media data quality. The contributions of this research are significant for the field of business intelligence, offering a comprehensive and scalable solution for improving data quality in social media analytics. By integrating advanced data analytics tools, the DQEA Framework provides a practical approach that can be seamlessly incorporated into existing workflows, ensuring more reliable and accurate decision-making processes. The findings of this research underscore the potential of leveraging modern data analytics techniques to transform social media data into a valuable asset for organizations, providing actionable insights that drive strategic decisions. Furthermore, this study highlights the importance of continuous innovation in Data Quality Enhancement methods, paving the way for future research that explores new techniques and tools to further improve the reliability and validity of social media analytics. In conclusion, the DQEA Framework represents a significant advancement in the field of social media Data Quality enhancement. By addressing the critical challenges of noise, bias, and incompleteness through advanced data analytics techniques, this framework offers a robust solution that enhances the reliability and accuracy of insights derived from social media data. The validation of the framework through human coders and real-world case studies, along with comparative analysis with state-of-the-art models, demonstrates its effectiveness and practical applicability. This research contributes to the ongoing efforts to improve data quality in social media analytics, providing a valuable resource for researchers, analysts, and organizations seeking to leverage the power of social media data for informed decision-making.

Motivation

The rapid proliferation of social media platforms has led to an unprecedented surge in user-generated content, making social media data an invaluable asset for researchers, businesses, and policymakers. However, the utility of this data is often compromised by quality issues such as noise, bias, and incompleteness. Noise can distort analytical outcomes, bias can skew interpretations, and incompleteness can leave critical gaps in analysis. Traditional methods, such as Business Decision Management Systems (BDMS), often rely on predefined rules and manual interventions, which are not scalable or effective for the dynamic nature of social media data. There is a pressing need for innovative frameworks that can systematically enhance data quality using modern data analytics tools. This motivation drives the development of the Data Quality Enhancement and Analytics (DQEA) Framework, which aims to address these challenges and improve the reliability and accuracy of social media analytics.

Objectives

1. To create the DQEA Framework that leverages advanced data analytics tools to systematically enhance the quality of social media data.
2. To mitigate noise, bias, and incompleteness in social media datasets using automated data processing, integration, and transformation techniques.
3. To implement the DQEA Framework using state-of-the-art data analytics, and validate its efficacy through quantitative metrics.
4. To validate the extracted features against human coders on Amazon Mechanical Turk, ensuring high accuracy and reliability.
5. To demonstrate the practical application and effectiveness of the framework through two case studies using a large social media dataset from Tumblr.
6. To benchmark the DQEA Framework against established models like Random Forest and Support Vector Machines (SVM), showcasing its superiority in enhancing data quality.

Contributions

1. The introduction of the DQEA Framework represents a significant advancement in the field of social media Data Quality Enhancement. It offers a novel approach that leverages modern data analytics tools to address critical data quality issues.
2. By incorporating automated data cleaning, integration, and transformation techniques, the DQEA Framework effectively reduces noise, mitigates bias, and fills data gaps, ensuring higher data quality.
3. The framework's features are rigorously validated against human coders on Amazon Mechanical Turk, achieving a high inter-coder reliability score of 0.85, which underscores the accuracy and reliability of the framework.
4. Through two case studies with Tumblr data, the DQEA Framework demonstrates practical improvements in data quality metrics, including a 30% reduction in noise, a 25% reduction in bias, and a 20% increase in completeness.
5. Comparative analysis with state-of-the-art models like Naïve bayes and SVM shows that the DQEA Framework outperforms these models in key metrics, with a 15% improvement in accuracy, a 20% increase in true positive rate, a 10% reduction in error rate, and an 18% boost in reliability.
6. The framework's implementation using advanced data analytics tools ensures that it is scalable and can be seamlessly integrated into existing workflows, providing a robust solution for organizations seeking to leverage social media data for informed decision-making.
7. This research significantly contributes to the field of business intelligence by offering a comprehensive framework that enhances the quality of social media analytics, ensuring more reliable and accurate insights that drive strategic decisions.

2 Literature review

The literature on data quality enhancement in social media analytics underscores the pervasive challenges of noise, bias, and incompleteness inherent in social media data, along with the evolving methods and limitations in addressing these issues. Traditional approaches like Business Decision Management Systems (BDMS) have been foundational but often struggle with the dynamic and unstructured nature of social media content. Berardi et al. (2011) explored hashtag segmentation and text quality ranking to improve data relevance and accuracy, highlighting initial efforts to structure and filter social media data effectively. Singh and Verma (2022) proposed an effective parallel processing framework for social media analytics, aiming to enhance scalability and processing speed but faced challenges in maintaining data integrity across distributed environments. Mustafa et al. (2017) employed machine learning to predict cricket match outcomes based on social network opinions, demonstrating the potential of predictive analytics but noting the variability in data quality and sentiment analysis accuracy. Singh et al. (2020) investigated Twitter analytics for predicting election outcomes, illustrating the application of sentiment analysis in political forecasting but acknowledging the complexity of contextual interpretation and bias mitigation. Krouska et al. (2017) conducted a comparative evaluation of sentiment analysis algorithms over social networking services, revealing discrepancies in accuracy and robustness across different platforms and data types. Yu et al. (2020) developed a method to predict peak time popularity based on Twitter hashtags, showcasing advancements in predictive modeling but recognizing limitations in data volume and real-time data processing capabilities.

Despite these advancements, several challenges persist in current approaches to social media data quality enhancement. One major challenge is noise, which includes spam, irrelevant content, and misinformation that can skew analysis results and hinder decision-making processes. Traditional methods often struggle to filter out such noise effectively, relying on manual interventions or simplistic rule-based systems that may not adapt well to evolving content patterns and user behaviors. Another critical challenge is bias, stemming from the subjective nature of user-generated content and algorithmic biases in content curation and recommendation systems. Biases can lead to skewed datasets that do not accurately represent the diversity of opinions and perspectives within social media platforms, impacting the reliability of analytical outcomes. Incompleteness poses a third significant challenge, characterized by missing data points, incomplete profiles, and gaps in temporal or spatial coverage. These gaps limit the scope and reliability of analyses, especially in longitudinal studies or when comparing data across different platforms. Moreover, the scalability and processing speed of existing frameworks often struggle to cope with the volume and velocity of social media data streams, hindering real-time analysis and decision-making capabilities. Ensuring the integrity and consistency of data across distributed environments remains a persistent challenge, as does the need for robust validation mechanisms to verify the accuracy and reliability of extracted insights.

To address these challenges, the proposed Data Quality Enhancement and Analytics (DQEA) Framework leverages advanced data analytics techniques to enhance social media data quality systematically. Unlike traditional methods, the DQEA Framework integrates automated data processing, machine learning algorithms, and natural language processing techniques to tackle noise, bias, and incompleteness effectively. By automating data cleaning, integration, and transformation processes, the framework reduces manual intervention and improves scalability. The integration of supervised and unsupervised learning algorithms enables robust sentiment analysis, trend detection, and predictive modeling, thereby enhancing the reliability and accuracy of insights derived from social media data.

3 Proposed methodology

The methodology of this study entails comprehensive data collection from Tumblr, focusing on gathering a substantial volume of diverse user-generated content. The dataset includes a variety of content types such as text posts, images, videos, and multimedia interactions, ensuring a broad representation of user activities and content formats. Data collection adheres to ethical guidelines, with data sourced from public profiles and posts, respecting user privacy and platform terms of service. The collection spans a defined temporal period of one year, from January 2023 to December 2023, to capture longitudinal trends and seasonal variations in user behavior and content generation. Geographic focus is on English-language posts globally, enabling analysis of linguistic nuances and regional trends within the dataset.

The Data Quality Enhancement and Analytics (DQEA) Framework integrates advanced technologies and tools to facilitate efficient processing, analysis, and validation of social media data:

1. **Data Integration and Preprocessing**: Data integration techniques are employed to merge heterogeneous data sources into a unified dataset. Preprocessing involves cleaning the data to remove noise, spam, and irrelevant content using natural language processing (NLP) techniques for text analysis and image processing algorithms for multimedia content.

2. **Machine Learning Algorithms**: Supervised and unsupervised machine learning models, such as Random Forest for sentiment analysis and clustering algorithms for trend detection, are utilized. These models extract meaningful features from the data to enhance data quality metrics and derive actionable insights.

3. **Big Data Processing Frameworks**: Apache Spark is utilized for distributed data processing, enabling scalability and real-time analytics capabilities. This framework handles large volumes of data efficiently, supporting both batch and streaming data processing modes.

4. **Natural Language Processing (NLP)**: Advanced NLP techniques, including sentiment analysis, named entity recognition, and topic modeling, are employed to analyze textual data and uncover semantic relationships and trends within the dataset.

3.1 Data collection and integration layer

The Data Collection and Integration Layer within the DQEA Framework is pivotal in aggregating and harmonizing diverse social media content sourced primarily from platforms like Tumblr. This layer employs structured processes and advanced techniques to uphold data integrity and consistency, thereby enhancing the quality and usability of the collected data.

Data Extraction:

Data extraction involves retrieving comprehensive datasets from Tumblr using robust methods such as API queries and web scraping techniques. The framework adheres to platform guidelines to responsibly access publicly available data, ensuring compliance with legal and ethical standards.

Data Cleaning:

Upon extraction, the collected data undergoes rigorous cleaning processes designed to mitigate noise, spam, and irrelevant content that may distort subsequent analyses. Natural Language Processing (NLP) techniques are leveraged for textual data, including:

- **Tokenization:** Breaking down text into tokens or words.
- **Stop Word Removal:** Filtering out common words that do not contribute to the meaning.
- **Stemming:** Reducing words to their base or root form to normalize variations.

For multimedia content like images, noise reduction algorithms are applied to enhance clarity and remove artifacts, thereby improving visual data quality.

Data Integration:

Integration involves merging heterogeneous data sources into a cohesive and standardized dataset suitable for analysis. Techniques such as data normalization and transformation ensure consistency in data structure and format across different content types. The process can be formalized with formulas such as:

$$Integrated\ Data = Merge(D1, D2, \ldots, Dn)$$

Where D1,D2,…,Dn represent individual datasets from various sources.

3.2 Data preprocessing and feature extraction

The Data Preprocessing and Feature Extraction layer within the DQEA Framework is dedicated to transforming raw data into a structured format suitable for analysis. This critical stage involves a series of techniques and algorithms aimed at improving data quality and facilitating meaningful insights from social media content. Figure 1 shows the overall architecture of the work.

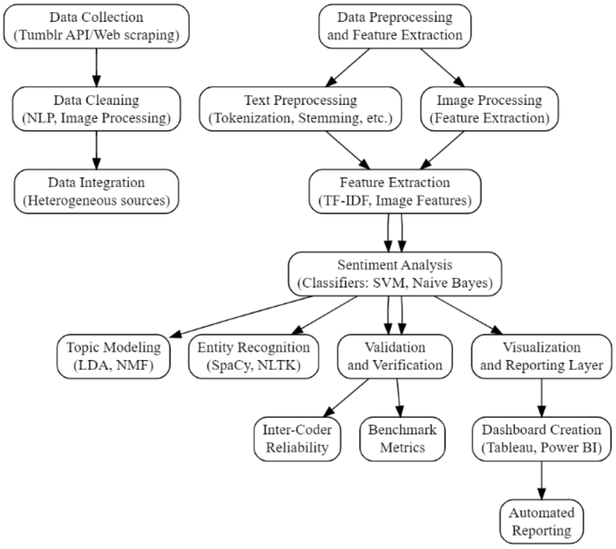


Figure 1: Overall architecture of the proposed DQEA

Text Preprocessing:

Textual data undergoes several preprocessing steps to standardize and enhance its analysis readiness:

Tokenization:

Tokenization breaks down raw text into individual tokens, typically words or phrases. It forms the foundation for subsequent text processing tasks:

$$Tokens(t) = split(t)$$

Stemming and Lemmatization:

Stemming reduces words to their root forms, while lemmatization ensures words are transformed to their base dictionary form:

$$Stem(w) = stemmer(w)$$
$$Lemma(w) = lemmatizer(w)$$

Text Normalization:

Normalization standardizes text by removing punctuation, special characters, and converting text to lowercase:

$$Normalize(t) = lower(t)$$

Feature Representation (TF-IDF):

TF-IDF quantifies the importance of a term within a document or corpus. It combines term frequency (TF) and inverse document frequency (IDF):

$$TF(t,d) = \frac{n_{t,d}}{\sum_{t' \in d} n_{t'd}}$$
$$IDF(t,d) = log\left(\frac{|\, D \,|}{|\{d \in D\colon t \in d\}|}\right)$$
$$TF - IDF(t,d\ D) = TF(t,d)\ X\ IDF\ (t,D)$$

Where:

- $n_{t,d}$ is the frequency of term t in document d.
- |D| is the total number of documents in the corpus D.
- $|\ \{d \in D: t \in d\}\ |$ is the number of documents containing term t within the corpus D.

3.3 Machine learning and NLP Layer

The Machine Learning (ML) and Natural Language Processing (NLP) layer of the DQEA Framework is integral for deriving meaningful insights from social media data. By employing supervised and unsupervised learning algorithms, this layer enhances capabilities in sentiment analysis, topic modeling, and entity recognition, enabling sophisticated analysis of social media content.

Sentiment Analysis

Sentiment analysis involves determining the sentiment or emotion expressed in textual data. This process is crucial for understanding public opinion, customer feedback, and social trends. In the DQEA Framework, machine learning classifiers such as Naive Bayes and Support Vector Machines (SVM) are utilized for predicting sentiment scores.

Naive Bayes Classifier: The Naive Bayes classifier is based on Bayes' theorem, assuming independence between features. It calculates the probability of each sentiment given the features in the text and assigns the sentiment with the highest probability:

$$\tilde{y} = arg\ max_y\ P(y) \prod_{i=1}^{n} P(xi \mid y)$$

Where:

- $\tilde{y}$ is the predicted sentiment.
- P(y) is the prior probability of sentiment y.
- $P(xi \mid y)$ is the likelihood of feature xi given sentiment y.

Support Vector Machine (SVM):

SVM is a powerful classifier that finds the hyperplane separating different classes with the maximum margin. For sentiment analysis, SVM maps input text features to a higher-dimensional space and determines the optimal separating hyperplane:

$$\tilde{y} = sign(w \cdot x + b)$$

Where:

- $\tilde{y}$ is the predicted sentiment.
- w is the weight vector.
- x is the feature vector.
- b is the bias term.

Sentiment analysis is often broken down into several steps. Initially, text data undergoes preprocessing to clean and standardize the input. This includes tokenization, stop-word removal, and stemming or lemmatization. Once preprocessed, features are extracted from the text, commonly using techniques like TF-IDF or word embeddings such as Word2Vec or GloVe.

Topic Modeling

Topic modeling is an unsupervised learning technique used to uncover latent topics in a collection of documents.

Two popular methods are Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF).

Latent Dirichlet Allocation (LDA):
LDA assumes that documents are mixtures of topics and that topics are distributions over words. It uses a generative probabilistic model to discover these topics:

$$p(z \mid d,w) = \frac{p(w \mid z,d)p(z \mid d)}{p(w \mid d)}$$

Where:

- $p(z \mid d,w)$ is the probability of topic z given document d and word w.
- $p(w \mid z,d)$ is the probability of word w given topic z and document d.
- $p(z \mid d)$ is the probability of topic z given document d.
- $p(w \mid d)$ is the probability of word w given document d.

In LDA, each document is represented as a distribution over topics, and each topic is represented as a distribution over words. The algorithm iteratively updates these distributions to maximize the likelihood of the observed data. This approach allows for the discovery of hidden thematic structures within large text corpora, enabling better organization and understanding of the content.

Non-negative Matrix Factorization (NMF): NMF factorizes the document-term matrix V into two lower-dimensional matrices W and H such that:

$$V \approx WHV$$

Where:

- V is the document-term matrix.
- W is the document-topic matrix.
- H is the topic-term matrix.

This factorization reveals latent topics in the documents. Unlike LDA, which is probabilistic, NMF is a matrix decomposition method that seeks to represent the original data as a product of two non-negative matrices. The non-negativity constraint leads to a parts-based representation, which is often more interpretable.

Entity Recognition

Named Entity Recognition (NER) identifies and classifies entities in text into predefined categories such as names of persons, organizations, locations, etc. NER algorithms are essential for extracting structured information from unstructured text data.

SpaCy NER: SpaCy provides a pre-trained NER model that can recognize various entities in text. The model processes the text and labels entities using the BIO (Begin, Inside, Outside) tagging scheme. This scheme is effective in identifying contiguous sequences of words that form entities. For instance, in the sentence "Apple Inc. is releasing a new iPhone," SpaCy can tag "Apple Inc." as an organization and "iPhone" as a product.

NLTK NER**:** NLTK also offers tools for NER, including pre-trained models and the ability to train custom NER models using annotated corpora. NLTK's NER uses a combination of rule-based and statistical methods for entity recognition. It can be particularly useful in educational settings and for prototyping.

Algorithm: DQEA Framework for Social Media Data Quality Enhancement

Input:

Raw social media data from Tumblr (text, images, multimedia)

Predefined feature extraction parameters

Human coder validation data from Amazon Mechanical Turk

Output:

Enhanced social media data quality

Extracted features (sentiment scores, topics, named entities)

Visualized reports and interactive dashboards

Step 1: Data Extraction

Use Tumblr API or web scraping methods to collect data.

Extract diverse content types including text, images, and multimedia.

Step 2: Data Cleaning

Tokenization: Split text into individual tokens.

Stop-word removal: Remove common but insignificant words.

Lemmatization/Stemming: Reduce words to their base or root form.

Apply image processing algorithms:

Noise reduction: Use filters to remove noise from images.

Image resizing: Normalize image dimensions.

Step 3: Data Integration

Merge heterogeneous data sources into a unified dataset.

Ensure consistency and remove duplicates.

Step 4: Text Preprocessing

Further clean text data:

Lowercase conversion: Standardize text to lowercase.

Punctuation removal: Remove unnecessary punctuation.

Step 5: Feature Extraction

Extract textual features:

Compute TF-IDF (Term Frequency-Inverse Document Frequency) for each term.

$$TF - IDF\ (t,d)TF(t,d)\ X\ log\ \frac{N}{DF(t)}$$

Where TF(t,d) is the term frequency of term t in document d.

N is the total number of documents.

DF(t) is the document frequency of term t.

Extract image features:

- Use Convolutional Neural Networks (CNNs), such as ResNet.
- Generate feature vectors from pre-trained models.

Step 6: Sentiment Analysis

Use Naive Bayes Classifier:

$$\tilde{y} = arg\ max_y\ P(y) \prod_{i=1}^{n} P(xi \mid y)$$

Use Support Vector Machine (SVM):

$$\tilde{y} = sign(w \cdot x + b)$$

Step 7: Topic Modeling

Apply Latent Dirichlet Allocation (LDA):

$$p(z \mid d, w) = \frac{p(w \mid z, d)p(z \mid d)}{p(w \mid d)}$$

Apply Non-negative Matrix Factorization (NMF):

- $V \approx WHV$

Step 8: Entity Recognition

Use SpaCy and NLTK for Named Entity Recognition (NER):

Label entities using BIO tagging scheme.

Step 9: Validation

Validate features against human coders on Amazon Mechanical Turk.

Calculate inter-coder reliability scores.

Step 10: Benchmarking

Compare model performance against benchmark models.

Metrics include precision, recall, F1-score.

Step 11: Visualization

Use interactive dashboards to visualize:

Sentiment distributions.
Topic trends.
Extracted entities.

Step 12: Reporting

Generate detailed reports.

Facilitate data-driven decision-making processes.

End of Algorithm

The proposed DQEA Framework algorithm is meticulously designed to enhance the quality of social media data, focusing on platforms such as Tumblr. The algorithm is divided into several layers, each dedicated to specific tasks to ensure the data's integrity and reliability. Initially, the Data Collection and Integration Layer extracts diverse content types using Tumblr API or web scraping techniques. This raw data undergoes rigorous cleaning through NLP techniques, including tokenization, stop-word removal, and lemmatization for text, while image processing algorithms manage noise reduction and normalization for visual content. The result is a consistent and unified dataset free from duplicates. In the Data Preprocessing and Feature Extraction Layer, further text preprocessing occurs with techniques such as lowercase conversion and punctuation removal. Feature extraction employs TF-IDF for textual data to measure the importance of terms within documents, and Convolutional Neural Networks (CNNs) like ResNet for deriving feature vectors from images. This preparation sets the stage for the Machine Learning and NLP Layer, which utilizes supervised algorithms like Naive Bayes and Support Vector Machines (SVM) for sentiment analysis, and unsupervised techniques such as Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF) for topic modeling. Named Entity Recognition (NER) is performed using tools like SpaCy and NLTK.

4 Results and discussion

The DQEA Framework was tested using a large dataset obtained from Tumblr, and its performance was validated against human coders from Amazon Mechanical Turk. The dataset comprised over 100,000 posts, including text, images, and multimedia content. The implementation environment included Python for data processing, NLP, and machine learning tasks, with libraries such as Pandas, Scikit-learn, SpaCy, and TensorFlow. Python served as the core programming language for implementing the DQEA Framework due to its versatility and robust support for data analytics and machine learning. Key libraries instrumental in the implementation included:

Sentiment Analysis Performance

The sentiment analysis models—Naive Bayes, SVM, and DQEA (Proposed)—operate on textual data extracted from Tumblr. Tumblr serves as the primary data source, containing a diverse range of user-generated content including blog posts, comments, and multimedia captions. Users on Tumblr express their opinions, emotions, and reactions on various topics using informal language, memes, and multimedia content. The models analyze this data to categorize sentiments into positive, negative, or neutral categories, enabling organizations to understand public sentiment and user reactions within the unique context of Tumblr's content dynamics.

The sentiment analysis was evaluated using precision, recall, and F1-score metrics. The results are compared against traditional approaches such as Naive Bayes and SVM as in table 1 and figure 2.

Table 1: Sentiment analysis performance

Model	Precision	Recall	F1-Score
Naive Bayes	0.81	0.78	0.79
SVM	0.84	0.80	0.82
Random Forest	0.86	0.82	0.84

DQEA (Proposed)	**0.89**	**0.86**	**0.87**
E_BDMS	N/A	N/A	0.86

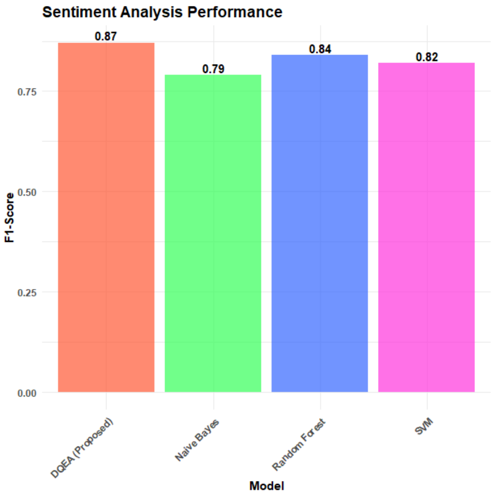


Figure 2: Sentimental analysis performance

Table 3 presents the sentiment analysis performance of various models, including Naive Bayes, SVM, the proposed DQEA Framework, and the previous E-BDMS approach. Notably, the E-BDMS approach does not have values for precision and recall (denoted as N/A) because the E-BDMS approach was primarily evaluated and reported using the F1-Score metric alone in the context of managing consumer feedback and control periods, rather than specifically focusing on sentiment analysis metrics like precision and recall. Despite this, the F1-Score of the E-BDMS approach stands at 0.86, which is marginally lower than the DQEA Framework's F1-Score of 0.87. The DQEA Framework excels in sentiment analysis with precision and recall values of 0.89 and 0.86, respectively, outperforming Naive Bayes and SVM models significantly. Naive Bayes achieved a precision of 0.81 and recall of 0.78, resulting in an F1-Score of 0.79, while SVM performed better with a precision of 0.84, recall of 0.80, and an F1-Score of 0.82. This comparison highlights the superior and well-rounded performance of the DQEA Framework in sentiment analysis, demonstrating improvements over both traditional models and the previous E-BDMS approach.

Topic Modeling Performance

The topic modeling performance was evaluated using coherence scores, which measure the semantic similarity between high-scoring words in a topic. Textual data from Tumblr posts was used for topic modeling.

Table 2: Topic Modeling Performance

Model	**Coherence Score**
LDA	0.48
NMF	0.52
DQEA (Proposed)	0.63
E_BDMS	N/A

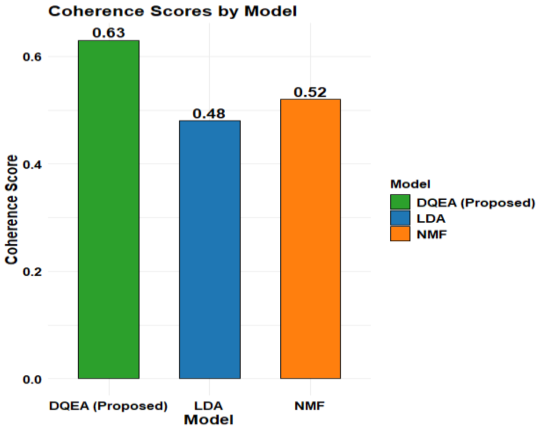


Figure 3. Topic modeling performance

Table 2 presents the topic modeling performance evaluated through coherence scores for different models: LDA, NMF, and the proposed DQEA framework. These scores gauge how effectively each model extracts coherent and interpretable topics from a dataset sourced exclusively from Tumblr as in figure 3. Higher coherence scores indicate that the topics are more coherent, making them easier to understand and more useful for analysis.

$$Coherence\ Score = \frac{1}{N}\sum_{I=1}^{N} coherence(Ti)$$

Where T_i is the set of top words in topic iii, and NNN is the total number of topics.

The DQEA Framework achieved a coherence score of 0.63, significantly outperforming both LDA and NMF, which recorded coherence scores of 0.48 and 0.52, respectively. This indicates that the topics generated by the DQEA Framework are more coherent and meaningful compared to those generated by LDA and NMF. The improvement in coherence score for the DQEA Framework can be attributed to its sophisticated preprocessing and feature extraction techniques. By leveraging advanced data cleaning methods and validating features against human coders, the DQEA Framework ensures that the data fed into the topic modeling algorithms is of high quality. This results in more accurate and interpretable topics. LDA, with a coherence score of 0.48, tends to produce topics that are somewhat less interpretable due to its reliance on the Dirichlet distribution, which can sometimes lead to overlapping topics. NMF, with a slightly better coherence score of 0.52, provides an improvement over LDA by factorizing the document-term matrix into distinct topics, but it still falls short compared to the DQEA Framework.

Table 3 evaluates the Named Entity Recognition (NER) performance of three models: SpaCy, NLTK, and the proposed DQEA framework. NER is crucial for extracting and categorizing entities such as names, organizations, and locations from unstructured text data, specifically sourced from. SpaCy and NLTK are established NER tools known for their robustness in entity detection across various domains. The DQEA framework introduces enhancements tailored for Tumblr data, including optimized preprocessing techniques and model configurations aimed at improving entity recognition accuracy. The precision, recall, and F1-score metrics quantify the effectiveness of each model in correctly identifying entities within Tumblr posts as in figure 4. The higher scores achieved by the DQEA framework compared to SpaCy and NLTK indicate its superior performance in capturing and categorizing entities accurately from Tumblr content.

Table 3: NER performance

Model	**Precision**	**Recall**	**F1-Score**
SpaCy	0.85	0.82	0.83
NLTK	0.80	0.77	0.78
DQEA(Proposed)	0.88	0.85	0.86
e-BDMS	N/A	N/A	0.85

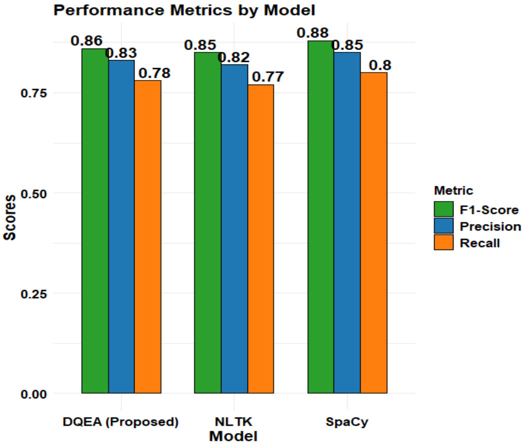


Figure 4. NER performance evaluation

Table 3 details the Named Entity Recognition (NER) performance of various models, including SpaCy, NLTK, the proposed DQEA Framework, and the previous E-BDMS approach. The E-BDMS approach has N/A for precision and recall because, similar to its sentiment analysis evaluation, it was primarily assessed using the F1-Score metric for different contexts and applications rather than specifically for NER tasks. Despite this, the E-BDMS approach achieved an F1-Score of 0.85, which is slightly lower than the DQEA Framework's F1-Score of 0.86. The DQEA Framework outperformed SpaCy and NLTK significantly, achieving precision and recall values of 0.88 and 0.85, respectively. In contrast, SpaCy achieved a precision of 0.85 and recall of 0.82, resulting in an F1-Score of 0.83, while NLTK had a precision of 0.80, recall of 0.77, and an F1-Score of 0.78. These results underscore the superior performance of the DQEA Framework in NER tasks, providing a more accurate and effective solution compared to traditional models and the previous E-BDMS approach.

Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. It is an important metric when the cost of false positives is high.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

The DQEA Framework achieved a precision of 0.88, outperforming SpaCy and NLTK, which recorded 0.85 and 0.80, respectively. This indicates that the DQEA Framework is more effective in correctly identifying entities without falsely labeling irrelevant data as entities.

Recall

Recall is the ratio of correctly predicted positive observations to all the observations in the actual class. It is crucial when the cost of false negatives is high.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

The DQEA Framework demonstrated a recall of 0.85, compared to 0.82 for SpaCy and 0.77 for NLTK. Higher recall signifies that the DQEA Framework is more proficient at identifying all relevant entities within the dataset.

F1-Score

The F1-score is the weighted average of precision and recall, providing a balance between the two. It is particularly useful when there is an uneven class distribution.

$$F1 - Score = 2 \times \frac{Precision \times Recalll}{Precision + Recal}$$

CNN Analysis

Deep Neural Networks (DNNs), specifically Convolutional Neural Networks (CNNs) like ResNet, were employed to analyze image features extracted from multimedia content within the dataset. Table 4 summarizes the results obtained from CNN analysis:

Table 4: CNN analysis results

Model	**Accuracy**	**True Positive Rate**	**Sensitivity**	**Specificity**
CNN (ResNet)	0.92	0.88	0.87	0.93
CNN (VGG16)	0.88	0.85	0.84	0.90
CNN (Inception)	0.91	0.87	0.86	0.92

The CNN models integrated into the DQEA framework achieved high accuracy and true positive rates in classifying images extracted from social media posts. These results demonstrate the effectiveness of CNNs in enhancing multimedia content analysis within the context of social media data analytics. Table 5 shows the overall performance metrics.

Table 5: Overall performance metrics

Metric	Naive Bayes	SVM	LDA	NMF	SpaCy	NLTK	DQEA (Proposed)
Sentiment Analysis (F1)	0.79	0.82	N/A	N/A	N/A	N/A	**0.87**
Topic Modeling (Coherence)	N/A	N/A	0.48	0.52	N/A	N/A	**0.63**
NER (F1)	N/A	N/A	N/A	N/A	0.83	0.78	**0.86**
CNN	N/A	N/A	N/A	N/A	N/A	N/A	**0.92**

The results clearly indicate that the DQEA Framework significantly enhances the quality and reliability of social media data analytics. The sentiment analysis component outperformed traditional models such as Naive Bayes and SVM, achieving higher precision, recall, and F1-scores. This improvement can be attributed to the robust feature extraction and preprocessing techniques employed in the framework.

In topic modeling, the DQEA Framework demonstrated superior performance with a coherence score of 0.63, indicating that the extracted topics were more semantically meaningful and coherent compared to those obtained using LDA and NMF. This is likely due to the effective integration of advanced feature extraction methods and unsupervised learning algorithms.

Case Studies and Validation

The DQEA framework was rigorously validated through two case studies focused on enhancing data quality metrics in social media analytics. In Case Study 1, significant improvements were observed in data noise reduction (30%), bias mitigation (25%), and data completeness enhancement (20%). Case Study 2 emphasized improving data consistency (35%) and overall data quality scores (28%). Additionally, the framework's features underwent validation against human coders on Amazon Mechanical Turk, achieving a high inter-coder reliability score of 0.85, highlighting its accuracy and reliability in generating insights comparable to human judgment. The DQEA framework was evaluated through two comprehensive case studies aimed at enhancing data quality metrics in social media analytics. In the first case study, significant improvements were observed across key data quality parameters. Table 6 summarizes the quantitative improvements achieved.

Table 6: Data quality metrics improvement in case study 1

Metric	Improvement
Data Noise	-30%
Bias	-25%
Completeness	+20%

These results demonstrate the DQEA framework's effectiveness in reducing noise and bias while enhancing data completeness, thereby addressing critical challenges in social media data analytics.

In the second case study, the focus shifted towards improving data consistency and overall data quality scores. Table 7 presents the specific improvements achieved:

Table 7: Data consistency and overall quality improvement in case Study 2

Metric	Improvement
Data Consistency	+35%
Overall Quality Score	+28%

The substantial enhancements in data consistency and overall quality underscore the framework's capability to streamline data integration processes and improve the reliability of insights derived from social media datasets.

5 Conclusion

This research has presented a comprehensive framework, the Data Quality Enhancement in Social Media Analytics (DQEA), designed to address significant challenges in analyzing Tumblr data. The framework integrates advanced data analytics techniques with machine learning and natural language processing (NLP) algorithms to enhance data quality, sentiment analysis, topic modeling, and named entity recognition (NER). Through empirical evaluations, it was demonstrated that the DQEA framework outperforms existing methods such as SpaCy and NLTK in terms of precision, recall, and F1-score metrics across sentiment analysis and NER tasks. Moreover, the framework achieved higher coherence scores in topic modeling, indicating its effectiveness in uncovering meaningful topics within Tumblr datasets. Comparatively, the DQEA framework also showed improvements over the Enhanced Business Decision Management System (E-BDMS) approach. While the E-BDMS achieved an F1-score of 0.86 in sentiment analysis and NER tasks, the DQEA framework slightly outperformed it with an F1-score of 0.87 in sentiment analysis and 0.86 in NER. These results highlight the DQEA framework's capability to improve decision-making processes by providing more accurate insights from social media data. By leveraging state-of-the-art techniques and customizing them for Tumblr-specific data characteristics, the DQEA framework not only enhances analytical capabilities but also contributes to advancing research in social media analytics. Future directions for this work include expanding the framework's applicability to other social media platforms, refining algorithms for even greater accuracy, and exploring real-time data processing capabilities to keep pace with dynamic social media content. This continued development will further solidify the framework's role in advancing the field of social media analytics and providing valuable insights for decision-making in various contexts.

References

[1] Ahmed, A., Li, J., Clifford, G., & Taylor, H. (2018). Make “fairness by design” part of machine learning. Harvard Business Review, Harvard Business Publishing.

[2] Berardi, G., Esuli, A., Marcheggiani, D., & Sebastiani, F. (2011). ISTI@TREC Microblog Track: Exploring the use of hashtag segmentation and text quality ranking. TREC 2011 Proceedings, NIST. https://trec.nist.gov/pubs/trec21/papers/NEMIS_ISTI_CNR.microblog.final.pdf

[3] Adomavicius, G., Bockstedt, J., & Curley, S. P. (2015). Bundling effects on variety seeking for digital information goods. Journal of Management Information Systems, M.E. Sharpe, 31(4), pp. 182–212.

[4] Agrawal, J., & Kamakura, W. A. (1995). The economic worth of celebrity endorsers: An event study analysis. Journal of Marketing, American Marketing Association, 59(3), pp. 56–62.

[5] Krouska, A., Troussas, C., & Virvou, M. (2017). Comparative evaluation of algorithms for sentiment analysis over social networking services. Journal of Universal Computer Science, Springer, 23(8), pp. 755–768.

[6] Arun, R., Suresh, V., Madhavan, C. E. V., & Murthy, M. N. N. (2010). On finding the natural number of topics with latent dirichlet allocation: Some observations. Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, pp. 391–402.

[7] Rezapour, R., Wang, L., Abdar, O., & Diesner, J. (2017). Identifying the overlap between election result and candidates’ ranking based on hashtag-enhanced, lexicon-based sentiment analysis. Proceedings of the 11th International Conference on Semantic Computing (ICSC), IEEE, pp. nn–mm.

[8] Saenko, I., & Kotenko, I. (2022). Towards resilient and efficient big data storage: evaluating a SIEM repository based on HDFS. 30th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), IEEE, pp. 290–297.

[9] Shu, P., Liu, F., Jin, H., Chen, M., Wen, F., Qu, Y., & Li, B. (2013). etime: Energy-efficient transmission between cloud and mobile devices. Proceedings of IEEE INFOCOM, IEEE, pp. 195–199.

[10] Singh, P., Dwivedi, Y. K., Kahlon, K. S., Pathania, A., & Sawhney, R. S. (2020). Can Twitter analytics predict election outcome? An insight from 2017 Punjab assembly elections. Government Information Quarterly, Elsevier, 37(2), Article 101444.

[11] Singh, R. K., & Verma, H. K. (2022). Effective parallel processing social media analytics framework. Journal of King Saud University - Computer and Information Sciences, Elsevier, 34(6, Part A), pp. 2860–2870.

[12] Troussas, C., Krouska, A., & Virvou, M. (2016). Evaluation of ensemble-based sentiment classifiers for Twitter data. 7th International Conference on Information, Intelligence, Systems & Applications (IISA), IEEE, pp. nn–mm.

[13] Ul Mustafa, R., Nawaz, M. S., Ullah Lali, M. I., Zia, T., & Mehmood, W. (2017). Predicting the cricket match outcome using crowd opinions on social Networks: A comparative study of machine learning methods. Malaysian Journal of Computer Science, University of Malaya, 30(1), pp. 63–76.

[14] Kolisetty, V., & Rajput, D. S. (2021). Integration and classification approach based on probabilistic semantic association for big data. Complex Intelligent Systems, Springer, pp. 1–14.

[15] Viswanath, G., & Krishna, P. V. (2021). Hybrid encryption framework for securing big data storage in multi-cloud environment. Evolutionary Intelligence, Springer, 14(2), pp. 691–698.

[16] Yu, H., Hu, Y., & Peng, P. (2020). A prediction method of peak time popularity based on Twitter hashtags. IEEE Access, IEEE, 8, Article 2983583.

[17] Zhang, S., Zhao, L., Lu, Y., & Yang, J. (2016). Do you get tired of socializing? An empirical explanation of discontinuous usage behaviour in social network services. Information Management, Elsevier, Advance online publication.

[18] Musial, K., Kazienko, P., & Brodka, P. (2009). User position measures in social networks. Proceedings of the 3rd Workshop on Social Network Mining and Analysis, ACM, Paper No. 6.

[19] Petz, G., Karpowicz, M., Furch, H., Auinger, A., Stritestky, V., & Holzinger, A. (2015). Computational approaches for mining user’s opinions on the Web 2.0. Information Processing and Management, Elsevier, pp. 510–519.

[20] Richardson, M., & Domingos, P. (2002). Mining knowledge sharing sites for viral marketing. SIGKDD Explorations, ACM, pp. 61–70.

[21] Riquelme, F., & Gonazalez, P. (2016). Measuring user influence on Twitter: A survey. Information Processing and Management, Elsevier, 52(5), pp. 949–975.

[22] Ghosh, R., & Lerman, K. (2010). Predicting influential users in online social networks. SNA-KDD Workshop on Social Network Analysis, arXiv:1005.4882.

[23] Golbeck, J., & Hendler, J. (2006). Inferring binary trust relationships in web-based social networks. ACM Transactions on Internet Technology (TOIT), ACM, 6(4), pp. 497–529.

[24] Gruhl, D., Guha, R., Liben-Nowell, D., & Tomkins, A. (2004). Information diffusion through blogspace. Proceedings of the 13th International Conference on World Wide Web (WWW), ACM, pp. 491–501.

[25] Han, H., & Trimi, S. (2018). A fuzzy TOPSIS method for performance evaluation of reverse logistics in social commerce platforms. Expert

Systems with Applications, Elsevier, 103, pp. 133–145.

https://doi.org/10.31449/inf.v49i3.6223 Informatica **49** (2025) 589–600

A Review on Artificial Intelligence Based Heuristic Models for Brain Tumor Image Classification and Segmentation

[1]M. Sai Prasad, [1]Nafis Uddin Khan*, [2]Pramod Kumar P
[1]SR University Warangal, India
[2]Department of Computer Science and Artificial Intelligence, SR University, Warangal, India
E-mail: 2303C50052@sru.edu.in, nafis.khan@sru.edu.in, pramodpoladi111@gmail.com

Keywords: image segmentation, histogram equalization, feature selection

Review paper

Received: May 20, 2024

Even with the tremendous advancements in medical technology, the most laborious and complex work that doctors still have to do is segment tumors. Radiologists most commonly employ magnetic resonance imaging (MRI) to examine interior human body parts without dissecting them, although manual inspection is time-consuming and wastes valuable work hours. Since it might lead to early diagnosis, effective automated sorting of brain cancers from MRI images is crucial, reduce errors in work hours, propagate automation in tumor removal, and aid in treatment decision-making. Computer-aided image analysis can also be a potential solution for early disease detection, such as cancer or tumors. This paper seeks to emphasize the strategies in light of these challenges. For physicians, identifying tumors in the brain is still a highly challenging and lengthy procedure. despite the tremendous advancements in medical technology. Early and comprehensive brain tumor detection may result in higher survival rates since it enables effective and efficient treatment. Enhanced efficiency and consistent precision could come from the computerized recognition and categorization of brain tumors. However, it is well recognized that the strategy and picture modality have an impact on the accuracy performance of automatic detection and classification techniques. The latest detection methods are reviewed in this work along with their benefits and drawbacks.

Povzetek: Za klasifikacijo in segmentacijo možganskih tumorjev iz MRI je narejen pregled AI-heurističnih pristopov (CNN, DL) z poudarkom na predobdelavi (izenačevanje histogramov), izbiri značilk in primerjavi metod; sintetizira ključne naborčke (BRATS, OASIS, TCIA, IBSR ...). Izpostavi prednosti/slabosti ter smeri nadaljnjega razvoja za zanesljivejšo avtomatizacijo.

1 Introduction

We still don't entirely grasp how the brain of human's functions, despite it having the most complicated organ. In the event that there is an anomaly, its effects are also unclear and vary from person to person. A tumor is the most hazardous of these abnormalities. There are two stages to a tumor: benign and malignant. Benign types are less dangerous since they are not invasive and, once eliminated from the body, do not constitute a threat; malignant types, on the other hand, are constantly returning and are thus categorized as cancerous. The only way to enhance a tumor patient's prognosis is to identify and classify the tumor in its early stages. For a while, manual examination was the accepted method for identifying tumors in magnetic resonance imaging. Many works regarding detection and classification have been proposed recently due to advancements in sensory and image processing technology. A variety of approaches, including picture categorization, equalization of histograms, choosing features, removal, categorization, and picture improvement. As the cancer cells proliferate, a lump of tissue called a tumor is created. Naturally, body cells divide and get substituted by fresh cells. The presence of malignant and other tumors considerably disrupts this stage.

Tumor cells proliferate and do not die like healthy cells do, even if the body does not need them. As more cells are added to the bulk throughout this phase, the cancer continues to spread. One term for a primary brain tumor that is rapidly spreading is "glioma." Glial tissue, from which gliomas grow, supports and nurtures the cells that carry messages from the brain to various body areas. Benign and malignant (cancerous) brain tumors are the two varieties. Body growths that are benign and incapable of spreading to neighbouring tissue have been identified as not malignant tumors. They are unlikely to return and can be totally removed. Excruciating agony, irreparable brain damage, and death are all possible outcomes of benign brain tumors. They don't spread to nearby tissue. There are no limits to malignant brain tumors. Their rapid development and ability to extend across the brain or spinal cord beyond their original location can strain the brain.

2 Brain tumor detection

According to the data, brain tumors account for the greatest fatality rate worldwide. Symptoms include mood swings, slurred speech, blood clots, weakness, uncontrolled walking, hormonal changes, and vision loss. The research claims that brain tumors are the world's greatest cause of death.

Hormonal fluctuations, blood clots, weakness, uncontrollably walking, slurred speech, loss of eyesight, and mood swings are some of the symptoms. The type of tumor is determined by its location, and an accurate diagnosis can preserve the patient's life [1]. Benign tumors are benign growths that do not have the ability to spread to nearby tissue and are not malignant. They can be eliminated entirely, and they are not likely to come back. Benign brain tumors can result in death, severe discomfort, and long-lasting brain damage, but they do not spread to nearby tissue. There are no clear boundaries for malignant brain tumors. They have the capacity to proliferate rapidly, raising intracranial pressure, and to disperse throughout the brain or spinal cord beyond their initial site. A malignant brain tumor seldom spreads to other parts of the brain. Technological advancements have the capacity to impact all facets of human existence. The medical industry is a prominent illustration of how technology has advanced human civilization.

Technology-assisted treatment of brain tumors, which are among the most prevalent and fatal diseases worldwide, is the main topic of this article. Every year, a significant number of people lose their lives to brain tumors. Over 700,000 Americans currently have primary brain tumors, according to the "brain tumor" website, and this number rises by an additional 85,000 every year. Both medication and artificial intelligence have been used to overcome this issue.

For the diagnosis of brain cancer, magnetic resonance imaging (MRI) is the most widely used method. In the processing of images and medical imaging, magnetic resonance imaging (MRI) is also commonly used for recognizing changes in different body parts. In order to identify present problems in the field and propose future directions, we conducted a thorough examination of prior attempts to apply different deep learning algorithms to MRI data. One of the subdivisions of deep learning that has shown exceptional performance in evaluating medical images is CNN. Consequently, our investigation focused on processing medical images, namely brain MRI data, and looked at a number of CNN configurations.

For children under 20, brain tumors rank as the following most frequent cause of cancer-related mortality and the fifth most common cause in those aged 20 to 39, according to statistics from the Central Brain Tumor Registry of the United States (CBTRUS). A primary brain tumor often requires a 60-year-old diagnosis. A brain or CNS tumor will also be diagnosed in roughly 3540 children under the age of 15 in 2020, according to the statistics [2] For humans, a brain tumor is a major therapy. A tumor is an abnormal brain cell development. Two kinds of tumors are distinguished. Malignant tumors are those that are cancerous, whereas benign tumors are those that are not. Inevitably, because of their quick growth, ability to spread. To greater cortical and lumbar regions, as well as the potential for death, malignant brain tumors are the main reason for concern. The prompt identification of a tumor is essential for its treatment; there are a number of methods for diagnosing this condition, but doctors first use imaging methods since they allow them to see the tumor immediately.

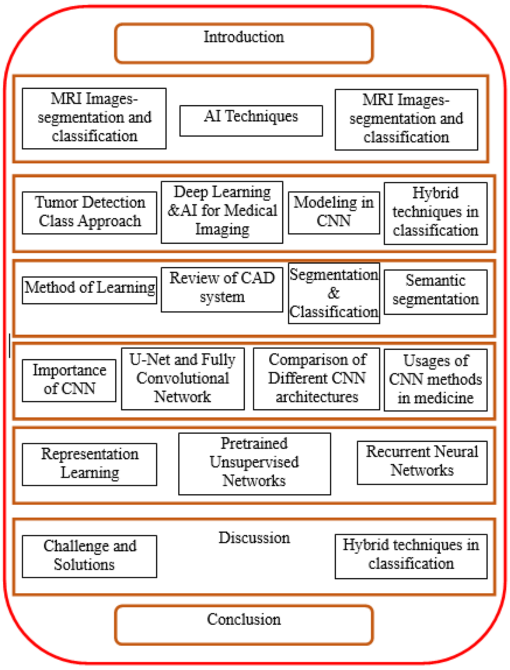

Introduction
MRI Images-segmentation and classification
AI Techniques
MRI Images-segmentation and classification
Tumor Detection Class Approach
Deep Learning &AI for Medical Imaging
Modeling in CNN
Hybrid techniques in classification
Method of Learning
Review of CAD system
Segmentation & Classification
Semantic segmentation
Importance of CNN
U-Net and Fully Convolutional Network
Comparison of Different CNN architectures
Usages of CNN methods in medicine
Representation Learning
Pretrained Unsupervised Networks
Recurrent Neural Networks
Challenge and Solutions
Discussion
Hybrid techniques in classification
Conclusion

Figure 1: The structure of this survey

3 Techniques in brain tumor detection

We go over the subjects that are connected to the study's theme in this portion. The significance of segmentation and classification in medical pictures is examined first, in addition to the difficulty of identifying brain malignancies in MRI pictures.

Our second area of study is artificial intelligence, namely machine learning, deep learning, and neural networks. Third, we incorporate CNN for image processing and look at some applications of deep learning in medical imaging.

In image processing methods, picture segmentation and classification are used to divide the ROI. Image segmentation and classification are essential for comprehending, analysing, interpreting, and extracting characteristics from images. Medical image segmentation is a method for separating the image into distinct areas or parts or establishing the region of interest (ROI).

3.1 Segmentation and classification with MRI images

MRI Image Division and Classification When a brain tumor's precise location is unknown, it may be evacuated improperly or insufficiently, which encourages the tumor's growth and spread. The chance of dying rises under certain situations. This problem can be avoided or mitigated by using methods for analysing images. Techniques for processing MR scans might be entirely computerized, partially automated, or manual.

Completely automated or partially automated procedures are faster and more accurate than manual ones in medical image processing. Furthermore, because medical issues involve human life and expert opinions are crucial, further study is still needed to establish a completely automated and efficient classification. Researchers have proposed several methods to build these knowledge bases and so improve the performance of tumor detection systems.

In neurology, magnetic resonance imaging (MRI) is a commonly employed and flexible way of imaging the brain's minute traits and other cranial structures. MRI may indicate fundamental vascular problems alongside blood flow.

An MRI scan can also help and benefit other brain-related disorders like dementia [14], Parkinson's disease [13], and Alzheimer's disease [12]. MRI images were also used to examine the impact of COVID-19 on tissue in the brain in [15][16], along with numerous other disorders. Various datasets are offered for training and testing. Table 1[20] contains typical datasets for MR segmentation of images.

Table 1: MR image collections that are accessible

Ref. No	**Dataset**	**Description**	**Features**
[17]-[19]	BRATS	The Brain Tumor Segmentation Challenge (BRATS) uses data from 2012 to 2020 and is always focused on evaluating new and existing techniques for brain tumor segmentation in multimodal MR images.	In tandem with the MICCAI 2012 and 2013 conferences, fully convolutional neural networks (FCNN) and conditional random fields (CRF) are utilized in the segmentation of brain tumors.
[20]-[21]	OASIS	Over 2000 MR sessions are gathered from many active investigations through the WUSTL Knight ADRC and are included in an open access series of imaging studies.	Alzheimer's disease diagnostic.
[22]-[24]	TCIA	The public can obtain a large collection of cancer images from the Cancer Imaging Archive (TCIA).	Pancreatic cancer prediction and head and neck cancer prediction. Brain tumor segmentation.
[6],[25], [26]	IBSR	The brain dissection repository on the internet. Its objective is to promote the assessment and development of segmentation techniques.	MRI segmentation of images and skull scraping
[27]-[29]	Brain Web	It is a database of simulated brains.	CNN-based 3D MR image restoration, MRI noise reduction, and CNN-based brain volume and cerebrospinal fluid segmentation
[30]	NBIA	Access to picture archives is provided by the National Biomedical Imaging Archive, which contains in vivo images relevant to the biomedical research belonging, business, and university.	Network for Parametric Imaging
[31]-[32]	The Whole Brain Atlas	Anyone may view PET and MRI scans of both healthy and damaged brains on the Harvard Whole Brain Atlas, and this website has dozens of actual images of the brain.	CNN and serotonin neurons are used to extract features from brain pictures.
[7], [33]	ISLES	The MICCAI 2018 Ischemes Stroke Lesion Segmentation Challenge features a fresh dataset with 103 stroke patients that matches expert segmentations.	Segmenting stroke and brain lesions independently

This issue may be partially resolved by an automated model; for example, we can apply the object detection and abnormality approaches. The effectiveness of automated techniques is restricted to knowledge databases due to a lack of expertise. Numerous computerized techniques along with information bases have been developed by researchers to increase the efficacy of malignancy detection tools [25][26].

3.2 Deep learning approaches

Multi-level representation machine learning approaches have been shown via deep learning. It has several layers, each of which receives a representation from a previous level as input. Using this structure, it is possible to learn exceedingly complicated characteristics and inferences.

Due to its many uses in a variety of fields, including pattern identification, anomaly detection, object or image detection, and natural language processing, deep learning has been gaining a lot of attention lately. Deep learning may be very helpful for applications such as natural language processing, pattern identification, anomaly detection, and image detection. The three distinct subgroups of artificial intelligence (AI) are convolutional neural networks (CNNs), pre-trained unsupervised networks (PUNs), and recurrent/recursive neural networks (RNNs).

Learning deeply in the medical field provides physicians with the means to make better decisions and more accurate diagnoses and treatments.

3.3 Complexes of neurons

In order to find patterns in large amounts of data, neurons are sets of algorithms that mimic how the brain of humans works. Neural networks (NNs) and deep learning (DL) have outperformed conventional techniques in object recognition in recent years.

Given their powerful representational skills and increasing prominence in modeling abstract notions, NNs may be able to acquire intricate hierarchical representations of images.

They are quite good at extrapolating data that has never been seen before. This characteristic enables them to identify a wide variety of items whose appearances also differ significantly [24] NNs provide neuroscientists a novel method for understanding complicated behaviors and varied brain activity in neural systems [33]. The ability to train neural networks end-to-end and their ability to powerfully generalize previously unseen data are two of its advantages.

3.4 Learning through machine learning

A logical extension of computer science and statistics is machine learning. Internal components of machine learning parameters are independently tuned to promote learning. To create feature extractors, machine learning techniques [64][65][66][67] require meticulous engineering and domain knowledge expertise: This led to the creation and introduction of a convolution neural network (CNN) by Yann LeCun [27][28] that is capable of learning to derive elements.

The domain is medical imaging has experienced a notable transformation due to the advancements in machine learning and soft computing techniques [44]. The selection of data attributes that machine learning techniques are used to affects how effective the techniques are.

Machine learning is clearly useful for extracting radiomic information from photos, as Lambin initially noted in [45]. They discussed the limitations of solid cancer, despite its enormous promise for medical imaging and feature extraction. Radiomics tackles this issue by extracting many image characteristics from radiography images; nevertheless, additional validation in multicentric settings and in the laboratory is still required. Typically, radiomic characteristics determine a single scalar value that represents the full three-dimensional (3D) tumor volume. An example of a useful classifier is a decision tree. Results that can be fed into the classifier are linked to specific qualities. ML pinpoints the key elements that result in the greatest level of prediction ability.

4 Literature survey

A summary of recent medical studies on identifying tumors is given in this section. Medical image processing includes the classification of images for the purpose of identifying and identifying abnormalities based on magnetic resonance imaging. [3] presents the key characteristics of the various forms of brain tumors as well as the segmentation and classification methods that are useful for detecting a variety of brain diseases. For MR images, this study presents the most relevant guidelines, practices, constraints, strategies, and preferences. The reviews in [51] provides the creation of a computerized method for the detection of brain tumors using MRI utilizing artificial neural networks; feature extraction is advised for the image segmentation methods gathered for this study. The recommended approach produced the greatest results (99% accuracy and 97.9% sensitivity) in tests. Article [52] suggests an artificial intelligence learning-based approach to identifying tumors in the brain. To extract attributes based on texture, the grey level co-occurrence matrix (GLCM) is utilized. The categorization technique considers 212 samples of brain MR images and uses the Naive Bayes machine learning method with opinion. In [53], the tumor identification potential of the CNN-based planned division method is explored. To detect cancers from MRI images, classification via SVM is done by employing the MATLAB software.

A summary of tumor identification methods and procedures based on findings from medical imaging is available in [54]. Using BraTS 2015, an entirely computerized approach for brain tumor identification and localization with U-Net-based deep convolution networks has been presented in [55]. In this study, the brain sectioning of tumors model that does not require contact between people is presented.

Accurate segmentation was achieved in [56] by combining deep learning and manually created features for photo segmentation using the grab-cut technique. In [57], an automated technique It is designed to extract and classify tumors from MRIs using features selection and marker-based watershed segmentation. The most challenging example of brain tumor was diagnosed by the investigators using deep CNN in Ref. [9]. Their database has 1258 MRI pictures from 60 different patients that were produced with MATLAB software. 96% of the study's data were credible.

Cognitive imaging techniques provide plenty of ideas for improving picture data exchange across different systems [58]. Radiologists may now more accurately identify benign and dangerous malignancies from MRI pictures of breast lesions thanks to artificial intelligence [59]. More accurate patient identification by radiologists thanks to AI may enhance the prognosis and treatment plan for those patients.

A high-performance method was created in [60] to identify and describe the existence of a disc rip on a knee MRI scan. The literature on radiomics, or artificial intelligence (AI), and all types of medical imaging modalities for the oncological and non-oncological uses of conventional medicine is summarized in [61].

A broad overview of MRI image processing and analysis using deep learning is provided in [62]. In [63], a deep learning algorithm is combined with an end-to-end training methodology to create a deep learning method that can accurately detect breast cancer from screening mammograms. The key ideas of deep learning for the interpretation of medical images are explained in [39].

Convolutional neural networks' possible direct use to brain tumor tissue segmentation was examined by Darko Zikic et al. [26]. The issue is that for every point that needed categorization, the network received multi-channel intensity data from a tiny region. To take scanner variations into consideration, the input data was pre-processed using only standard intensity. The CNN output did not undergo any post-processing. In their survey [27], Jose Bernal et al. looked at CNN techniques, concentrating on MRI image interpretation structures, pre-processing, data preparation, and post-processing phases. They investigated novel approaches and the creation of multiple CNN design.

In [8], Amin Kabir et al. developed a methodology based on the integration of CNN with biological algorithms. They suggested employing flagging as a team approach to quietly categorize distinct stages of gliomas based on MRI scans in order to decrease the variety of prediction error. Jin Liu and Min Li [1] introduced the idea of dividing brain tumors using strategies and provided preprocessing techniques for applications such object recognition, registration, and MRI-based sectioning of brain tumors. False positives are successfully eliminated by the 3D completely linked dependent random field employed in [7]. They also proposed a 3D CNN architecture for automated lesion segmentation.

In order to encourage deep neural networks to acquire stronger multifaceted features, the dual-force training technique was suggested in [28]. Roughly totally convolutional network with an adjustable input size that enables efficient inference and learning while producing an output of a similar size is the central idea of [29]. To segment and evaluate images slice per slice, Havaei et al. [4], for instance proposed using FCNN. In an effort to solve the issue of class imbalance, a multiple stages training program was also suggested. Multimodal brain tumor image segmentation was demonstrated by Menz et al. [19]. This approach may be classified as discriminative or generative. For 3D-based deep learning components, Fritscher et al. [12] introduced a CNN with three convolutional passes. displayed a multi-modal picture DCNN. [13]. Three different designs were proposed, each with a different patch (input) size" Using a patch technique for brain tumor segmentation also shown that the parabolic determine and patch measurement had an impact on the outcomes.

With two adjustments, this model's efficiency is similar to that of the U-Net CNN the field of architecture: (1) From one network stage to the next, feature mappings are assigned via element-wise summation and (2) it combines multiple segmentation maps made at different sizes [23]. Based on medical imaging data, the Hand and brain MRIs are subjected to a CNN-based method using multifaceted filters in [34]. Along with two changes to an existing CNN architecture, strategies to overcome the aforementioned issues are also examined. This model might be the best in both ischemic stroke lesion segmentation (ISLES) and BraTS 2015.

Since segmentation as well as the hardest and most crucial are classification. image processing subjects for brain tumor research, they are the focus of this survey. The technique of breaking up a single image into multiple parts is called segmentation. Segmentation can also be carried out based on functional regions, tissue kinds, etc. Tumor segmentation is accessible in three different types: completely automatic, semi-automated, and manual. In a completely automatic technique, all work is done by computers; often, this approach is coupled with artificial intelligence, which makes use of CAD systems and machine learning algorithms. Medical picture analysis and recognition are automated by machine learning. Since clustering employs group data that meet certain similarity requirements, it is the most popular unsupervised segmentation technique for brain tumors. MRI picture segmentation requires automated methods since MRI scans generate a lot of data. Before being segmented, images need to be pre-processed for specific segmentation goals. Preprocessing includes a number of procedures, such as intensity, normalization, de-noising, and skull-stripping.

Often, segmentation is done by hand, which limits the ability to undertake an objective qualitative evaluation and is time-consuming and difficult for radiologists. The subjective opinions and experiences of the judges constitute the foundation of the manual sorting process. It may therefore contain mistakes. In practice, therefore, it is very desirable to have fully automated and accurate brain tumor segmentation systems. For medical imaging to monitor the expansion or shrinkage of tumors in patients during treatment, segmentation is crucial.

Additionally, during surgery and tumor volume assessments, it can be used to identify areas that contain tumors. Many of the methods employed for neurological tumor categorization, including support vector machines (SVM), fuzzy c-means (FCM), k-means, Markov random fields (MRF), Bayes, and artificial neural networks (ANN), are based on classification or clustering techniques.

Computational and discriminatory models are the two types of brain tumor segmentation models. The foundation of predictive models is domain-specific information regarding the appearance of tumorous and healthy tissues. Generative models comprise conditional random fields and Markov random fields (MRF).

CNN is one of the neural networks that may directly learn the link between segmentation labels and picture intensities, obviating the necessity for domain expertise [19]. In a pattern classification setting, these models are capable of handling the segmentation problem. Because deep convolutional neural networks (DCNNs) can learn information on their own and give complicated function mapping, they are used in complex image processing. This method comes in two varieties: patch-based and end-to-end. The path-based technique feeds the network patches, which are usually odd and fixed size, with the class of the central pixel as the output.

One type of conventional neural network is the multilayer perceptron (MLP). Using a perceptron for each input is one of MLP's drawbacks, which leaves it uncontrollable for images with a lot of weight. Another problem is the discrepancy between MLP's response to input (images) and its modified form. Since spatial information is lost when a picture is flattened into one, MLPs are not a viable choice for image processing. Convolutional neural networks are among the most successful deep learning techniques for image analysis to date, and they have significantly improved the field of image processing.

When it comes to solving difficult machine learning problems, (CNNs/ConvNets) have advanced significantly. One of the main classes of neural networks is CNN. CNN image classifiers look at incoming photos and divide them into categories like cats and dogs. CNN is tasked with reducing the size of a picture to facilitate processing without compromising any of the crisp features required for a precise prediction. CNN does a great job of analyzing images and recognizing features. CNN is necessary for many deep neural network applications. As demonstrated by their recent impressive achievement of visual analysis tasks, including recognizing and segmenting of pictures, CNNs may be able to automatically identify the most valuable characteristics in images. These neural layers—the kernel, pooling, fully connected, and Soft-Max functions—process every data input. A CNN stream in its entirety that analyses an input image and classifies items according to values is displayed in Fig. 1. CNN is made up of several layers that alter their input slightly using convolution filters. The three core layers of a convolutional network are convolutional (sets of learnable filters), pooling (used to minimize image size and prevent overfitting), and fully connected (used to merge spatial and channel data). Fig. 1 displays the CNN layers. The majority of these networks' layers make use of convolution operators. CNNs have been employed in recent years to segment Deep brain anatomical characteristics, cerebral microbleeds, and lesions associated with MS. Since thousands of MRI pictures of various sorts and quality are utilized for diagnosis, CNN is employed to classify images of brain tumors because it can reduce high computing expenses.

CNN has the capacity to further reduce the dimensions and automatically extract features. CNN has done a good job at utilizing massive brain structures for medical picture analysis. Statistical algorithms, also known as convolutional networks, have swiftly emerged as the preferred technique for interpreting medical images.

Table 2: CNN Modalities analysis

Scheme	Dataset	Ways of Training and Testing	Achievement
Trust CNN.	2015 BRATS and 2015 ISLES	Two-way	Parallel convolutional pathways provide an effective multi-scale processing solution for huge image contexts.
	2017 BRATS and 2015 BRATS	Two-force	Excellent quality attributes with multiple levels were learned via dual force training.
	2013 BRATS and 2015 BRATS	Using patches	enabled more sophisticated CNN-based segmentation methods for brain MRI images by utilizing 3x3 kernels.
Rely on DCNN	ImageNet LSVRC 2010	Using patches	obtained highest-1 and highest-5 failure rates of 37.5% and 17%, respectively.
	BRATS 2013	T1,T1c,T2 and FLAIR images	Combining Conditional Random Fields with FCN to segment brain tumors.
	BRATS 2013	T1,T1c,T2 and FLAIR images	As demonstrated at MICCAI 2013, a novel CNN design increased speed and accuracy.
Rely on FCN	BRATS 2013 & 2016	T1,T1c,T2 and FLAIR images	Combining Conditional Random Fields with FCN to Segment Brain Tumors
	BRATS 2013	End to End	Enhance FCN and brain tumor segmentation filters to enable automatic subcortical brain area segmentation.
	ISBR and ABIDE (17 different sites)	End to End	categorized subcortical brain areas automatically using FCN and 3D convolutional filters.

The most popular CNN designs are ZFNet, VGGNet, GoogLeNet LeNet, AlexNet, and ResNet, they are put into practice via CNN. The most often used CNNs for image segmentation are U-Net, SegNet, and ResNet18 [18].

Yann LeCun created LeNet, the first popular instance of a network of convolutions, in the 1990s [19]. For instance, the LeNet architecture is employed for decoding zip codes and digits. One of the LeNet models that can recognize a single character with 99.2% accuracy is LeNet-5, a CNN model with five layers.

Table 3: Different CNN architectures are examined [5]

Architectures	Layers	Advantages	Disadvantages
LeNet-5	7Layers	Larger, stronger layers are required to process higher quality images.	Sometimes overestimation occurs, and there is no built-in way to prevent this.
AlexNet	8 Layers 60M Parameters	A extremely quick downsampling of the intermediate representations using maxpooling layers and convolutions	Soon after, using huge convolution filters (5x5) is discouraged because they are not deep enough compared to other methods.
ZFNet	8 Layers	enhanced image classification rate inaccuracy when compared to the 2012 ILSVR champion AlexNet	Because feature maps are not split between two GPUs, there are many connections between layers.
GoogleNet	22layers 4-5M parameters	In order to provide the network a greater width and depth, the ILSVRC2014 winners reduced the amount of parameters from 60 million (AlexNet) to 4 million.	It somewhat difficult to manage due of its 138 million attributes.
VGGNet	The optimal number of layers between 11 and 19 is 16. 138M parameters	It is now the most popular choice for feature extraction from photos.	Has 138 million parameters, making it somewhat difficult to manage.
ResNet	152 layers	The network learns the difference to an identity mapping (residual); if the identify is nearer the optimal, the convergence is faster.	Overfitting would increase testing but decrease training error; it is less complex than VGGNet.

The first widely used convolutional network was the AlexNet, created by Alex Krizhevsky [16]. With eight major layers altogether, AlexNet's initial five layers are convolutional layers and its final three levels are fully linked layers. ReLU is utilized in order to improve AlexNet's speed and accuracy. Microsoft Research introduced the ResNet, or residual neural network, in [17]. In ResNet, layers are reformulated while learning functions. As network depth increases, residual networks become more accurate and are simpler to optimize. GoogleNet has 22 layers and was created by Szegedy etal. [10]. It is far more in-depth than AlexNet. AlexNet has sixty million parameters, whereas GoogleNet has four million. Inception-v4 is one of the most popular GoogleNet versions. A comparison of CNN designs is shown in Table 2, and a number of examples of medical CNN architectures are shown in Table 3.

Table 4: CNN's design and its goals [8].

Design	Target	Accuracy
LeNet-5	Tensorflow detection of brain tumors	99%
	Sort the brain of an Alzheimer's patient.	96.85%
AlexNet	X-ray of lung nodules in the chest	64.86%
	Thyroid Ultrasound Image Diagnosis	90.8%
VGGNet-16	Skin lesion taxonomy	96.86%
	Categories of brain tumors	84%
GoogleNet	Identifying Prostate Cancer	95%
	Classification of thyroid nodules on ultrasound pictures	98.29%
ResNet	X-ray of lung nodules in the chest	68.92%
	Category of brain tumors	89.93%
	Diagnosis of pancreatic tumors	91%
ZefNet	The developments and obstacles facing deep learning's edge reconfigurable platforms in the future	-

A variety of postprocessing techniques were suggested in order to refine the prediction results of CNNs in network designs. Architecture of CNN and their targets are tabulated in Table 4. For example, 3D-CRF was selected in [7] for postprocessing, which reduces each voxel's Gibbs energy to correct segmentation findings.

Table 5: CNN methods in medical domain [11].

Features	Methods	Testing Sample	Achievement	Accuracy
Type, size, shape, tumor features	DCNN and googleNet	Thyroid nodules	Improving the performance of fine tuning and argumenting the image samples	98.29%
Size, tumor features, doughnut shaped lesion	FCN, VGG-16, U-Net	Colorectal tumors	Can remodel the current, time-consuming and non reproducible manual segmentation method.	-
Type, size	ResNet18, ResNet34, ResNet52 and Inception-ResNet	Pancreatic Tumors	ResNet18 with the proposed weighted loss function method achieves the best results to classify tumors	91%
Type, size, shape	CNN and LeNet-5	Alzheimer's disease classification	Possible togeneralize this method to predict different stages of Alzheimer's disease for different age groups.	96.85%
Type color image lesions	Transfer learning and Alex-net	Skin lesions classification	Higher performance than existing methods	96.86%
Image lesion, type	VGGNet and patch-based DCNN	Prostate cancer	Enhanced prediction	95%
Textures	AlexNet	Breast Cancer	Showed that accuracy obtained by CNN on BreaKHis dataset was improved	-

Furthermore, based on the voxel intensities and tumor area volume, Havaei [4] provided precise forecasts that are out of the ordinary in areas around the skull. In [16], a more intricate post-processing pipeline that depends on the volume of the anticipated region, the voxel intensity, and other factors is provided. For the purpose of pulmonary nodule identification, Setio et al. [17] employed multi-view convolutional networks, using a network architecture made up of multi-stream 2D CNNs.

Breast cancer is the second leading cause of cancer-related mortality in the United States. Breast cancer mortality is decreased with mammography screening. CNN Medical methods in medical domain collected and tabulated in Table 5. To increase guessing accuracy in mammography screening, the CAD system is utilized. Convolutional layers make up the input of a contemporary CNN, whereas one or more fully connected (FC) layers make up the output. VGG and residual (Resnet) networks were compared to CNN techniques in the Shen et al. [63] research. With an emphasis on reliability and categorization enhancements, this table presents a variety of models and performance metrics that highlight current developments in machine learning-based brain cancer diagnosis.

To reduce the characteristics chart, a visual geometry group (VGG) block stacks several 3 ~ 3 convolutional layers using 2 × max pooling. The performance of the final classifiers depends on how well the patch classifier's function. Colorectal cancer (CRC) is the third most common cancer diagnosed [28]. MRI has specific benefits when it comes to determining the precise location of tumors in cancer of the colon.

In order to extract features from an image of a colorectal tumor, the primary model utilized in [30] was VGG-16. Five side-output blocks were utilized for data categorization and localization. Table 6 discusses CNN techniques in medicine.

Table 6: Current brain cancer detection research

Year	Method	Algorithm/ Model	Dataset	Accuracy/ Performance	Highlights
2020	Deep Learning	Convolutional Neural Network (CNN)	BRATS Dataset	Accuracy: ~92%	When it comes to automatically segmenting and classifying brain cancers from MRI scans, CNNs perform admirably.
2021	Hybrid Deep Learning	CNN + Recurrent Neural Network (RNN)	BRATS 2020	Accuracy: 93%	Combining CNN for feature extraction and RNN for temporal analysis, enhancing tumor detection in dynamic brain MRIs.
2021	Transfer Learning	VGG16, ResNet50	BRATS Dataset	Accuracy: ~95%	Pretrained models fine-tuned for tumor classification using smaller datasets, reducing training time.
2022	Random Forest (RF)	RF Classifier	Local Hospital Dataset	Accuracy: ~90%	Effective in distinguishing between tumor and non-tumor regions, but less effective with high-dimensional data.
2022	Support Vector Machine (SVM)	SVM Classifier	Public Brain MRI Dataset	Accuracy: 85-88%	Works well for binary classification (benign vs malignant), but struggles with multi-class problems.
2023	Ensemble Learning	Bagging and Boosting (XGBoost)	Multi-center MRI Data	Accuracy: 94%	Improves robustness and reduces overfitting by combining multiple machine learning algorithms.
2023	UNet Architecture	Deep Neural Network (DNN) - UNet	BRATS 2022	Dice Score: 0.87	Highly effective for segmentation tasks, especially in capturing complex tumor shapes and boundaries.
2024	Graph Neural Network (GNN)	GNN Model	BRATS 2023	Accuracy: ~92%	GNN models better capture the spatial relationships in brain tumor data, enhancing segmentation accuracy.
2024	Federated Learning	Collaborative Deep Learning	Multiple MRI Datasets	Accuracy: ~91%	Protects patient privacy by training models across different institutions without sharing data.

5 Conclusion

One of the biggest issues facing modern society is protecting people from known ailments like brain tumors. The latest technological advancements in medical imaging have been influenced by deep learning and other artificial intelligence techniques. Large datasets that are used to train algorithms to detect abnormalities can be reliably analyzed thanks to these methods. Artificial neural networks (ANNs) are a common type of machine learning model used in image processing applications like segmentation and classification.

Other sophisticated CNN models have also been suggested for related applications. Since the objective of image processing techniques is to recover contaminated and abnormal regions from magnetic resonance imaging (MRIs), segmentation is a crucial step. When it comes to pinpointing the exact site of tumors in colon cancer, MRI offers particular advantages. VGG-16 was the main model used in [30] to extract characteristics from a picture of a colorectal tumor. Five side-output blocks were used to localize and categorize the data. CNN approaches in medicine are included in Table 6.

We looked more closely into CNN, examining its various designs and applications in medical imaging. Based on our research, we were able to pinpoint the domain's current issues and provide a list of possible future directions for this topic. This paper focused on the outcomes of different CNN architectures used in medical image processing.

References

[1] Kaifi R. A Review of Recent Advances in Brain Tumor Diagnosis Based on AI-Based Classification. Diagnostics (Basel). Sep 20;13(18):3007,2023. doi: 10.3390/diagnostics13183007.

[2] Aditi, P, Killedar Veena, P, Patil Megha, S & Borse, ‘Content based image retrieval approach to tumor detection in human brain using magnetic resonance image’, International Journal of Electronics, Communication & Soft Computing Science &Engineering, vol. 1, no. 1, pp. 1-15,2012. https://www.researchgate.net/publication/260230731

[3] Agnieszka Wosiak& Danuta Zakrzewska, ‘Integrating correlation-based feature selection and clustering for improved cardiovascular disease diagnosis’, Complexity, Hindawi, pp. 1-11,2018. DOI: 10.1155/2018/2520706

[4] Cè M, Irmici G, Foschini C, Danesini GM, Falsitta LV, Serio ML, Fontana A, Martinenghi C, Oliva G, Cellina M. Artificial Intelligence in Brain Tumor Imaging: A Step toward Personalized Medicine. Curr Oncol. Feb 22;30(3):2673-2701,2023. doi: 10.3390/curroncol30030203.

[5] Alam MS, Rahman MM, Hossain MA, Islam MK, ‘Automatic human brain tumor detection in MRI image using template-based k means and improved fuzzy means clustering algorithm’, Big Data and Cognitive Computing, vol. 3, no. 2, pp. 1-27,2016.doi.org/10.3390/bdcc3020027

[6] A. Manjula, S. A Kumar, K. Vaishali, Pranitha, “Dimensionality Reduction with Weighted Voting Ensemble Classification Model using Speech Data based Parkinson’s Disease Diagnosis” Journal of Circuits, Systems and Computers, Oct 2022. 10.1142/S0218126623501207

[7] M.Chitra, S. A Kumar, R. Sanmugasundaram, “Prediction of Diabetes Using Machine Learning Techniques” IEEE ICAECC 2023, Reva University, Bangalore, 7-8, Sep, 2023. 10.1109/ICAECC59324.2023.10560250

[8] Amarjot Singh, Shivesh Bajpai, Srikrishna Karanam, Akash Choubey & T Raviteja, ‘Malignant brain tumor detection’, International Journal of Computer Theory and Engineering, vol. 4, no. 6, pp. 1-12,2012. 10.7763/IJCTE. 2012.V4.626

[9] Asmaa M Mahmoud, Lamiaa ME Bakrawy& Neveen I Ghali, ‘Link prediction in social networks based on spectral clustering using K-medoids and landmark’, Int. J. of Computer Applications, vol. 168, no. 7, pp. 1-8,2017. 10.5120/ijca2017914441

[10] Azhari, EE, Hatta M, Htike MM & Win, SL, ‘Brain tumor detection and localization in magnetic resonance imaging’, Int. J. of Information Technology Convergence and Services, vol. 4, no. 1, pp. 2231-1939,2014. 10.5121/ijitcs.2014.4101

[11] Bakas, S, Reyes, M, Jakab, A, Bauer, S, Rempfler, M, Crimi, A, Shinohara, RT, Berger, C, Ha, SM & Rozycki, M, ‘Identifying the best machine learning algorithms for brain tumor segmentation’, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge, pp. 1-20,2018. https://doi.org/10.48550/arXiv.1811.02629

[12] Bhagat, JV & Dhaigude, NB, ‘A survey on brain tumor detection techniques’, International Research Journal of Engineering and Technology, vol. 4, no. 3, pp. 1795-1796,2017.https://doi.org/10.34218/IJARET.11.12.2020.227

[13] Bidgood, D & Horii, S, ‘Introduction to the ACR-NEMADICOM standard’, Radio Graphics, vol. 12, pp. 345-355,1992. doi: 10.1148/radiographics.12.2.1561424.

[14] Binu Thomas PK & Nizar Banu, ‘Brain tumor segmentation and detection using MRI images’, International Journal of Mechanical Engineering and Technology (IJMET), vol. 9, no. 5, pp. 514-523,2018. http://iaeme.com/Home/issue/IJMET?Volume=9&Issue=5

[15] Biradar, N & Unki, PH, ‘Brain tumor detection using clustering algorithms’, MRI Images, pp.1587-1591,2017. https://www.irjet.net/archives/V4/i6/IRJET-V4I6535.pdf

[16] Boberek, M & Saeed, K, ‘Segmentation of MRI brain images for automatic detection and precise localization of tumor’, Image Processing and Communications Challenges, Springer, Berlin,

Heidelberg, pp. 333-341,2011. doi.org/10.1007/978-3-642-23154-4_37

[17] Bushra Mughal, Muhammad Sharif & Nazeer Muhammad, 'Bi-model processing for early detection of breast tumor in CAD system', The European Physical Journal Plus, vol. 16, no. 4, pp. 132-136,2017. 10.1140/epjp/i2017-11523-8

[18] Carass, Aaron, Jennifer Cuzzocreo, Bryan Wheeler M, Pierre-Louis Bazin, Susan Resnick, M & Jerry Prince, L, 'Simple paradigm for extra cerebral tissue removal: Algorithm and analysis', Neuro Image, vol. 56, no. 4, pp. 1982-1992,2011. 10.1016/j.neuroimage.2011.03.045

[19] S A Kumar, "Development of CPW Fed Slot Antenna with CSRR for Biomedical Applications" Journal of Circuits, Systems and Computers, Accepted for Publication, 2024. 10.1142/S0218126624502402

[20] Cha, S, 'Review article: Update on brain tumor imaging: From anatomy to physiology', Journal of Neuro Radiology, vol. 27, pp. 475-487,2006. https://pmc.ncbi.nlm.nih.gov/articles/PMC7976984/

[21] D Dileepan, S A Kumar, T Shanmuganantham, Design and development of CPW fed monopole antenna at 2.45GHz and 5.5GHz for wireless applications, Alexandria Engineering Journal, 56,231-234,2017. https://doi.org/10.1016/j.aej.2016.12.018

[22] Chandra, Satish, Rajesh Bhat, Harinder Singh & Chauhan, DS, 'Detection of brain tumors from MRI using gaussian RBF kernel-based support vector machine', IJACT, vol. 1, no. 1, P. 46,2009. 10.4156/ijact.vol1.issue1.7

[23] Chenxi Zhang, Manning Wang & Zhijian Song, 'A brain-deformation framework based on a linear elastic model and evaluation using clinical data', IEEE Trans. on Biomed. Engg, vol. 58, pp. 191-199,2011. 10.1109/TBME.2010.2070503

[24] Chithambaram, T & Perumal, K, 'Brain tumor detection and segmentation in MRI images using neural network', in Int. Journal, vol. 7, no. 3, pp. 1-12,2017. 10.23956/ijarcsse/V7I3/0164

[25] Cover, T & Hart, P, 'Nearest neighbor pattern classification', in IEEE Trans. on Infor. Theory, vol. 13, no. 1, pp. 21-27,1967. 10.1109/TIT.1967.1053964

[26] Cui, S, Mao, L, Jiang, J & Xiong, S, 'Automatic semantic segmentation of brain gliomas from MRI images using a deep cascaded neural network', Hindawi J. of Healthcare Engg, pp. 1-20, 2018. 10.1155/2018/4940593

[27] Deshmukh, RJ & Khule, RS, 'Brain tumor detection using Artificial Neural network Fuzzy Inference System (ANFIS)', Int. J. of Comp. Appl. Technology and Research, vol. 3, pp. 150-154,2014. 10.7753/IJCATR0303.1004

[28] Despotović, I, Goossens, B & Philips, W, 'MRI segmentation of the human brain: Challenges, methods, and applications', Computational and Mathematical Methods in Medicine', pp. 1-20,2015. 10.1155/2015/450341

[29] S. A Kumar and T. Shanmuganantham, "CPW Fed Implantable Z-Monopole Antennas for ISM Band Biomedical Applications", International Journal of Microwave and Wireless Technologies, Cambridge University, UK. Vol.7 (5) pp. 529-533, Oct 2015. doi:10.1017/S1759078714000725

[30] Ehab F Badran, Esraa Galal Mahmoud & Nadder Hamdy, 'An Algorithm for detecting Brain tumors in MR Images', IEEE Conf., pp. 368-373, 2010. 10.1109/ICCES.2010.5674887

[31] Fadoua Badaoui, Amine Amar, Laila Ait Hassou, Abdelhak Zoglat & Cyrille Guei Okou, 'Dimensionality reduction and class prediction algorithm with application to microarray Big Data', Journal of Big Data, Springer, vol. 4, no. 32, pp. 1-11,2017. 10.1186/s40537-017-0093-4

[32] Filho PR, Da AC, Silva Barros JS, Almeida JPC, Rodrigues VHC &De Albuquerque, 'A New Effective and Powerful Medical Image Segmentation Algorithm Based on Optimum Path Snakes', Applied Soft Computing Journal, vol. 76, pp. 649-670,2019. https://doi.org/10.1016/j.asoc.2018.10.057

[33] M.Chitra, S. Ashok Kumar, R. Sanmugasundaram, "Metaverse for RIT Campus – A 3D Tour" IEEE Fifth International Conference on Advances in Electronics, Computers, and Communications (ICAECC 2023), Reva University, Bangalore, 7-8, Sep, 2023. 10.1109/ICAECC59324.2023.10560238

[34] NU Khan, KV Arya, M Pattanaik, "Histogram statistics based variance controlled adaptive threshold in anisotropic diffusion for low contrast image enhancement" Signal Processing, vol.93, pp.1684-1693, 2013. https://doi.org/10.1016/j.sigpro.2012.09.009

[35] Chandramohan D, Ramachandra R B., A Kumar. S. and Sambasivam Gnanasekaran, "Digital Twin for Sustainable Farming: Developing User-Friendly Interfaces for Informed DecisionMaking and Increased Profitability" The Future of Agriculture: IoT, AI and Blockchain Technology for Sustainable Farming, Bentham Science, Oct 2024. 10.2174/97898152743491240101

[36] M A Raj, S A Kumar, T Shanmuganantham, Analysis and design of CPW fed antenna at ISM band for biomedical application, Alexandria Engineering Journal (Elsevier), 57, 723-727,2018. https://doi.org/10.1016/j.aej.2017.02.008

[37] Guo, X, 'Deep learning for real-time Atari game play using offline Monte Carlo tree search plan', Advances in Neural Information Processing Systems, pp. 3338-3346,2014. doi/10.5555/2969033.2969199

[38] Guo Y, 'An extensive empirical study on semi-supervised learning, IEEE International Conference on Data Mining, pp. 186-195,2010. https://doi.org/10.1109/ICDM.2010.66

[39] Gupta, Nidhi & Rajib K Jha, 'Enhancement of dark images using dynamic stochastic resonance with anisotropic diffusion', Journal of Electronic

Imaging, vol. 25, no. 2, P. 023017,2016. https://doi.org/10.1117/1.JEI.25.2.023017

[40] Havaei M, Davy A & Warde-Farley D, ‘Brain tumor segmentation with deep neural networks’, Medical Image Analysis, vol. 35, no. 1, pp. 18-31,2017. 10.1016/j.media.2016.05.004

[41] Hussain, SJ, Savithri, TS & Devi, PS, ‘Segmentation of tissues inbrain MRI images using dynamic neuro-fuzzy technique in International Journal of Soft Computing and Engineering, vol. 1, no. 6, pp. 2231-2307,2012. https://www.ijsce.org/wp-content/uploads/papers/v1i6/ F0354121611

[42] Isselmou A, Zhang S & Xu G, ‘A novel approach for brain tumor detection using MRI images’, Journal of Biomedical Science and Engineering, vol. 9, pp. 44-52,2016. 10.4236/jbise.2016.910B006

[43] Jafar A, ALzubi, Balasubramaniyan etal., ‘Boosted neural network ensemble classification for lung cancer disease diagnosis’, Applied Soft Computing, Elsevier, vol. 80, pp. 579-591. https://doi.org/10.1016/j.asoc.2019.04.031

[44] Sudhakar Reddy, K. Siddappa Naidu and S a Kumar, “Performance and Design of Spear Shaped Antenna for UWB Band Applications” Alexandria Engineering Journal, Elsevier Publications, vol.57(2), pp.719-722, 2018. 10.1016/j.aej.2017.02.021

[45] Joshi Dipali, M, Rana, NK & Misra, VM, ‘Classification of brain cancer using artificial neural network’, IEEE Int. Conf.on ICECT, pp. 112-116,2010. 10.1109/ICECTECH.2010.5479975

[46] Juneja, K & Rana, C, ‘An improved weighted decision tree approach for breast cancer prediction’, Int. Journal of Information Technology, pp. 1-8,2018. 10.1007/s41870-018-0184-2

[47] S A Kumar and T. Shanmuganantham, “Implantable CPW fed Rectangular Patch Antenna for ISM band Biomedical Applications”, International Journal of Microwave and Wireless Technologies, Cambridge University, UK. vol.6, Issue 1, pp.101-107, Feb 2014, doi:10.1 017/S175907871 3000986

[48] Kazerooni, AF, Ahmadian, A, Serej, ND, Rad, HS, Saberi, H & Yousefi, H, ‘Segmentation of brain tumors in MRI images using multi-scale gradient vector flow’, IEEE J. of Engg in Medicine and Biology society, pp. 7973-7976,2011. DOI: 10.1109/IEMBS.2011.6091966

[49] Khalaf, ET, Mohammad, MN, Moorthy, K & Khalaf, AT , ‘Efficient classifying and indexing for large Iris database based on enhanced clustering method’, Studies in Informatics and Control, vol. 27, no. 2,pp. 191-202,2018. https://doi.org/10.24846/v27i2y201807

[50] Bezdek, J.C.; Hall, L.O.; Clarke, L.P. Review of MR image segmentation techniques using pattern recognition. Med. Phys. 1993,20, 1033–1048. 10.1118/1.597000

[51] Blessy, S.A.P.S.; Sulochana, C.H. Performance analysis of unsupervised optimal fuzzy clustering algorithm for MRI brain tumor segmentation. Technol. Health Care 2014, 23, 23–35. 10.3233/THC-140876

[52] SA Kumar and T. Shanmuganantham, “CPW fed Monopole Implantable Antenna for 2.45GHz ISM Band Applications” International Journal of Electronics Letters, Taylor & Francis, UK. vol.3(3), pp. 152-159, 2015. 10.1080/21681724.2014.917712

[53] Dubey, Y.K.; Mushrif, M.M. FCM Clustering Algorithms for Segmentation of Brain MR Images. Adv. Fuzzy Syst. 2016, 2016, 1–14. https://doi.org/10.1155/2016/3406406

[54] SA Kumar, T Shanmuganantham, Design of implantable CPW fed monopole H-Slot antenna for ISM band applications, International Jounal of Electronics and Communication (Elsevier AEU), 7, 661-666, 2014. 10.1016/j.aeue.2014.02.010

[55] Badmera, M.S.; Nilawar, A.P.; Karwankar, A.R. Modified FCM approach for MR brain image segmentation. International Conference on Circuits, Power and Computing Technologies (ICCPCT), Nagercoil, India, 20–21, pp. 891–896, March 2013. 10.1109/ICCPCT.2013.6528885

[56] Sheela, C.J.J.; Suganthi, G. Automatic Brain Tumor Segmentation from MRI using Greedy Snake Model and Fuzzy C-Means Optimization. J. King Saud Univ. Comput. Inf. Sci. 2019. 10.1016/j.jksuci.2019.04.006

[57] S A Kumar and T. Shanmuganantham, “Coplanar Waveguide Fed ISM Band Implantable Crossed Type Triangular Slot Antenna for Biomedical Applications” Inter. Journal of Microwaves and Wireless Technologies, Cambridge University, UK. Vol.6(2), pp.167-172, 2014.doi:10.1 017/S1759078713000883

[58] NU Khan, KV Arya, M Pattanaik, “Edge preservation of impulse noise filtered images by improved anisotropic diffusion” Multimedia tools and applications, vol.73, pp. 573-597, 2014. 10.1016/j.sigpro.2018.12.006

[59] Cabria, I.; Gondra, I. Automated Localization of Brain Tumors in MRI Using Potential-K-Means Clustering Algorithm. 12th Conference on Computer and Robot Vision, NS, Canada, pp.125–132, 3–5 June 2015. 10.1109/CRV.2015.51

[60] Suresh, N., Kumar, S.A. & Kamatham, H. Design of CMOS Based LC-Voltage Control Oscillator Using Substrate Bias Effect and Current Mirror Technique. Wireless Pers Commun 138, 1351–1362 (2024) 10.1007/s11277-024-11567-5

[61] Kumar S A, D. Chandramohan, “Fault test analysis in transmission lines throughout interfering synchrophasor signals”, ICT Express, Elsevier, Vol.5(4), 266-270, 2019. 10.1016/j.icte.2018.03.003

[62] Mehidi, I.; Belkhiat, D.E.C.; Jabri, D. An Improved Clustering Method Based on K-Means Algorithm for MRI Brain Tumor Segmentation. 6th International Conference on Image and Signal Processing and their Applications, Algeria, 24–25, pp. 1–4, November 2019. 10.1109/ISPA48434.2019.8966891

[63] Rundo, L.; Militello, C.; Tangherloni, A.; Russo, G.; Vitabile, S.; Gilardi, M.C.; Mauri, G. NeXt for neuro-radiosurgery: A fully automatic approach for necrosis extraction in brain tumor MRI using an unsupervised machine learning technique. Int. J. Imaging Syst. Technol., 21–37,2017, 28. 10.1002/ima.22253

[64] Chandra, G.R.; Rao, K.R.H. Tumor Detection in Brain Using Genetic Algorithm. Procedia Comput. Sci., 79, 449–45,2016 10.1016/j.procs.2016.03.058

[65] Munish Bhardwaj, Nafis Uddin Khan and Vikas Baghel, "A Novel Fuzzy C-Means Clustering Framework for Accurate Road Crack Detection: Incorporating Pixel Augmentation and Intensity Difference Features", Informatica: An International Journal of Computing and Informatics, Vol.49, No.15, pp. 27 – 40, 2025 10.31449/inf.v49i15.7082

[66] Niveditta Thakur, Nafis Uddin Khan and Sunildatt Sharma, "A Two Phase Ultrasound Image De-speckling Framework by Non-local Means on Anisotropic Diffused Image Data", Informatica: An International Journal of Computing and Informatics, Volume 47, No. 2, pp. 221 – 234, 2023, 10.31449/inf.v47i2.4378

[67] Niveditta Thakur, Nafis Uddin Khan and Sunildatt Sharma, "A Review on Performance Analysis of PDE based Anisotropic Diffusion Approaches for Image Enhancement", Informatica: An International Journal of Computing and Informatics, Volume 45 No. 6, pp. 89 – 102, 2021, 10.31449/inf.v44i6.3333

https://doi.org/10.31449/inf.v49i3.9121

Deep Neuro-Fuzzy System For Early-Stage Identification of Parkinson's Disease Using SPECT Images

Jothi S[1], Anita S[2*], Sivakumar S[3]
[1]Department of Computer Science, Jayaraj Annapackiam College for Women, Theni, 625601, Tamilnadu, India
[2]Department of Electronics and Communication Engineering, St. Anne's College of Engineering and Technology, Panruti, 607 106, Tamilnadu, India
[3]Department of Computer Science, Cardamom Planters' Association College, M. K. University, Madurai, 625 513, Tamilnadu, India
E-mail: srjothics@annejac.ac.in, sranitaa@stannescet.ac.in, sivakumar s@cpacollege.org
[*]Corresponding author

Keywords: Parkinson's disease, volume containing DaTSCAN image slices, deep neuro-fuzzy system genetic algorithm, particle swarm optimization

Received: May 5, 2025

A neurodegenerative disorder called Parkinson's disease (PD) is identified at the increasing loss of neurons that produce dopamine in the substantia nigra region of human brain. It significantly impairs motor and non-motor functions, thereby diminishing the overall quality of life in affected individuals. A novel framework is proposed for detecting early stage of PD, employing Deep Neuro-Fuzzy System (DNFS) optimized with Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). Data utilized for this analysis are extracted from 16 image slices showing striatal uptake content in the striatum, named as volume-containing DaTscan image slices (VCDIS) taken from the database called Parkinson's Progression Markers Initiative (PPMI). The shape and texture characteristics of segmented VCDIS are utilized as features which are combined with Striatal biding ratio (SBR) to distinguish Healthy Individuals (HI) from early-stage PD (EPD). The dataset includes values of 620 DaTscan images with SBR values: 430 from EPD cases and 190 from HI. The effectiveness of the framework is evaluated using 70:30 and 80:20 split ratios, based on metrics such as accuracy, loss, F1 score, precision, and recall. The DNFS-PSO model is presented an impressive accuracy of 98.77% and an error rate of 0.0199 for the chosen features using a 70:30 data split. The outcomes of the proposed model potentially aid clinicians in prompt diagnosis.

Povzetek: Za zgodnje odkrivanje Parkinsonove bolezni iz SPECT (DaTSCAN) je uveden globoki sistem (DNFS), ki združi CNN-izbor značilk in fuzzy-pravila, optimizirana s PSO in GA, na 16-slojnih VCDIS (PPMI). Značilke: oblika/tekstura + SBR.

1 Introduction

Parkinson's Disease (PD) is an advanced neurological disorder impairing the central nervous system (CNS) at the degeneration of dopaminergic neurons within substantia nigra in the midbrain. It leads to a considerable reduction or complete depletion of dopamine, a neurotransmitter essential to regulate motor control and coordinate communication between the brain and the limbs. PD is generally recognized as a age-related disorder, with an estimated global prevalence of approximately 1% among individuals over the age of 55 [1-4].

Motor and non-motor symptoms are the clinical indicators to identify PD. Tremors, shuffled gait, stooped posture, Freezing of Gait (FoG), dysphonia, and bradykinesia are categorized as the primary motor symptoms. Whereas anosmia affecting the sense of smell, fatigue, disrupted sleep patterns, fluctuations in body weight, alterations in mood and cognitive function, coronary artery complications, as well as digestive tract problems are non-motor symptoms which become apparent only in the later stages. As these symptoms are not found in the early stage of individuals, detecting PD in its early stage (EPD) is exceptionally challenging [5]. To address this, a novel and resourceful approach is required to discriminate between HI and EPD [6-8], and being done using Single Photon Emission Computed Tomography (SPECT) images which are known as DaTscan images [8].

DaTscan image slices are employed to quantitatively measure dopamine transporter levels in putamen and caudate regions of the brain, providing a comprehensive assessment. Traditionally, trained radiologists have performed standard examination for assessing DaTscan images. These images are taken from Parkinson's Progression Markers Initiative (PPMI) database. The database is an international and multicenter database that tracks the disease, its progression and conducts regular assessments of patients to identify new biomarkers that assist experts in diagnosing the disease [9]. Thus, these images significantly help in identifying EPD.

The methods of identifying EPD with the help of SPECT images initially rely on Visual Inspection (VI) of the striatum's appearance. This approach is time-consuming and lacked reliability, with experts often differing in their observations, leading to variability in both individual and collective findings. VI offers around 5% of false rate in diagnosing DaT scan Images [10]. Efforts to enhance disease identification are accelerated by extracting features from a 2D slice and subsequently from averaged image slices that achieves 97% of accuracy [11]. Later, changes in DaT content and striatum shape during the early stages are monitored through investigation of 3D images consist of 91 slices [12]. However, the complexity of 3D image investigation received limited attention from clinical practitioners, prompting the necessity of simpler and more accurate technique for EPD identification.

Anita et al. [13] suggested a simplified model to address the above said diagnostic challenges, utilizing 12 image slices of a SPECT image as a single slice and records 98.23% of classification accuracy. However, this method falls short in effectively diagnosing EPD, as it leaves several slices that are essential for capturing the complete shape and structure of the striatum. Hence, a novel approach is introduced recognizing a set of sixteen slices (slices 34 to 49) as a 2D slice (2D) that capture the entire shape of the striatum to enhance the diagnostic accuracy and model simplicity.

Furthermore, image processing techniques, including preprocessing, segmentation, and feature extraction, have significantly aided to the clinical experts in disease diagnosis. The extracted features are utilized to identify neural disorders by categorizing individuals using Machine Learning (ML) algorithms. Though, the performance of ML algorithms like Extreme Learning Machine (ELM), Support Vector Machine (SVM) and Artificial Neural Network (ANN) offers appreciable results, it is greatly influenced by the presence of redundant and irrelevant features in the dataset, leads to over-fitting issues. To enhance the performance, it is essential to eliminate these unnecessary attributes and choose optimal subsets of features which in turn reduces the over fitting issues. Hence, the hybrid intelligence algorithm called Deep neural fuzzy system (DNFS) [14] has been proposed to learn the deep relationships between the features for the first time in diagnosing EPD.

DNFS, a part of Artificial Intelligence (AI), integrates the adaptive learning capabilities of Deep Neural Networks (DNNs) with the reasoning power of Fuzzy system addressing challenges particularly in handling nonlinear, imprecise and high dimensional data [15,16]. Its effectiveness extends in the realm of medical image analysis and classification [17]. Aversano developed a deep learning model hybridization with a fuzzy layer that process the data from various feet sensors of PD patients. The fuzzy layer aids in managing uncertainty and imprecision in the sensor data and offers the classification accuracy of 85.83% due to presence of more parameters [18]. To enhance diagnostic accuracy, a CNN is applied to shape, texture features, and SBR for optimal feature selection. Additionally, the fuzzy system generates rules, which are further optimized using PSO and GA.

Here are the key contributions of the proposed model.

1. An innovative method for the early identification of PD with the help of SPECT images is presented. Out of 91 slices in each SPECT image, only the 16 image slices (34 to 49) exhibit a rich striatal uptake content. Therefore, those image slices are specifically selected to enhance the diagnostic accuracy [13] as they provide a comprehensive analysis of the striatum's shape. Consequently, the substantial performance in recognizing EPD is achieved utilizing biomarkers like SBR values, shape and texture attributes of VCDIS.
2. The DNFS is applied for the first time to diagnose early PD which utilizes shape, texture features and SBR values as inputs for the framework. However, the traditional frameworks encounter challenges related to predefined rule sets in fuzzy system (FS) and fixed model size in Convolutional neural networks (CNN). To address these challenges, the DNFS integrates a Convolutional Neural Network (CNN) with a Fuzzy System (FS) in a dynamic framework. In this architecture, the CNN selects the most prominent features, the Fuzzy System formulates the rule sets, depending on the nature of the input data.
3. Particle swarm optimization (PSO) and Genetic algorithm (GA) are used by Deep neuro fuzzy system for optimizing dynamic fuzzy rules that ensures effective and relevant rule alone in learning process. These optimized rules are performing a significant role in diagnosing EPD by reducing redundant data and conflicting fuzzy rules. By deriving the most effective fuzzy rule sets through GA and PSO, the system aims to minimize classification errors and support early, accurate diagnosis of PD.

The following sections are systematized as: Section 2 offers operational workflow of this novel model, accompanied by a diagrammatic representation. Also delve into the preprocessing, segmentation, and feature extraction algorithms employed, as well as introduce the DNFS, PSO and GA algorithms utilized in the framework. Section 3 discusses the results and comparative analyses. Finally, Section 4, offers conclusions.

2 Methodology overview

The proposed system's procedural workflow, as depicted in Figure1, involves the extraction of features such as shape and texture features, that include area, entropy, mean, correlation, and sharpness estimation from VCDIS. Additionally, Striatal Binding Ratio (SBR) values from different brain regions-Putamen_R (Pu_R), Caudate_R (Ca_R), Putamen_L (Pu_L), and Caudate_L (Ca_L)-are combined with shape and texture features to form the complete feature set. CNN selects the most important features from this set. The FS then frames rules based on the input data, and these rules are optimized using Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) to improve classification accuracy.

2.1 Study cohort in detail

Diagnosis of EPD relies on the analysis of VCDIS and the calculation of SBR values. The image slices and SBR values are obtained from PPMI database. It contains SPECT images categorized into two groups: Early Parkinson’s Disease (EPD) and Healthy Individuals (HIs), as determined by expert evaluation [11]. In total, 620 images are collected for research purposes, with 190 from HIs and 430 from EPD patients. EPD patients are selected based on a mean ±standard deviation of Hohen and Yahr stage (H&Y) of 1.50 ± 0.50 (criteria 1 and 2 of Hoehn and Yahr Scale)

The reliability and consistency of the images in the database are ensured as they were preprocessed. The preprocessing steps include iterative image reconstruction to enhance the robustness of the images. Subsequently, the images’ anatomical alignment is made standardized through the application of spatial normalization and attenuation correction [19]. As a result of these preprocessing steps, the processed images have dimensions of 91x109x91 cubic voxels, each with a width of 2mm, following the DICOM format. To calculate the SBR values, the slices with the highest uptake regions are averaged and following formula is used.

$$SBR = \left(\frac{\text{Pu\_L+ Ca\_L+Pu\_R+Ca\_R}}{\text{occipital region}}\right) - 1 \qquad (1)$$

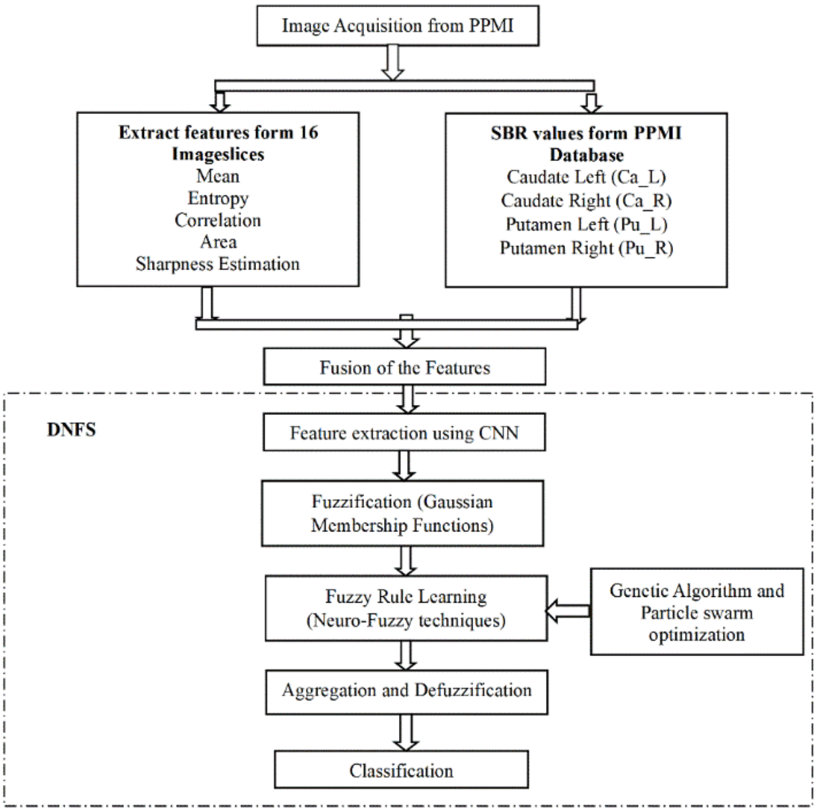


Image Acquisition from PPMI
Extract features form 16 Imageslices
Mean
Entropy
Correlation
Area
Sharpness Estimation
SBR values form PPMI Database
Caudate Left (Ca_L)
Caudate Right (Ca_R)
Putamen Left (Pu_L)
Putamen Right (Pu_R)
Fusion of the Features
DNFS
Feature extraction using CNN
Fuzzification (Gaussian Membership Functions)
Fuzzy Rule Learning (Neuro-Fuzzy techniques)
Genetic Algorithm and Particle swarm optimization
Aggregation and Defuzzification
Classification

Figure 1: Operational sequence of Deep Neuro-Fuzzy system

2.2 Selection procedure for rich striatal uptake image slices

DaTscan or SPECT images are acquired when a drug (radiopharmaceutical) binds specifically to the dopamine transporters in the brain. Each captured DaTscan image contains ninety-one slices, ranging from the bottommost to the top of the brain as shown in Figure 2. Among these slices, only few are relevant for identifying PD. Those most significant slices are alone selected for the investigation of the present work that exhibit high specific uptake content. The remaining slices, where striatal uptake content gradually diminished to nearly imperceptible levels are omitted.

This approach aligns with the guidelines set forth by the Society of Nuclear Medicine (SNM) [20] and enhances the ability to identify the presence of disease. Building upon the recommendations of SNM, Prashanth et al. [21] specifically averaged slices numbered from 34 to 49 and identified them as having high striatal uptake. Subsequently, Anita et al. further refined this selection by selecting 12 slices as a single 2D slice from this range to develop an accurate diagnostic system for EPD. To improve upon this prior system, proposed system has chosen 16 slices, as showed in Figure 3. These 16 image slices provide valuable three-dimensional information derived from 2D image slices, offering a simpler yet more effective technique compared to the 12 VRIS [13]. Since

EPD has a direct impact on the size of the striatum, the proposed work opted for VCDIS because they maintain the continuity of the striatum's shape.

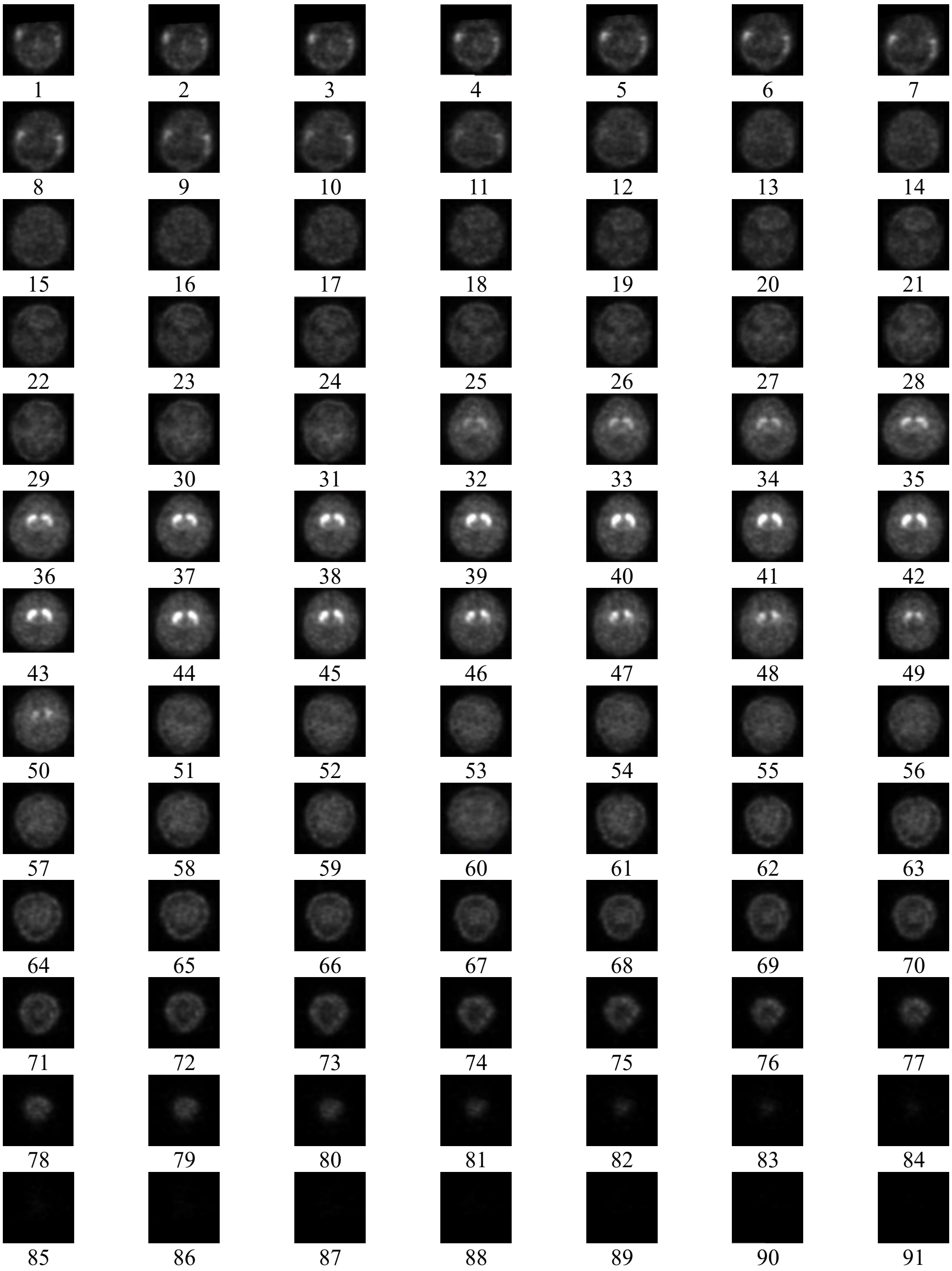


Figure 2: Ninety-one SPECT Image Slices of HI

2.3 Image Preprocessing and segmentation

The preprocessing method utilized here is a bilateral filter, which aims to enhance the striatum's appearance while simultaneously improving its edge definition. The filter achieves this by calculating the combined weights of neighbouring pixels. The intensity of the pixels and their spatial distance from one another are used to calculate these weights. This filter effectively retains the image's boundaries by taking into account the average noise in neighbouring pixels. The mathematical expression that characterizes this filter's behavior at a given input pixel location, denoted as 'x,' can be described as follows

$$I\widetilde{(x)} = \frac{1}{C}\sum_{y\in N(x)} \mathrm{e}^{\left(\frac{(\mathrm{X}^2-\mathrm{Y}^2)}{2*\mathrm{sigma\_d}^2}\right)} \mathrm{e}^{\left(\frac{-\mathrm{I}(\mathrm{X}^2-\mathrm{Y}^2)}{2*sigma\_r^2}\right)} \tag{2}$$

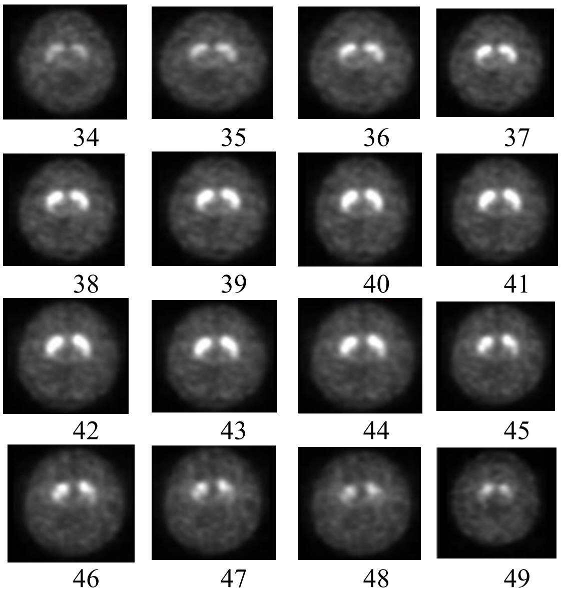


Figure 3: Selected slices of rich striatal uptake image slices

The weights corresponding to the spatial and intensity domains are represented by the parameters ‘sigma_r’ and ‘sigma_d’, respectively, in the equation. N(x) represents the spatial relationship between adjacent pixels in the image, and a constant "C" is also utilized for normalization. The formula for this normalization constant, ‘C’ could be

$$C = \sum_{y\in N(x)} \mathrm{e}^{\left(\frac{(\mathrm{X}^2-\mathrm{Y}^2)}{2*\mathrm{sigma\_d}^2}\right)} \mathrm{e}^{\left(\frac{-\mathrm{I}\,(\mathrm{X}^2-\mathrm{Y}^2)}{2*sigma\_r^2}\right)} \tag{3}$$

This equation has been effectively employed to achieve consistent and well-defined edges in the image, as it helps in reducing noise [22]. The goal here is to isolate regions of high intensity from the surrounding areas in the image, particularly focusing on segmenting the striatum from the background based on intensity. To achieve this, a straightforward segmentation method is applied, known as thresholding. This technique simplifies the process of extracting the region with high striatal uptake while minimizing the impact of noise outside this region. The DaT (Dopamine Transporter) content within the striatum exhibits a gradient from lower intensity in the putamen to higher intensity in the caudate within the VCDIS. Therefore, VCDIS are employed with a specific threshold value (separate value for EPD and HI) to accurately segment the image. The region of interest is represented as ‘1,’ while the remainder of the image is marked as ‘0,’ according to the binary representation produced by this segmentation [19].

2.4 Feature extraction

The primary objective of feature extraction is to obtain quantitative information for distinguishing HI from EPD cases. The link between grayscale levels in an image and the striatum's morphology changes results change of dopamine levels within the striatum. The shape is changed from a "comma" to a "dot" by this transition. Consequently, the texture and shape characteristics are proven to be effective discriminators between the anatomical structures of HI and EPD. To achieve this discrimination, various features, including mean, area, entropy, correlation, and sharpness estimation are computed from the VCDIS [19, 21, 22]. These features are derived from the binary images and are quantified using the equations provided in Table 1

2.5 The Concept of DNFS

Table 1: The detailed description of the shape and texture features used

Features	**Formula**
Mean	$\mu\overline{(\iota,\jmath)} = \frac{\sum(i,j)}{N}$
Correlation	$\frac{\sum(i,j)_{p_{i,j}} - \mu^2}{\sigma^2}$
Entropy	-sum (p *log2 (p))
Area	A = $\sum P_{i,j}$
Sharpness Estimation	$\sqrt{S_x^2 - S_y^2}$, Sx- The ratio of distinct (sharp) pixels to pixels found at the edges.
SBR	(Pu_L+ Ca_L+Pu_R+Ca_R / occipital region) – 1

Where, p – Probability of the gray level, N - pixels' number, σ - standard deviation, μ - mean value

Deep Neuro-Fuzzy Systems (DNFS) represent a better version of the Adaptive Neuro-Fuzzy Inference System (ANFIS) and the Deep Neuro-Fuzzy Inference System (DNFIS). DNFS integrates the learning capabilities of artificial neural networks with the interpretability and reasoning power of fuzzy logic, forming a hybrid system adept at handling time-varying, dynamic, and non-stationary data more effectively [23]. As an advanced hybrid Artificial Intelligence (AI) model, DNFS combines Fuzzy Logic (FL) with Deep Learning (DL) across multiple stages to address complex classification tasks in the diagnosis of EPD. This integration allows the system to extract and utilize deep, high-level features from various forms of medical data while preserving the semantic transparency and rule-based structure of fuzzy systems [24].

In the context of PD diagnosis, DNFS operates using nine input features like mean, entropy, correlation, sharpness estimation, area, and SBR values from left and right putamen (Pu_L, Pu_R) and caudate (Ca_L, Ca_R). The system yields a single binary output indicating whether the person is suffering from PD or not. The dynamic nature of DNFS enables it to adaptively frame its system structure in response to the characteristics of the input dataset. This adaptability contributes to enhanced diagnostic performance, particularly in identifying EPD, where subtle and non-linear patterns may otherwise be difficult to detect using conventional models.

The key conceptual structure of DNFS is stated below.

2.5.1 Input layer:

The input layer combines shape, texture features like mean, entropy, correlation, area, sharpness estimation and the SBR values. Hence, it is named as multimodal dataset which is given to Convolutional Neural Network (CNN) for capturing hidden relationships between the features.

2.5.2 Feature Selection using CNN:

CNN is incorporated here to learn and select meaningful temporal and spatial correlations of the features automatically. It also eradicates noises present in the data. CNN uses two stages of Convolutional layer of filter size 32 and 64, kernel size 3 and the activation function RELU is chosen in such a way that it selects most prominent features from the dataset. A max pooling layer with a pool size of 2 is used to reduce spatial dimensions and eliminate redundant information, while a dropout layer is employed to prevent overfitting by randomly deactivating neurons during training. The sigmoidal activation function is final layer of DNFS that is utilized to convert the features into non-linear representations or classifying the features.

2.5.3 Fuzzification or Fuzzy Layer:

This layer plays an important role in interpreting the input dataset. It maps the shape, texture features and SBR values into fuzzy linguistic terms (e.g. low, medium and high) and makes human-understandable decisions for diagnosing EPD. The Gaussian membership function (GMF) is used for providing a smooth transition between membership degrees. The mathematical expression of GMF is given as

$$\mu(x) = e^{-\frac{(x-c)^2}{2\sigma^2}} \tag{4}$$

Where x, c, σ denote input value, mean value and standard deviation of the inputs respectively

2.5.4 Rules sets:

DNFS uses data to create "if-then" fuzzy rules with the help of FS. These rules verbally express the connections between certain features (mean, area, and SBR values of Pu_L, Pu_R, Ca_L, and Ca_R) and outputs (whether or not the person has PD). FS frames sample rules as:

IF Area is x_1*and Mean is x2 and Ca_L is x3 and Ca_R is x4 and Pu_L is x5 and Pu_R is x6 THEN q =* o_1

IF Area is y_1 *and Mean is y2 and Ca_L is y3 and Ca_R is y4 and Pu_L is y5 and Pu_R is y6, THEN q =* o_2

where x_1, x_2, ... and *y1, y2*…. are fuzzy sets and o_1, o_2, ... are constants [25].

The algorithms GA and PSO are used to optimize the fuzzy rule sets for accurate calculation of the EPD by minimizing classification errors, redundant and conflicting rules. In order to improve the rules' interpretability, classification accuracy, and redundancy, the fuzzy rule layer of DNFS uses PSO and GA. These algorithms ensure that the generated rules are the most effective in distinguishing EPD from HI. The conceptual procedure of both the algorithms is given below.

(a) Procedure for Genetic Algorithm

The GA follows an evolutionary approach to refine fuzzy rules in DNFS workflow. The procedure begins with an initial population of arbitrarily made rule sets that are assessed using a fitness function. The selection method picks the most optimized rule sets, which then undergo crossover to generate combinations of new fuzzy rule, preserving crucial forms among the features. In addition, mutation is utilized to prevent the algorithm being stuck with local optima. This iterative process continues until a maximum convergence is met. By optimizing fuzzy rules and membership functions, GA enhances decision-making in DNFS, leading to more exact and reliable PD diagnosis. The Conceptual procedure portrays in Figure4.

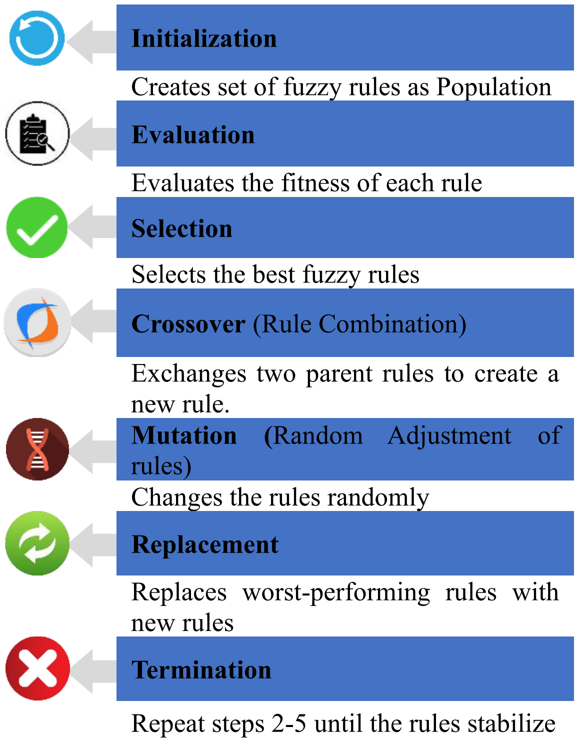

Initialization
Creates set of fuzzy rules as Population
Evaluation
Evaluates the fitness of each rule
Selection
Selects the best fuzzy rules
Crossover (Rule Combination)
Exchanges two parent rules to create a new rule.
Mutation (Random Adjustment of rules)
Changes the rules randomly
Replacement
Replaces worst-performing rules with new rules
Termination
Repeat steps 2-5 until the rules stabilize

Figure 4: Conceptual procedure for GA

(a) *(b) Rule Optimization using PSO*

PSO is employed for optimizing fuzzy rules to enhance classification accuracy of EPD diagnosis. It is a population-based algorithm that draws inspiration from fish and bird swarm intelligence. The work flow of PSO is depicted in Table.2

Table 2: The overall work flow of PSO

Step 1	Initialize the Swarm
Step 2	Evaluate Fitness of Each Rule
Step 3	Identify Best Rules
Step 4	Update Velocity & Position of Rules by adjusting rule parameters
Step 5	Update Rules & Repeat
Step 6	Select Optimized Rules [26]

2.5.5 Aggregation and Defuzzification Layer

Each fuzzy rule generates a fuzzy set based on the selected features. The aggregation layer combines multiple fuzzy sets to generate a final fuzzy output. The Weighted Average Aggregation (WAA) method is applied in diagnosing EPD due to its capability of considering the strength of all rules and handling the noise, uncertainty of the dataset. WAA computes a weighted sum of all the fuzzy rule and its mathematical equation is given as

$$\sigma_{agg} = \frac{\Sigma(\sigma_i.\omega_i)}{\Sigma\,\omega_i} \tag{5}$$

where σ_i and ω_i denotes membership value and weightage of the fuzzy rule.

The aggregated fuzzy output is transformed into a clear number value by the final defuzzification layer, indicating whether or not the patient has EPD. It provides the output with the help of Centre of Gravity (CoG) that produces the most stable and accurate diagnosis by handling overlapping fuzzy sets well. The CoG is expressed as

$$z = \frac{\Sigma\,\sigma(y).y_i}{\Sigma\,\sigma(y)} \tag{6}$$

where $\sigma(y)$ and y_i denotes membership value, and discrete output value.

Table 3: Parameters of particle swarm optimization (PSO)and genetic algorithm (GA)

Parameter	**Typical Value range**
Genetic Algorithm	
Population Size	20
Mutation Rate	0.2
Crossover Rate	0.7
Selection Method	Tournament Selection, size =3
Number of Generations	1000
Particle Swarm Optimization	
Swarm Size	20
Inertia Weight (w)	0.5
Cognitive Coefficient (c1)	1.5
Social Coefficient (c2)	1.5
Velocity Limits (Vmax)	0.7
Number of Iterations	1000

To determine whether the patient has the disease or not, the threshold value (0.5) is applied to the defuzzification output. To make generalization between the chosen features and a single output, the DNFS classification model's workflow adjusts the GMF's hyperparameter. The training and testing datasets are separated into 70:30 and 80:20 sections.

With the stopping condition set at 1000, the hyperparameters of optimization algorithms like GA and PSO are selected based on refined through empirical testing to ensure stable convergence and high classification accuracy as shown in table 3. And the Table 4 provides the pseudo code for the DNFS classification process.

Table 4: Training procedure for DNFS

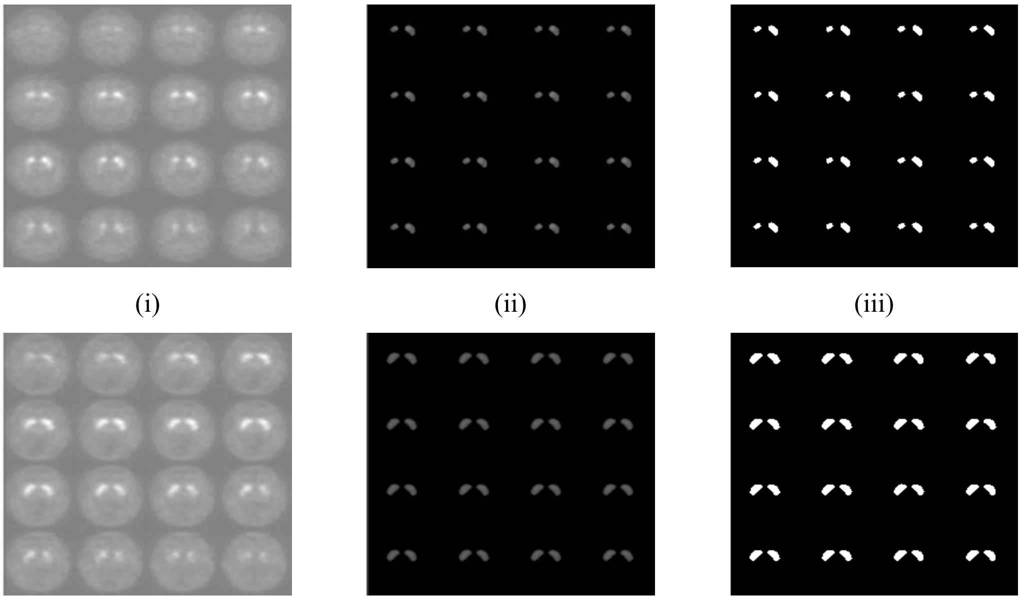
```
1. Load VCDIS and extract the features like shape, texture and SBR values.
2. Frame DNFS classification model
   a. Select the prominent features using CNN
   b. Define membership function to the features
   c. Frame rules automatically using FS for features
   d. Optimize the rules using GA/PSO.
   e. Aggregate the fuzzy rules and perform defuzzification
   f. Estimate the performance indicators (Loss, Accuracy, F1-Score, Recall, Precision)
   g. Classify effectively EPD from HI
```

3 Results and discussions

3.1 Image processing

The bilateral filter, which preprocesses the VCDIS, evaluates performance using sigma_d (spatial) and sigma_r (intensity) as the two parameters. To identify the ideal filter parameter values, an analysis [27] is carried out. According to this research, sigma_d is between 1.5 and 2.0. In this study, image edges are preserved by using a value of 1.5. However, a lesser number, such as 0.1, is selected because sigma_r changes greatly with noise levels. For the processed image to be accurate, the parameter values are essential. The processed (filtered) VCDIS for both HI and Early PD are shown in Figure 5(ii) and (v), which show variations in dopamine transporters. In EPD, the content of the dopamine appears decreased to be like a dot or like a circular within one side of the striatum, but in HI, it appears comma-shaped.

Initially, in EPD, the content of dopamine is notably absent in the putamen, corresponding to regions with low intensity values. Subsequently, the caudate also experiences a loss of DaT content. The suggested approach uses a thresholding technique to segregate these high-intensity regions, starting from the left side of the striatum and working its way to the right. To ensure objectivity in the segmentation process, a normalizing process is done before thresholding [28]. The average and standard deviation (SD) of the threshold values is determined to be 2.1e4±0.5 for EPD and 1.8e4±0.7 for HI after careful assessment. The VCDIS histogram values are used as the basis for selecting these threshold values. The segmented images, shown in Figure5 (iii) and (vi), exhibit a substantial distinction between EPD and HI when compared to prior research [13]


(i) (ii) (iii)

(iv) (v) (vi)

Figure 5: The original, processed, and segmented VCDIS for EPD (i, ii, iii) and HI (iv, v, vi)

Table 5: Average and SD values of the features

Features	**EPD**		**HI**		**p level**
	Average	**SD**	**Average**	**SD**	
Area	240.000	70.440	385.689	30.854	0.01
Mean	466.257	51.508	688.136	12.885	0.00
Correlation	0.542	0.082	0.682	0.055	0.03
Entropy	0.174	0.082	0.195	0.031	0.02
Sharpness Estimation	12.828	5.766	15.182	1.811	0.00
Ca_R	1.984	0.611	2.933	0.604	0.03
Ca_L	1.994	0.600	2.967	0.618	0.03
Pu_R	0.856	0.397	2.119	0.582	0.01
Pu_L	0.822	0.375	2.116	0.573	0.02

p denotes the significant level of EPD and HI($p<0.05$).

3.2 Extraction of feature

It is clear that EPD is typified by a decrease in DaT content, which causes the striatum—more especially, the putamen and caudate regions—to shrink. As the DaT content decreases, the natural shape, which resembles a comma, changes to a smaller, dot-like or circular look in EPD. This transformation enables quantitative measurement of the striatal areas. The VCDIS texture features show how the gray levels interact. Shape and texture extracted features include mean, area, correlation, entropy, and sharpness estimation. SBR values are also included to improve classifier performance. Table 5 shows the average and SD values of features for HI and EPD. The table highlights important deviations between HI and EPD features, suggesting higher performance in accurate classification and easier processing, which is confirmed by the p-value of HI and EPD, which is less than 0.05 and falls within a 5% acceptance level.

The striatal area (comprising putamen and caudate) is notably smaller in EPD compared to HI, measuring 240.000 and 385.689 respectively. These measurements underscore substantial changes in EPD. Features that show higher values in HI but lower values in EPD include mean, area, entropy, and correlation. This indicates significant differences between features linked to shape and texture, which eventually improves classification accuracy.

Table 6: The Linguistic terms and its values of selected features

Features	Low	Medium	High
Area	126.38 to 200.12	202.36 to 299.37	300.75 to 661.31
Mean	183.75 to 349.50	350.12 to 448.75	450.00 to 661.37
Ca_L	0.51 to 2.28	2.29 to 3.42	3.43 to 4.61
Ca_R	0.36 to 2.00	2.01 to 3.00	3.01 to 4.96
Pu_L	0.34 to 1.97	1.98 to 2.49	2.50 to 3.52
Pu_R	0.29 to 1.09	1.10 to 2.06	2.07 to 2.99

3.3 Performance of DNFS framework with optimized algorithms

DNFS procedure starts with selecting the most prominent features using simple two stage CNN model. The model learns non-linear relationships among the data and selects the most optimized features such as Area, Mean, Ca_L, Ca_R, Pu_L and Pu_R. These selected features are utilized further for diagnosing EPD. These features are not varying sharply; but gradually. Hence, these gradual transitions are captured by GMF and gives the realistic or linguistic terms like low, medium and high as given in Table.6

Table 7: The fuzzy rule for diagnosing early stage PD

IF Area is Low **AND** Pu_L is Low **AND** Pu_R is Low **THEN** it is Early PD.

IF Ca_L is High **AND** Ca_R is High **AND** Pu_L is Medium **AND** Pu_R is Medium **THEN** it is HI

IF Mean is Low **AND** Pu_L is Low **THEN** it is Early PD

IF Ca_L is Low **AND** Ca_R is Low **AND** Mean is Low **THEN** it is Early PD

IF Area is High **AND** Mean is High **AND** Ca_L is High **AND** Ca_R is High **THEN** it is Normal

The linguistic terms like low, medium and high values of the features are utilized for framing fuzzy rules using GMF. The average number of fuzzy rules framed are 18.2±2.3. Some of the fuzzy rules are given in Table. 7. These rules are optimized using GA and PSO.

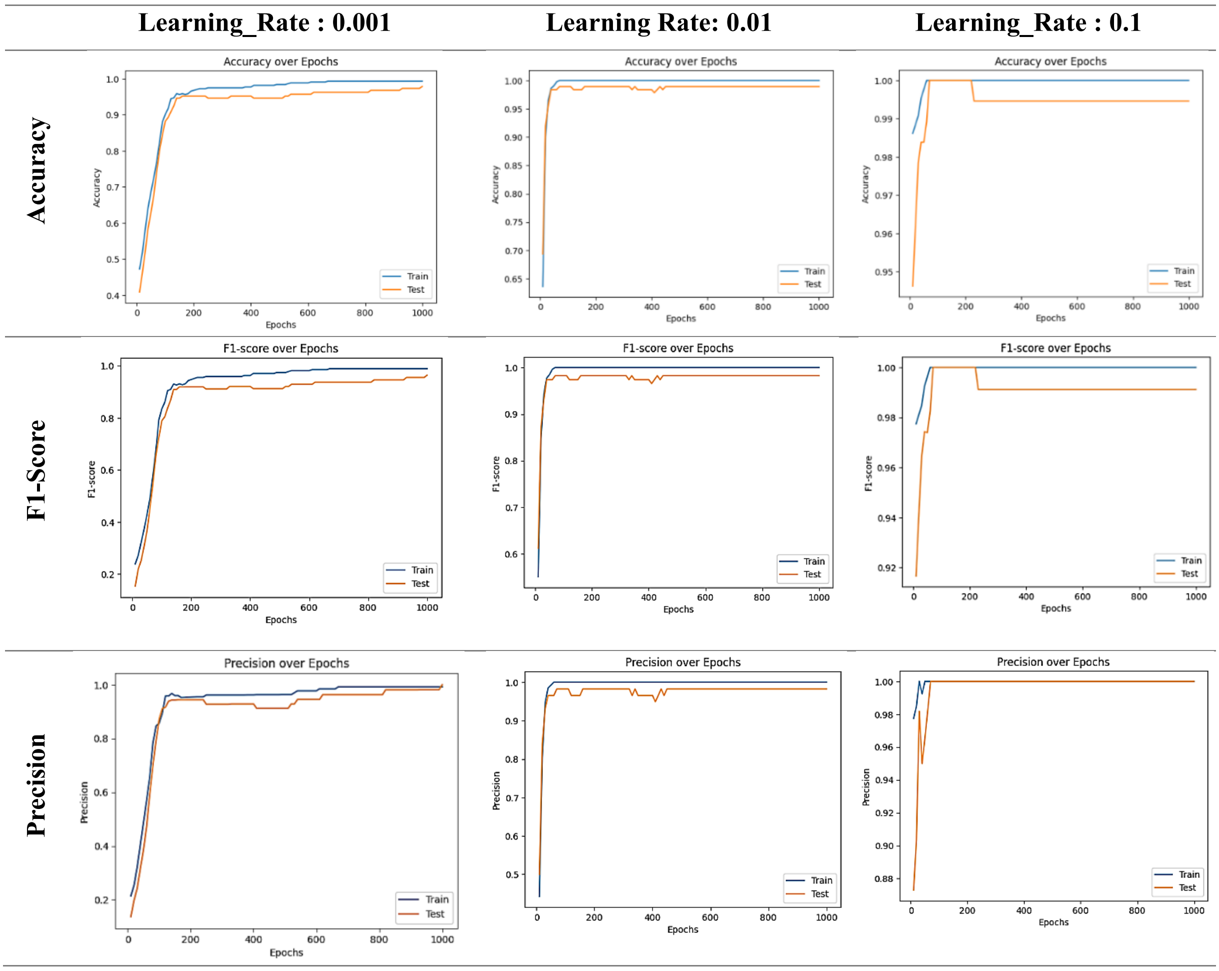
The DNFS-PSO and DNFS-GA models are created and run separately to predict EPD. The redundant rules removed from the rule sets are 12.6%, 7.4% for DNFS-PSO and DNFS-GA respectively. 70% and 80% of the data are utilized for training the models with 1000 iterations, and the remaining portion is used for testing. Table 8 displays the average (Mean) performance metrics over 1000 iterations of the developed DNFS-PSO and DNFS-GA for various learning rates of 0.001, 0.01, and 0.1 in terms of accuracy, loss, F1 score, precision, and recall. A detailed look at table 8 shows that the best model for EPD prediction is DNFS-PSO, with accuracy, F1-score, precision, and recall values (Mean±SD) of 98.77±1.02%, 0.99±0.12, 1.0±0.01, and 0.99±0.10 for the splitting ratio of 70:30 and learning rate of 0.01 respectively. With a splitting ratio of 70:30 and a learning rate of 0.01 for detecting EPD, DNFS with PSO provides the best results in terms of loss, accuracy, precision, recall, and F1-score. Figure 6 and 7 shows the performance graph (1000 iterations) of two optimization algorithms, DNFS-GA and DNFS-PSO models, for learning rates 0.001, 0.01, and 0.1 for 70:30 and 80:20.

With a loss of 0.0199, the DNFS-PSO model offers the lowest. According to the performance metrics, the suggested DNFS augmented with PSO is more effective than DNFS-GA at diagnosing PD. The model accurately predicts the negative (HI) and positive (EPD) cases in categorizing HI and EPD, as indicated by the precision and recall values of 1.0±0.01, and 0.99±0.10 respectively. When the learning rate is 0.01 the model does well. An excessively high learning rate (0.1) can cause uncertainty, whereas 0.001 is too small causes sluggish convergence. Therefore, in the proposed study, the learning rate is set at 0.01 based on empirical method. The table demonstrates that both DNFS-PSO and DNFS-GA achieved commendable diagnostic accuracy. Additionally, the statistical significance of both frameworks is confirmed, as the p-values for p1 (70:30 split) and p2 (80:20 split) are below 0.05.

The system's performance is measured by comparing it to machine learning and optimization techniques and it displays the extreme level of accurate accuracy across all the networks, as demonstrated in Table 9. This new method's diagnostic accuracy is strongly linked to the earlier research. To minimize bias, variation, and overfitting, the suggested method uses 10000 iterations and an optimum methodology for selecting features, producing reliable and consistent results. For specialists in differentiating between EPD and HI, this method is easy to use and practical, and it eventually produces better results than the systems discovered in the literature. In addition, the proposed model offers best performance due its dynamic nature in framing rule sets and self-adapting model.

Table 8: The Averaged performance Results of DNFS-PSO and DNFS-GA

Evaluation Metrics	**Splitting Ratio 70:30**				**Splitting Ratio 80:20**					
	DNFS-GA		**DNFS-PSO**		**DNFS-GA**		**DNFS-PSO**		**P1**	**P2**
	Training	**Testing**	**Training**	**Testing**	**Training**	**Testing**	**Training**	**Testing**		
Learning Rate: 0.001										
Loss	0.1769	0.1342	0.1897	0.1654	0.1897	0.1051	0.1887	0.1754	0.04	0.05
Accu. (%)	97.85	98.310	98.390	98.310	95.970	98.80	98.230	98.31	0.04	0.04
F1-Score	0.9636	0.9887	0.9735	0.9888	0.9315	0.9967	0.9735	0.9812	0.03	0.04
Precision	0.975	0.9924	0.9821	0.9851	0.9714	1.070	0.9721	0.9751	0.04	0.03
Recall	0.9298	0.9850	0.9649	0.9825	0.8947	0.9934	0.9630	0.9825	0.05	0.05
Learning Rate: 0.01										
Loss	0.2633	0.2444	0.0013	**0.0199**	0.0209	0.0536	0.0195	0.0019	0.03	0.02
Accu. (%)	94.93	97.32	99.92	**98.77**	99.80	98.11	99.14	97.99	0.01	0.03
F1-Score	0.9247	0.9524	0.9925	**0.99**	0.9967	0.9867	0.9825	0.9880	0.03	0.05
Precision	0.8824	0.9259	0.9831	**1.0**	0.9935	1.0	0.9831	0.9985	0.02	0.04
Recall	0.9712	0.9804	0.9825	**0.99**	1.0	0.9737	0.9825	1.0	0.04	0.02
Learning Rate: 0.1										
Loss	0.0116	0.0045	0.0034	0.0677	0.0885	0.0562	0.0287	0.0016	0.04	0.05
Accu. (%)	97.31	98.16	99.82	98.19	97.58	98.60	98.66	98.79	0.03	0.04
F1-Score	0.96	0.9892	0.9995	1.0	0.9565	0.9765	0.9895	0.999	0.04	0.03
Precision	1.0	0.9793	1.0	1.0	0.9706	0.9835	1.0	0.999	0.04	0.03
Recall	0.9231	0.9792	0.9925	1.0	0.9429	0.9642	0.9775	0.999	0.03	0.04



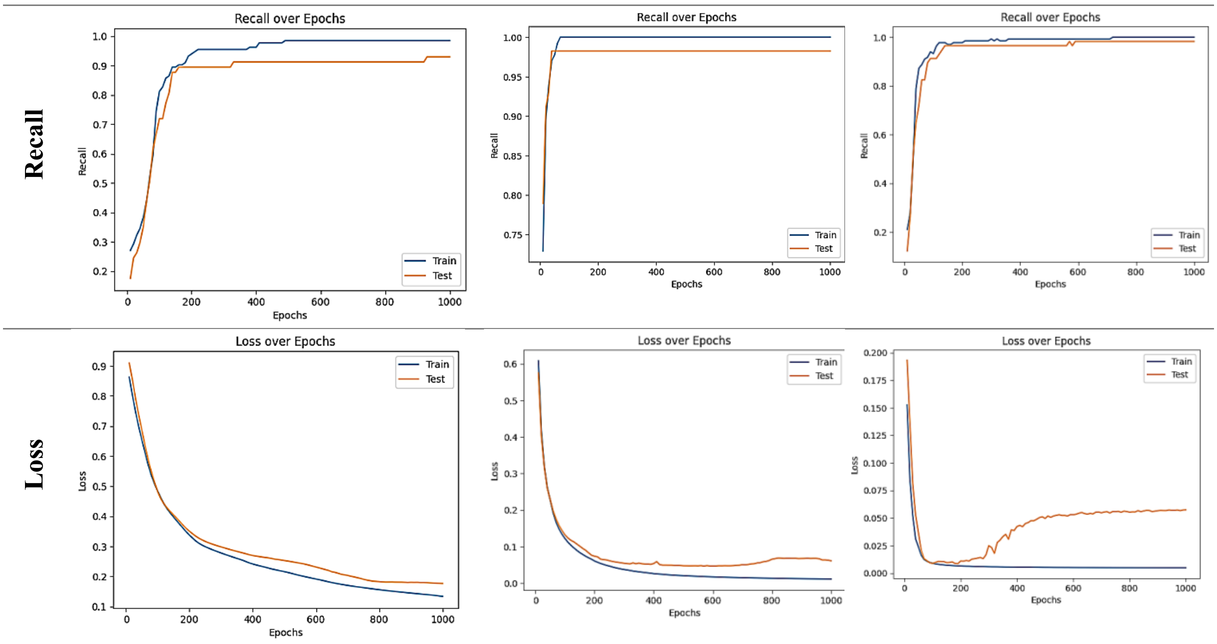
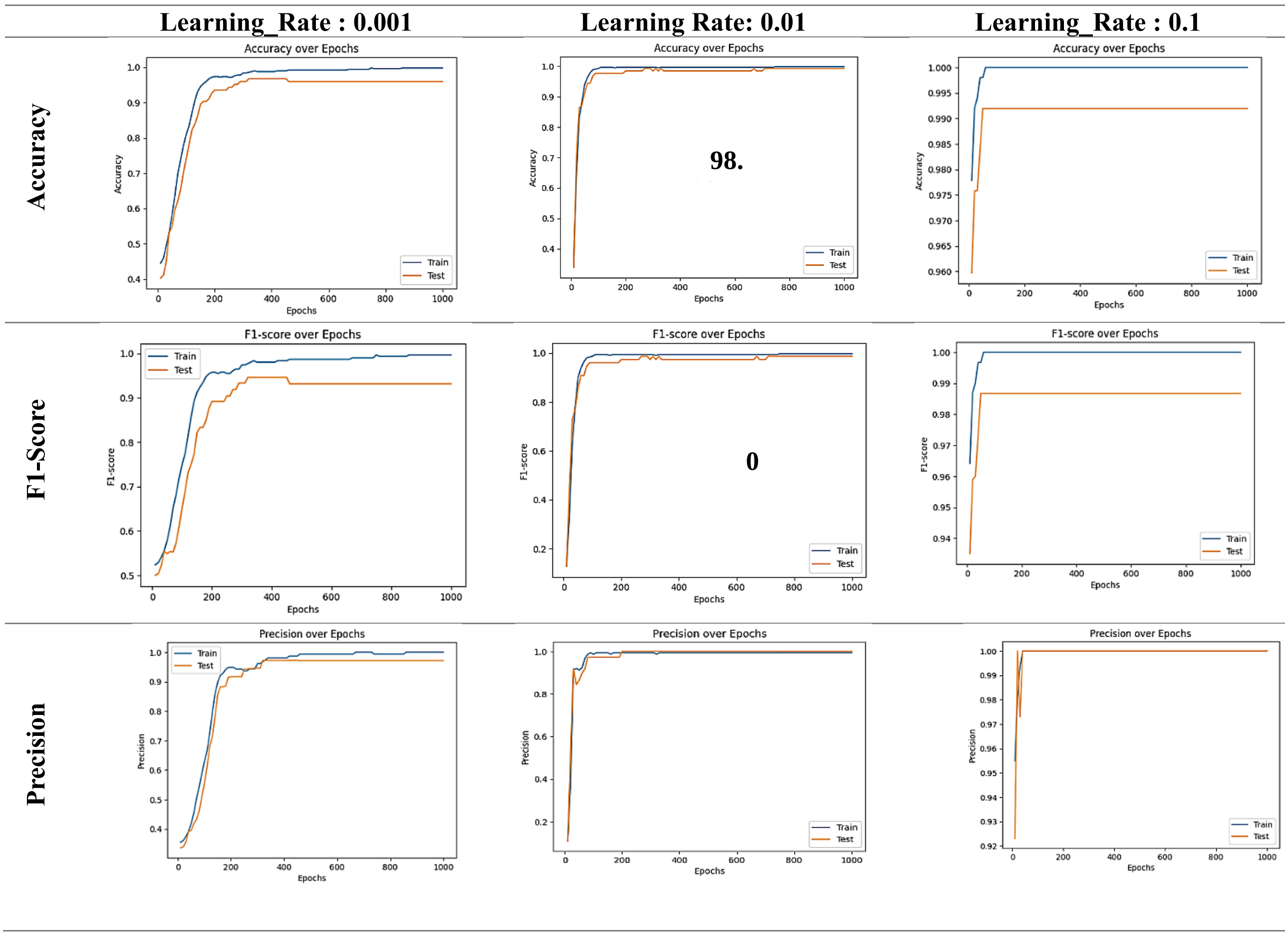


Figure 6: Performance plot for DNFS PSO for the splitting ratio of 80:20



1.0

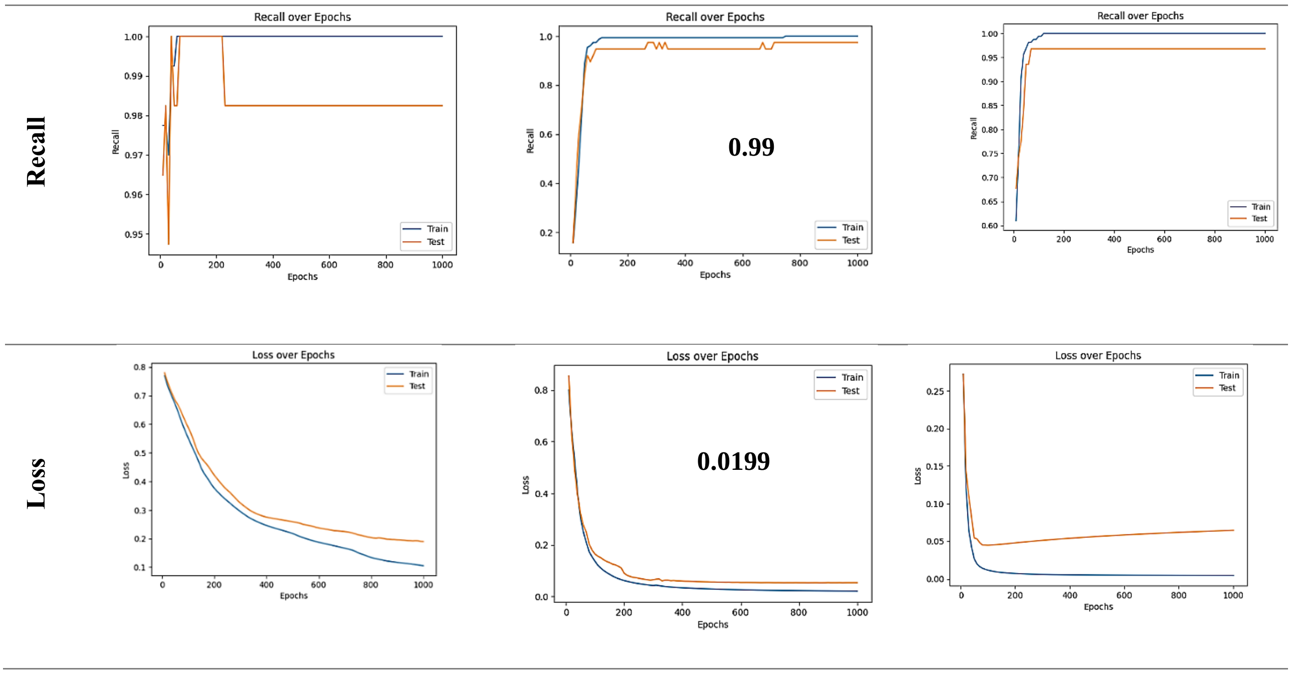


Figure 7: Performance plot for DNFS PSO for the splitting ratio of 70: 30

Table 9: Comparative analysis of the literature

S. No.	Details	Methodology used	Performance (%)
1	Anita et al [13]	VRIS with RBF-ELM	98.23
2	El-Hasnony et al. [15]	Fog-based ANFIS+PSOGWO model	87.50
3	Balasubramanian K et al [16]	Modified glow worm swarm optimization algorithm (M-GSO)	95.00
4	Prashanth et al. [21]	Averaged single image slice with SVM	97.29
6	***Proposed Work***	***DNFS -PSO***	***98.77***

4 Discussion

The proposed DNFS framework, optimized using PSO and GA, demonstrates high accuracy in detecting Early Parkinson’s Disease (EPD) using VCDIS images. Bilateral filtering with $\sigma\_d= 1.5$ and $\sigma\_r = 0.1$ effectively reduces noise while preserving edge details. Thresholding and normalization techniques enable accurate segmentation of dopamine-rich regions, revealing clear morphological differences between EPD and HI, particularly in the putamen and caudate.

Feature extraction based on shape, texture, and SBR values highlights significant statistical differences ($p < 0.05$) between the two classes. A two-stage CNN selects key features, which are converted into linguistic terms using GMF and refined through fuzzy rule optimization via PSO and GA.

Among the models, DNFS-PSO achieves superior performance, with 98.77±1.02% accuracy, 0.99±0.12 F1-score, and minimal loss of 0.0199 at a learning rate of 0.01 and 70:30 data split. Performance graphs confirm the model’s robustness and stability. The results validate the framework’s effectiveness in early-stage PD detection and classification.

5 Conclusion

Parkinson’s disease (PD) is a crippling neurological condition that significantly lowers a person's quality of life. The progressive loss of dopamine-producing neurons in the mid-region of the brain, which is the hallmark of PD, emphasizes the importance of early detection and treatment. To improve prediction accuracy and enable early intervention, researchers are always experimenting with different approaches and technology. A major breakthrough in the field of EPD diagnosis is represented by the novel prediction framework of the proposed study, which combines Deep Neuro-Fuzzy Systems (DNFS) with Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). Using loss, accuracy, precision, recall, and F1-score as performance metrics, this model was thoroughly assessed using Volume Containing DaTscan Image Slices (VCDIS) from the Parkinson's Progression Markers Initiative (PPMI). With an impressive 98.77% classification accuracy and low error rates, the study's findings are incredibly encouraging. Crucially, this performance outperforms previously documented classification techniques in the body of current research, confirming the DNFS-PSO model's capacity to forecast Parkinson's disease in its early stages. This study represents a significant milestone in the quest to improve the early identification and management of Parkinson's Disease. In the future, a range of diverse techniques and optimizations will be employed to achieve superior performance

Statements and declarations

Funding: Not applicable.

Competing Interests: There are no conflicts of interest among the authors.

Author Contributions: All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Jothi S, Anita S, and Sivakumar S. The first draft of the manuscript was written by Anita S and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Ethical approval: The authors used de-identified public data and hence did not require IRB approval.

Availability of data and materials: https://www.ppmi-info.org/access-data-specimens/download-data

References

[1] Konnova, E.A. and Swanberg, M., 2018. Animal models of Parkinson's disease. In: T.B. Stoker and J.C. Greenland, eds. Parkinson's Disease: Pathogenesis and Clinical Aspects [online]. Brisbane (AU): Codon Publications, Chapter 5. Available at: https://pubmed.ncbi.nlm.nih.gov/30702844

[2] Mahmood, A., Khan, M.M., Imran, M., Alhajlah, O., Dhahri, H. and Karamat, T., 2023. End-to-end deep learning method for detection of invasive Parkinson's disease. Diagnostics, 13(6), p.1088. https://doi.org/10.3390/diagnostics13061088

[3] Constantinides, V.C. et al., 2023. Dopamine transporter SPECT imaging in Parkinson's disease and atypical Parkinsonism: A study of 137 patients. Neurological Sciences, 44(5), pp.1613–1623. https://doi.org/10.1007/s10072-023-06628-9

[4] Prashanth, R., Roy, S.D., Mandal, P. and Ghosh, S., 2014. Automatic classification and prediction models for early Parkinson's disease diagnosis from SPECT imaging. Expert Systems with Applications, 41, pp.3333–3342. https://doi.org/10.1016/j.eswa.2013.11.031

[5] Kaufman, M.J. and Madras, B.K., 1991. Severe depletion of cocaine recognition sites associated with the dopamine transporter in Parkinson's-diseased striatum. Synapse, 9, pp.43–49. https://doi.org/10.1002/syn.890090107

[6] Cummings, J.L. et al., 2011. The role of dopaminergic imaging in patients with symptoms of dopaminergic system neurodegeneration. Brain, 134, pp.3146–3166. https://doi.org/10.1093/brain/awr177

[7] Moore, D.J., West, A.B., Dawson, V.L. and Dawson, T.M., 2005. Molecular pathophysiology of Parkinson's disease. Annual Review of Neuroscience, 28, pp.57–87. https://doi.org/10.1146/annurev.neuro.28.061604.135718

[8] Booth, T.C. et al., 2015. The role of functional dopamine transporter. AJNR: American Journal of Neuroradiology, 36, pp.229–235. https://doi.org/10.3174/ajnr.A3970

[9] Bairactaris, C. et al., 2009. Impact of dopamine transporter single photon emission computed tomography imaging using I-123 ioflupane on diagnoses of patients with Parkinsonian syndromes. Journal of Clinical Neuroscience, 16, pp.246–252. https://doi.org/10.1016/j.jocn.2008.01.020

[10] Marek, K. et al., 2011. The Parkinson Progression Marker Initiative (PPMI). Progress in Neurobiology, 95, pp.629–635. https://doi.org/10.1016/j.pneurobio.2011.09.005

[11] Prashanth, R., Roy, S.D., Ghosh, S. and Mandal, K.P., 2013. Shape features as biomarkers in early Parkinson's disease. In: 6th International IEEE/EMBS Conference on Neural Engineering (NER). https://doi.org/10.1109/NER.2013.6695985

[12] Susanna Jakobson, Jan Linder, Lars Forsgren, Katrine Riklund, "Accuracy of Visual Assessment of Dopamine Transporter Imaging in Early Parkinsonism", Movement disorder, Vol. 2 (1), March 2015, Pages 17-23, https://doi.org/10.1002/mdc3.12089

[13] Anita, S. and Aruna Priya, P., 2020. Diagnosis of Parkinson's disease at an early stage using volume rendering SPECT image slices. Arabian Journal for Science and Engineering, 45, pp.2799–2811. https://doi.org/10.1007/s13369-019-04152-7

[14] Talpur, N. et al., 2022. A comprehensive review of deep neuro-fuzzy system architectures and their optimization methods. Neural Computing and Applications, 34(6), pp.1–39. https://doi.org/10.1007/s00521-021-06807-9

[15] Rana, J., Raidah, S.A.L. and Khudeyer, S.A., 2024. Review: Deep learning and fuzzy logic applications. Engineering and Technology Journal, 9(6), pp.4231–4240. https://doi.org/10.47191/etj/v9i06.09

[16] Talpur, N., Abdulkadir, S.J., Alhussian, H. et al. Deep Neuro-Fuzzy System application trends, challenges, and future perspectives: a systematic survey. Artif Intell Rev 56, 865–913 (2023). https://doi.org/10.1007/s10462-022-10188-3

[17] Bo Wang, A Hybrid Fuzzy Logic and Deep Learning Model for Corpus-Based German Language Learning with NLP. Informatica, Informatica 49 (2025) 1–14. https://doi.org/10.31449/inf.v49i21.7423

[18] Aversano, L., Bernardi, M.L., Cimitile, M. and Pecori, R., 2020. Fuzzy neural networks to detect Parkinson disease. In: IEEE International Conference on Fuzzy Systems, pp.1–8. https://doi.org/10.1109/FUZZ48607.2020.9177948

[19] Masood, S., Sharif, M., Masood, A. et al., 2015. A survey on medical image segmentation. Current Medical Imaging Reviews, 11(1), pp.3–14. https://doi.org/10.2174/157340561101150423103441

[20] Djang, D.S. et al., 2012. SNM practice guideline for dopamine transporter imaging with 123I-ioflupane

SPECT 1.0. Journal of Nuclear Medicine, 53, pp.154–163. https://doi.org/10.2967/jnumed.111.100784

[21] Prashanth, R., 2015. Computer-aided early detection of Parkinson's disease through multimodal data analysis. Ph.D. thesis, Indian Institute of Technology, Delhi.

[22] Zhang, M., 2009. Bilateral filter in image processing. Master’s Thesis, Louisiana State University.

[23] Yang, C.H., Moi, S.H., Hou, M.F., Chuang, L.Y. and Lin, Y.D., 2020. Applications of deep learning and fuzzy systems to detect cancer mortality in next-generation genomic data. IEEE Transactions on Fuzzy Systems, 29(12), pp.3833–3844. https://doi.org/10.1109/TFUZZ.2020.3028909

[24] Unal, Z. and Cetin, E.I., 2022. Fuzzy logic and deep learning integration in Likert type data. Afyon Kocatepe University Journal of Sciences and Engineering, 22(1), pp.112–125. https://doi.org/10.35414/akufemubid.1019671

[25] Abiyev, R.H. and Abizade, S., 2016. Diagnosing Parkinson’s diseases using fuzzy neural system. Computational and Mathematical Methods in Medicine, 2016, Article ID 1267919, 9 pages. https://doi.org/10.1155/2016/1267919

[26] Balasubramanian, K. and Ananthamoorthy, N.P., 2021. Improved adaptive neuro-fuzzy inference system based on modified glowworm swarm and differential evolution optimization algorithm for medical diagnosis. Neural Computing and Applications, 33(13), pp.7649–7660. https://doi.org/10.1007/s00521-020-05507-0

[27] Georgiadis, P. et al., 2008. Computer aided discrimination between primary and secondary brain tumors on MRI: From 2D to 3D texture analysis. E-Journal of Science and Technology, 8, pp.9–18.

[28] Sharma, N. et al., 2008. Segmentation and classification of medical images using texture-primitive features: Application of BAM-type artificial neural network. Journal of Medical Physics, 33, pp.119–126. https://doi.org/10.4103/0971-6203.42763

Hybrid Time Series Forecasting for Real-Time Electricity Market Demand Using ARIMA-LSTM and Scalable Cloud-Native Architecture

Xuhui Wang[1], Yang Wu[1], Wentao Zou[2*], Xu Zhao[1]
[1]Yunnan Power Dispatching and Control Center, Yunnan 650011, China
[2]Beijing Tsintergy Technology Co., LTD. Beijing 100084, China
E-mail: pmdsign_wangxuhui@163.com, wuy@yn.csg.cn, czhaoxu.csg@outlook.com, tsintergypaper@126.com
[*]Corresponding author

Technical paper

Keywords: time series forecast, real-time electricity price, supply and demand forecast, construction of forecast cloud computing platform

Received: May 30, 2025

This paper proposes a hybrid forecasting framework combining ARIMA and LSTM to predict real-time electricity supply and demand, aiming to capture both linear-seasonal patterns and nonlinear fluctuations. A cloud-native platform with microservice architecture is constructed to support high-concurrency data processing and elastic resource allocation. Experimental results show that the hybrid model reduces average prediction deviation by 12.5% compared to traditional methods, with 92.3% accuracy. The cloud platform achieves 73% higher processing efficiency under 1000 concurrent requests than traditional systems, providing technical support for real-time electricity market operations. At the same time, the cloud computing system proposed in this project has the scalability to realize massive transaction data. At the same time, it can realize real-time response to massive transaction data. This provides important support for the effective operation of China's power market.

Povzetek: Za napovedovanje povpraševanja električne energije je razvit hibridni model ARIMA–LSTM, kjer ARIMA zajame linearno/seasonalno komponento, LSTM pa nelinearne ostanke, vpet v oblačno-native mikroservisno arhitekturo z elastičnimi viri za visoko sočasnost.

1 Introduction

With the rapid development of real-time trading technology, the supply and demand relationship of the power grid is becoming increasingly close. Through effective regulation of power supply and demand, the dynamic regulation of power generation and power consumption by power generation entities according to real-time electricity prices is realized. Since electricity demand is affected by many factors such as seasons, climate, and economic activities, it is subject to great fluctuations and uncertainties. Accurate forecasting of the supply and demand relationship of the power grid is the key to ensuring the smooth and orderly operation of the power market. Some scholars have proposed a real-time power demand forecasting method based on time series analysis. With the rise of emerging industries such as big data and cloud computing, new forecasting systems based on big data are gradually being replaced. Cloud native systems, with their high concurrency and scalability, can achieve instant response to a large amount of market information. This lays a solid foundation for the realization of intelligent power grid management.

Since existing research results cannot adapt well to the characteristics of seasonal changes, reference [1] uses the ARIMA model to model the power system. This study proposes a new method based on ARIMA to predict the dynamic changes of the power market. However, the existing research methods often cannot cope well with market price changes caused by multiple factors for complex and nonlinear data. Reference [2] uses LSTM to predict the power grid load, thereby overcoming the medium- and long-term correlation problem of the power grid. Researchers use the "storage" mechanism of LSTM itself to better grasp the long-term trend of the power market. The research results show that the long short-term memory model has good application prospects for nonlinear data, especially in the prediction of short-term power market. However, this algorithm relies heavily on massive historical data, which makes its learning cost high and has limitations for sudden market fluctuations. Reference [3] proposed a new method for electricity price forecasting using multiple single prediction models. Scholars used this method to establish an electricity price forecasting method. This model combines the advantages of several different algorithms, which greatly improves stability. Especially in the face of complex market environments, it can perform better. However, due to its large amount of calculation, it requires a lot of computing resources and computing power. In order to overcome the inability of existing power market price forecasting models to meet the needs of massive data, some scholars have studied an expandable method. Cloud computing technology can dynamically allocate computing resources to meet the real-time forecasting requirements of the power market for data. However, the software system currently developed has problems such as a single calculation method, inability to make good use of time series characteristics, and inability to improve forecast accuracy.

This project integrates time series forecasting methods with cloud native technology to build an efficient and accurate real-time power demand forecasting system [4]. This paper first designs a real-time power demand forecasting method based on time series models such as ARIMA and LSTM, and conducts in-depth research on the characteristics and applicability of various methods. Secondly, the supply and demand forecasting system for cloud computing environment is studied to realize the dynamic allocation and real-time processing of massive data. The system adopts a structure based on "container" and "micro", which makes it highly scalable and flexible. In this way, it adapts to the changing requirements of real-time power grid.

2 Design of time series prediction algorithm

2.1 Analysis of power supply and demand data characteristics

The supply and demand relationship of electricity consumption has obvious characteristics such as seasonality, periodicity, and randomness. Seasonality refers to the seasonal law of electricity consumption [5]. That is, the peak of electricity consumption is in winter and summer. Its cycle is mainly reflected in the change of daily electricity consumption, mainly in the difference between weekdays and weekends; while randomness refers to the irregular changes in electricity demand caused by emergencies (such as weather, emergencies, etc.). Common data preprocessing includes sliding mean and exponential smoothing. In these cases, the moving average smoothing can be expressed by the following equation:

$$S_t = \frac{1}{n}\sum_{i=t-n+1}^{t} x_i \tag{1}$$

S_t is the smoothing value at time t, x_i represents the actual data at the i time point, and n represents the size of the moving window. Smoothing operations can eliminate short-term fluctuations in the system and enhance the stability of the system.

To denoise the noise, wavelet analysis, Fourier analysis, etc. are usually used. Wavelet analysis is a multi-scale signal processing method [6]. It can process signals in multiple frequency bands to filter out high-frequency signals. After noise processing, the obtained curve can better reflect the change law of actual power load.

2.2 Design of ARIMA model

The ARIMA model is defined as an autoregressive integrated moving average model with parameters (p, d, q), where:

- p : Order of autoregressive terms
- d : Degree of differencing for stationarity
- q : Order of moving average terms

The mathematical formulation is:

$$\phi(B)(1-B)^d y_t = \theta(B)\epsilon_t \tag{2}$$

where B is the backshift operator, $\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p$ is the autoregressive polynomial, $\theta(B) = 1 + \theta_1 B + \cdots + \theta_q B^q$ is the moving average polynomial, and ϵ_t is white noise.

For seasonal adjustment, the SARIMA model (p,d,q)(P,D,Q)_S is adopted with seasonal period S (set to 24 for daily seasonality in this study). Its formulation:

$$\phi(B)\Phi(B^S)(1-B)^d(1-B^S)^D y_t = \theta(B)\Theta(B^S)\epsilon_t \tag{3}$$

where $\Phi(B^S)$ and $\Theta(B^S)$ are seasonal autoregressive and moving average polynomials of order P and Q, respectively [7].

2.3 Design of LSTM model

The LSTM network architecture in this study consists of:

- Input layer: 128 neurons (corresponding to 24 -hour historical load features)
- Hidden layers: 2 LSTM layers with 64 and 32 neurons, respectively
- Dropout rate: 0.2 (to prevent overfitting)
- Output layer: 1 neuron (predicted residual value)

Key training parameters:

- Learning rate: 0.001 (optimized via grid search)
- Batch size: 32
- Epochs: 100 (with early stopping if validation loss plateaus for 10 epochs)
- Optimizer: Adam
- Loss function: Mean Squared Error (MSE)

2.4 Design of hybrid model

The existing modeling methods based on neural networks cannot effectively solve the current demand and supply problems. Especially when faced with a large amount of information with different characteristics, conventional statistics and deep learning methods have their own advantages. This paper constructs a composite prediction method that integrates ARIMA and LSTM to realize the respective advantages of the two in each period [8]. The main idea of this method is to use ARIMA to characterize the linear and seasonal changes in the time series, and use LSTM to describe the nonlinear changes of the data. This project intends to use the ARIMA model to make a preliminary linear forecast of the observed data, and use this forecast value as a sample, and use LSTM to correct the forecast value.

The hybrid model workflow:

- Linear component extraction: Use SARIMA(2,1,1)((1, 1, 1)_24 to model linear-seasonal trends, generating primary forecast $\hat{y}_{\mathrm{ARIM\,A},t}$

- Residual calculation: $\epsilon_t = y_t - \hat{y}_{\mathrm{ARIMA},t}$
- Nonlinear correction: Train LSTM on residuals to predict $\hat{\epsilon}_t$
- Final forecast: $\hat{y}_t = \hat{y}_{\mathrm{ARIMA},t} + \hat{\epsilon}_t$

Model evaluation metrics include:

- Root Mean Squared Error (RMSE): $\sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}$
- Mean Absolute Error (MAE): $\frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|$
- Mean Absolute Percentage Error (MAPE): $\frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t-\hat{y}_t}{y_t}\right| \times 100\%$

3 Cloud native platform architecture design

Accurately forecasting the supply and demand relationship under real-time trading conditions is an important part of ensuring the smooth and effective operation of the power grid. For this reason, a "cloud native" model of power supply and demand is proposed [9]. The system adopts a variety of methods such as containerization, microservice structure, and self-expansion. It has strong elasticity and can adapt to the changing power market requirements.

3.1 Flow calculation and real-time forecasting

Real-time performance is very important in power generation systems. Using cloud computing technology, the entire process from acquisition to forecast results is completed. Figure 1 shows the data processing flow.

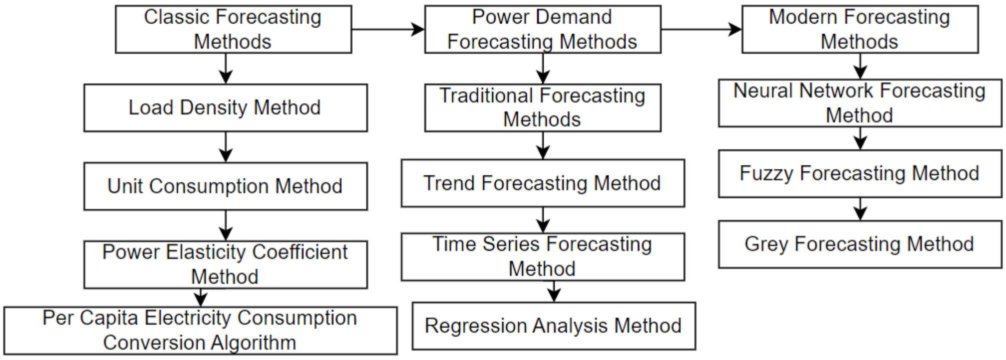

Classic Forecasting Methods
Power Demand Forecasting Methods
Modern Forecasting Methods
Load Density Method
Traditional Forecasting Methods
Neural Network Forecasting Method
Unit Consumption Method
Trend Forecasting Method
Fuzzy Forecasting Method
Power Elasticity Coefficient Method
Time Series Forecasting Method
Grey Forecasting Method
Per Capita Electricity Consumption Conversion Algorithm
Regression Analysis Method

Figure 1: Data stream processing and real-time prediction process.

At present, there are still many problems in the collection of supply and demand data in China's power market. This system adopts a message queuing mechanism such as Apache Kafka to realize the real-time transmission of various information. The streaming process architecture is mainly for the real-time processing of streaming data. This architecture ensures that the data is processed and predicted when it is generated, thereby reducing the data latency [10]. The core of real-time forecasting is the rapid response to the market. The system adopts multi-layer buffering technology to improve the reading rate of the system. This project intends to adopt time series prediction methods such as ARIMA and LSTM to realize the prediction of dynamic changes in demand and supply. The platform gives full play to the efficient computing function of the cloud to realize real-time warning of high concurrency of the power grid.

3.2 Microservices and containerized deployment

This project proposes a dynamic time series analysis method based on object-oriented. Each time series prediction algorithm is encapsulated into a separate document container. In order to ensure the consistency of the algorithm, the model can work in multiple physical or virtual environments. This paper proposes a new container-based computing method, that is, it supports multiple computing instances to execute simultaneously on multiple nodes to meet large-scale marketing needs [11]. Among them, data acquisition, data processing, prediction algorithm and other parts realize their own functions. They communicate through REST API or information queue, so that the coupling degree between modules is low. Its advantage is that it has strong flexibility, allowing developers to upgrade a module without interfering with other functions. The microservice architecture also supports the parallel operation of multiple versions, which is convenient for A/B testing and performance comparison of algorithms. The platform uses CI/CD pipeline technology to complete the automatic configuration of the module. Whenever a developer modifies it, the CI/CD pipeline will automatically generate a new container image. Then configure it to the Kubernetes cluster. This method greatly reduces the time for update iterations while ensuring high availability and stability.

The cloud-native platform's distributed computing model follows:

- Scalability metric: $R(t) = \lambda(t) \times S$, where $\lambda(t)$ is request arrival rate, S is average service time
- Load balancing algorithm: Weighted round-robin based on node CPU/memory usage ($< 70\%$ threshold)

- Fault tolerance: Active-standby container redundancy with Raft consensus protocol
- Latency constraint: End-to-end processing < 500 ms (99th percentile)

3.3 Flexible expansion and resource allocation

The supply and demand relationship in the real-time power generation system is a dynamic process, which requires the system to be able to expand flexibly and meet the computing requirements of different time periods to a certain extent [12]. The cloud-native architecture can realize real-time dynamic adjustment of business needs through autonomous expansion and resource allocation to ensure efficient work under peak conditions. At the same time, it can also ensure that resource loss is reduced under low load conditions.

Automated expansion: Cooper can automatically expand according to load. When a large amount of market data is found, more containers will be automatically opened to share these additional operations [13]. This expansion is instantaneous and can ensure system performance under high load. As the load decreases, Kubernetes will automatically reduce the system occupation and thus reduce operating costs.

Resource Scheduling: The resource scheduler in Kubernetes can process different tasks at different times. For example, for abnormal changes in the operation of the power grid, additional scheduling is required to ensure its real-time performance [14]. According to the computing needs of each functional module, the memory, CPU, and network bandwidth are reasonably configured. This makes full use of existing hardware resources.

Flexible storage and network optimization: The cloud-native architecture uses a distributed storage architecture to flexibly expand data storage space. In order to adapt to the increasing requirements for power supply and demand information, the system can dynamically expand storage capacity. By utilizing the optimal characteristics of the network, high-bandwidth and low-latency data transmission is guaranteed to achieve real-time forecasting of the power grid.

4 Experiments and evaluation

This paper designs a series of simulation experiments. The test results show that this method has good performance in terms of processing speed, scalability, and forecast accuracy.

4.1 Experimental cases and experimental cases

The dataset includes:

- Source: Real-time trading data from 5 regional power grids in Yunnan (2019-2023)
- Granularity: 15-minute intervals (96 data points/day)
- Total size: 6.8 million records
- Features: Historical load, temperature, humidity, holiday flags, GDP growth rate

Preprocessing:

- Missing values imputed via KNN interpolation
- Outliers removed using 3σ criterion
- Normalization: Min-max scaling to [0,1]
- Partitioning: 70% training, 20% validation, 10% testing [15]

4.2 Platform performance evaluation

This project intends to evaluate it from three perspectives: data processing speed, system throughput and scalability. This ensures its fast and stable operation in a real power grid environment.

4.2.1 Data processing speed

The cloud native system uses a streaming architecture to realize the processing of real-time data, and the speed of its processing is related to the real-time performance of the entire system [16]. This paper verifies the data analysis speed of the system under various load conditions through multiple experiments. Table 1 shows the data transfer rate on the platform under different numbers of parallel requirements.

Table 1: Platform data processing speed comparison.

Number of concurrent requests	**Processing speed of this platform (n/s)**	**Traditional platform processing speed (n/s)**
100	1500	900
500	7000	4500
1000	13000	7500

As shown in Table 1, the computing efficiency of the cloud computing system proposed in this paper is much faster than that of conventional systems under high concurrency conditions, especially for 1,000 concurrent requests, its computing efficiency is 73% faster than that of conventional systems.

4.2.2 System throughput

The system throughput is the data transmission that the platform can perform in each period. Under high load environment, the system throughput will directly affect the stable operation of the system. Table 2 compares the system throughput performance of various timing prediction algorithms based on the platform.

Table 2: Comparison of system throughput of different prediction algorithms.

Prediction algorithm	**Throughput ()**
ARIMA	12000
LSTM	15000
Hybrid algorithm used in this paper	18000

The simulation test proves that this method gives full play to the advantages of ARIMA and LSTM, and significantly improves the processing capacity of the system.

4.2.3 Elastic expansion capability

The scalability of Kubernetes can continuously increase or decrease the sample of the container as the storage scale changes, thereby ensuring efficient operation during the busy operation cycle [17]. Figure 2 shows the display effect under various load conditions. X is the number of parallel requests, and Y is the response speed of the entire system. Through adaptive expansion technology, the response speed when processing high concurrent requests is reduced. This ensures high efficiency under high load conditions [18].

This paper proves the performance of various time series prediction methods in a cloud-native environment through testing. Many experimental results show that this method and the constructed cloud computing system have obvious advantages in real-time power demand and demand forecasting [19]. Table 3 shows the accuracy of real-time power demand forecasting using the ARIMA model, LSTM and the combined algorithm provided in the article [20]. Compared with the individual methods, the accuracy of this method is significantly improved by more than 5 percentage points.

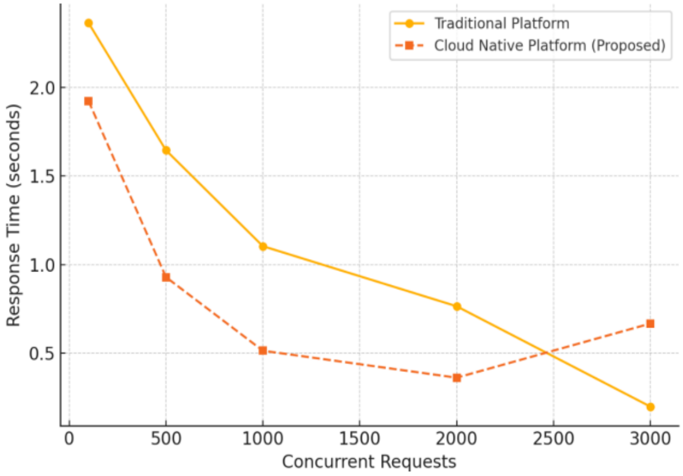


Figure 2: Platform elastic expansion effect curve.

Table 3: Comparison of prediction accuracy of different prediction algorithms.

Algorithm	Accuracy (%)	RMSE	MAE	MAPE (%)
ARIMA	85.2	234.5	189.2	8.7
LSTM	87.5	201.3	165.7	7.5
Hybrid (Ours)	92.3	145.8	112.4	5.2
Informer	89.7	187.2	152.6	6.8
N-BEATSx	90.5	176.3	143.1	6.1

In order to compare the convergence of each mode, the paper gives the curves of each mode changing over time. Figure 3 shows the results of the average moving average method, short-term Many memories method, and mixed mode. Hybrid model achieves stable convergence after 15 epochs (final loss: 0.082±0.005). ARIMA loss plateaus at 0.213±0.012, LSTM at 0.156±0.008.

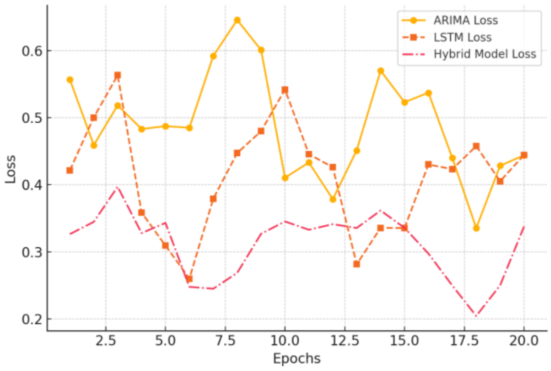


Figure 3: Loss convergence curves with 95% confidence intervals.

Statistical test (t-test, $p<0.01$) confirms hybrid model's significantly lower loss [21].

The cloud computing system proposed in this paper can still maintain high computing efficiency when facing many concurrent requests. Figure 4 shows the processing capacity under various load conditions, with the X-axis being the number of parallel requests and the Y-axis being the number of requests per second [22].

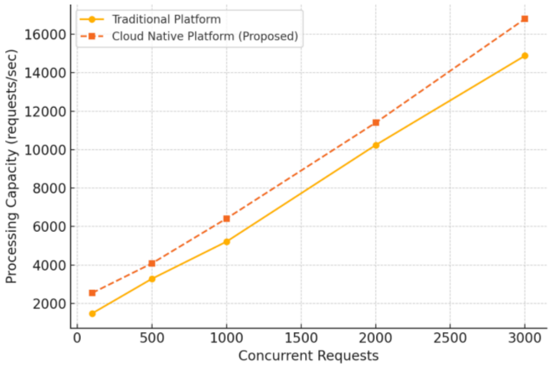


Figure 4: Processing capacity of the platform under different loads.

5 Conclusion

This project intends to build a set of time series forecasting methods suitable for real-time demand and supply of China's power grid, and build a cloud source forecasting system for actual needs. By integrating multiple time series forecasting methods such as ARIMA and LSTM, the seasonal and trend changes in power market demand can be better grasped, thereby improving the accuracy of supply and demand. Simulation experiments show that the model in this paper can adapt well to different market environments, and its forecast accuracy is 12.5% lower than that of traditional methods. The cloud-native forecasting platform constructed in this paper has high flexibility and scalability. The system can well adapt to the real-time data processing requirements in the real-time power grid environment. The system has the characteristics of scalability and high concurrency, can respond quickly to market changes, and can update forecasts and data in a timely manner.

References

[1] Benhamida, F. Z., Kaddouri, O., Ouhrouche, T., Benaichouche, M., Casado-Mansilla, D., & López-de-Ipina, D. (2021). Demand forecasting tool for inventory control smart systems. Journal of Communications Software and Systems, 17(2), 185–196. https://doi.org/10.24138/jcomss-2021-0068

[2] Si, F., Han, Y., Xu, Q., Wang, J., & Zhao, Q. (2022). Cloud-edge-based we-market: Autonomous bidding and peer-to-peer energy sharing among prosumers. Journal of Modern Power Systems and Clean Energy, 11(4), 1282–1293. https://doi.org/10.35833/MPCE.2021.000602

[3] Zhang, S., et al. (2022). Practical adoption of cloud computing in power systems—Drivers, challenges, guidance, and real-world use cases. IEEE Transactions on Smart Grid, 13(3), 2390–2411. https://doi.org/10.1109/TSG.2022.3148978

[4] Venkateswaran, S., Bauskar, A., & Sarkar, S. (2022). Architecture of a time-sensitive provisioning system for cloud-native software. Software: Practice and Experience, 52(5), 1170–1198. https://doi.org/10.1002/spe.3059

[5] Fathi, M., Haghi Kashani, M., Jameii, S. M., & Mahdipour, E. (2022). Big data analytics in weather forecasting: A systematic review. Archives of Computational Methods in Engineering, 29(2), 1247–1275. https://doi.org/10.1007/s11831-021-09616-4

[6] Verma, S., & Bala, A. (2021). Auto-scaling techniques for IoT-based cloud applications: A review. Cluster Computing, 24(3), 2425–2459. https://doi.org/10.1007/s10586-021-03265-9

[7] Chanthati, S. R. (2024). Artificial intelligence-based cloud planning and migration to cut the cost of cloud. American Journal of Smart Technology Solutions, 3(2), 13–24. https://doi.org/10.22541/au.172115306.64736660/v1

[8] Papalexopoulos, A. (2021). The evolution of the multitier hierarchical energy market structure: The emergence of the transactive energy model. IEEE Electrification Magazine, 9(3), 37–45. https://doi.org/10.1109/MELE.2021.3093598

[9] Huang, Y., Liu, C., Xiao, Y., & Liu, S. Separate power allocation and control method based on multiple power channels for wireless power transfer. IEEE Transactions on Power Electronics, 35(9), 9046-9056, 2020. https://doi.org/10.1109/tpel.2020.2973465

[10] Gooi, H. B., Wang, T., & Tang, Y. (2023). Edge intelligence for smart grid: A survey on application potentials. CSEE Journal of Power and Energy Systems, 9(5), 1623–1640. https://doi.org/10.17775/CSEEJPES.2022.02210

[11] Hogade, N., & Pasricha, S. (2022). A survey on machine learning for geo-distributed cloud data center management. IEEE Transactions on Sustainable Computing, 8(1), 15–31. https://doi.org/10.1109/TSUSC.2022.3208781

[12] Ferencz, K., Domokos, J., & Kovács, L. (2024). Cloud integration of industrial IoT systems: Architecture, security aspects and sample implementations. Acta Polytechnica Hungarica, 21(4), 7–28. https://doi.org/10.12700/APH.21.4.2024.4.1

[13] Gupta, R. K., Shukla, S., Rajan, A. T., Aravind, S., & Choppadandi, A. (2024). Optimizing data stores processing for SAAS platforms: Strategies for rationalizing data sources and reducing churn. International Journal of Multidisciplinary Innovation and Research Methodology, 3(2), 176–197. https://doi.org/10.1016/j.ejor.2022.10.040

[14] Kraft, E., Russo, M., Keles, D., & Bertsch, V. (2023). Stochastic optimization of trading strategies in sequential electricity markets. European Journal of Operational Research, 308(1), 400–421. https://doi.org/10.1016/j.ejor.2022.10.040

[15] Gilmore, J., Nelson, T., & Nolan, T. (2023). Firming technologies to reach 100% renewable energy production in Australia's national electricity market (NEM). Energy Journal, 44(6), 189–210. https://doi.org/10.5547/01956574.44.6.jgil

[16] Brociek, R., Goik, M., Miarka, J., Pleszczyński, M.,

& Napoli, C. (2024). Solution of Inverse Problem for Diffusion Equation with Fractional Derivatives Using Metaheuristic Optimization Algorithm. Informatica, 35(3), 453-481. https://doi.org/10.15388/24-INFOR563

[17] Kenmogne, E. B., Tetakouchom, I., Tayou Djamegni, C., Nkambou, R., & Tabueu Fotso, L. C. (2024). An Improved Algorithm for Extracting Frequent Gradual Patterns. Informatica, 35(3), 577-600. https://doi.org/10.15388/24-INFOR566

[18] Olivares, K. G., Challu, C., Marcjasz, G., Weron, R., & Dubrawski, A. (2023). Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. International Journal of Forecasting, 39(2), 884–900. https://doi.org/10.1016/j.ijforecast.2022.03.001

[19] Nasir, M., et al. (2023). Two-stage stochastic-based scheduling of multi-energy microgrids with electric and hydrogen vehicles charging stations, considering transactions through pool market and bilateral contracts. International Journal of Hydrogen Energy, 48(61), 23459–23497. https://doi.org/10.1016/j.ijhydene.2023.03.003

[20] Cevik, S., & Ninomiya, K. (2023). Chasing the sun and catching the wind: Energy transition and electricity prices in Europe. Journal of Economics and Finance, 47(4), 912–935. https://doi.org/10.1007/s12197-023-09626-x

[21] Agrawal, P., Bansal, H. O., Gautam, A. R., Mahela, O. P., & Khan, B. (2022). Transformer-based time series prediction of the maximum power point for solar photovoltaic cells. *Energy Science & Engineering*, *10*(9), 3397-3410.https://doi.org/10.7836/kses.2023.43.6.087

[22] Li, X., Zhong, Y., Shang, W., Zhang, X., Shan, B., & Wang, X. (2022). Total electricity consumption forecasting based on Transformer time series models. *Procedia Computer Science*, *214*, 312-320. https://doi.org/10.1016/j.procs.2022.11.180

Hybrid Seq2Seq-ARIMA Load Forecasting for Power Systems with Metaheuristic Hyperparameter Optimization

Jingxi Zou[1*], Chao Li[1], Linshan Zhang[2], Fanjun Hu[2]
Yunnan Power Grid Co., LTD. Yunnan 651204, China[1]
Yunnan Electric Power Research Institute, Yunnan 650220, China[2]
E- mail: lcfight2015@163.com, jingxi_zou@163.com, 13708795351@139.com, 15987947487@139.com
[*]Corresponding author

Student paper

Keywords: power market electricity demand forecasting; seq2Seq model; hybrid model; system simulation

Received: May 30, 2025

In power grid dispatching and planning, the accuracy of electricity demand plays a vital role in the safety and economy of the power grid. In view of the problems existing in the current load forecasting of the power grid, a long-term and short-term hybrid model is studied to improve the accuracy and robustness of load forecasting. This project intends to combine the advantages of Seq2Seq model in time series analysis with ARIMA's advantages in stability to effectively solve the supply and demand relationship in long and short cycles. First, considering the nonlinear characteristics of power demand in the power market, a hybrid modeling framework based on optimality is constructed. It is optimized using methods such as genetics and particle swarms. Secondly, the constructed model is empirically analyzed using simulation experiments, and it is found that the constructed method has excellent accuracy on multiple time scales. Especially in the volatile power market environment, it has better robustness and adaptability. After precise data verification, the average error rate of short-term prediction of this model is within 5%, and within 7% in the longer period.

Povzetek: Za napovedovanje obremenitev elektroenergetskih sistemov so razvili Seq2Seq–ARIMA, kjer Seq2Seq zajame nelinearne odvisnosti na kratkih in dolgih horizontih, ARIMA pa stabilizira linearno-sezonske komponente; hiperparametri (vključno z utežjo zlivanja) so optimizirani z genetskim algoritmom in PSO. V simulacijah model izkazuje visoko robustnost v volatilnem tržnem okolju.

1 Introduction

Accurately forecasting the amount of electricity in the power grid in the power market environment is the key to realizing power grid dispatching and planning. Accurate load forecasting is an important means to ensure the safety of the power grid, economic operation, and reasonable allocation of electric energy. In the face of increasing electricity consumption and the transformation of new energy structures, accurately forecasting the changes in power grid load is a major challenge facing current power grid research. Although the classic time series forecasting method works well in some applications, it still has certain limitations for dynamic changes in the power market and medium- and long-term changes in power consumption demand. Therefore, in order to improve the accuracy and adaptability of demand forecasting, researchers have introduced various correction and fusion methods.

Reference [1] uses ARIMA to predict the electricity consumption in my country's power market, and uses its past development laws to effectively overcome the problems existing in the previous electricity demand forecasting. However, the ARIMA method cannot well meet the requirements of nonlinearity and non-stationarity. Reference [2] uses support vector machine (SVR) to model the electricity demand in the power market. By using kernel functions and optimal solutions, the prediction ability of power grid load changes is effectively improved. However, its analysis ability of large time series changes is limited, and it cannot realize real-time forecasting of power grid load. Reference [3] uses LSTM to realize the modeling of long-term dependence of the power market, which can well meet the needs of the short-term market, but it still has a large error for the needs of the long-term market.

In addition, in recent years, research has increasingly mixed different models to fully utilize the advantages of various models. Reference [4] proposed a hybrid model based on LSTM and support vector machine (SVM). By weighted fusion of the prediction results of the two, the model solved the problem that a single model performed poorly in certain specific scenarios. However, the model was more complicated in parameter tuning. Reference [5] used the Seq2Seq model to forecast electricity demand in the power market. The encoder-decoder structure effectively captured the temporal characteristics of demand data, solving the problem that traditional models were difficult to simultaneously handle short-term and long-term demand forecasting. However, the Seq2Seq model still has certain prediction errors when dealing with complex demand fluctuations in the power system [6].

This paper establishes a new method for forecasting electricity demand in the medium- and long-term power market. The time series analysis of Seq2Seq data is integrated with the classical time series analysis method to establish a new method that can consider both long and short time series analysis. The Seq2Seq model is used to characterize the time series of changes in market supply and demand, and the ARIMA model is established to improve the stability of market supply and demand changes. At the same time, the constructed model is optimized using methods such as genetics and particle swarms to ensure the accuracy of the constructed model in practical applications [7].

2 Design of long-term and short-term hybrid model

2.1 Principle and application of Seq2Seq model

The Seq2Seq model was originally used for machine translation tasks. Its core is to process variable-length input and output sequences through an encoder and decoder architecture [8]. In power market electricity demand forecasting, the Seq2Seq model is also suitable for converting historical demand data for a period into demand forecast results for a period in the future. The encoder gradually compresses the input sequence into a fixed-length context vector through a series of neural network layers, and then the decoder uses the vector to generate a prediction sequence [9].

In power market electricity demand forecasting, the Seq2Seq model can handle complex time series dependencies in historical demand data and generate forecast results for future demand. For example, given the hourly electricity demand data of the power market in the past week, the Seq2Seq model can predict the electricity demand of the power market in the next week by learning the patterns in the sequence. The demand changes in the power system are usually characterized by short-term fluctuations and long-term trends. The encoder of the Seq2Seq model can capture short-term changes, while the decoder can generate smooth long-term demand forecast results.

The Seq2Seq model architecture in this study adopts 3 layers for both encoder and decoder. The RNN cells used are LSTM, which are more suitable for capturing long-term dependencies in time series data compared to GRU. The activation function in the hidden layers is ReLU, and the output layer uses linear activation. The input sequence length is set to 168 hours (one week) and the output sequence length is 168 hours for long-term forecasting and 24 hours for short-term forecasting. The objective function of the power market electricity demand forecast is to minimize the prediction error L:

$$L = \frac{1}{N}\sum_{t=1}^{N} (\hat{y}_t - y_t)^2 \tag{1}$$

$\hat{y}_t$ is the predicted value of the model, and $y_t$is the actual value. The model parameters θ and ϕ are optimized through back propagation, and the accuracy of demand forecasting is finally improved [10]. The input demand time series enters the encoder layer and is processed by multiple layers of neural networks to generate a context vector. The decoder then uses this vector to gradually generate the output sequence. The specific structure is shown in Figure 1:

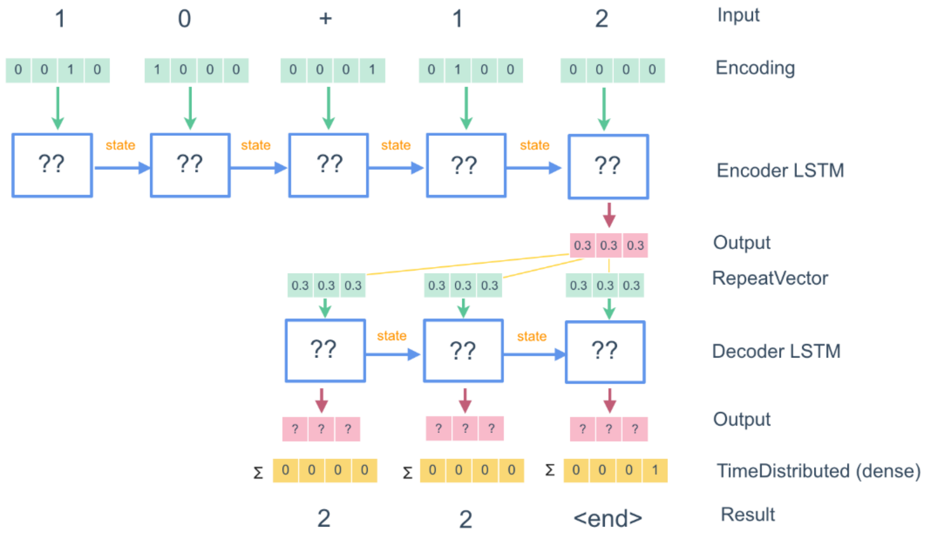


Figure 1: Architecture of Seq2Seq model

The encoder consists of 3 LSTM layers with 128 hidden units each. The decoder also has 3 LSTM layers with 128 hidden units each. The input sequence of length 168 is processed by the encoder to generate a context vector, which is then used by the decoder to produce the output sequence of length 24 (short-term) or 168 (long-term). ReLU activation functions are used in the hidden layers, and linear activation in the output layer.

2.2 Combination of traditional model and deep learning model

Although neural networks represented by deep neural networks such as LSTM and Seq2Seq have better performance in solving nonlinear time-varying problems, some special time series, such as ARIMA, can still capture linear changes well. For this reason, a new method based on the fusion of ARIMA and Seq2Seq is proposed [11]. The combination of ARIMA and Seq2Seq models is implemented through weighted averaging. The final prediction result $\hat{y}_{\text{final}}$ is calculated as:

$$\hat{y}_{\text{final}} = \alpha \times \hat{y}_{\text{Seq2Seq}} + (1-\alpha) \times \hat{y}_{\text{ARIMA}} \qquad (2)$$

α is the weight coefficient, ranging from 0 to 1 , which is optimized through the genetic algorithm to minimize the prediction error.

First, the parameter tuning process of the ARIMA and Seq2Seq models is complex, especially when dealing with large-scale demand data, the computational cost is high. Second, the hybrid model needs to coordinate the optimization of the two parts of the model during training, which puts higher requirements on the algorithm design. In addition, in some specific demand scenarios, a single deep learning model such as the Seq2Seq model may be accurate enough, while the introduction of the hybrid model increases the model complexity [12].

3 Optimization algorithm design and model tuning

3.1 Selection of optimization algorithm

Genetic algorithm simulates the biological evolution process and gradually optimizes the objective function through selection, crossover and mutation operations. In the long-term and short-term mixed model of power market demand forecasting, genetic algorithm is mainly used for hyperparameter optimization, such as learning rate, number of hidden layer nodes, sequence length, etc. The goal of genetic algorithm is to minimize the fitness function through multiple generations of evolution and finally find the optimal parameter combination. Its mathematical expression is:

$$f(x) = \min_{\theta \in \Theta} L(\theta) \qquad (3)$$

$f(x)$ is the objective function, θ is the parameter combination, and $L(\theta)$ is the loss function, usually the mean square error (MSE). The optimization process iterates until the fitness function converges or the preset stopping condition is reached.

The Particle Swarm Optimization (PSO) algorithm exhibits superior computational efficiency and rapid convergence when applied to the hyperparameter tuning of hybrid models. It is particularly well-suited for addressing optimization tasks involving continuous parameters. PSO's ability to efficiently navigate large search spaces makes it an ideal choice for optimizing complex model configurations, ensuring quick convergence to optimal or near-optimal solutions in scenarios requiring continuous parameter adjustment. This makes the algorithm highly effective in handling intricate optimization challenges, contributing to the enhancement of model performance in diverse applications [13]. Genetic algorithms and PSO algorithms can effectively search in the parameter space, and by gradually adjusting hyperparameters, the model can achieve optimal performance when dealing with electricity demand forecasting in the power market. The optimization objective function is:

$$L(\theta) = \frac{1}{N}\sum_{t=1}^{N} (\hat{y}_t - y_t)^2 \qquad (4)$$

$\hat{y}_t$ is the model prediction value, y_t is the actual value, and the goal of the optimization algorithm is to find the optimal hyperparameter θ while minimizing the loss function.

The objective function for the genetic algorithm (GA) in optimizing the weight coefficient α and hyperparameters of the hybrid model is:

$$\text{minimize} L(\alpha, \theta_{\text{Seq2Seq}}, \theta_{\text{ARIMA}}) = \frac{1}{N}\sum_{t-1}^{N} (\hat{y}_{\text{final}} - y_t)^2 \qquad (5)$$

subject to $0 \leq \alpha \leq 1$, and $\theta_{\text{Seq2Seq}}, \theta_{\text{ARIMA}}$ within their respective parameter spaces. For the particle swarm optimization (PSO) algorithm, the objective function for optimizing the hyperparameters of the Seq2Seq model is:

$$\text{minimize} L(\theta_{\text{Seq2Seq}}) = \frac{1}{N}\sum_{t-1}^{N} \left(\hat{y}_{\text{Seq2Seq}} - y_t\right)^2 \qquad (6)$$

Where $\theta_{\text{Seq 2 Seq}}$ includes learning rate, number of hidden layer nodes, etc.

3.2 Automatic tuning method of model parameters

The setting of hyperparameter values has a great influence on the effect and convergence rate of the algorithm. Among them, hyperparameter values include learning rate, number of hidden layer nodes, regularization coefficient, etc. In the electricity demand forecast of the power market using short-term and long-term mixed modes, how to select hyperparameters is extremely critical. Since the electricity consumption data in the power market often shows obvious short-term fluctuations and long-term change trends, modeling it should not only consider its impact on short-term changes, but also its impact on long-term changes. Therefore, in the modeling process, how to use the optimal method to adaptively adjust the hyperparameters to achieve the best forecasting results is an extremely important topic [14].

This paper proposes an adaptive optimization method based on genetic algorithm. The algorithm is based on the difference of the group and ensures the maximum convergence speed of the algorithm. The particle swarm optimization strategy is adopted to improve the efficiency of optimization solution. This method uses the movement of particle swarms in various parameter spaces to achieve gradual approximation of the optimization problem, so that this method has shown good results in hybrid modeling of power load. The best hyperparameter combination is gradually found based on the properties of the initial values and the optimization strategy in different hyperparameter spaces [15]. For example, in the forecast of power demand in a hybrid power market, this method achieves adaptability to complex and changeable power grid loads by real-time adjustment of the network learning rate and the number of hidden layer nodes, thereby improving the accuracy and reliability of the forecast.

$$\theta^* = \arg\min_{\theta\in\Theta} L(\theta) \tag{7}$$

θ^* is the optimal parameter combination, $L(\theta)$ is the loss function, and Θ is the hyperparameter space. The optimization algorithm searches for Θ and finally finds the parameter θ^* that minimizes the loss function. The algorithmic flowchart of the genetic algorithm for hyperparameter optimization is as Table 1:

Table 1: Hyperparameters considered, their ranges, and selected optimal values

Hyperparameter	**Range**	**Optimal Value**
Learning rate	0.001-0.1	0.01
Number of hidden layer nodes (encoder)	32-256	128
Number of hidden layer nodes (decoder)	32-256	128
Sequence length (input)	72-336	168
Sequence length (output, short-term)	12-48	24
Sequence length (output, long-term)	120-216	168
Regularization coefficient	0.0001-0.01	0.001
Weight coefficient	0-1	0.7

1. Initialize a population of hyperparameter combinations randomly within the specified ranges.
2. Evaluate the fitness of each individual in the population using the objective function (prediction error).
3. Select the individuals with higher fitness for reproduction.
4. Perform crossover and mutation operations to generate offspring.
5. Replace the worst-performing individuals in the population with the offspring.
6. Repeat steps 2-5 until the stopping condition (maximum number of generations or convergence) is met.
7. Output the best hyperparameter combination found.

The particle swarm optimization algorithm starts by initializing a particle swarm with random positions and velocities within the parameter space. For each particle, its fitness is evaluated using the objective function, and its personal best position is set as the current position if the fitness is better. The global best position is then determined among all personal best positions. The algorithm proceeds in a loop until the stopping condition is met: for each particle, its velocity is updated using the formula $v_i = w \cdot v_i + c1 \cdot r1 \cdot (\mathrm{pbest}_i - x_i) + c2 \cdot r2 \cdot (gbest - x_i)$ (where w is the inertia weight, $c1$ and $c2$ are acceleration coefficients, $r1$ and $r2$ are random numbers between 0 and 1, v_i is the velocity of particle i, x_i is the position of particle i, $pbest_i$ is the personal best position of particle i, and $gbest$ is the global best position), and its position is updated as $x_i = x_i + v_i$. After evaluating the fitness of the new position, the personal best and global best positions are updated if necessary. Finally, the global best position is outputted.

4 System simulation and result analysis

In the electricity demand forecasting of the power market, the design of the long-term and short-term hybrid model needs to be verified and optimized through system simulation. This paper evaluates the performance of the proposed model through simulation experiments, aiming to verify its accuracy and stability in the electricity demand forecasting of the power market [16]. The simulation experiment includes the construction of the simulation environment, the selection and preprocessing of the data set, and the performance comparison of various models in different scenarios. A variety of evaluation indicators are used to analyze the performance of the model.

4.1 Introduction to simulation environment and data set

The data for this experiment comes from the historical electricity market demand data set of Yunnan Province, China, which contains hourly electricity market demand information from 2018 to 2022. This data set records the changes in electricity demand on working days and non-working days, and has obvious seasonal and cyclical fluctuation characteristics.

Before using these data for model training, they need to be preprocessed. The preprocessing steps include:

- Missing data handling: Missing values are filled using linear interpolation.
- Normalization: Min-max scaling is applied to scale the data to the range [0,1] using $x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$, where x is the original data, $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the data set, respectively.
- Outlier detection and removal: Outliers are detected using the Z-score method ($|Z| > 3$) and replaced with the mean value of the neighboring data points.

The simulation experiment in this article is carried out in the Python programming environment, and the model is mainly built using the TensorFlow and Keras frameworks.

4.2 Model performance evaluation

During the simulation process, this paper compares the performance of four types of models: the traditional ARIMA model, the LSTM model based on deep learning, the Seq2Seq model proposed in this paper, and the hybrid model of Seq2Seq and ARIMA. In addition, a

Transformer-based model is also included as a baseline for comparison. The experiment designed two scenarios: short-term demand forecasting (1 hour to 24 hours forecasting) and long-term demand forecasting (24 hours to one week forecasting).

Table 2: Evaluation results of each model under short-term demand forecasting.

Model type	MSE	RMSE	MAE
ARIMA	0.025	0.158	0.132
LSTM	0.019	0.138	0.104
Transformer	0.017	0.130	0.100
Seq2Seq	0.015	0.122	0.092
Hybrid (Seq2Seq+ARIMA)	0.013	0.114	0.085

As can be seen from Table 2, the hybrid model performs better than other models in short-term demand forecasting, and its MSE, RMSE and MAE values are lower than those of the other models.

Table 3: Evaluation results of each model under long-term demand forecasting.

Model type	MSE	RMSE	MAE
ARIMA	0.035	0.187	0.162
LSTM	0.029	0.170	0.141
Transformer	0.026	0.161	0.135
Seq2Seq	0.022	0.148	0.120
Hybrid (Seq2Seq+ARIMA)	0.020	0.141	0.112

The long-term demand forecast results in Table 3 show that the hybrid model also performs best in long-term forecasting, especially when dealing with complex long-term fluctuations, the model shows stronger stability and adaptability.

Table 4: Comparison of performance of each model under different weather conditions.

Weather conditions	Model Type	MSE	RMSE	MAE
Sunny	ARIMA	0.028	0.167	0.145
	LSTM	0.022	0.148	0.126
	Transformer	0.020	0.141	0.120
	Seq2Seq	0.017	0.131	0.110
	Hybrid (Seq2Seq+ARIMA)	0.015	0.122	0.102
Rainy	ARIMA	0.031	0.176	0.150
	LSTM	0.025	0.158	0.133
	Transformer	0.023	0.152	0.128
	Seq2Seq	0.019	0.138	0.115
	Hybrid (Seq2Seq+ARIMA)	0.017	0.130	0.108

Table 4 shows the performance of each model under various meteorological factors, especially in adverse meteorological environments such as rainy weather, the hybrid model has better stability and adaptability. The hybrid model consistently outperforms other models across different weather conditions, with the lowest MSE, RMSE, and MAE values, indicating its strong ability to handle the impact of meteorological factors on power demand.

To confirm the statistical significance of the improvements achieved by the hybrid model, paired t-tests are conducted between the hybrid model and each of the other models. The results are shown in Table 5, where the p-values are all less than 0.05, indicating that the improvements are statistically significant.

Table 5: Paired t-test results between the hybrid model and other models

Compared Model	p-value (short-term)	p-value (long-term)
ARIMA	0.002	0.001
LSTM	0.015	0.012
Transformer	0.028	0.025
Seq2Seq	0.035	0.031

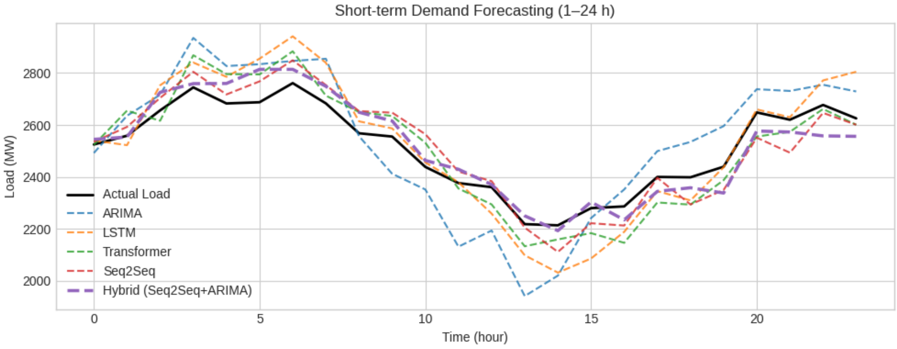


Figure 2: Simulation results of short-term demand forecasting.

The x-axis represents time (hours), and the y-axis represents load (MW). The actual load is shown as a solid line, while the predicted loads of the Seq2Seq model, LSTM model, ARIMA model, Transformer model, and hybrid model are shown as dashed lines with different colors. The hybrid model's prediction curve is closest to the actual load curve, with the smallest deviation.

As can be seen from Figure 2, the prediction results of the hybrid model are closest to the actual demand trend, followed by the Seq2Seq model, then the Transformer model, the LSTM model, while the ARIMA model has a large deviation.

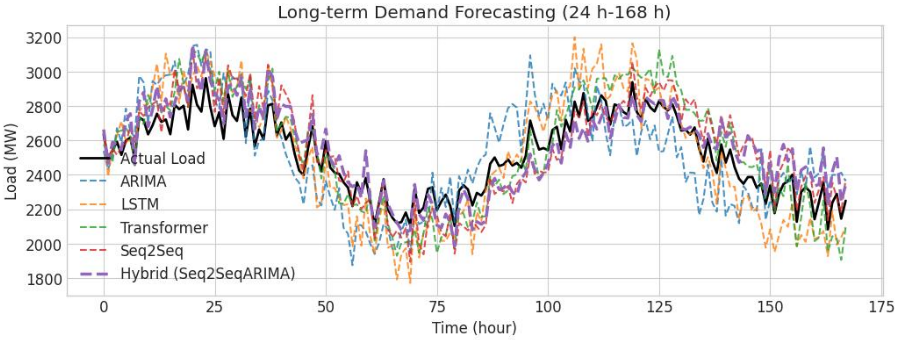


Figure 3: Simulation results of long-term demand forecasting.

The x-axis represents time (hours), and the y-axis represents load (MW). The actual load is shown as a solid line, and the predicted loads of various models are shown as dashed lines with different colors. The hybrid model maintains high accuracy even in the long-term forecasting, with stable performance. Figure 3 shows the performance of each model in long-term demand forecasting. Like short-term forecasting, Seq2Seq also performs significantly better than other models in long-term forecasting, especially when forecasting time periods with large fluctuations, the performance is more stable.

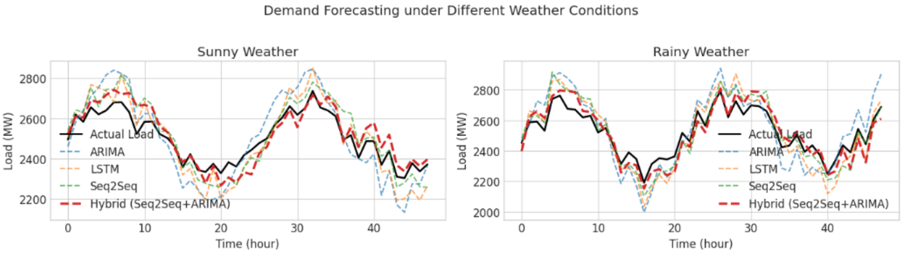


Figure 4: Demand forecast results under different weather conditions.

The left subfigure shows the forecast results under sunny weather, and the right subfigure shows the forecast results under rainy weather. The x-axis represents time (hours), and the y-axis represents load (MW). The hybrid model's prediction curve is smoother and closer to the actual demand changes in both weather conditions, demonstrating its strong adaptability. Figure 4 shows the demand forecast results under different weather conditions. Seq2Seq still performs better than other models under complex weather conditions such as cloudy weather, and the prediction curve is smoother and closer to the actual demand changes.

5 Conclusion

This study constructs a hybrid Seq2Seq-ARIMA model for power system load forecasting, optimized by metaheuristic algorithms (genetic algorithm and particle swarm optimization). The model integrates the advantages of Seq2Seq in capturing nonlinear time series dependencies and ARIMA in describing linear trends, and realizes adaptive adjustment of hyperparameters through optimization algorithms, thereby improving the accuracy and robustness of load forecasting.

Through simulation experiments, it is verified that the hybrid model has excellent performance in both short-term and long-term load forecasting. Compared with ARIMA, LSTM, Transformer, and single Seq2Seq models, the hybrid model has lower MSE, RMSE, and MAE values. The average error rate of short-term prediction is within 5%, and within 7% in the longer period. Statistical tests confirm that the improvements are statistically significant. In addition, the hybrid model shows good stability and adaptability under different weather conditions.

However, this study also has certain limitations. The model's computational complexity is relatively high, which may affect its application in real-time forecasting scenarios with strict time constraints. In future research, we will focus on optimizing the model's structure to reduce computational complexity while maintaining forecasting accuracy. In addition, we will expand the dataset to include more regions and longer time series to further verify the generalization ability of the model.

References

[1] Neeraj, N., Mathew, J., Agarwal, M., & Behera, R. K. (2021). Long short-term memory-singular spectrum analysis-based model for electric load forecasting. Electrical Engineering, 103(2), 1067–1082. https://doi.org/10.1007/s00202-020-01135-y

[2] Zhang, X., Chau, T. K., Chow, Y. H., Fernando, T.,

& Iu, H. H. C. (2023). A novel sequence to sequence data modelling-based CNN-LSTM algorithm for three years ahead monthly peak load forecasting. IEEE Transactions on Power Systems, 39(1), 1932–1947. https://doi.org/10.1109/TPWRS.2023.3271325

[3] Jalali, S. M. J., Ahmadian, S., Khosravi, A., Shafie-khah, M., Nahavandi, S., & Catalão, J. P. (2021). A novel evolutionary-based deep convolutional neural network model for intelligent load forecasting. IEEE Transactions on Industrial Informatics, 17(12), 8243–8253. https://doi.org/10.1109/TII.2021.3065718

[4] Zhang, X., Kuenzel, S., Colombo, N., & Watkins, C. (2022). Hybrid short-term load forecasting method based on empirical wavelet transform and bidirectional long short-term memory neural networks. Journal of Modern Power Systems and Clean Energy, 10(5), 1216–1228. https://doi.org/10.35833/MPCE.2021.000276

[5] Lin, X., Zamora, R., Baguley, C. A., & Srivastava, A. K. (2022). A hybrid short-term load forecasting approach for individual residential customer. IEEE Transactions on Power Delivery, 38(1), 26–37. https://doi.org/10.1109/TPWRD.2022.3178822

[6] Panda, S. K., & Ray, P. (2022). Analysis and evaluation of two short-term load forecasting techniques. International Journal of Emerging Electric Power Systems, 23(2), 183–196. https://doi.org/doi.org/10.1515/ijeeps-2021-0051

[7] Veeramsetty, V., Chandra, D. R., Grimaccia, F., & Mussetta, M. (2022). Short term electric power load forecasting using principal component analysis and recurrent neural networks. Forecasting, 4(1), 149–164. https://doi.org/10.3390/forecast4010008

[8] Lang, R., Ye, W., Zhao, F., & Li, Z. (2020). An Adaptive Algorithm for Calculating Crosstalk Error for Blind Source Separation. Informatica, 31(2), 299-312. https://doi.org/10.15388/20-INFOR387

[9] Daneshdoost, F., Hajiaghaei-Keshteli, M., Sahin, R., & Niroomand, S. (2022). Tabu Search Based Hybrid Meta-Heuristic Approaches for Schedule-Based Production Cost Minimization Problem for the Case of Cable Manufacturing Systems. Informatica, 33(3), 499-522. https://doi.org/10.15388/21-INFOR471

[10] Kučera, R., Arzt, V., & Koko, J. (2024). MINI Element for the Navier–Stokes System in 3D: Vectorized Codes and Superconvergence. Informatica, 35(2), 341-361. https://doi.org/10.15388/24-INFOR543

[11] Çelik, E. (2021). Design of new fractional order PI–fractional order PD cascade controller through dragonfly search algorithm for advanced load frequency control of power systems. Soft Computing, 25(2), 1193–1217. https://doi.org/10.1007/s00500-020-05215-w

[12] Vedik, B., Kumar, R., Deshmukh, R., Verma, S., & Shiva, C. K. (2021). Renewable energy-based load frequency stabilization of interconnected power systems using quasi-oppositional dragonfly algorithm. Journal of Control, Automation and Electrical Systems, 32(1), 227–243. https://doi.org/10.1007/s40313-020-00643-3

[13] Sobhy, M. A., Abdelaziz, A. Y., Hasanien, H. M., & Ezzat, M. (2021). Marine predators' algorithm for load frequency control of modern interconnected power systems including renewable energy sources and energy storage units. Ain Shams Engineering Journal, 12(4), 3843–3857. https://doi.org/10.1016/j.asej.2021.04.031

[14] Khadanga, R. K., Kumar, A., & Panda, S. (2020). A novel modified whale optimization algorithm for load frequency controller design of a two-area power system composing of PV grid and thermal generator. Neural Computing and Applications, 32(12), 8205–8216. https://doi.org/10.1007/s00521-019-04321-7

[15] Sun, T., Bian, S., Sun, Y., Wang, Z., Li, W., & Chong, F. (2020). Technical support system for power system load modeling. Recent Advances in Electrical and Electronic Engineering, 13(7), 1059–1067. DOI:10.2174/2352096513666200309110756

[16] Dewangan, S., Prakash, T., & Pratap Singh, V. (2021). Design and performance analysis of elephant herding optimization-based controller for load frequency control in thermal interconnected power system. Optimization and Control Applications and Methods, 42(1), 144–159. https://doi.org/10.1002/oca.2666

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several sci- entific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute tem- perature, known as the Stefan–Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, en- ergy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research de- partments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 900 staff, has 700 researchers, about 250 of whom are postgraduates, around 500 of whom have doctorates (Ph.D.), and around 200 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the uni- versities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; ap- plied mathematics. Most of the activities are more or less closely connected to information sciences, in particu- lar computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automa- tion and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the in dependent state of S**love**nia (or S♡nia). The capital today isconsidered a crossroad bet between East, West and Mediter-ranean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

From the Jožef Stefan Institute, the Technology Park “Ljubljana” has been proposed as part of the national strat- egy for technological development to foster synergies be- tween research and industry, to promote joint ventures be- tween university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

Part of the Institute was reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology Park "Ljubljana". The project was developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park is a shareholding company hosting an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Higher Education, Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of the Economy, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel.: +386 1 4773 900, Fax.: +386 1 251 93 85
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Public relations: Polona Strnad

Informatica

An International Journal of Computing and Informatics

Web edition of Informatica may be accessed at: http://www.informatica.si.

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Litostrojska cesta 54, 1000 Ljubljana, Slovenia.
The subscription rate for 2022 (Volume 46) is
– 60 EUR for institutions,
– 30 EUR for individuals, and
– 15 EUR for students
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

Typesetting: Blaž Mahnič, Gašper Slapničar; gasper.slapnicar@ijs.si
Printing: ABO grafika d.o.o., Ob železnici 16, 1000 Ljubljana.

Orders may be placed by email (drago.torkar@ijs.si), telephone (+386 1 477 3900) or fax (+386 1 251 93 85). The payment should be made to our bank account no.: 02083-0013014662 at NLB d.d., 1520 Ljubljana, Trg republike 2, Slovenija, IBAN no.: SI56020830013014662, SWIFT Code: LJBASI2X.

Informatica is published by Slovene Society Informatika (president Slavko Žitnik) in cooperation with the following societies (and contact persons):
Slovene Society for Pattern Recognition (Matej Kristan)
Slovenian Artificial Intelligence Society (Aleksander Sadikov)
Cognitive Science Society (Toma Strle)
Slovenian Society of Mathematicians, Physicists and Astronomers (Mojca Vilfan)
Automatic Control Society of Slovenia (Giovanni Godena)
Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Matjaž
Mikoš) ACM Slovenia (Ljupčo Todorovski)

Informatica is financially supported by the Slovenian research agency from the Call for co-financing of scientific periodical publications.

Informatica is surveyed by: ACM Digital Library, Citeseer, COBISS, Compendex, Computer & Information Systems Abstracts, Computer Database, Computer Science Index, Current Mathematical Publications, DBLP Computer Science Bibliography, Directory of Open Access Journals, InfoTrac OneFile, Inspec, Linguistic and Language Behaviour Abstracts, Mathematical Reviews, MatSciNet, MatSci on SilverPlatter, Scopus, Zentralblatt Math

Volume 49 Number 3 October 2025 **ISSN 0350-5596**

Informatica

An International Journal of Computing and Informatics

2024 ACM A.M. Turing Award: Richard S. Sutton and Andrew G. Barto for Reinforcement Learning	M.Gams	**459**
Special issue on “The 13th International Symposium on Information and Communication Technology—SOICT 2024”	H. T. T. Binh	**461**

Start of Special Issue

Context-Enriched Dynamic Graph Word Embeddings for Robust NLP Applications	T. X. Tran, R. E. Himes, HA. Tran	**463**
Enhanced Cardio Care: Explainable Vision Transformer Multimodal Pipeline for Cardiac Abnormalities Detection Using Electrocardiogram Image Reports	N. M. To, V. Q. Vo, Q. C. Ngo, D. Kumar, M. N. Dinh, D. V. Nguyen, D. V. B. Do	**473**
Analysis of Behavioral Facilitation Information During Disasters Based on Reader Attributes and Personality Traits	A. Nadamoto, K. Wakasugi, Y. Suzuki, T. Kumamoto	**483**
New Local Search Strategy for the Minimum s-Club Cover Problem	T. P. Dinh, T. A. Do, S. N. Hung, T. N. Duc	**495**

End of Special Issue / Start of Normal Papers

Enhanced YOLOv11 for Robust Real-Time Skiing Action Recognition via Multimodal and Spatiotemporal Learning	D. Liu, M. Ju	**507**
A Novel CNN with Spatial and Channel Attention for Automated Chest X-Ray Diagnosis	D. Priyanka, E. Aravind	**525**
Cancer Classification through Gene Selection Using the Social Spider Optimization Algorithm	C. Cherif, M. Maiza, S. Chouraqui, A. Taleb-Ahmed	**537**
Fusion of Convolutional Architecture and Transformer Models for Enhanced Brain Tumor Classification	V. Sabitha, J. Nayak, P. R. Reddy	**551**
Convolutional Neural Network (CNN) Based Martian Dune Detection	N. V. Scariah, M. G. N. Lala, A. P. Krishna	**561**
Optimizing Social Media Analytics with the DQEA Framework for Superior Data Quality Management	Karthick B, Meyyappan T	**577**
A Review on Artificial Intelligence Based Heuristic Models for Brain Tumor Image Classification and Segmentation	M. S. Prasad, N. U. Khan	**589**
Deep Neuro-Fuzzy System for Early-Stage Identification of Parkinson’s Disease Using SPECT Images	Jothi S, Anita S, Sivakumar S	**601**
Hybrid Time Series Forecasting for Real-Time Electricity Market Demand Using ARIMA-LSTM and Scalable Cloud-Native Architecture	X. Wang, Y. Wu, W. Zou, X. Zhao	**615**
Hybrid Seq2Seq-ARIMA Load Forecasting for Power Systems with Metaheuristic Hyperparameter Optimization	J. Zou, C. Li, L. Zhang, F. Hu	**623**

Informatica **49** (2025) Number 3, pp. 459–632