Princess Sumaya جامعة
University الأميرة سميّة
for Technology للتكنولوجيا

صندوق دعم البحث العلمي والابتكار
Scientific Research and Innovation Support Fund

Jordanian Journal of Computers and Information Technology

September 2025 VOLUME 11 NUMBER 03 ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

www.jjcit.org jjcit@psut.edu.jo

An International Peer-Reviewed Scientific Journal Financed
by the Scientific Research and Innovation Support Fund

Jordanian Journal of Computers and Information Technology (JJCIT)

The Jordanian Journal of Computers and Information Technology (JJCIT) is an international journal that publishes original, high-quality and cutting edge research papers on all aspects and technologies in ICT fields.

JJCIT is hosted and published by Princess Sumaya University for Technology (PSUT) and supported by the Scientific Research Support Fund in Jordan. Researchers have the right to read, print, distribute, search, download, copy or link to the full text of articles. JJCIT permits reproduction as long as the source is acknowledged.

Aims and Scope

The JJCIT aims to publish the most current developments in the form of original articles as well as review articles in all areas of Telecommunications, Computer Engineering and Information Technology and make them available to researchers worldwide. The JJCIT focuses on topics including, but not limited to: Computer Engineering & Communication Networks, Computer Science & Information Systems and Information Technology and Applications.

Indexing

JJCIT is indexed in:

ELSEVIER Scopus | INDEXED IN EMERGING SOURCES CITATION (Web of Science) CLARIVATE ANALYTICS | DOAJ DIRECTORY OF OPEN ACCESS JOURNALS | crossref Text and Data Mining

OCLC WorldCat | BASE | ACADEMIA | Google scholar

Editorial Board Support Team

JJCIT Address

Website: www.jjcit.org
Email: jjcit@psut.edu.jo
Address: Princess Sumaya University for Technology, Khalil Saket Street, Al-Jubaiha
B.O. Box: 1438 Amman 11941 Jordan
Telephone: +962-6-5359949
Fax: +962-6-7295534

EDITORIAL BOARD

INTERNATIONAL ADVISORY BOARD

“Opinions or views expressed in papers published in this journal are those of the author(s) and do not necessarily reflect those of the Editorial Board, the host university or the policy of the Scientific Research Support Fund”.

"ما ورد في هذه المجلة يعبر عن آراء الباحثين ولا يعكس بالضرورة آراء هيئة التحرير أو الجامعة أو سياسة
صندوق دعم البحث العلمي والابتكار".

A Hybrid CNN-transformer Approach for Precise Three-class Diabetic Retinopathy Classification

Samira Ait Kaci Azzou[1], Djamila Boukredera[2] and Sifeddine Baouz[3]

(Received: 4-Feb.-2025, Revised: 11-Apr.-2025, Accepted: 16-Apr.-2025)

Abstract

This study evaluates the effectiveness of Vision Transformers (ViTs) and hybrid deep-learning architectures for diabetic retinopathy (DR) classification, addressing the challenge of inter-stage ambiguity in traditional systems. While convolutional neural networks (CNNs) such as ResNet50 excel at localized feature extraction in retinal images, ViTs offer superior global contextual modeling. To synergize these strengths, we propose a hybrid architecture integrating ResNet50's granular feature extraction with ViTs' global relational reasoning. Three models are designed and evaluated: (1) an auto-tuned ResNet50, (2) a hyperparameter-optimized ViT and (3) a hybrid model combining both architectures. To reduce ambiguity between neighboring stages, we simplified the traditional five-stage classification into three clinically relevant categories: no DR, early DR (mild/moderate) and advanced DR (severe/proliferative). Trained and validated on the APTOS dataset, the ResNet50 model achieves precision scores of 93.0% (No DR), 82.0% (Early DR) and 86.0% (Advanced DR). The standalone ViT demonstrates relative improvements, attaining 98.0%, 91.0% and 93.0%, respectively. The hybrid model surpasses both, achieving 98.0% average precision across all classes, with gains of +7.0% (early DR) and +5.0% (advanced DR) over the standalone ViT. The proposed hybrid model achieved an impressive value of 99.5% on all metrics (accuracy, precision and recall) for identifying DR (binary classification) and a value of 98.3% for 3-stage classification. It was also concluded that the proposed method achieved better performance in DR detection and classification compared to conventional CNN and other state-of-the-art methods. The proposed hybrid approach significantly reduces confusion between classes, demonstrating its potential for accurate classification of the different stages of DR.

Keywords

Diabetic retinopathy, Vision transformer, Transfer learning, Artificial intelligence.

1. Introduction

Diabetic retinopathy (DR) is a disease that affects the blood vessels of the retina and can result in blindness. It is a serious complication in diabetic patients [1]-[2]. DDR is identified by the emergence of several types of lesions on the retina. The lesions include microaneurysms (MAs), hemorrhages (HMs) and soft and hard exudates (EXs) [3]. Positive RD is split into several stages. (1) Microaneurysms indicate the mild phase, (2) Moderate stage reveals a stage where blood vessels begin to lose their ability to transport, (3) Severe stage includes blood vessel obstructions and (4) Proliferative stage represent the advanced phases of RD, as shown in Figure 1.

According to the International Diabetes Federation [4], there are around 537 million diabetics, with this figure anticipated to increase to 643 millions by 2030 and 783 millions by 2045. Furthermore, most individuals with diabetes remain undiagnosed for DR, because this disease is often asymptomatic until an advanced stage [5]. In order to diagnose and treat DR, regular retinal screening is essential for diabetic patients. Classification issues associated with DR can be divided into two categories: binary classification and five-class classification. Binary classification focuses on distinguishing between sick and healthy retinas in color fundus images, as established by [6]-[8]. Conversely, five-class classification methodologies strive to categorize images into five distinct classes: Class 0- no DR, Class 1- mild DR, Class 2- moderate DR, Class 3- severe DR and Class 4 -proliferative DR [9]-[10], as resumed in Figure 1. Manual examination of retinal images is carried out using traditional methods to detect the presence of DR, which requires experienced and professional ophthalmologists. In

1. S. Ait Kaci Azzou is with University of Bejaia, Faculty of Exact Sciences, LIMED Laboratory, 06000 Bejaia, Algeria. Email: samira.aitkaciazzou@univ-bejaia.dz
2. D. Boukredera is with University of Bejaia, Faculty of Exact Sciences, LMA Laboratory, 06000 Bejaia, Algeria. Email: djamila.boukredera@univ-bejaia.dz
3. S. Baouz is with University of Bejaia, Faculty of Exact Sciences, Department of Computer Science, 06000 Bejaia, Algeria. Email: sifeddine.baouz@se.univ-bejaia.dz

addition, there is a high probability of misdiagnosis during the manual examination, which is time-consuming and costly.

Automated methods have emerged as viable solutions to enable early identification of Diabetic Retinopathy (DR) and avoid permanent blindness [11]-[12], overcoming problems related to manual classification. In this case, machine learning has shown to be the most effective technique to overcome this problem [13].

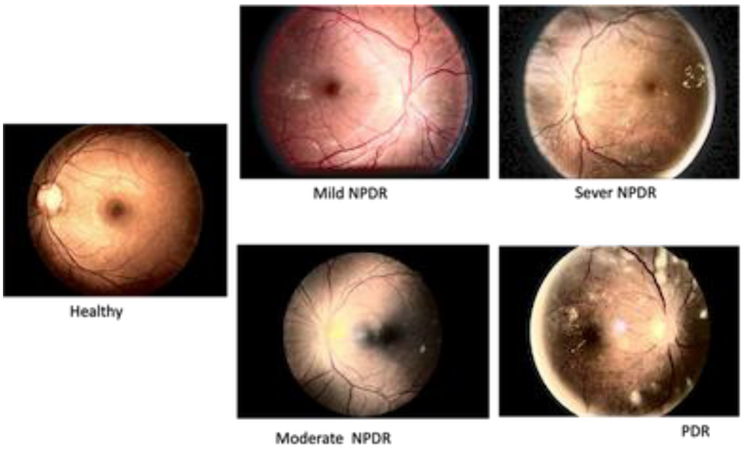


Figure 1. Fundus images representing phases of diabetic retinopathy from the Aptos dataset.

Deep-learning (DL) methods, particularly transfer-learning models, like VGG16, InceptionV3 and ResNet50, have shown considerable promise in analyzing medical images [14-[17]. Convolutional neural networks (CNNs), which underpin these models, mainly concentrate on local features in the input images, which restricts their capability to effectively recognize long-range dependencies and global contextual connections. Vision Transformers (ViTs) have emerged as a revolutionary substitute, addressing these constraints by utilizing self-attention mechanisms to capture long-range dependencies and global contextual associations throughout whole images. While transfer learning-based approaches [18]-[19] have been widely adopted for diabetic-retinopathy (DR) severity classification, existing methods struggle with diagnostic accuracy in early-stage DR, where subtle lesion patterns (e.g. microaneurysms, mild hemorrhages) necessitate both fine-grained feature extraction and global contextual understanding of the retinal image.

To address these challenges and evaluate the effectiveness of ViTs for DR classification, we propose and compare three architectures, each differing in its feature extraction method:

1) ResNet50-based model: A CNN baseline optimized *via* Bayesian hyperparameter tuning for localized feature extraction.
2) ViT-based model:A standalone ViT model tailored for global dependency modeling.
3) Hybrid architecture: A novel fusion of ResNet50 and ViT, combining their complementary strengths.

We further redefine the traditional five-stage DR grading system into three clinically relevant classes: no DR, early DR (encompassing mild and moderate stages) and advanced DR (comprising severe and proliferative stages). This regrouping minimizes confusion between closely related stages, enhancing classification accuracy. Experiments carried out on the APTOS 2019 dataset [20] demonstrate that the hybrid architecture achieves 98.0% precision across all classes, reducing misclassification between adjacent stages by 15%–20% compared to standalone models. ViTs alone outperform ResNet50, with relative improvements of 11.0% (early DR) and 8.1% (advanced DR) in precision. The hybrid architecture significantly enhances early-stage detection of DR, leading to better clinical results.

To sum up, our contributions are as follows:

1) Three novel architectures for DR detection and classification:
   - AtRD/AtR3C: Auto-tuned ResNet50 models with Bayesian hyper-parameter optimization, achieving 99.22% detection accuracy and 94.26% 3-class severity-classification accuracy.
   - ViRD/ViR3C: Vision Transformer (ViT) models leveraging global attention, attaining 97.73%

detection accuracy and 92.97% classification accuracy.

- Revi-RD/Revi-3C: A hybrid CNN-Transformer architecture combining both precedent architectures. It achieves 99.55% detection accuracy and 98.26% 3-class severity-classification accuracy.

2) Redefined DR grading into (0: no DR, 1: early DR, 2: advanced DR), reducing ambiguity in traditional 5-stage grading between neighbor classes.
3) State-of-the-Art Performance:
   - The proposed models are validated on the APTOS 2019 dataset and compared against one another, highlighting the effectiveness of ViTs and the complementary advantages of the CNN-ViT hybrid architecture.
   - The earliest stages were detected with greater accuracy, especially in the hybrid model.
   - We effectively optimized each model's performance as compared to previous methodologies. With the hybrid approach, we greatly outperformed previous results.

The rest of the article is organized as follows: Section 2 reviews relevant research conducted on the DR classification. Section 3 details the methodology, including data pre-processing, the proposed approach and performance measures. The results are presented and analyzed in Section 4. Finally, Section 5 presents the key conclusions and recommendations for future works.

2. Literature Review

Early identification of Diabetic Retinopathy (DR) remains a significant challenge. Researchers have investigated several techniques to address this issue. Classifying DR from retinal images falls into two main categories. Binary classification determines whether or not DR exists, whereas multi-class classification indicates the disease's specific stage. This latter method needs the model to differentiate minor visual variations between DR stages, making it a more difficult task. Several studies have investigated both binary and 5-class classification of DR using machine-learning (ML) [13], [21]-[22], deep-learning (DL) [14], [23]-[25], transfer-learning techniques (TL) [8], [26]-[28] and more recently vision-transformer methods [29]-[30]. However, research into the classification of DR into three classes remains limited. Public retinal-image datasets, such as Idrid, EyePACS, Messidor and Aptos, have been instrumental in these studies for detecting and diagnosing DR. This work will specifically focus on recent advancements in transfer learning (TL) and Vision Transformer (ViT) applied to DR detection and classification on the Aptos dataset.

2.1 Transfer Learning in DR Classification

Dekhil et al. [31] proposed a customized CNN based on a transfer-learning technique for a 3-class classification task. It consists of a pre-processing stage, VGG16 and fully connected layers. To adapt the pre-trained model, they retrained all the layers, achieving a validation accuracy of 77%. In their study [32], Rao et al. evaluated five CNN classifiers; namely, Inception-V3, VGG19, VGG16, Resnet50 and InceptionResNetV2. Resnet50 achieved the highest accuracy (95.59%) for a binary classification. InceptionResNetV2 excelled at multi-class classification. It reached an accuracy of 88.14% for classifying DR into three stages and 85% accuracy for a five-stage classification. Gangwar and Rav [33] proposed an hybrid model incorporating a custom convolutional neural network (CNN) block added to the pre-trained Inception-ResNet-v2. For training these hybrid models, they utilized two Kaggle datasets: Messidor-1 and the APTOS 2019. The achieved test accuracy was 72.33% for Messidor-1 and 82.18% for the APTOS 2019 dataset, respectively. Islam et al. [34] proposed an architecture based on supervised contrastive learning, utilizing the pre-trained Xception model, the APTOS dataset and Messidor-2. They achieved an accuracy of 98.36% for binary classification and 84.364% for multi-class classification. Their study revealed an improvement in performance compared to previous architectures, including ResNet50, Inception and other earlier models. Oulhadj et al. [35] proposed an automatic method based on deep learning. It consists of two main steps; the first one is the pre-processing. The second one is the classification. Four CNN models (Densenet-121, Xception, Inception-v3 and Resnet50) are employed to detect the DR-severity stage. The authors implemented a voting mechanism using the APTOS 2019 dataset. They achieved a final accuracy of 85.28%. Mondal et al. [36] also suggested a deep-learning strategy for detecting diabetic retinopathy that combines the DenseNet101 and ResNet models. Experiments were carried out using the APTOS19 and

DIARETDB1 datasets. Their approach produced an accuracy of 86.08% for five-class classification and 96.98% for binary classification. Many CNN-based techniques have proved their ability to extract subtle image features surpassing traditional methods. While CNNs excel at extracting discriminative local features, crucial for recognizing subtle image characteristics, they struggle to process long-range information due to their inherent local receptive field mechanism. This limitation hinders their ability to fully understand the complex patterns associated with diabetic retinopathy. To address CNNs' difficulties in collecting long-range dependencies within retinal images, Vision Transformers (ViTs) have emerged as a potential solution.

2.2 ViT in DR Classification

Dosovitskiy et al. [37] introduced the Vision Transformer (ViT) for image classification, motivated by the effectiveness of transformers in natural-language processing [38]. ViTs have surpassed traditional convolutional neural networks in a variety of computer-vision tasks by considering images as sequences of patches and exploiting self-attention. Despite the promising potential of ViTs, their application in DR classification remains relatively unexplored and studies specifically focused on DR classification are still limited. Recently, the remarkable representation capabilities of transformers received increasing interest in medical-image analysis [39]-[40]. For DR classification, Wu et al. [41] employed ViTs to prove their superior performance compared to CNNs. Additionally, Mohan et al. [42] proved that dividing the fundus images into non-overlapping portions maintains information about the position of each patch. A different dataset was used to test the effectiveness of DR classification. For example, Nazih et al. [43] provided a ViT-based deep-learning pipeline for recognizing the severity stages of DR. ViT requires big datasets for successful learning; therefore, they utilized the FGADR (fine-grained annotated diabetic retinopathy) dataset, which comprises 1,842 fundus images, to build their model. Experimental results of their ViT model using F1-score, accuracy, and recall metrics were 82.5%, 82.5% and 82.5%, respectively. In [29], Gu et al. classified DR using ViT on the DDR dataset. The performances of the model using specificity, sensitivity and accuracy metrics were 82.45%, 81.40% and 82.35%, respectively. Khan et al. [44] presented an automated approach for DR-severity classification using a fine-tuned Compact Convolutional Transformer (CCT) model, which combines convolutional layers with transformer mechanisms. The model was trained on a huge dataset created by combining five datasets (Aptos, Idrid, Messidor2, DDR and Kaagel Dr dataset). Different pre-processing and augmentation techniques were used to improve image quality. The model achieved an accuracy of 84.5%, outperforming both the ViT (81.56%) and the shifted window transformer (Swin) (82.23%). Different ViT architectures are tested in the study conducted by Karkera et al. [45]. Four pre-trained image transformers: ViT, DeiT, CaiT and BEiT, were trained on a dataset called DBtr. The researchers then combined all four models to predict the severity stages of DR. The combined approach achieved an accuracy of 94.63% outperforming the results obtained by each of the individual models. Recently, Oulhadj et al. [46], proposed a hybrid architecture combining a fine-tuning vision transformer and a capsule network for automatic prediction of the severity level of diabetic retinopathy. The approach was evaluated using four datasets, including APTOS, Messidor-2, DDR and EyePACS and attained the best accuracy scores on the Aptos dataset: 88.18%. Lian and Liu in [47] combined a convolutional neural network (Inception-Resnet-v2) with a vision transformer. The model attained an accuracy of 93.2% using Messidor1 for binary classification and an accuracy of 89.1% using the Aptos dataset for 5-stage classification. Yang et al. [48] have developed a Transformer model based on multiple instance learning (MIL) to classify diabetic retinopathy (DR). Their model divides high-resolution retinal pictures into 224 × 224 pixel patches, which are then processed by a Vision Transformer (ViT) to extract local characteristics. A Global Instance Computing Block (GICB) then combines information from many patches, improving the model's capacity to understand global relationships within the image. The model obtained 93.2% accuracy for binary classification on the Messidor1 dataset and 85.65% accuracy for 5- stage classification on the Aptos dataset, surpassing the Mil-ViT proposed by Yu et al. [49]. Dihin et al. [50] used a combination of Wavelet and multi-Wavelet transforms with the Swin-transformer model. The study highlights the innovative use of the multi-Wavelet transform for feature extraction, integrated into the Swin transformer. The model obtained 96% accuracy for binary classification on the Kaggle APTOS 2019 dataset. The Swin-T model with multi-Wavelet transformation achieved a 98% recall and 96% F1-score for binary classification. However, the model's accuracy decreased in multi-class classification (82%). Approaches based solely on CNNs or ViTs struggle to combine the detection of local lesions with the analysis of the global anatomical context, which accentuates the ambiguity between classes. To demonstrate the efficacy of hybridization in overcoming these limitations, this study proposes a hybrid CNN-ViT architecture that combines fine feature extraction and contextual modeling. Further, we redefine DR staging into three-tier clinically actionable categories - no DR, early DR and advanced DR - to improve the accuracy of classification, which remains under-explored in the literature.

3. Methodology

This section presents three deep-learning architectures for the classification of diabetic retinopathy (DR). Each model was trained for binary detection (0: No DR, 1: DR) and three-stage severity classification (0: No DR, 1: Early DR, 2: Advanced DR). The first proposed architecture employs transfer learning with ResNet-50 for feature extraction. AtRD and AtR3C, respectively, handle binary and 3-class classification. The second proposed architecture uses ViTs for feature extraction. ViRD and ViR3C deal with binary and 3-class classification, respectively. Finally, we propose a hybrid architectures, ReVi-RD and ReVi-3C, for detection and 3-class classification, respectively, combining the strengths of both previous models. As illustrated in Figure 2, each model follows a similar pipeline composed of several processes:

- Pre-processing process that balances the dataset and enhances the quality of input images.
- Feature extraction is performed using the chosen architecturen (Rsnet50 and ViT).
- A multi-layer neural network classifies the image into two or 3-class classification. In the following part, we give more details for each of these processes.

3.1 Datset Description

A Kaggle dataset titled APTOS 2019 Blindness Detection (APTOS stands for Asia Pacific Tele Ophthalmology Society) was used to train and evaluate the models [20]. This dataset was collected by Aravind Eye Hospital in rural areas of India with the objective of developing high-performance tools for the automated diagnosis of diabetic retinopathy and enhancing the hospital's ability to identify potential patients. The dataset consists of 3,662 retinal images, categorized into five stages of diabetic retinopathy (DR)(see Figure 3b): no DR, mild DR, moderate DR, severe DR and proliferative DR, which are annotated with values ranging from 0 to 4. However, one of the main limitations of this dataset is the significant class imbalance, particularly for the severe NPDR category, which contains only 193 images. Additionally, the images vary in size and exhibit considerable variations due to their collection in a real-world multi-center environment. These variations arise from differences in camera settings across centers and the presence of noise, both in the data and in the annotations.

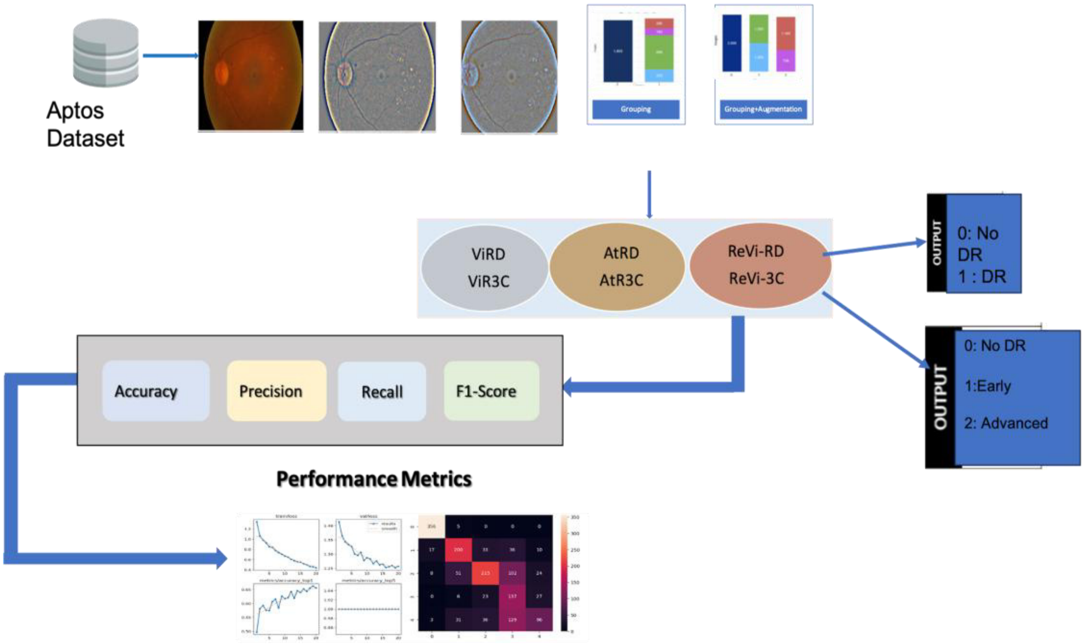
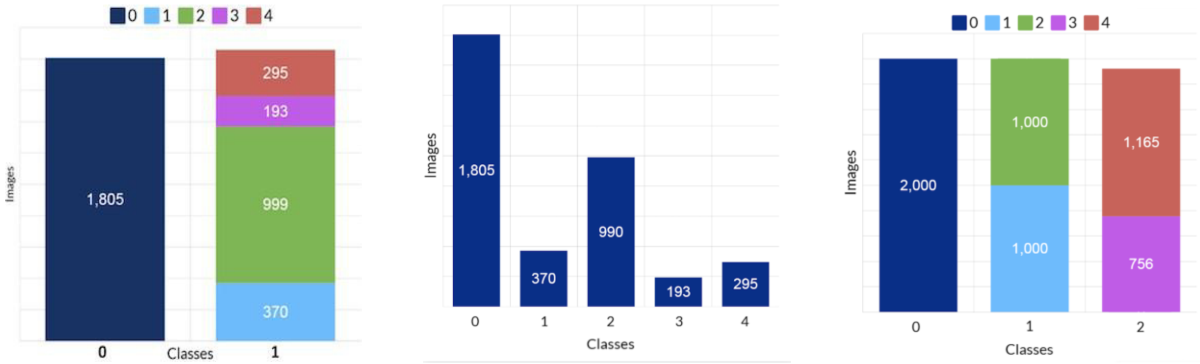


Figure 2. Proposed-approach pipeline from data pre-processing to class prediction.

3.2 Dataset Preparation

Our goal is to develop a model that can detect the existence of DR and classify its severity. As shown in Figure 3a and Figure 3b, the classes were grouped and re-annotated according to the classification task (binary or three-class classification, respectively). However, achieving an accurate model performance necessitates overcoming the persistent problem of data imbalance. For DR detection, we use a binary classification (No DR, DR). This grouping successfully balances the dataset, as shown in Figure 3a.



(a) Binary Aptos dataset (b) Aptos dataset before augmentation (c) 3-class Aptos dataset

Figure 3. Aptos dataset before and after aggregation and augmentation.

However, for three-stage classification, the problem of data imbalance persists. To address this issue, we use data-augmentation techniques that create additional images.

3.3 Data Augmentation

We employ data-augmentation techniques to expand the database and provide additional images of the different DR stages as illustrated in Figure 3c. Each original image underwent multiple augmentation transformations, resulting in five augmented images. These transformations include distortions, horizontal and vertical flips, as well as brightness adjustments. The purpose is twofold: expanding the dataset's variability while meticulously preserving the essential DR characteristics. This enables machine-learning models to learn and identify retinopathy features regardless of the image's position or lighting conditions. Figure 4 shows a sub-set of the generated images by the augmentation process.

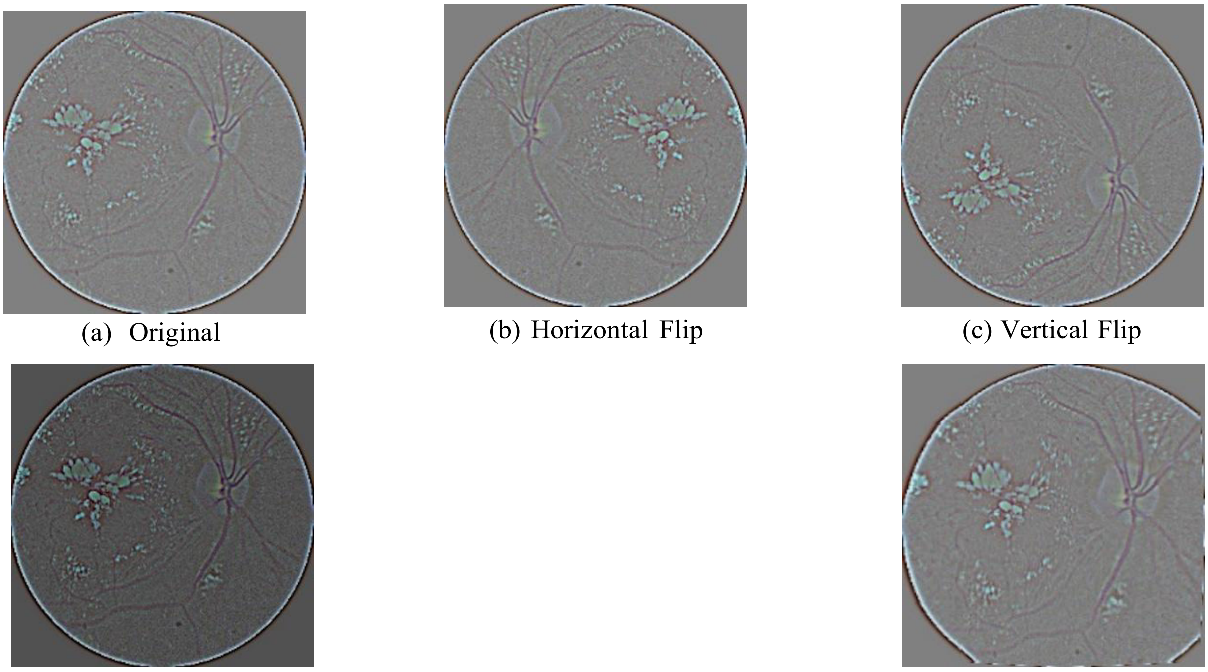
3.4 Image Pre-processing

Due to their many sources, the fundus images in the dataset show significant heterogeneity in terms of size, noise levels and distortion. These variations present significant problems for accurate analysis and reliable lesion detection. To overcome these obstacles and improve the quality of feature extraction, we propose a multi-stage pre-processing process (see Figure 5). The different stages of pre-processing that we have carried out are:

1) The initial step involves resizing all images to a uniform size of 224x224 pixels. This standardization facilitates subsequent analyses and the extraction of characteristics.
2) Each resized color image was converted into gray scale, followed by convolution using a Gaussian blur filter, as illustrated in Figure 5b [51].This step is designed to reduce noise and accentuate features, in particular by improving the visibility of exudate, red lesions and blood vessels.
3) A circular-cropping [52] technique was used to remove non-informative black pixels (background or noise) and retain only the regions of interest, as shown in Figure 5c.
4) Finally, normalization was performed on the pre-processed images to ensure consistent scaling of all pixel values, thereby enhancing the efficiency and stability of model training. This data normalization process aims to standardize the distribution of the images.

3.5 Fine Tuning

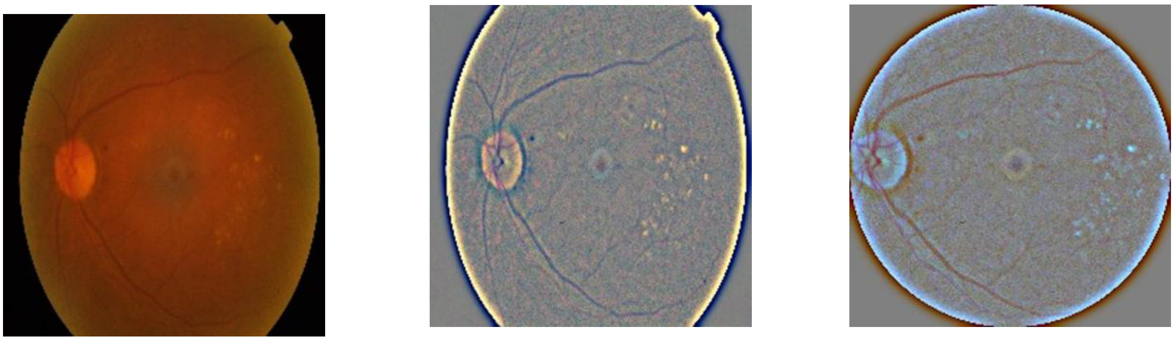
Pre-trained models, such as ResNet50 and Vision Transformers (ViTs), require fine-tuning to meet the specific demands of DR detection and classification. For proposed models—AtRD/AtR3C (ResNet50-



(a) Original (b) Horizontal Flip (c) Vertical Flip

(d) Brightness (e) Grid Distortion

Figure 4. Data-augmentation illustration.



(a) Original (b) Gaussian Blur (c) Circle Crop

Figure 5. Pre-processing phases.

based) and ViRD/ViR3C (ViT-based), we employed a two-phase optimization. First, the pre-trained architectures were fine-tuned on the APTOS dataset, enabling them to capture discriminative retinal features, such as microaneurysms, hemorrhages and exudates, by adapting their weights to the morphological patterns of DR. Second, we applied Bayesian optimization to systematically refine critical hyperparameters, including image resolution, batch size and learning rate, ensuring robust classification performance across DR-severity classes while minimizing overfitting. This dual-phase strategy optimizes both the models' feature-extraction capabilities and training dynamics.

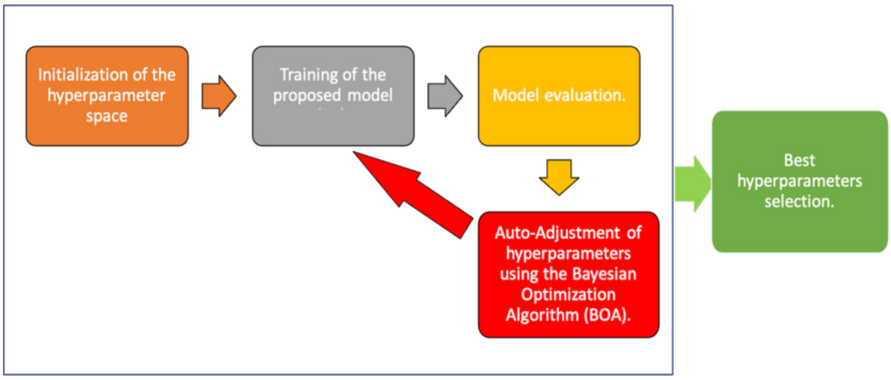

Initialization of the hyperparameter space
Training of the proposed model
Model evaluation.
Auto-Adjustment of hyperparameters using the Bayesian Optimization Algorithm (BOA).
Best hyperparameters selection.

Figure 6. Auto-hyperparameter-tuning process.

As shown in Figure 6, the fine-tuning process using Bayesian optimization aims to efficiently identify the optimal hyperparameter configuration for our architectures based on transfer learning. For efficient optimization, the network is trained with a limited number of epochs while exploring various hyperparameter combinations within a pre-defined range. This approach prioritizes identifying the hyperparameter set that yields the best score on the validation of metric set.

3.6 DR Classification Using AtRD and AtR3C: Approach-based Transfer Learning

Transfer learning, unlike training from scratch, aims to transfer knowledge that has been learned from another data set to a target problem. In this study, we adopted ResNet50, a convolutional neural network pre-trained on the ImageNet dataset, as the backbone for feature extraction.

ResNet-50 is a specific variant of Residual Neural Networks (ResNets), developed by Kaiming He et al. in 2015 [53] for image recognition. It consists of 50 layers structured into convolutional layers and identity blocks. The key innovation of ResNet-50 lies in the use of residual connections, also known as skip connections (see Figure 7), which enable the network to bypass certain layers. This approach facilitates the training of very deep networks by mitigating the vanishing-gradient problem. ResNet-50 adopts an optimized architecture in which each residual block contains three convolutional layers (1×1, 3×3 and 1×1 convolutions) instead of the two used in earlier ResNet variants. The 1×1 convolutions serve to reduce and expand dimensionality, improving computational efficiency, while the 3×3 convolution captures spatial features. Several factors contribute to the model's success: its large receptive fields, which capture more contextual information for each pixel; the separation between localization and classification stages; its computational efficiency at deeper layers; and its effective encoding schemes that rely on low-complexity arithmetic operations.

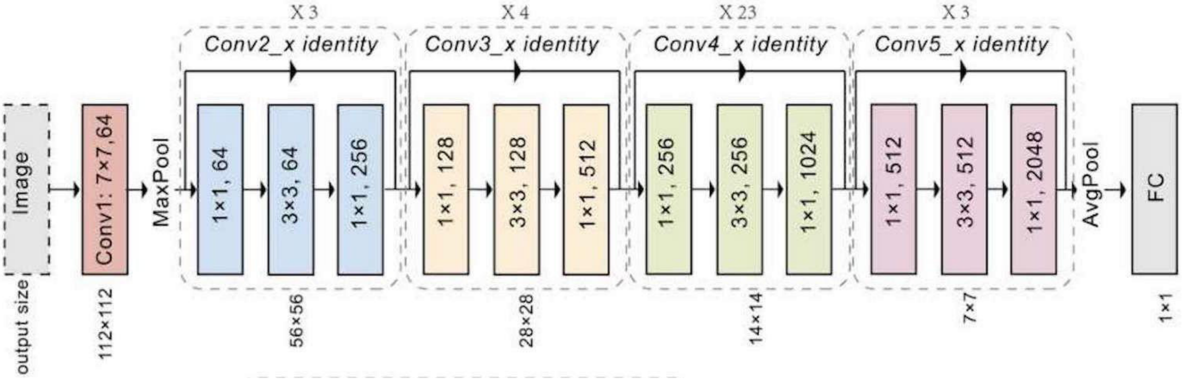


Figure 7. Resnet50 architecture [53].

While ResNet50 excels in general image classification, its final fully connected layer—originally configured for 1,000-class ImageNet classification—was unsuitable for our specialized binary and three-class DR classifications. In response, we designed the AtRD and AtR3C architectures, which retain the feature-extraction capabilities of ResNet50 while incorporating domain-specific adaptations. As illustrated in Figure 8, we replaced the final classification layer of ResNet50 with a customized multi-layer perceptron (MLP) comprising five additional layers (Flatten, Dense, Dropout, Dense, Dense). The final dense layer contains two nodes for binary classification or three nodes for 3-class classification.

3.7 DR Classification Using ViRD and ViR3C: Approach-based ViT

Taking advantage of ViT's ability to model long-range dependencies, we propose ViRD and ViR3C, two ViT-based architectures, for the detection and classification of DR. Figures 9 illustrates the proposed architecture.

The important components of the transformer are multi-head self-attention (MSA) and multi-layer perception (MLP). Multi-head attention in the Figure 10 is the core part of the Transformer. The ViT model considers an image submitted as a series of image patches.

Here are the key steps in its operation:

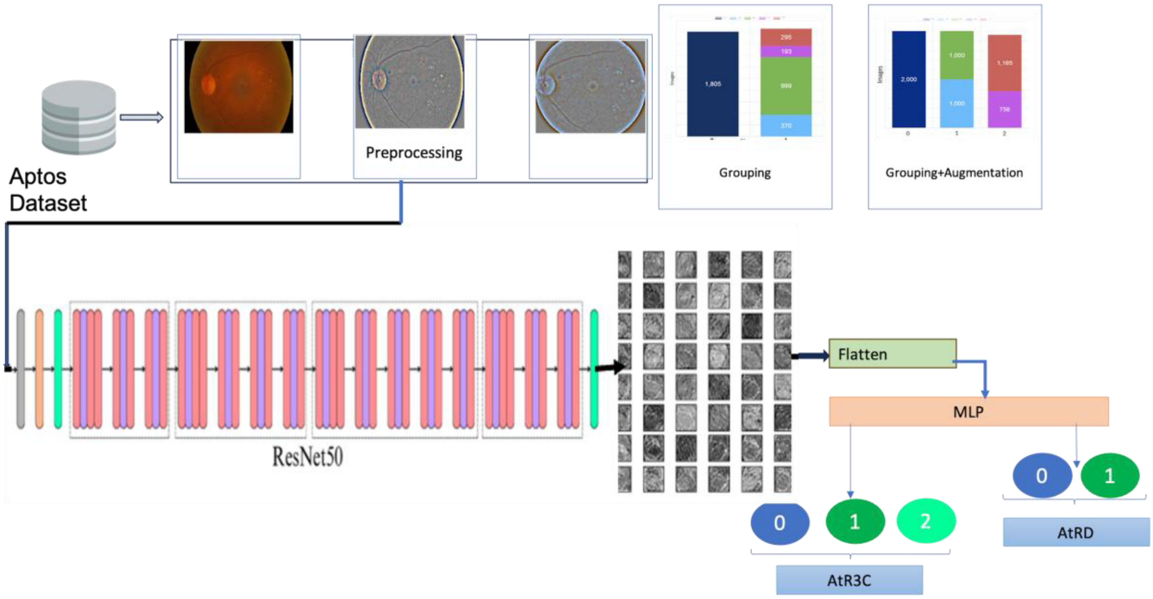


Figure 8. Proposed architecture-based ResNet50: AtRD and AtR3C.

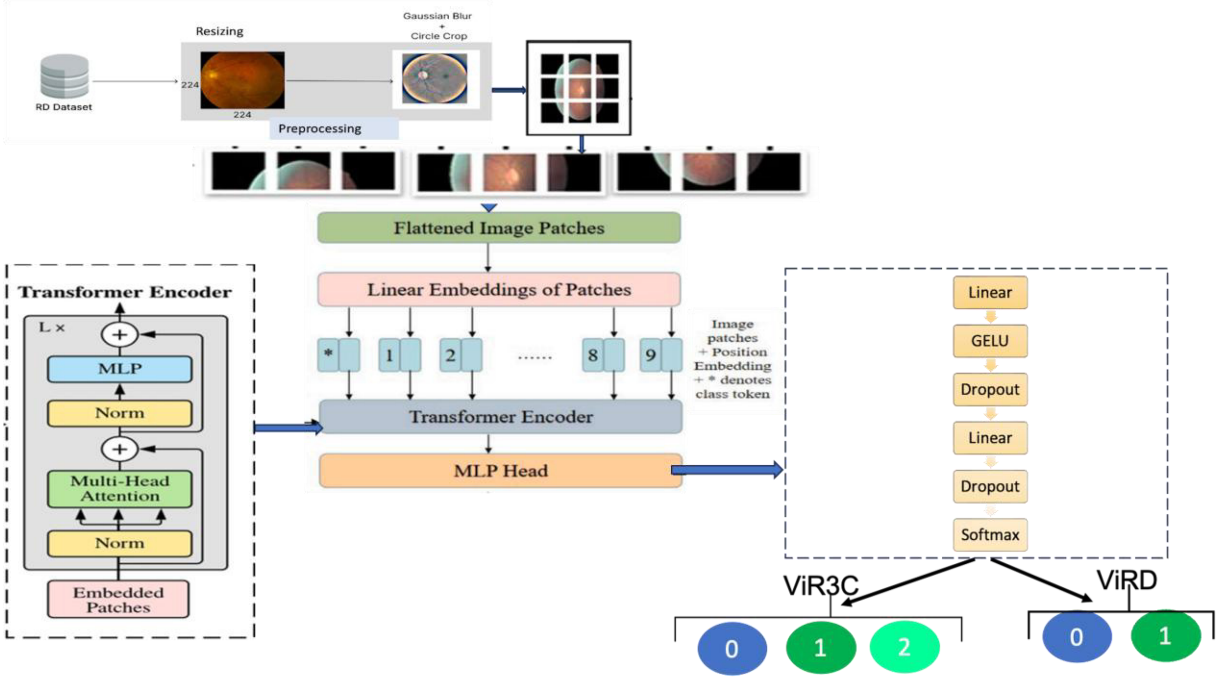


Figure 9. Proposed architecture-based ViT: ViRD and ViR3C.

Image Splitting into Patches: After pre-processing and resizing to 224*224, input picture I is divided into a series of flattened patches Xip (for i = 1, 2, ..., np), each with a size of p × p × C, C=3 corresponding to the three RGB channels in the image I; p = 16, resulting in np =(224 × 224/16*16)= 196 patches. Each patch Xip is flattened and transformed into a 1D vector X_0 of dimension pxpx3= 162x3=768 using linear embedding.

$$X_0 = [x_1, x_2, \dots, x] \in \mathbb{R}^{196\times(768)} \qquad (1)$$

Linear Projection of Patches (Patch Embedding): Each flattened patch is projected into a space of dimension *D* using a learnable matrix $E \in \mathbb{R}^{(768)\times D}$. For the *i*-th patch x_i, the embedding is given by $z_i = xi.$E. E represents the projection weight matrix, with dimensions 768×*D*, where 768 is the flattened patch dimension and *D* is the dimension of the projection space. D defines the dimension of the transformer's input tokens, which serve as the basis for self-attention mechanisms. In basic ViTs, D is commonly set to 768.

$$Z_0 = \left[z_1, z_2, \dots, z_{np}\right] \in \mathbb{R}^{196\times D} \qquad (2)$$

Class Token and Positional Embedding Initialization: As illustrated in Figure 9, the positional information **Pos** $\in \mathbb{R}^{196\times D}$ added into each embedded patch, allowing ViT to better understand the spatial relationships within the input data. The ViT model also incorporates a classification token (z[cls]) inside the embedded patches. This is a randomly initialized, learnable parameter used to aggregate global information for classification. It essentially acts as a decoder.

The input to the Transformer encoder is constructed as:

$$Z = [z[cls], z_0] \in \mathbb{R}^{(196+1\times D)} \quad (3)$$

After adding positional encoding, the final input to the encoder becomes:

$$Z_f = Z + POS \in \mathbb{R}^{(197\times D)} \quad (4)$$

The resulting embedding matrix Z_f, enriched with both visual and positional information, is then fed into a Transformer encoder stack.

Transformer Encoders: The Transformer Encoder is composed of two main layers: Multi-head Self-Attention (MSA) and Multi-layer Perceptron (MLP). The resulting embedding matrix, Z_f , is then fed into a stack of six Transformer encoder blocks. Each block consists of a multi-head self-attention (MSA) module with eight attention heads, followed by a multi-layer perceptron (MLP). Layer normalization and residual connections are applied before and after each sub-layer.

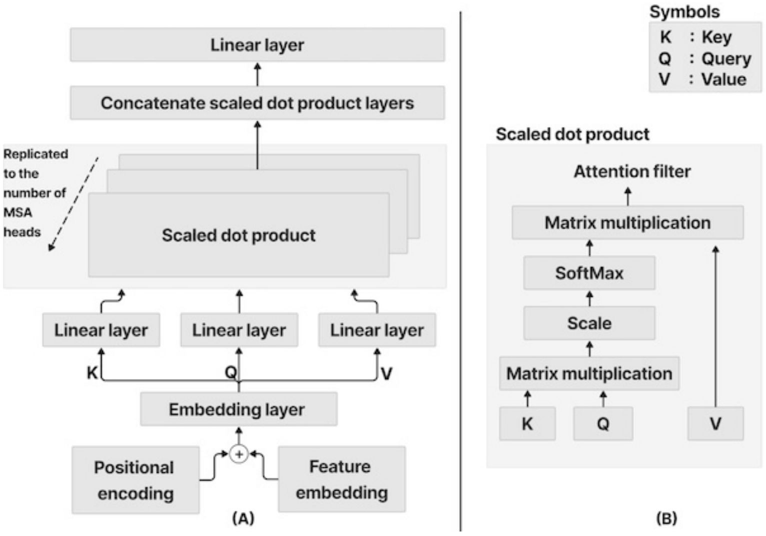


Figure 10. MSA process: (a) MSA process with several attention layers; (b) Scaled dot-product attention [38].

The multi-head attention mechanism (MSA) is a form of self-attention that allows the model to concentrate on information from different sub-spaces of representation at various positions. To calculate attention scores, MSA uses several scaled dot-product attention mechanisms, as shown in Figure 10. The complete MSA operation is summarized as:

$$MSA(Q, K, V) = Concat(h_1, h_2, \dots, h_n).W_0 \quad (5)$$

where *Concat* denotes the concatenation of all attention-head outputs; *n* is the number of attention heads. h_i is the output of the *i*-th self-attention head. The concatenated output is then projected back to the original embedding space using a final weight matrix W_0.

The output of each attention head h_i is computed as:

$$h_i = Attention(Q, K, V) = softmax(\frac{Q_i K_i^T}{\sqrt{d_k}})V_i \quad (6)$$

A softmax function is applied to derive the attention weights for the value matrices. This softmax operation normalizes the resulting scores, ensuring that they are positive and sum to unity. We then multiply the attention weights with value matrix (V_i) to get the self-attention output h_i.

The query Q_i, key K_i and value V_i vectors for each head ($i \in \{1, \dots, n\}$) are obtained by multiplying the input embedding matrix Z_f by three distinct weight matrices, effectively projecting the input embeddings into different representation sub-spaces for each attention head.

$$Q_i = Z_f W_i^Q \qquad K_i = Z_f W_i^K \qquad V_i = Z_f W_i^V$$

The outputs from all the heads are subsequently merged and forwarded to an MLP layer for further processing. Each MLP and MSA operation is preceded and followed by residual blocks and normalization layers to guarantee stability and model optimization. MLP comprises two fully-connected linear layers and between these layers, a non-linear activation function is applied. This function introduces non-linearity, allowing the model to learn more intricate patterns in the data. A common choice for this activation function in ViT is the Gaussian Error Linear Unit (GELU). GELU has a smoother, more continuous shape than the ReLU function, which can make it more effective at learning complex patterns in the data [38].

$$GeLU = 0.5.x + tanh\left[\sqrt{\frac{2}{\pi}}.(x + 0.0447x^2)\right] \tag{7}$$

We introduce two dropout layers to regularize the model and prevent overfitting. Finally, we extract the [Cls] token from the Transformer Encoder output and pass it through a classification head to obtain class predictions y. In order to classify DR into 2 or 3 severity stages, we use a head classification output layer composed of 2 or 3 neurons for ViRD and ViR3C, respectively. We applied a softmax function to get a probability distribution to classify fundus images over the two or three severity stages of DR (see Figure 9).

$$y = softmax(z[Cls]) \tag{8}$$

3.8 DR Classification Using ReVi-RD and ReVi-3C: A Novel Hybrid Approach

To enhance the precision of DR classification, we suggest a novel hybrid architecture that merges the benefits of Vision Transformers (ViTs) and Resnet50. Retinal-image features can be captured locally and globally by ReVi-RD and ReVi-3C models by integrating pre-trained ViRD/ViR3C with pre-trained AtRD/AtR3C models.

The hybrid approach is illustrated in Figure11. To construct this hybrid model, we use the weights of the pre-trained AtRD or AtR3C models to extract local features. We remove the MLP (final layers) of these models and replace it with the pre-trained ViRD or ViR3C, as described in Figure 12. In the following part, we describe our hybrid approach, illustrated in Figure11 and Figure 12, from input images to final classification.

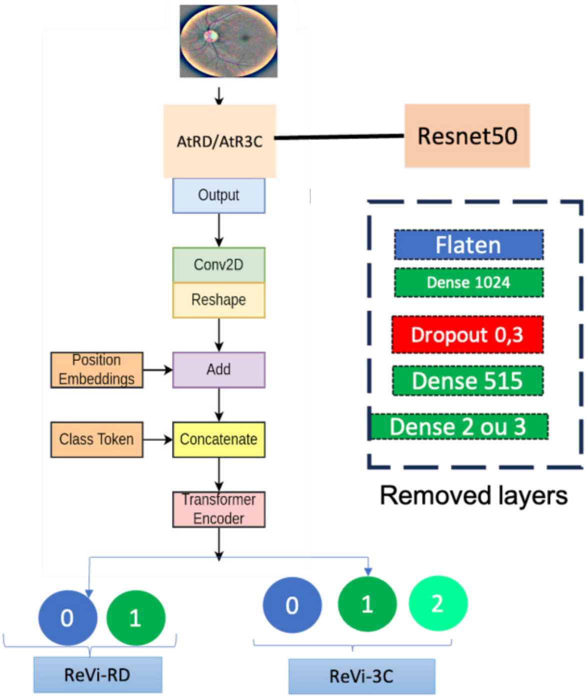


Figure 11. Hybrid architectures: ReVi-RD and ReVi-3C.

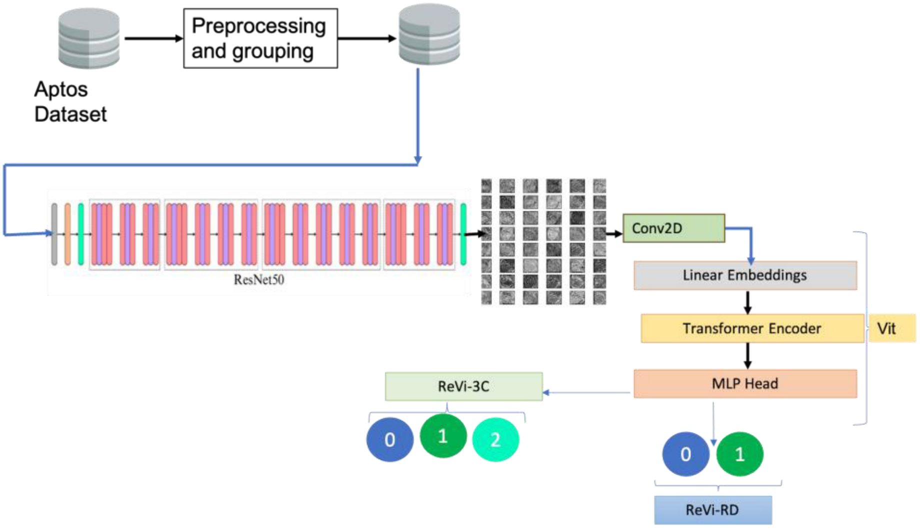


Figure 12. Detailed architecture of ReVi-RD and ReVi-3C.

- Input: an RGB image of 224×224 pixels, represented by a shape tensor [224, 224, 3], which is a standard input size for the ResNet50 model, is introduced into the pre-trained model.
- After pre-processing, AtrD/Atr3C are used to extract local spatial features from input images of size 224×224×3. The final classification layers of AtrD/Atr3C are removed and replaced with a transformer-based head.

  The output from an intermediate layer (specifically, the 7th layer from the end) of the modified ResNet50 model is extracted. The resulting feature map is 7×7×768 in size. This feature map retains high-dimensional representations of localized patterns while compressing spatial resolution to 7×7 grids, each with 768 channels.
- Reshaping for Vision Transformer (ViT): The resulting feature map of dimensions 7×7×768 is reshaped into a sequence of flattened patches, transforming the 7×7×768 feature map into a sequence of 49 tokens, each of 768 dimensions [49, 768]. Here, the 7×7 spatial grid is reinterpreted as 49 non-overlapping "patches", each represented as a 768-dimensional vector. This step adapts the output into a format compatible with transformer-based processing.
- Position Embedding and Class Token: To inject spatial information into the transformer, we add a learnable position embedding to the 49 patches, preserving their spatial relationships. Then, we concatenate a learnable [CLS] token (classification token) to the sequence, increasing its length to 50 ([50, 768]). A final sequence of length 50 is then processed by a Transformer Encoder.
- Transformer Encoder: the sequence of length 50 is fed through a series of 6 Transformer encoder blocks. Each block comprises a multi-head self-attention mechanism with 8 attention heads, followed by an MLP that includes layer normalization and residual connections.
- Classification Head: After the Transformer encoder, we performed a layer normalization and extracted the output corresponding to the class token. Then, we projected the final representation into the class space (2 for ReVi-RD or 3 For ReVi-3C) *via* a dense layer, yielding raw classification scores, which are then transformed into class probabilities using a softmax function.

4. Experimental Results

In this section, a detailed discussion of the experimental results obtained is carried out to prove the effectiveness of the Vits and hybrid models proposed for the classification of DR. The experiment was conducted using the Python environment on a server equipped with an Intel(R) Xeon(R) CPU @ 2.20GHz processor, 13 GB of RAM and a GPU P100 16GB provided by Kaggle platform. We use the Aptos dataset to train and test our architectures. To prevent data leakage, the dataset was explicitly split into two sub-sets with the ratio of 80:20 to make the training and testing datasets. Additionally, to address class imbalance, data augmentation was applied only to the training set, ensuring that artificially generated samples did not leak into validation or test sets.

The model underwent multiple independent trials, each with a unique random seed for dataset shuffling and partitioning. This approach introduced variability in data order and distribution across trials, enabling a thorough assessment of the model's stability.

For the ResNet50-based model, we used the Adam optimizer, while the ViT-based model utilized the AdamW optimizer. We employed categorical cross-entropy as the loss function, suiatable for our multi- class classification task with softmax activation. The learning rate was automatically selected through hyperparameter tuning and the optimal value obtained was 0.0001 for model based on Resnet50 and 0.00002 for model based on ViT. This value was fixed during training to ensure stable convergence.

4.1 Evaluation Metrics

To assess the detection performance of the proposed models, we use the most commonly used metrics: accuracy, precision, specificity or recall (sensitivity) and F1 score. Their mathematical expressions are given in Table 1. TP, TN, FP and FN are true positives, true negatives, false positives and false negatives, respectively.

4.2 Obtained Hyperparameters after Auto-tunning

After image pre-processing, we fine-tuned the architectures to get the best hyperparameters which are presented in Table 2 for AtRD and AtR3C, and in Table 3 for ViRD and ViR3C.

Table 1. Performance metrics.

Metrics	**Formula**
Accuracy (Acc)	$Acc = \frac{TP + TN}{TP + TN + FP + FN}$
Precision (Positive Predictive Value)	$Precision = \frac{TP}{TP + FP}$
Recall (Sensitivity)	$Recall = \frac{TP}{TP + FN}$
F1-score	$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}$
Specificity (True Negative Rate)	$Specificity = \frac{TN}{TN + FP}$

Table 2. Best hyperparameters obtained for AtRD and AtR3C.

Hyperparameter	**Value**
Image size	224x224
Batch size	32
Warmup epochs	5
Warmup learning rate	0.00001
Epochs	50
Learning rate	0.0001
Weight decay	0.02
Early stopping patience	15
Reduced LR patience	5
Regularizer	0.02

All the proposed architectures are trained using their obtained hyperparameters.

Their performance based on test data was evaluated using the five metrics: accuracy, precision, recall (sensitivity), F1-score and specificity.

4.3 Diabetic Retinopathy Detection Performance

As the first experiment, we compare the performance of AtRD, ViRD and ReVi-RD to evaluate their effectiveness in DR detection and assess the impact of the features extracted by each model. The results reported in Table 4 summarize the evaluation metrics obtained for detecting DR. We can notice that AtRD and ReViRD architectures demonstrate exceptional performance, exceeding 99% across all metrics (accuracy, precision, recall, F1-score), showcasing their robustness in DR detection. The exceptional performance of AtRD can be attributed to the efficient tuning of hyperparameters. The ViRD model achieves slightly lower, but still impressive results, surpassing 97.7% across all metrics. This disparity arises from the inherent data requirements of ViTs, which typically demand larger datasets to fully leverage their global attention mechanisms compared to transfer-learning models [37]. The hybrid ReViRD model outperforms both standalone architectures , underscoring the synergistic benefits of combining ResNet50's localized feature extraction with ViTs' ability to model long-range dependencies.

The detection performance of the AtRD, ViRD and hybrid ReVi-RD models is compared using their confusion matrices (see Figure 13) and the evaluation metrics summarized in Table 5. The AtRD model achieves high sensitivity in retinopathy detection (99.2% true positive rate), but exhibits a specificity of 96.81%, corresponding to a 3.2% false positive rate in healthy-patient classification. While this underscores its efficacy in identifying pathological cases, the elevated misdiagnosis rate for normal patients highlights limitations in distinguishing subtle non-pathological variations. In contrast, ViRD demonstrates balanced specificity (98.0% overall), with a slightly reduced 2.7% false negative rate for retinopathy cases. Although with an area under curve (AUC) of 99.1% (see Figure 14a), the ViT model is excellent at capturing global context through self-attention; it sometimes misses subtle local features that are critical for identifying retinopathy. This reliance on global context means that, in cases where pathological signs are very localized or subtle, the model might not sufficiently distinguish them from normal variations.

Table 3. Best hyperparameters obtained for the ViRD and Vi3C.

Parameter	Value
Image size	224x224
Batch size	16x16
Train batch size	32
Test batch size	64
Warmup steps	500
Warmup learning rate	0.00001
Epochs	20
Learning rate	0.00002
Weight decay	0.01

Table 4. Performance comparison of proposed models for DR detection (%).

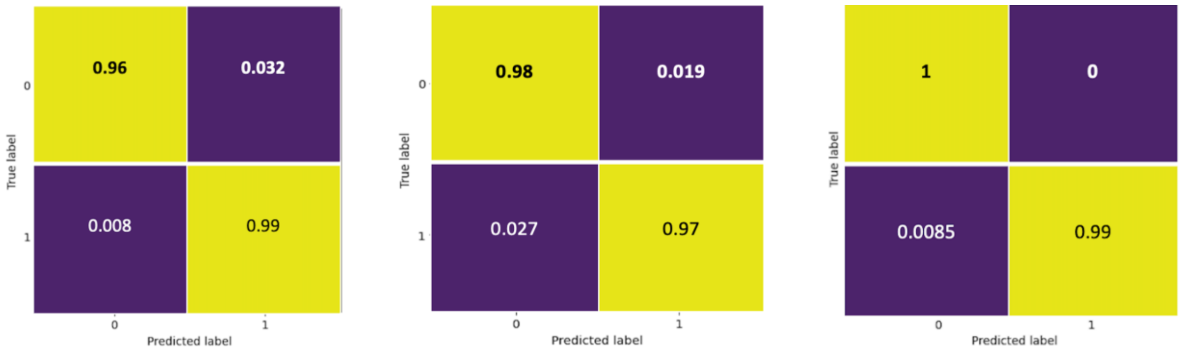
Metric	AtRD	ViRD	ReVi-RD
Accuracy (%)	99.22	97.73	99.55
Precision (%)	99.66	97.72	99.51
Recall (%)	99.23	97.73	99.58
F1-Score (%)	99.40	97.73	99.54
Specificity(Average) (%)	98.01	98.00	99.50

The hybrid ReVi-RD architecture addresses these limitations by synergistically combining CNN-driven local feature extraction (AtRD) and ViT-based global dependency modeling (ViRD). This integration achieves near-perfect classification: a 1.0% false negative rate for retinopathy and 0.0% false positives rate for healthy cases (Table 4). With a specificity of 99.50%, ReVi-RD minimizes unnecessary diagnoses while maintaining exceptional sensitivity, outperforming both AtRD (98.01%) and ViRD (98.00%) in robustness. Class-specific metrics (Table 5) further elucidate these distinctions. AtRD shows moderate precision-recall harmonization (F1-scores: 97.7% for both classes), constrained by CNN architectures' focus on localized textures rather than on lesion correlations. ViRD improves balance, achieving 98.00% F1-scores for both classes *via* global attention, yet remains vulnerable to localized oversights. ReVi-RD's hybrid design transcends these trade-offs, leveraging CNN-localized granularity and ViT-global context to optimize feature representation. This dual capability enables superior accuracy in diabetic-retinopathy classification, particularly for cases requiring simultaneous fine-grained and global analysis.

The hybrid ReVi-RD resolves residual trade-offs, achieving near-perfect metrics (100% F1-score for both classes, 99–100% precision/recall and 99.9% AUC, as shown in Figure 15a). Its dominance stems from synergizing AtRD localized feature extraction with ViRD global-context modeling, effectively eliminating misclassifications (only 0.85% of non-healthy cases mislabeled). For clinical deployment, ViRD's standalone performance—particularly its precision gains for critical non-healthy cases—validates ViTs as an important tool for severity staging, while ReVi-RD's hybrid architecture sets a new benchmark for applications requiring ultra-reliable classification. These results emphasize the necessity of integrating CNNs and ViTs in medical imaging, where both local granularity and global coherence are essential for accurate, interpretable diagnoses.

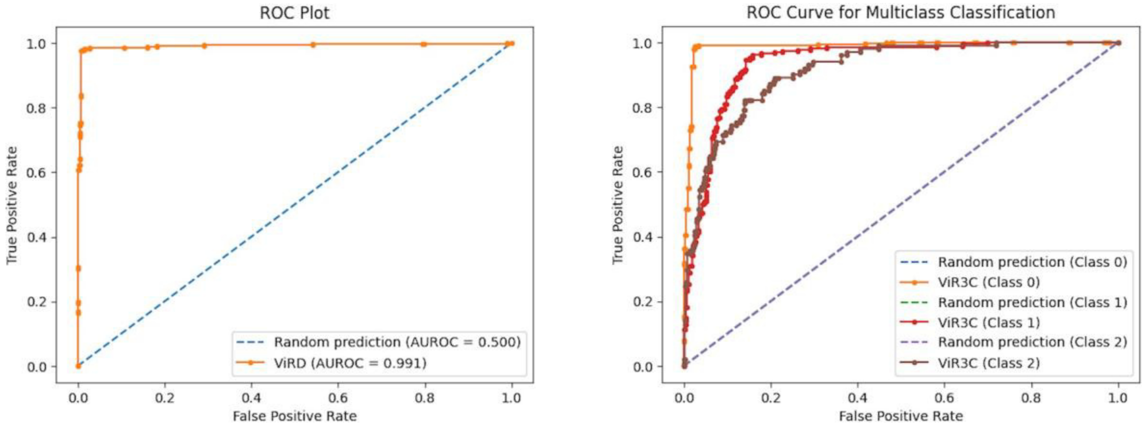
Table 5. Class-wise performance of proposed models for DR detection (%).

	AtRD		**ViRD**		**ReVi-RD**	
Metrics	**Class 0**	**Class 1**	**Class 0**	**Class1**	**Class 0**	**Class1**
Precision (%)	97.60	97.90	97.00	98.00	99.00	100.00
Recall (%)	97.90	97.60	98.00	97.00	100.00	99.00
F1-score (%)	97.70	97.70	98.00	98.00	100.00	100.00
Specificity (%)	99.21	96.81	98.00	98.00	100.00	99.00



(a) (b) (c)

Figure 13. The confusion matrices: (a) AtRD, (b) ViRD and (c) ReVi-RD.



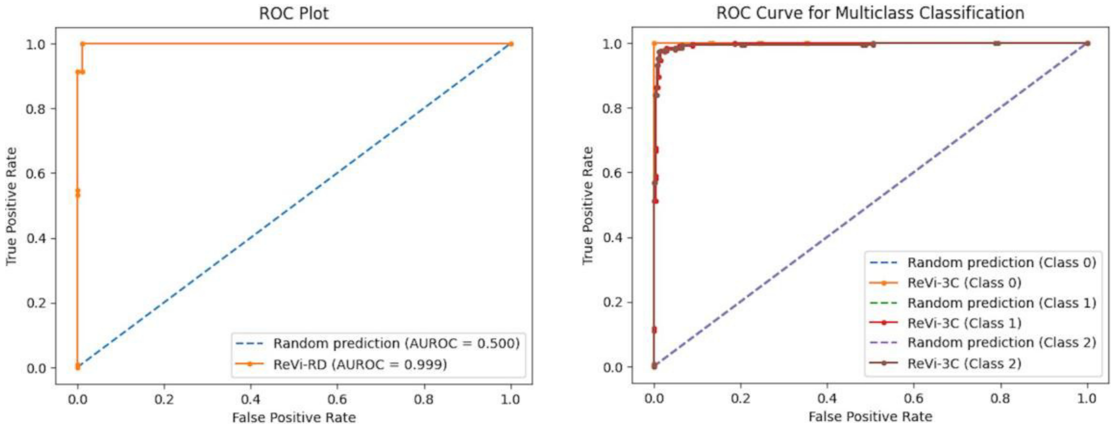
(a) (b)

Figure 14. ROC curve for (a) ViRD and (b) ViR3C.

4.4 Diabetic Retinopathy Classification Performance

In the following experiment, we test the generalization capacity of the suggested models for the difficult task of classifying data into three different stages of severity in order to evaluate its potential.

Table 6 summarizes the evaluation metrics for staging RD into 3 classes. AtR3C and ViR3C offers a well-balanced performance across precision, recall and F1-score, as well as about 94% and 93% across all metrics, respectively. ReVi-3C produced remarkable results, achieving an average of nearly 98% across all metrics and classes, including an area under the curve (AUC) of 99% per class, as shown in Figure 15b. This indicates that the model's predictions are balanced and reliable across the different performance measures.



(a) (b)

Figure 15. ROC curve for (a) ReVi-RD and (b) ReVi-3C.

Table 6. Performance evaluation of proposed models for 3-class DR classification (%).

Metric	**AtR3C**	**ViR3C**	**ReVi-3C**
Accuracy (%)	94.26	92.97	98.26
Precision (%)	94.41	93.77	98.43
Recall (%)	94.09	93.22	98.21
F1-score (%)	94.24	93.46	98.32
Specificity (Average) (%)	93.70	96.60	98.67

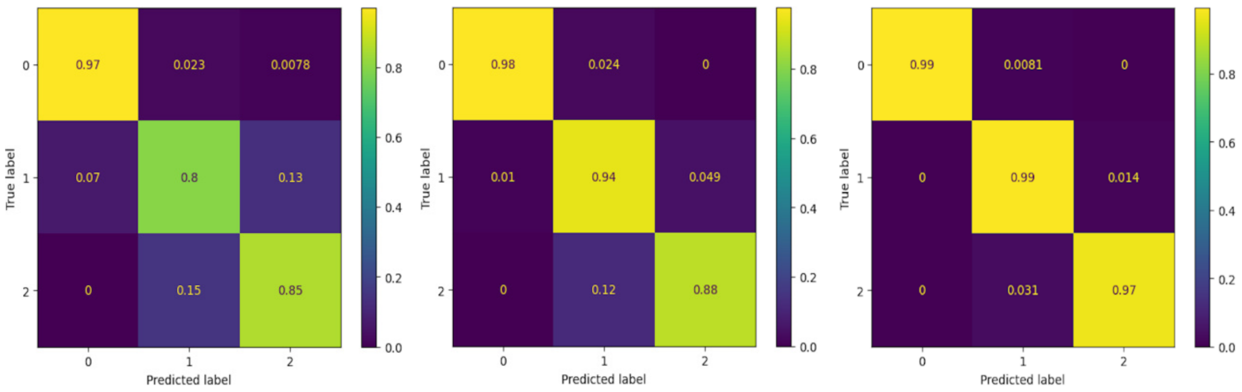
In order to evaluate the effectiveness of the suggested models (AtR3C, ViR3C and ReVi-3C), we examined the confusion matrices (see Figure 16), to provide details on the distribution of errors and classification accuracy across the three severity classes. As illustrated in Table 7, AtR3C model excels at identifying class 0 cases, achieving a precision of 97%, which means that nearly all predictions for this category are accurate. However, a specificity of 91.40% indicates that the model encounters difficulties with class 1. Specifically, 13% of cases are mislabeled as class 2 and 7% are incorrectly classified as class 0. Similarly, 15% of class 2 cases are mistakenly assigned to class 1. These patterns reveal a critical limitation: the model struggles to differentiate between adjacent severity levels, particularly distinguishing class 1 (moderate severity) from class 2 (high severity). This confusion suggests that AtR3C may lack the nuance needed to separate closely related categories, a gap that could impact its reliability in scenarios requiring precise severity staging. On the other hand, the ROC-curve in Figure 14b corresponding to class 0 lies very close to the top-left corner of the plot. This indicates that ViR3C is very accurate at detecting patients without DR.

The model demonstrated exceptional specificity of 99.5% for class 0 (healthy patients), minimizing false positives (0.5%) and thus avoiding misdiagnosis in unaffected individuals, which is essential for reliable screening. For class 1, specificity reached 92.8%, with 7.2% false positives, reflecting moderate difficulty in isolating this intermediate category. In contrast, class 2 (severe stage) has a high specificity of 97.5%, drastically limiting critical over-diagnosis and avoiding unwarranted invasive treatment.

For unhealthy cases, early-stage DR (class 1) is correctly identified in 94% of instances, though a 1% misclassification as healthy poses a risk of missed diagnoses, while advanced-stage DR (class 2) shows 88% accuracy, with 12% confused as early-stage DR, but none misclassified as healthy, highlighting robust performance for severe cases, but some overlap in staging severity. These results highlight the model's potential for accurately diagnosing early-stage DR and shows that the misclassification error mainly concerns stages 1 and 2.

Compared to AtR3C, ViR3C enhances the detection of healthy cases by reducing the misclassification rate of healthy individuals as non-healthy from 3% with AtR3C to 2% with ViR3C. This improvement highlights the power of ViTs in better detecting primitives across the entire set of images. We can decrease the errors by combining the strengths of the two architectures.

The hybrid ReVi-3C model dramatically outperforms its predecessors, AtR3C and ViR3C, achieving near-flawless classification across all severity levels: 99% precision for class 0 and class 1 and 97% for class 2, marking a substantial leap in accuracy. Misclassification errors are reduced to negligible levels, with only 3% of class 2 cases mistakenly labeled as class 1, while confusion between class 0 and class 1 is virtually eliminated. These results highlight the critical role of hybrid architectures in addressing multi-class challenges, where subtle inter-class differences demand precise discrimination.



(a) (b) (c)

Figure 16. The confusion matrix: (a) AtR3C, (b) ViR3C and (c) ReVi-3C.

Table 7. Class-wise performance of proposed models for 3-class DR classification (%).

	AtR3C			ViR3C			REVi-3C		
Metrics	**Class 0**	**Class 1**	**Class 2**	**Class 0**	**Class 1**	**Class 2**	**Class 0**	**Class 1**	**Class 2**
Precision (%)	98.00	82.00	86.00	98.00	91.00	93.00	100.00	98.00	98.00
Recall (%)	97.00	80.00	85.00	98.00	94.00	88.00	99.00	99.00	97.00
F1-score (%)	95.00	81.00	85.00	98.00	92.00	90.00	100.00	98.00	97.00
Specificity (%)	96.50	91.40	93.20	99.5	92.80	97.50	100.00	97.00	99.00

4.5 Results' Conclusion

The results obtained and their subsequent interpretation demonstrate that the proposed hybrid architectures (Revi-RD and Revi-3c) achieved remarkably high performance in both sensitivity and specificity. This success can be attributed to the effective exploitation of the complementary strengths of local feature extraction (by Resnet50) and global modeling of spatial dependencies (by ViTs).

5. Comparison of Our Approaches with the State-of-the-Art

To benchmark our approach, we compared our results with those of other state-of-the-art methods that have utilized transfer learning on the APTOS dataset for DR severity-level classification. Our models were benchmarked against Convolutional Neural Networks (CNNs) [32], [54], ensemble transfer learning [55], Supervised Contrastive Learning [34], a Deep Dual Branch model [56], Swin Transformer [50] and hybrid models combining Multiple Instance Vision Transformer (Milv4) [49] and Vision Transformer with Inception [47]. The comparison is carried out utilizing performance parameters including accuracy, precision, recall or sensitivity and F1-score across both binary and three-class classification tasks. All the methods illustrated in Table 8 are explained in the Related Works section. We can clearly say that our results are better and more enhanced than state-of-the-art results.

- **2-stage Classification**

AtRD model delivers a balanced performance (99.22% accuracy, 99.60% precision, 99.41% F1-score) surpassing recent models, such as those of Shakibania et al. [56]. (98.50% accuracy) and Islam et al. [34] (98.36%). Athira et al. [55] achieved a slightly higher accuracy of 99.80%, as they also used an ensemble deep-learning approach with auto-tuning, but did not provide an F1-scor. In comparison, AtRD (99.22%) and ReVi-RD (99.55%) surpass nearly all previous works. However, the hybrid ReVi-RD model, with 99.55% accuracy, 99.51% precision and 99.54% F1-score, outperforms all existing approaches.

- **3-stage Classification**

AtR3C model did well in the 3-class-classification test, achieving accuracy, recall and F1-score values of 94.41%, 94.09% and 94.24%, respectively. Our results are somewhat superior to those of Athira et al. [55], who reported a slightly lower F1-score of 93.00%, but attained precision and recall of 94.00% each, noting that Athira did not report class performance. On the other hand, ViR3C attains an F1-score of 93.46%, demonstrating the potential of Vision Transformers (ViTs) in DR classification, though these models require more data than CNNs based on transfer learning. ReVi-3C, a hybrid architecture, achieves an impressive F1-score of 98.32%, representing an absolute improvement of 10.3% over Rao et al. and a 5.1% gain over Athira et al. This significant performance boost validates the effectiveness of hybrid models, where CNNs excel in localized feature extraction, while ViTs capture global contextual patterns. The importance of our method is underscored by the lack of research on the three-class classification of diabetic retinopathy (DR). Revi-3C's encouraging performance highlights its potential for DR detection, especially in its early stages, leading to better diagnostic results.

Table 8. Comparison of the proposed approaches with relevant previous works: binary and 3-stage classifications (unit %).

Architecture	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Binary classification				
Esfahani [54]	86.00	85.00	86.00	85.00
Rao et al. [32]	96.56	97.00	97.00	96.56
Islam et al. [34]	98.36	98.37	98.36	98.37
Athira et al. [55]	99.80	99.00	99.00	99.00
Shakibania et al. [56]	98.50	97.61	99.46	/
Our AtRD	99.22	99.60	99.23	99.41
Dihin et al. [50]	96.00	/	98.00	96.00
Yang et al. [48]	93.2	/	86.9	/
Lian and Liul [47]	95.3	/	94.2	/
Our ViRD	97.73	97.72	97.73	97.73
Our ReVi-RD	**99.55**	99.51	**99.58**	**99.54**
3-class Classification				
Rao et al. [32]	/	88.00	88.00	88.02
Athira et al. [55]	94.00	94.00	93.00	
Our AtR3C	94.26	94.41	94.09	94.24
Our ViR3C	92.97	93.77	93.22	93.46
Our ReVi-3C	98.26	98.431	98.21	98.32

6. CONCLUSION

This study highlights the potential of Vision Transformers (ViTs) and hybrid architectures in advancing diabetic retinopathy (DR) classification, particularly for early detection. By simplifying the traditional five-stage DR classification into three classes—no DR, early DR (mild/moderate) and advanced DR (severe/proliferative), we reduced ambiguity between adjacent stages. To this end, we proposed three architectures: (1) a Resnet50-based model with Bayesian hyperparameter optimization (AtRD, AtR3C), (2) a fine-tuned Vision Transformer model (ViRD, ViR3C) and (3) a hybrid architecture (ReVi-RD, ReVi-3C) that combines the strengths of both approaches. Experimental results show that while our architecture-based ViTs improve class differentiation, our hybrid model achieves superior accuracy and precision, demonstrating the advantage of integrating both local feature extraction and global attention mechanisms. This impressive result points to a high potential for accurate DR detection, which might greatly improve early diagnosis and care. However, several limitations should be noted. The use of the APTOS dataset alone for model training and evaluation may not fully represent the variety of fundus images encountered in real clinical settings. Consequently, it remains to generalize the models by training and evaluating on diverse datasets. Furthermore, the work does not fully address the difficulties of interpreting the models. It is essential to develop methods that enable clinicians to understand and trust the decisions made by the model. For future work, we aim to extend our model to five-stage DR classification to align with standard clinical grading. Additionally, we plan to enhance generalization by training and evaluating on diverse datasets, ensuring robustness across different populations and imaging conditions. Furthermore, we will investigate how to apply explainable AI approaches to improve the clarity of our model and encourage its application in medical environments.

ACKNOWLEDGEMENTS

This work was sponsored by the General Direction of Scientific Research and Technological Development, Ministry of Higher Education and Scientific Research (DGRSDT), Algeria.

REFERENCES

[1] E. Mehmet et al., "Diabetes Mellitus: A Review on Pathophysiology, Current Status of Oral Medications and Future Perspectives," Acta Pharmaceutica Sciencia, vol. 55, no. 1, pp. 61–82, 2017.

[2] J. Gu et al., "Recent Advances in Convolutional Neural Networks," Pattern Recognition, vol. 77, pp. 354–377, 2018.

[3] T. H. Fung et al., "Diabetic Retinopathy for the Non-ophthalmologist," Clinical Medicine, vol. 22, no. 2, pp. 112–116, 2022.

[4] D. J. Magliano et al., IDF Diabetes Atlas, 10th Edition, ISBN-13: 978-2-930229-98-0, 2022.

[5] F. Shaheen, B. Verma and M. Asafuddoula, "Impact of Automatic Feature Extraction in Deep Learning Architecture," Proc. of the 2016 IEEE Int. Conf. on Digital Image Computing: Techniques and Applications (DICTA), pp. 1–8, Gold Coast, Australia, 2016.

[6] R. Adriman, K. Muchtar and N. Maulina, "Performance Evaluation of Binary Classification of Diabetic Retinopathy through Deep Learning Techniques Using Texture Feature," Procedia Computer Science, vol. 179, pp. 88–94, 2021.

[7] R. Rajkumar et al., "Transfer Learning Approach for Diabetic Retinopathy Detection Using Residual Network," Proc. of the 2021 6th IEEE Int. Conf. on Inventive Computation Technologies (ICICT), pp. 1189–1193, Coimbatore, India, 2021.

[8] S. Karthika et al., "Enhancing Diabetic Retinopathy Diagnosis with ResNet-50-based Transfer Learning: A Promising Approach," Annals of Data Science, vol. 11, no. 1, pp. 1–24, 2024.

[9] L. Dai et al., "A Deep Learning System for Detecting Diabetic Retinopathy across the Disease Spectrum," Nature Communications, vol. 12, no. 1, p. 3242, 2021.

[10] B. Tymchenko, P. Marchenko and D. Spodarets, "Deep Learning Approach to Diabetic Retinopathy Detection," arXiv preprint, arXiv: 2003.02261, 2020.

[11] P. Vashist et al., "Role of Early Screening for Diabetic Retinopathy in Patients with Diabetes Mellitus: An Overview," Indian Journal of Community Medicine, vol. 36, no. 4, pp. 247–252, 2011.

[12] K. Aggarwal et al., "Has the Future Started? The Current Growth of Artificial Intelligence, Machine Learning and Deep Learning," Iraqi J. for Comp. Sci. and Math., vol. 3, no. 1, pp. 115– 123, 2022.

[13] M. Bader Alazzam, F. Alassery and A. Almulihi, "Identification of Diabetic Retinopathy through Machine Learning," Mobile Information Systems, vol. 2021, no. 1, pp. 1–8, 2021.

[14] C. Mohanty et al., "Using Deep Learning Architectures for Detection and Classification of Diabetic Retinopathy," Sensors, vol. 23, no. 12, p. 5726, 2023.

[15] C. Sharma and S. Parikh, "Comparison of CNN and Pre-trained Models: A Study," [Online], Available: https://www.researchgate.net/publication/359850786_Comparison_of_CNN_and_Pre-trained_models_A_Study, 2022.

[16] S. R. Salian and S. D. Sawarkar, "Melanoma Skin Lesion Classification Using Improved Efficientnetb3," Jordanian J. of Computers and Inform. Technol. (JJCIT), vol. 8, no. 1, pp. 45-56, 2022.

[17] I. Khoulqi and N. Idrissi, "Cervical Cancer Detection and Classification Using MRIS," Jordanian J. of Computers and Inform. Technol. (JJCIT), vol. 8, no. 2, pp. 141 – 158, 2022.

[18] I. Kandel and M. Castelli, "Transfer Learning with Convolutional Neural Networks for Diabetic Retinopathy Image Classification. A Review," Applied Sciences, vol. 10, no. 6, 2021.

[19] G. Selvachandran et al., "Developments in the Detection of Diabetic Retinopathy: A State-of-the-Art Review of Computer-aided Diagnosis and Machine Learning Methods," Artificial Intelligence Review, vol. 56, no. 2, pp. 915–964, 2023.

[20] S. D. Karthik, Maggie, "Aptos 2019 Blindness Detection," 2019.

[21] R. Casanova et al., "Application of Random Forests Methods to Diabetic Retinopathy Classification Analyses," PLOS One, vol. 9, no. 6, p. e98587, 2014.

[22] T. M. Usman et al., "A Systematic Literature Review of Machine Learning-based Risk Prediction Models for Diabetic Retinopathy Progression," Artificial Intell. in Medicine, vol. 143, p. 102617, 2023.

[23] W. L. Alyoubi et al., "Diabetic Retinopathy Detection through Deep Learning Techniques: A Review," Informatics in Medicine Unlocked, vol. 20, p. 100377, 2020.

[24] S. Sengupta et al., "Ophthalmic Diagnosis Using Deep Learning with Fundus Images: A Critical Review," Artificial Intelligence in Medicine, vol. 102, p. 101758, 2020.

[25] H. Jiang et al., "Eye Tracking-based Deep Learning Analysis for the Early Detection of Diabetic Retinopathy: A Pilot Study," Biomedical Signal Processing and Control, vol. 84, p. 104830, 2023.

[26] R. Vij and S. Arora, "A Novel Deep Transfer Learning Based Computerized Diagnostic Systems for Multi-class Imbalanced Diabetic Retinopathy Severity Classification," Multimedia Tools and Applications, vol. 82, no. 22, pp. 34847–34884, 2023.

[27] P. Bijam and S. Deshmukh, "A Review on Detection of Diabetic Retinopathy Using Deep Learning and Transfer Learning-based Strategies," Int. Journal of Computer (IJC), vol. 45, no. 1, pp. 164–175, 2023.

[28] S. Z. Beevi, "Multi-level Severity Classification for Diabetic Retinopathy Based on Hybrid Optimization Enabled Deep Learning," Biomed. Signal Process. and Control, vol. 84, p. 104736, 2023.

[29] Z. Gu et al., "Classification of Diabetic Retinopathy Severity in Fundus Images Using the Vision Transformer and Residual Attention," Comput. Intell. and Neurosci., vol. 2023, no. 1, p.1305583, 2023.

[30] H. E. Kim et al., "Transfer Learning for Medical Image Classification: A Literature Review," BMC Medical Imaging, vol. 22, no. 1, p. 69, 2022.

[31] O. Dekhil et al., "Deep Learning-based Method for Computer Aided Diagnosis of Diabetic Retinopathy," Proc. of the 2019 IEEE Int. Conf. on Imaging Systems and Techniques (IST), pp. 1–4, Abu Dhabi, UAE, 2019.

[32] M. Rao, M. Zhu and T. Wang, "Conversion and Implementation of State-of-the-Art Deep Learning Algorithms for the Classification of Diabetic Retinopathy," arXiv preprint, arXiv: 2010.11692, 2020.

[33] A. K. Gangwar and V. Ravi, "Diabetic Retinopathy Detection Using Transfer Learning and Deep Learning," Proc. of Evolution in Computational Intelligence: Frontiers in Intelligent Computing: Theory and Applications (FICTA 2020), vol. 1, pp. 679–689, 2021.

[34] M. R. Islam et al., "Applying Supervised Contrastive Learning for the Detection of Diabetic Retinopathy and Its Severity Levels from Fundus Images," Computers in Biology and Medicine, vol. 146, p. 105602, 2022.

[35] M. Oulhadj et al., "Diabetic Retinopathy Prediction Based on Deep Learning and Deformable Registration," Multimedia Tools and Applications, vol. 81, no. 20, pp. 28709–28727, 2022.

[36] S. S. Mondal et al., "EDLDR: An Ensemble Deep Learning Technique for Detection and Classification of Diabetic Retinopathy," Diagnostics, vol. 13, no. 1, p. 124, 2022.

[37] A. Dosovitskiy et al., "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale," arXiv preprint, arXiv: 2010.11929, 2020.

[38] A. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems, vol. 30, no. 1, pp. 261–272, 2017.

[39] J. Chen et al., "TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation," arXiv preprint, arXiv: 2102.04306, 2021.

[40] X. Wang et al., "Transpath: Transformer-based Self-supervised Learning for Histopathological Image Classification," Proc. of 24th Int. Conf. on Medical Image Computing and Computer Assisted Intervention (MICCAI 2021), pp. 186–195, Part VIII 24, Strasbourg, France, 2021.

[41] J. Wu, R. Hu, Z. Xiao, J. Chen and J. Liu, "Vision Transformer-based Recognition of Diabetic Retinopathy Grade," Medical Physics, vol. 48, no. 12, pp. 7850–7863, 2021.

[42] N. J. Mohan, R. Murugan, T. Goel and P. Roy, "Vit-DR: Vision Transformers in Diabetic Retinopathy Grading Using Fundus Images," Proc. of the 2022 IEEE 10th Region 10 Humanitarian Technology Conf. (R10-HTC), pp. 167–172, Hyderabad, India, 2022.

[43] W. Nazih et al., "Vision Transformer Model for Predicting the Severity of Diabetic Retinopathy in Fundus Photography-based Retina Images," IEEE Access, vol. 11, pp. 117546–117561, 2023.

[44] I. U. Khan et al., "A Computer-aided Diagnostic System to Identify Diabetic Retinopathy Utilizing a Modified Compact Convolutional Transformer and Low-resolution Images to Reduce Computation Time," Biomedicines, vol. 11, no. 6, p. 1566, 2023.

[45] T. Karkera et al., "Detecting Severity of Diabetic Retinopathy from Fundus Images: A Transformer Network-based Review," Neurocomputing, vol. 597, p. 127991, 2024.

[46] M. Oulhadj et al., "Diabetic Retinopathy Prediction Based on Vision Transformer and Modified

Capsule Network," Computers in Biology and Medicine, vol. 175, p. 108523, 2024.

[47] J. Lian and T. Liu, "Lesion Identification in Fundus Images *via* Convolutional Neural Network-vision Transformer," Biomedical Signal Processing and Control, vol. 88, p. 105607, 2024.

[48] Y. Yang, Z. Cai, S. Qiu and P. Xu, "A Novel Transformer Model with Multiple Instance Learning for Diabetic Retinopathy Classification," IEEE Access, vol. 12, pp. 6768 - 6776 2024.

[49] S. Yu et al., "Mil-vt: Multiple Instance Learning Enhanced Vision Transformer for Fundus Image Classification," Proc. of the 24th Int. Conf. on Medical Image Computing and Computer Assisted Intervention (MICCAI 2021), pp. 45–54, Part VIII 24, Strasbourg, France, 2021.

[50] R. A. Dihin et al., "Diabetic Retinopathy Classification Using Swin Transformer with Multi Wavelet," Journal of Kufa for Mathematics and Computer, vol. 10, no. 2, pp. 167–172, 2023.

[51] S. V. M. Sagheer and S. N. George, "A Review on Medical Image Denoising Algorithms," Biomedical Signal Processing and Control, vol. 61, p. 102036, 2020.

[52] S. H. Abbood et al., "Hybrid Retinal Image Enhancement Algorithm for Diabetic Retinopathy Diagnostic Using Deep Learning Model," IEEE Access, vol. 10, pp. 73079–73086, 2022.

[53] K. He et al., "Deep Residual Learning for Image Recognition," Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, DOI 10.1109/CVPR.2016.90, 2016.

[54] M. T. Esfahani et al., "Classification of Diabetic and Normal Fundus Images Using a New Deep Learning Method," Leonardo Electronic J. of Practices and Techn., vol. 17, no. 32, pp. 233–248, 2018.

[55] T. Athira and J. J. Nair, "Diabetic Retinopathy Grading from Color Fundus Images: An Autotuned Deep Learning Approach," Procedia Computer Science, vol. 218, pp. 1055–1066, 2023.

[56] H. Shakibania et al., "Dual Branch Deep Learning Network for Detection and Stage Grading of Diabetic Retinopathy," Biomedical Signal Processing and Control, vol. 93, p. 106168, 2024.

ملخص البحث:

تعمل هذه الورقة على تقييم محوّلات الرّؤية وبِنى التّعلُّم العميق الهجينة من أجل تصنيف اعتلال الشّبكية لدى المصابين بمرض السُّكَّري، لمعالجة الغموض الّذي يكتنف التّمييز بين المراحل في الأنظمة التّقليدية. ففي حين تتفوّق الشّبكات العصبية الالتفافية في استخلاص السِّمات المحلّية في صوَر الشّبكية، فإنّ محوّلات الرّؤية توفر نمذجةً عالمية مثالية. وللاستفادة من نقاط القوّة تلك، نقترح بِنيةً هجينةً تجمع بين الشّبكات العصبية الالتفافية ومحوّلات الرّؤية بهدف التّصنيف الدّقيق لاعتلال الشّبكية لدى مرضى السُّكَّري. وقد قمنا ببناء ثلاثة نماذج لهذه الغاية وتقييمها: الأول يرتكز على الشّبكات العصبية الالتفافية وحدها، والثّاني يستند إلى محوّلات الرّؤية وحدها، والثالث نموذج هجين يجمع بينهما.

ولتقليل الغموض في التّمييز بين المراحل، قمنا بتقليل عدد مراحل اعتلال الشّبكية من خمس مراحل إلى ثلاث: لا اعتلال، واعتلال مبكّر (خفيف ومتوسط)، واعتلال متقدّم (شديد وقابل للانتشار). وقد تم تدريب النّماذج وتقييمها باستخدام مجموعة البيانات أبتوس (APTOS). ولدى مقارنة نتائج تقييم النّماذج الثّلاثة، تبين أنّ النّموذج الهجين يتفوق على النّموذجين الآخرين، محققاً دقّة تصنيف بلغت فيٍ معدّلها 98% في جميع أصناف الاعتلال الثلاثة. وقد حقّق النّموذج الهجين قيمةً ممتازة بلغت 99.5% في جميع مؤشّرات الأداء المتعلّقة بالتّصنيف الثّنائي (عدم وجود اعتلال؛ وجود اعتلال)، بينما بلغت تلك القيمة 98.3% في مؤشّرات الأداء المرتبطة بالتّصنيف الثّلاثي (لا اعتلال؛ اعتلال مبكّر؛ اعتلال متقدم). وخلاصة القول هي أنّ النّظام الهجين المقترح لتصنيف وجود أو عدم وجود اعتلال في الشّبكية لدى مرضى السُّكّري وتحديد درجة ذلك الاعتلال –إنْ وجد- تفوّق من حيث الأداء على الأنظمة التّقليدية الّتي تستخدم الشّبكات العصبية الالتفافية وتلك الّتي تستخدم طرقاً أخرى، حيث يعمل النّظام الهجين المقترح على التّقليل من اللّبْس في التّصنيف بين مراحل اعتلال الشّبكية، وذلك يبين أنّ لديه القدرة على التّصنيف الدّقيق للمراحل المختلفة لذلك الاعتلال.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 03, September 2025.

ON THE OPTIMIZATION OF UAV SWARM ACO-BASED PATH PLANNING

Areej J. Alabbadi[1] and Belal H. Sababha[2]

(Received: 26-Jan.-2025, Revised: 13-Apr.-2025, Accepted: 6-May-2025)

ABSTRACT

Unmanned Aerial Vehicles (UAVs) play a crucial role in various operations, especially where human life must be protected. Efficient path planning and autonomous coordination are critical for UAV swarms in dynamic 3D cooperative missions, where real-time adaptability is essential. This work addresses the challenge of optimizing UAV swarm operations by proposing a novel hybrid navigation system based on Ant Colony Optimization (ACO). The system efficiently balances path optimization with dynamic formation control, adapting to mission-specific requirements. A key contribution is the hybrid navigation approach, which prioritizes the desired formation of the swarm or the path length and flight time through a threshold- based mechanism, allowing real-time adaptation to changing environments. The system also introduces a comprehensive cost function that evaluates the quality of the path, time consumption, mission completeness and formation divergence. The experiments show that the system consistently provides high-quality paths, achieving around 97% path quality in most cases and never declines below 90%, even in challenging scenarios. The collision avoidance module ensures the completeness of the 100% mission, successfully navigating drones around obstacles and maintaining an optimal path. Furthermore, the formation conservation mechanism effectively maintained the desired swarm configurations while dynamically adapting to obstacles, with the formation change staying within 30% of the allowable range in most scenarios, highlighting the system's ability to preserve the desired formation even in dynamic environments. This research advances UAV swarm intelligence, enabling efficient and autonomous operations in complex 3D environments for diverse cooperative missions. The system's adaptability to formation requirements opens new possibilities for UAV swarm applications, improving navigation efficiency and enhancing formation control.

KEYWORDS

Ant colony optimization (ACO), 3D dynamic environment, UAV swarm, Hybrid navigation approach, Collision avoidance mechanism.

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are revolutionizing the industry. They enable rapid and more cost-effective completion of industrial activities while ensuring safety primarily due to their small size, affordable density and general simplicity of management and operation [1]. UAVs are an effective tool for carrying out operations in locations that are difficult to access. Performing in groups or swarms offers additional benefits. The ability to perform tasks that require flying over large areas, reducing the time required for specific operations, area coverage and coordinated impacts are only a few of the operational benefits that UAV swarms have over non-swarm systems [2]-[5].UAV swarms leverage aerial mobility, high-speed maneuverability and extensive coverage capabilities, making them essential for a variety of applications [6]-[8]. Hundreds of thousands of agents can collectively be controlled by swarm systems, while a single operator or a small team is focused on carrying out mission objectives. Humans can maintain operational control while delegating low-level routine choices to UAV agents. UAV swarms can provide the capability for quick communication and decision-making, as detailed in [3]. A UAV swarm is considerably more effective than one or even several human decision-makers in many situations. Because of many advantages, autonomous swarms are often much more effective, timely and responsive than human or human-operated robot groups.

Centralized and distributed control architectures are the two main categories into which cooperative multi-UAV autonomous control architectures are typically classified [9]. With the benefit of obtaining a globally optimal solution, the centralized-control method has dominated early research. However, this strategy has a fundamental weakness: the multi-UAV system will become uncontrollable should the decision-making layer fail due to the high dependence on the communication link. The distributed

1. A. J. Alabbadi (Corresponding Author) is with Electrical Engineering Department, Princess Sumaya University for Technology, Amman, 11941, Jordan. Email: are20208172@std.psut.edu.jo

2. B. H. Sababha is with Computer Engineering Department, Princess Sumaya University for Technology, Amman, 11941, Jordan.

control approach, which has the advantages of increased dependability, less computation and communication, becomes a study focus as UAV performance and autonomous capabilities develop [10]-[11].

The capability of assigning targets and building a 3D trajectory for each UAV in the swarm is an essential part of its operation. The general problems associated with 3D path planning for a single UAV have been addressed using a variety of techniques, including probabilistic road maps, A* algorithms, artificial potential fields, probabilistic navigation functions and many other techniques [12]-[15]. Most of these algorithms use sampling-based and graph-based search techniques, which work well in high-dimensional configuration spaces and are relatively simple to implement. It is also known that, given enough time, they are probabilistically completed in a way that increases the probability of discovering a solution. Many of these techniques have drawbacks, such as potential exposure to local minima and limitations imposed by constraints connected to the grid’s properties. These algorithms often need a balance between exploration and exploitation and are computationally demanding. Some of these algorithms lack robustness, which prevents them from functioning in situations with various dynamic obstacles and automated real-time applications [16]-[17].

Kennedy and Eberhart presented an introductory book on swarm intelligence, based on previous work on robot control and decentralized AI [18]. Swarm intelligence is the intelligent behavior that results from a collection of independent, heterogeneous agents acting as one system. In terms of the distribution of organizational structure, simplicity of individuals, flexibility of the action mode, and establishment of Swarm Intelligence (SI), various social organisms in nature (such as ant colonies, bee colonies, fish schools, and wolf packs) exhibit many characteristics that UAV swarms share [19]. The swarm can be conceptualized as a single entity or system in which intelligence develops through the specific behaviors of a group of people [20]. To develop novel distributed integrated algorithms for UAV swarm cooperative mission planning, some researchers simulated the sophisticated and structured collective behaviors of social organisms.

This research makes several significant contributions to the field of UAV swarm intelligence and distribution for cooperative missions. First, an ACO-based path-planning algorithm is developed. Then, a hybrid navigation and obstacle-avoidance algorithm is proposed. The hybrid navigation method adapts to different application requirements. By integrating a formation-conservation mechanism, the hybrid method monitors the relative positions of drones in real time and dynamically adjusts their positions to maintain a desired formation. This development adds versatility to the algorithm, as it can prioritize either formation conservation or optimized path planning based on the application’s specific needs.

2. Literature Review

With advancements in electronic intelligence and control sub-systems, UAVs have gained popularity and are widely used in various professional and recreational applications [21]. Although initially used primarily for military purposes, UAVs have expanded their presence in the commercial and industrial sectors [22]. This expansion can be attributed to technological advancements and improved power capacities, enabling customized structures, configurations, and equipment customized to specific tasks[23]-[24].

Engaging in risky or laborious tasks often requires the deployment of multiple UAVs. This requirement arises from the significant time commitment and limited autonomy of these small unmanned vehicles. Using multiple drones concurrently, each vehicle assuming the role of a backup in the event of failure, tasks can be performed in parallel, resulting in reduced overall time requirements compared to sequential execution with individual drones. This collective approach improves efficiency, productivity, and the ability to tackle challenging endeavors effectively. This strategy draws inspiration from the remarkable group dynamics observed in various natural biological models, such as birds or ants [25]. These organisms exhibit remarkable coordination and interaction among individuals, as they work together toward a shared objective: migrating to warmer regions or efficiently transporting food to their colonies. Swarm-based systems aim to harness the power of coordinated action and adaptability to solve complex problems.

Metaheuristic algorithms have emerged as powerful tools in artificial intelligence and mathematical optimization, gaining significant attention over the past two decades [26]. These algorithms exhibit stochastic behavior and offer optimal solutions with reduced computational effort compared to conventional techniques. Metaheuristic algorithms are problem-independent and can be broadly classified into four categories: swarm-based (SI), physics-based, evolutionary-based and human-based algorithms. SI algorithms, particularly, harness the collective intelligence observed in natural systems, such as birds, ants, fish, wolves, and other social organisms. These algorithms strike a balance between exploration and exploitation within the search space. Exploration involves a global search for exploration, while exploitation involves a local search in areas identified as promising during the exploration phase. SI algorithms aim to find optimal solutions to a wide range of problems by emulating these social behaviors.

Multi-UAV cooperative path planning aims to meticulously determine an optimal flight path for each UAV, starting from its initial point and ending at the terminal point. This planning process involves minimizing overall flight costs while simultaneously satisfying various constraints, including the distance between UAVs, arrival time, safety requirements, and UAV Performance Criteria. Chen et al. tackled the air-ground cooperation problem of Unmanned Ground Vehicles (UGVs) and UAVs by combining the Genetic Algorithm (GA) with ACO [27]. Their method effectively decoupled the routes of UGVs and UAVs, optimizing the heterogeneous delivery problem and obtaining optimal routes.

Kyriakakis et al. introduced a novel dynamic optimization problem for UAV search and rescue scenarios [28]. They developed a multi-swarm framework with additional UAV constraints and evaluated seven optimization algorithms. Yu et al. proposed a mutation-constrained adaptive selection Differential Evolution Algorithm (DE) to handle the optimization problem [29]. The algorithm aimed to find the optimal solution while satisfying these constraints. To plan feasible paths that cover an entire area for a UAV to maintain a constant flight level relative to the ground, Gonzalez et al. developed a coverage algorithm [30]. They used DE to evaluate the resulting paths and select the best path based on distance costs.

Wu et al. developed an improved fast convergence Artificial Bee Colony (ABC) algorithm to obtain the optimal path in a battlefield environment, considering conflicts and constraints [31]. Xu et al. developed an improved multi-objective Particle Swarm Optimization (PSO) algorithm [32]. Their approach calculated feasible and collision-free trajectories with variable minimum altitude, length, and angle rates.

Phung and Ha addressed the path-planning problem for multiple UAVs in complex environments with multiple conflicts [33]. They proposed the Spherical Vector-based PSO, which efficiently explores the configuration space of UAVs to generate the optimal path that minimizes the cost function. Tong et al. integrated the Pigeon-inspired Optimization (PIO) algorithm with DE mutation strategies for path-planning optimization [34]. Their approach considered three indices: path length, path sinuosity, and path risk. Qu et al. combined hybrid Grey Wolf Optimization (GWO) with a modified Symbiotic Organism Search (SOS) algorithm [35]. They simplified the GWO phase to improve the convergence rate and maintain the population's exploration ability. The SOS phase was synthesized with GWO to enhance the ability to exploit.

There have been significant recent advancements in UAV swarm research in the integration of AI algorithms to enhance decision-making and adaptability [36]-[37]. However, challenges remain in achieving robust solutions for complex tasks, especially in dynamic and uncertain environments. Key research gaps include the need for improved collision avoidance, navigation strategies, and path-planning algorithms that can effectively address real-world constraints, such as uncertainty, security restrictions, and dynamic obstacles, which until now were discussed as an open issue and a research challenge [38]. While existing studies have explored these areas individually, there is a need for integrated systems that can comprehensively address these challenges. The proposed system significantly contributes to UAV swarm research by integrating several essential components, including a collision-avoidance algorithm, a hybrid navigation approach, and a path-planning algorithm based on Ant Colony Optimization. The system showcases cooperative detection and avoidance capabilities, enabling UAV entities to collaborate effectively in detecting and avoiding collisions with both obstacles and other UAVs. It functions in a 3D dynamic environment, addressing uncertainties, security restrictions, and multiple objects. Utilizing ACO, the path-planning algorithm exhibits distributed-planning behavior, as it is applied to each target in the mission, ensuring optimized safety and cost objectives. The system's ability to maintain formations enables UAV swarms to preserve their desired shapes and spatial dimensions. These features set the system apart from other studies in the literature, demonstrating its versatility and potential for real-world applications in various cooperative missions.

3. Proposed System

The proposed system consists of four key modules: the ACO-based path-planning module, the hybrid-path navigation module, the collision-avoidance module, and the messaging module. Each module serves a specific purpose in cooperative mission planning. These components work together to optimize the mission performance of the UAV swarm. Figure 1 illustrates the key components of the proposed system.

The ACO module forms the core of the system, drawing inspiration from ants' foraging behavior. Using pheromone-based communication and local heuristics, it guides the decision-making process of individual UAVs. By balancing exploration and exploitation, the ACO module facilitates the search for optimal paths within the swarm. To enhance the adaptability and flexibility of the system, a developed approach called the Hybrid Approach is proposed. The Hybrid Approach introduces adaptability to the system by dynamically adjusting the path-planning strategy based on the desired swarm shape. The Obstacle Avoidance module integrates real-time obstacle detection and intelligent decision-making to ensure safe navigation. By employing collision-avoidance algorithms, the module guides UAVs to navigate around obstacles and complete their missions. The Messaging System facilitates effective communication and information sharing among UAVs.

3.1 ACO-Module

Ant Colony Optimization (ACO) was initially proposed by Dorigo et al. as a powerful multi-dimensional optimization algorithm that draws inspiration from the foraging behavior of specific species of ant [39]-[40].

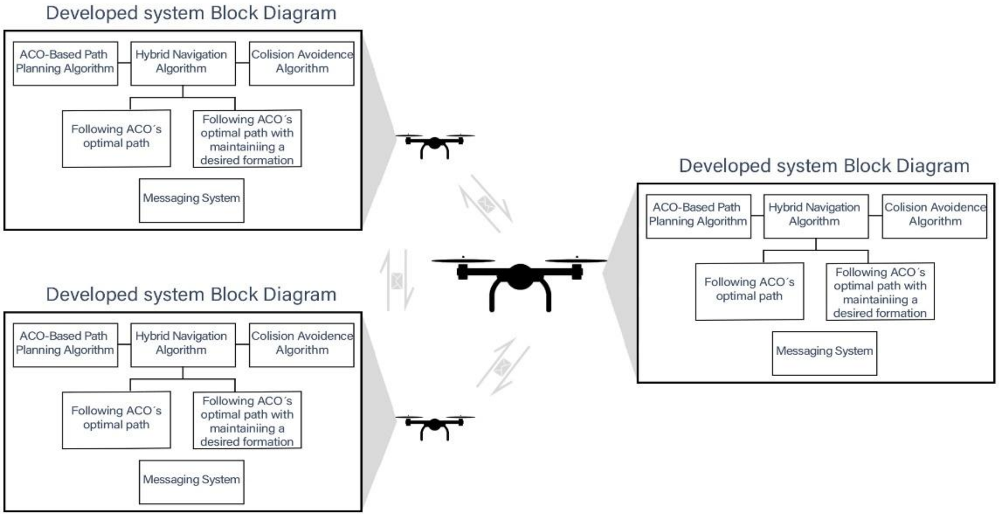

Developed system Block Diagram
ACO-Based Path Planning Algorithm
Hybrid Navigation Algorithm
Colision Avoidence Algorithm
Following ACO´s optimal path
Following ACO´s optimal path with maintainiing a desired formation
Messaging System
Developed system Block Diagram
ACO-Based Path Planning Algorithm
Hybrid Navigation Algorithm
Colision Avoidence Algorithm
Following ACO´s optimal path
Following ACO´s optimal path with maintainiing a desired formation
Messaging System
Developed system Block Diagram
ACO-Based Path Planning Algorithm
Hybrid Navigation Algorithm
Colision Avoidence Algorithm
Following ACO´s optimal path
Following ACO´s optimal path with maintainiing a desired formation
Messaging System

Figure 1. System block diagram.

Through collective intelligence, the ACO collaboratively determines the shortest path based on the density of the pheromone trail [41]. The strength of ACO lies in its ability to balance exploration and exploitation effectively. Randomly exploring ants ensures a diverse search-space coverage, enabling the algorithm to discover potential solutions. At the same time, the exploitation of the pheromone trails by other ants reinforces the convergence towards promising paths, promoting the identification of optimal solutions. This inherent balance between exploration and exploitation makes ACO highly robust and adaptive in dynamic problem domains.

To simulate the behavior of real ants, ACO models employ equations or algorithms to update and propagate the pheromone values dynamically. These updates reflect the collective behavior of the artificial ants and play a critical role in the convergence of the algorithm toward optimal or near-optimal solutions. The equation for the pheromone update is as follows:

$$\tau_{ij}(t+1) \leftarrow (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (1)$$

where:

- $\tau_{ij}(t+1)$: Represents the updated pheromone value on the path of component i to j at time t+1.
- $\tau_{ij}(t)$: Represents the current pheromone value on the path of component i to j at time t.
- ρ: The pheromone evaporation rate is a control of the rate at which pheromones decay.
- $\Delta\tau_{ij}(t)$: The pheromone deposit rate represents the amount of pheromone deposited on the path from component i to j at time t by the artificial ants constructing solutions.

ACO algorithms use mathematical models for simulating ant decision-making. Various models exist, often relying on state-transition rules and probabilistic methods. One widely used model is the ant system, which employs probabilities to choose paths. It balances pheromone intensity and heuristics, achieving the exploration-exploitation trade-off. The probability equation used in ant decision-making is as follows:

$$P_{ij} = \frac{(\tau_{ij}(t))^{\alpha} * (\eta_{ij}(t))^{\beta}}{\sum_{0}^{Allpossiblepaths} (\tau_{ij}(t))^{\alpha} * (\eta_{ij}(t))^{\beta}} \quad (2)$$

where:

- $P_{ij}(t)$*:* Represents the probability of selecting the path from component i to j at time t.
- $\eta_{ij}(t)$*:* Represents a problem-specific heuristic value associated with the path of component i to j at time t.
- α and β*:* Are parameters that control the relative importance of the pheromone trail and heuristic information, respectively.
- The denominator $[\sum_{0}^{Allpossiblepaths} (\tau_{ij}(t))^{\alpha} * (\eta_{ij}(t))^{\beta}]$ represents the sum of the probabilities for all possible paths or components at time t.

In decision-making, artificial ants consider pheromone information and problem-specific heuristics. Pheromone information, encoded in the pheromone trails, provides a collective memory of the paths previously explored by the ants. The higher the pheromone concentration on a path, the more attractive it becomes to subsequent ants.

The equation used to calculate the heuristic value, $\eta_{ij}(t)$, is problem-specific and depends on the characteristics of the path or component. One example of a commonly used heuristic is the inverse of the distance between two points, represented as:

$$\eta_{ij}(t) = \frac{1}{D_{ij}} \quad (3)$$

Where, D_{ij}: Represent the distance between the two points *i* and *j*.

In this research, the characteristics of swarm UAV path planning and the parameter values accordingly are considered carefully, as shown in Table 1.

Table 1. Parameter values for ACO in the proposed algorithm.

ACO Parameter	**Value**
Evaporation Rate	0.5
Pheromone Deposit Rate	1/Path length
Heuristic Information (β)	5
Importance of Pheromone Trails (α)	1
Initial Pheromone Rate	0.01
Number of Iterations	50

At initialization, each drone establishes its colony by populating several ants. These ants are then tasked with finding the optimal path from the drone’s start to its target point. The information sharing and cooperation among ants occurs exclusively within the bounds of the same colony, which belongs to a specific UAV. Each ant performs its path exploration within a colony, utilizing local and global search strategies to identify the most efficient route toward the target. The local search involves making decisions based on the immediate surroundings and information available locally within the drone’s colony. Meanwhile, global search entails updating pheromone trails to incorporate valuable information gathered during exploration. As a result, the swarm of drones operates with high degrees of decentralization and parallelism, significantly enhancing the overall efficiency and scalability of the system. The algorithm’s key steps are shown in Algorithm 1.

Algorithm 1 ACO-based Path Planning Algorithm

```
Initialize algorithm parameters
Set starting and target positions for the ant’s paths
Create a list of random points representing the map, including start and target nodes
Connect nodes with edges and set initial pheromone values
Initialize the pheromone matrix
Number of iterations $\Leftarrow 0$
while Number of iterations $<$ desired number of iterations do
    Populate ants on the map
    for each ant in the ant-list do
        Create a visit list and add the start point to it
        while ant is not at the target node do
            Move the ant to the next node based on Eq2
            Add the chosen point to the visit list
            Apply local search
            Apply global search
            Update the pheromone matrix based on Eq1
        end while
    end for
    Number of iterations $\Leftarrow$ Number of iterations $+ 1$
    if Number of iterations desired number of iterations then
        Calculate the distance of each ant’s shortest path
        Compare distances of shortest paths and output the optimal path
    end if
end while
```

3.2 Collision Avoidance Algorithm

The collision-avoidance process within the UAV swarm navigation system is an accurately designed multi-step procedure, supporting optimal path planning and obstacle avoidance. The collision-avoidance algorithm implemented in the proposed system builds upon a well-established approach presented in [42]-[43], known for its effectiveness in handling complex scenarios. To work for a swarm of UAVs instead of a single UAV, the modified obstacle avoidance algorithm is illustrated in Algorithm 2.

Algorithm 2 Collision Avoidance

```
Initialize each UAV with start point, target point, speed, rotation, scale and priority
The UAV moves to its current target
while UAV is moving to the target do
    Check if there is a potential collision on the UAV path
    if No potential collision then
        The UAV keeps moving to its target normally
    else
        Send a message to alert all drones in the swarm about the collision possibility
        Check if the UAV is considered to have the highest priority
        if UAV has the highest priority then
            Go back to The UAV moves to its current target and repeat
        else
            Generate a number of random points around the current position
            Calculate the distance to the target through the waypoints
            Find the nearest point with the minimum distance
            Check if the chosen point eliminates the potential collision
            if No, if the newly chosen point still leads to a potential collision then
                Go back to Generate a number of random points and repeat
            else
                Store the original target in the temporary target variable
                Set the target to the nearest point
                Go back to ”UAV moves to its current target” to move the UAV to the nearest point
                Check if the nearest point is reached
                if The nearest point is not reached then
                    Go back to ”UAV moves to its current target”
                else
                    Restore the original current target
                    Go back to ”UAV moves to its current target”
                end if
            end if
        end if
    end if
end while
```

3.3 Messaging Module

The messaging module in the system facilitates effective communication and coordination between drones within the UAV swarm. The messaging module implemented in the proposed system is based on a well- established approach presented in [43]. It is crucial to enable the swarm to operate as a cohesive unit, dynamically adapting to changing conditions and avoiding collisions while pursuing its mission objectives. Significant updates are made to enhance dynamic adaptability and swarm robustness. The system now adopts a distributed-path planning and hybrid navigation approach, allowing for more efficient and resilient performance.

The messaging module enables drones within the swarm to share their real-time positions. This continuous data exchange is essential for maintaining the desired formation during cooperative missions. Knowing the positions of the other drones, each UAV can adjust its trajectory to stay in the designated formation.

3.4 System Design

The system described in this sub-section is designed to control a swarm of drones operating within a specified environment. Its primary objective is to optimize the movement and coordination of the drones to achieve efficient and effective task completion. The system aims to minimize the distance traveled, maximize productivity and ensure safe operation by intelligently guiding the drones through commands and paths. The drone-swarm navigation system can be adapted to two different options based on application requirements. In the first option, formation conservation is not an application requirement, while in the second option, the application requires maintaining a specific formation or shape. In both options, the drones follow the optimal path generated by the ACO module, ensuring efficient navigation and collision avoidance within the environment.

- **Option one:** The system coordinates the movement of the drones, optimizes their paths using ACO, controls their movement using PID controllers and performs collision avoidance to ensure safe operation within the swarm, as shown in Algorithm 3.
- **Option two:** In the second option shown in Algorithm 4, additional functionality is introduced when the application requires maintaining a specific formation or shape.

3.5 Cost Function Evaluation

For an objective evaluation of the overall performance of the swarm, the following data is collected before the evaluation parameters are computed:

- Minimum Distance: The straight-line distance between each drone’s initial and final target positions.
- Total Travelled Distance: The cumulative distance traveled by each drone from its initial position to its final target.
- Total Travelled Time: The duration a drone needs to reach its final target.
- Number of Divergences: A divergence occurs when a drone deviates from its intended path.
- Number of Collisions: When two drones come into physical contact.

In addition, for option two, where formation conservation is required, an extra parameter is calculated:

- Average Distance Change: Measures how much each drone deviates from the desired formation.

The following evaluation parameters are formulated to capture the mission’s quality, efficiency, completion and formation conservation during cooperative missions:

- Path Quality (PQ): Evaluates the efficiency of the path-planning module. It is calculated using the following equation (Eq. 4):

$$PQ = \frac{1}{N} * \sum_{i=0}^{N-1} \frac{MinTravelledDistance_i}{TotalTravelledDistance_i} * 100\% \tag{4}$$

where:

- N: the total number of drones in the swarm.
- MinTravelledDistance$_i$: is the minimum distance traveled by drone i from its initial position to its target.
- TotalTravelledDistance$_i$: is the total distance traveled by drone i during its mission.

A higher value for this parameter indicates that the drone successfully optimizes its path, following the shortest route to its target.

Algorithm 3 System Behavior – Option 1

```
UAVs receive important mission information from the ground station, including start and target points, speed, rotation, scale, formation and priority.
Apply the ant colony algorithm for each UAV.
while the optimal path is not generated do
    Keep waiting
end while
```

```
Begin the main loop of the system
for each UAV do
    Each UAV’s initial target is set to the first node on its optimal path
    Each UAV sets the last point in the optimal path as the destination target
    Use a PID controller to calculate the drive forces for each axis (x, y, z)
    UAV moves to its current target
    while UAV is moving to the target do
        UAV checks if there is a potential collision on its path
        if No potential collision then
            Check if the current target has been reached
            if the current target is reached then
                Check if the current target is the destination target
                if the current target is the destination target then
                    Mission ends
                else
                    Update the target position to be the next node in the optimal path
                    Go back to UAV moves to its current target
                end if
            end if
        else
            Send a message to alert all drones in the swarm about the collision possibility
            Check if the UAV has the highest priority
            if UAV has the highest priority then
                Go back to UAV moves to its current target
            else
                Generate several random points around the current position
                Calculate the distance to the target through the waypoints
                Find the nearest point with the minimum distance
                Check if the chosen point eliminates the potential collision
                if Chosen point eliminates collision then
                    Store the original target in the temporary target variable
                    Set the target to the nearest point
                    Go back to UAV moves to its current target
                    Check if the nearest point is reached
                    if the nearest point is not reached then
                        Go back to UAV moves to its current target
                    else
                        Restore the original current target
                    end if
                else
                    Go back to the step of generating several random points and repeat
                end if
            end if
        end if
    end while
end for
Repeat the main loop until the UAV reaches its target
```

Algorithm 4 System Behavior – Option 2

```
UAVs receive mission information from the ground station, including start and target points, speed, rotation, scale, formation and priority.
Each UAV reads the start point for all other UAVs in the swarm.
Create a reference distance array that captures the distances between drones in the desired formation.
Apply the ant colony algorithm for all UAVs in the swarm.
Set the current target as the first node in the optimal path for the UAV and set the last point in the optimal path as the destination target.
Use a PID controller to calculate drive forces for each axis (x, y, z).
UAV moves to its current target.
while UAV is moving to the target do
    Read the current positions for all drones and create a current distance array, representing the current formation distances for the swarm.
    if The current distance array equals the reference distance array then
        The UAV keeps moving to its current position while checking for potential collisions and if the UAV reaches its target, continue to the next step.
    else
        Calculate the difference in distance between the UAV and all other UAVs in the swarm.
        if The difference in distances is less than the threshold then
            The UAV keeps moving to its current position while checking for potential collisions and if the UAV reaches its target, continue to the next step.
        else
            Generate several random points around the current position.
            Choose the nearest point.
            Check if the nearest point will maintain the UAV position.
            if The nearest point maintains the UAV position then
                Store the original target in the temporary target variable.
                Set the target to the nearest point.
                Go back to "UAV move to its current target".
                Check if the UAV reaches the nearest point.
                if UAV is not at the nearest point then
                    Go back to UAV move to its current target and repeat.
                else
                    Restore the original target.
                end if
            else
                Go back to Generate several random points and repeat.
            end if
        end if
    end if
end while
Repeat the main loop until the UAV reaches its target
```

- Mission Completeness (MC): Evaluates the collision-avoidance module's effectiveness and the UAV swarm's adaptability in successfully achieving its mission objectives. It is calculated using Eq. 5.

$$MC = \frac{N_{ReachedItsTarget}}{N} * 100\% \tag{5}$$

Where, $N_{ReachedItsTarget}$: is the count of drones successfully reaching their targets.

A higher value for this parameter indicates a success rate in achieving mission objectives, as many drones have reached their targets without collisions.

- Average of Divergence (AD): Measures how much each drone deviates from its original path to avoid collisions with other drones or with obstacles. It quantifies the quality of the new routes generated by the collision-avoidance module. Eq. 6 shows how this is calculated.

$$AD = \frac{\sum_{i=0}^{N-1} NumberOfDivergences_i}{N} \tag{6}$$

Where, *NumberOfDivergences$_i$*: is the number of times that drone i deviates from its original path.

- Swarm Flight Time (FT): Quantifies the efficiency of the UAV swarm in completing the mission, referring to a predefined time frame. It reflects how effectively all drones in the swarm work together to achieve mission objectives. This parameter is calculated as shown in Eq. 7.

$$FT = \frac{T}{TimeFrame} \tag{7}$$

Where, T: is the total time taken for all drones in the swarm to reach their respective targets.

A smaller value indicates a more cohesive and cooperative swarm, where drones work towards mission completion with minimal delays and divergences.

- Formation Change (FC): Evaluates how effectively drones in the swarm maintain their desired formation during cooperative missions. This parameter is calculated as shown in Eq. 8.

$$FC = \frac{1}{N} * \sum_{i=0}^{N-1} \frac{AverageofDistanceChange_i}{DefindThreshold} * 100\% \tag{8}$$

Where:

- *Averageof DistanceChange$_i$*: is the average distance change for each drone relative to the desired formation of the swarm.
- *DefinedThreshold*: is a predefined value that determines acceptable deviations from the desired formation.

A lower formation change value indicates better performance of the hybrid module, as it indicates that the drones successfully maintain their formation with minimal deviations from the desired configuration.

A cost function is formulated as a weighted sum (α,β,ω,γ,μ) of the four parameters in the first option and five parameters in the formation-conservation option, with each parameter assigned a specific weight to reflect its relative importance. The formula for the comprehensive cost function is given by Eq.9 and Eq.10 for option one and option two, respectively.

Option one:

$$CF = \alpha PQ + \beta(1 - FT) + \omega(1 - AD) + \gamma MC \tag{9}$$

Option two:

$$CF = \alpha PQ + \beta(1 - FT) + \omega(1 - AD) + \gamma MC + \mu(1 - FC) \tag{10}$$

These formulations ensure that the algorithm is evaluated based on its ability to optimize multiple key aspects simultaneously. A higher comprehensive cost function value indicates better performance.

3.6 System Complexity

The algorithm complexity measures how the performance and execution time of the algorithm scale with the increasing number of drones in the swarm. As the swarm size grows, the algorithm's efficiency becomes critical in ensuring real-time operation and mission success. Efficient algorithms with lower complexity ensure that the swarm can handle larger numbers of drones without compromising performance.

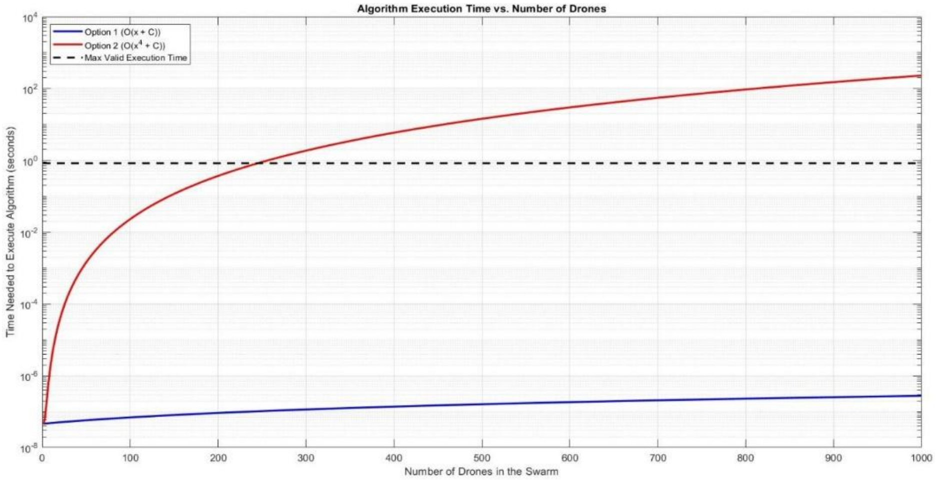


Figure 2. System execution time for both options as the number of drones in the swarm increases.

The execution time of the algorithm is a critical aspect that directly impacts the real-time operation of UAV swarms. In the first option, where formation conservation is not a specific application requirement, the algorithm complexity is O(X + C), where X is the number of drones in the swarm and C is the number of execution cycles, which remains constant regardless of the number of drones. The algorithm's scalability in this option is relatively better due to the linear complexity, making it suitable for swarms with a large number of drones.

In the second option, where formation conservation is essential, the algorithm complexity becomes O(X^4 + C). This increase in complexity is due to the additional calculations and coordination required to maintain the desired formation during cooperative missions. The formation-conservation constraint introduces non- linearity in the algorithm, which impacts its scalability as the number of drones increases.

To ensure real-time execution in both options, an upper bound for the execution time of the algorithm is established as follows:

- For Option One:

$$(X + C) * (executioncycletime) < (\frac{sensingrange}{dronespeed}) \tag{11}$$

- For Option Two:

$$(X^4 + C) * (executioncycletime) < (\frac{sensingrange}{dronespeed}) \tag{12}$$

By meeting this condition, the algorithm can guarantee safe and efficient navigation for the entire swarm, even in dynamic and densely populated environments. Figure 2 illustrates the algorithm execution time for both options as the number of drones in the swarm increases.

4. SIMULATION AND RESULTS

The proposed system is implemented using the UTSim simulator, which offers an adaptable platform for creating and configuring multiple instances of UAVs [43]. The simulation setup involved the implementation of flight scenarios in a 3D environment, where the UAVs were controlled using the proposed system. Before each mission, the initial locations and destinations of the UAVs were defined based on the specific scenario. The UAVs used in the experiments were all fixed in size, with a half-meter diameter. Their speeds were maintained at a constant value of 6 m/s throughout the missions. Due to the inherent characteristics of rigid bodies, the speed decreased when the UAVs changed direction or reached their destinations.

Each run was performed 35 times in the 3D space to ensure reliable results. In an obstacle-free environment where formation maintenance is not a mission requirement, the swarm exhibits perfect consistency across all 35 experimental runs, with zero variability within a confidence interval of 95%.

However, when operating in a dense obstacle environment with a 0.1 allowable distance change, the system displays slight variability. For a 30-drone swarm, the error margin remains below 0.007 for distance change and 0.009 for flight time. As the swarm size increases to 40 drones, the error in total distance remains below 0.12. Even with a swarm of up to 80 drones, the error margin for distance change stays below 0.01. These consistently low error margins across all test conditions provide strong evidence of the system's robustness and reliability in both controlled and complex environments.

4.1 Algorithm Constraints and Assumptions

Constraints play a vital role in shaping the behavior and performance of UAV swarms during missions. They are essential elements that impose limits and restrictions on various aspects of the swarm's operation, ensuring safe, efficient, and coordinated behavior. Several constraints were considered to study the system's performance under different scenarios:

- Maneuverability Constraints: The maximum turning angle (θmax) was set to 30 degrees on the x-axis while remaining unrestricted in the y and z-axes.

  $$\Theta_i(T) - \Theta max \leq 0 \tag{13}$$

  where: $\Theta_i(T)$: The turning angle of the i^{th} UAV at time T.

- Sensing Range Constraint: Each UAV sampled 25 points every time a reroute was computed. Rerouting was triggered when a passive obstacle was detected or when a higher-priority UAV was sensed. The sample points were taken within a customizable radius (Rs) of a circle/sphere set at 5 meters.

  $$Rs - Dij(T) \geq 0 \tag{14}$$

  where, $Dij(T)$: The distance between the i^{th} UAV and the j^{th} UAV or obstacle at time T.

- Collision-avoidance Constraints: The algorithm incorporates a safe distance, denoted as Dmin, between two UAVs or between a UAV and an obstacle. This distance defines the collider sensing range, represented by the radius of a circle or sphere centered at the UAV.

  $$Dmin - Dij(T) < 0 \tag{15}$$

  where, $Dij(T)$: The distance between the i^{th} UAV and the j^{th} UAV or obstacle at time T.

- Operating-range Constraint: The flight operation area was defined as 1 km * 1 km, providing a bounded environment for the swarm's missions.

- Time frame: the time frame is set to be a one-minute flight.

- For the first option's cost function, the (α,β,ω and γ) are 0.3, 0.3, 0.2, 0.2, respectively.

- For the second option's cost function, the (α,β,ω,γ and μ) are 0.2, 0.2, 0.2, 0.2, 0.2, respectively.

The experiment scenarios were designed to vary the number of drones within the flight area, ranging from 5 to 80 drones. The number of obstacles (moving and static) gradually increased, with the maximum number exceeding the total number of UAVs in the swarm, which is moving randomly in the environment. In the second option, various thresholds were tested to evaluate the performance of the hybrid navigation approach.

4.2 Effects of Different Safe Distances

This sub-section investigates the influence of varying safe distances on swarms of sizes ranging from 5 to 80 UAVs. The safe distance is incrementally increased from 1 meter to 3 meters for each case. This analysis provides insights into the optimal safe distance setting that maximizes the UAV swarm's efficiency and effectiveness in different scenarios. In this sub-section, all tests were conducted in obstacle-free environments and the safe-distance parameter of the system was adjusted and controlled from the ground station before each mission. The mission is designed, allowing tuning for a safe distance based on the distances between the drones and the total travel distance for each drone between the starting and target points, without considering the number of obstacles as a part of the mission design. This will be considered a design-preparation phase to set the safe distance to the next sections. These evaluations provided valuable insights into the system's performance and how the adjustable parameters influenced its behavior when encountering unexpected obstacles during missions.

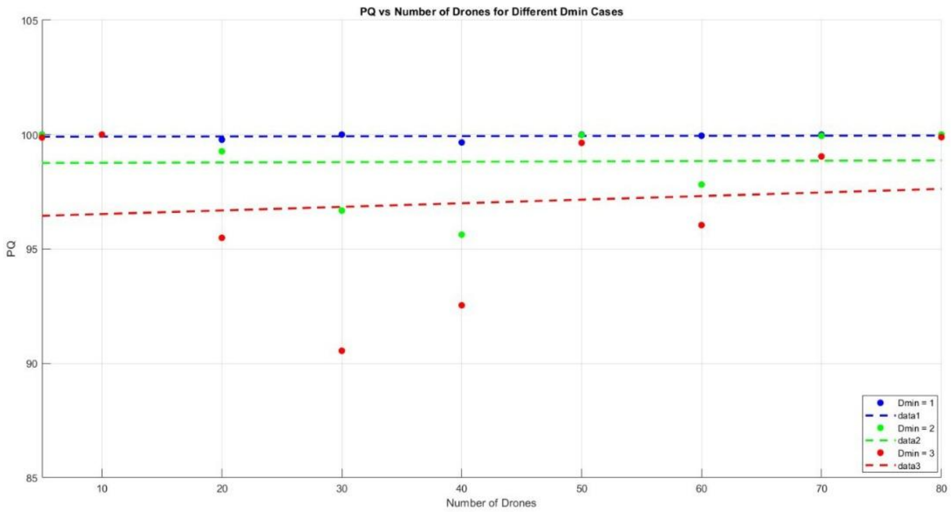


Figure 3. The path quality *vs.* the number of drones for different safe distances.

- Path Quality: The ACO module achieves a path quality of over 99% with 1m safe-distance scenarios. However, it reached 96% when the safe distance increased to 2m and 92% for the 3m safe distance, as shown in Figure 3. As the safe distance for drones increased, more drones had to make route diversions to avoid potential collisions, which increased the average of divergence, as shown in Figure 4, increasing the total traveled distance for each drone, which decreased the path quality. The formation and the distances between the start and target points for the drones are different from swarm to swarm, which explains the path quality and average of divergence behavior change for the same value of safe distance, since the distances between drones in the case with twenty UAVs are less than the distances between the drones in ten-UAVs. This increased the influence of large safe distances where the UAV needed to increase the number of divergences to save the safe distance simultaneously to avoid any potential collisions between the other UAVs in the swarm, which decreased the path quality. However, the path-quality values are close for all swarms because of the distributed approach in the ACO-based path-planning algorithm. The algorithm generates the optimal path for each drone based on its start and target point without considering the number of drones in the swarm.

- Swarm flight time: The increase in the average number of divergences leads to a greater total travel distance. This typically results in longer flight times for the swarms, as illustrated in Figure 5.

4.3 Effects of Number of Obstacles

The number of obstacles gradually increases. The number, speed, direction and all information of obstacles are unknown for the drones in the swarm to evaluate the system's adaptability to uncertainties. The obstacles move randomly in different directions and elevations. All cases are tested at a safe distance of 1 m.

- Path Quality: As illustrated in Figure 6, an increase in the number of obstacles does not significantly affect path quality in swarms with a small number of drones. This is because the drones maintain safe distances from each other and have a wide space within their operating range to locate the nearest point for collision avoidance. However, as the number of drones in the swarm increases, the distances between them decrease and the available operating space narrows, as shown in Figure 7. Consequently, the drones must find the nearest point to avoid collisions with obstacles while also considering a safe distance from other drones in the swarm. This necessity often increases the average number of divergences. Additionally, since the obstacles move randomly within the flight environment, their effects may vary across different scenarios.

- Swarm FT: Increasing the number of obstacles affected the mission time and the values of the cost function. However, since the obstacles are moving randomly in the flight environment, the effect does not show the same behavior in all scenarios, as shown in Figure 8 and Figure 9.

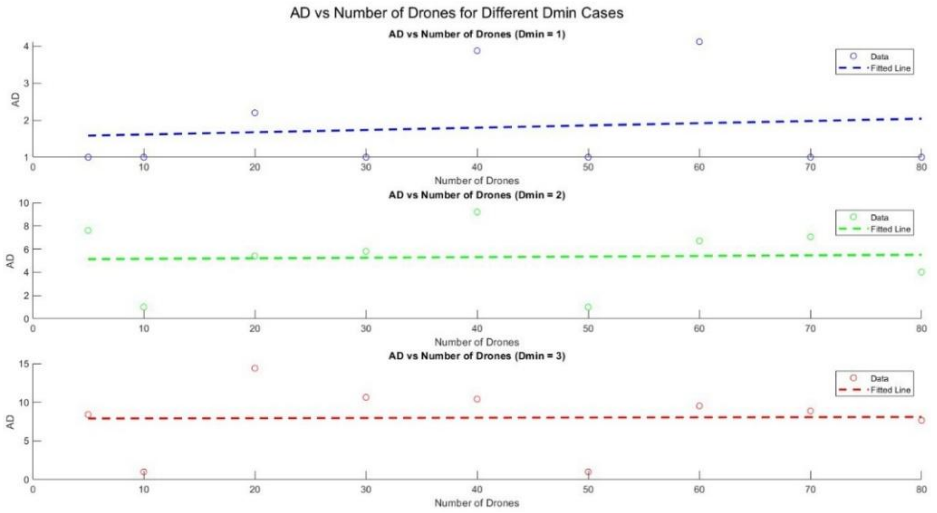


Figure 4. Average of divergence *vs.* the number of drones for different safe distances.

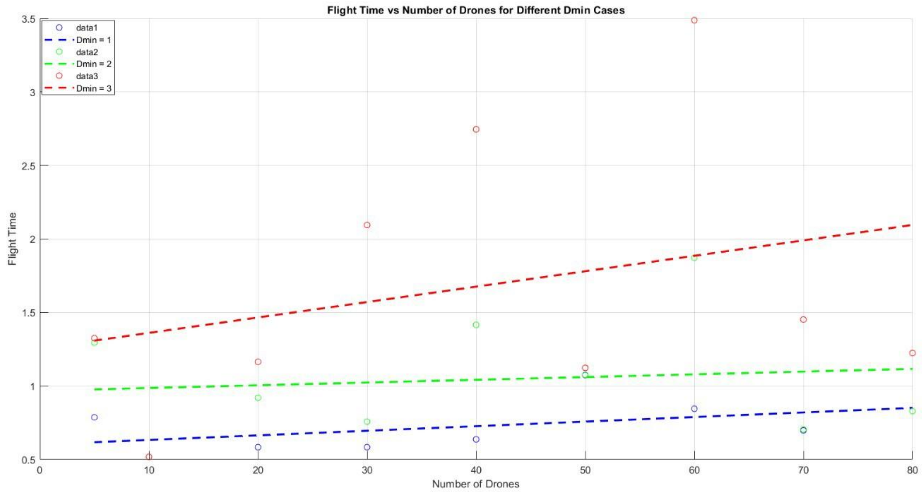


Figure 5. Swarm flight time divergence *vs.* the number of drones for different safe distances.

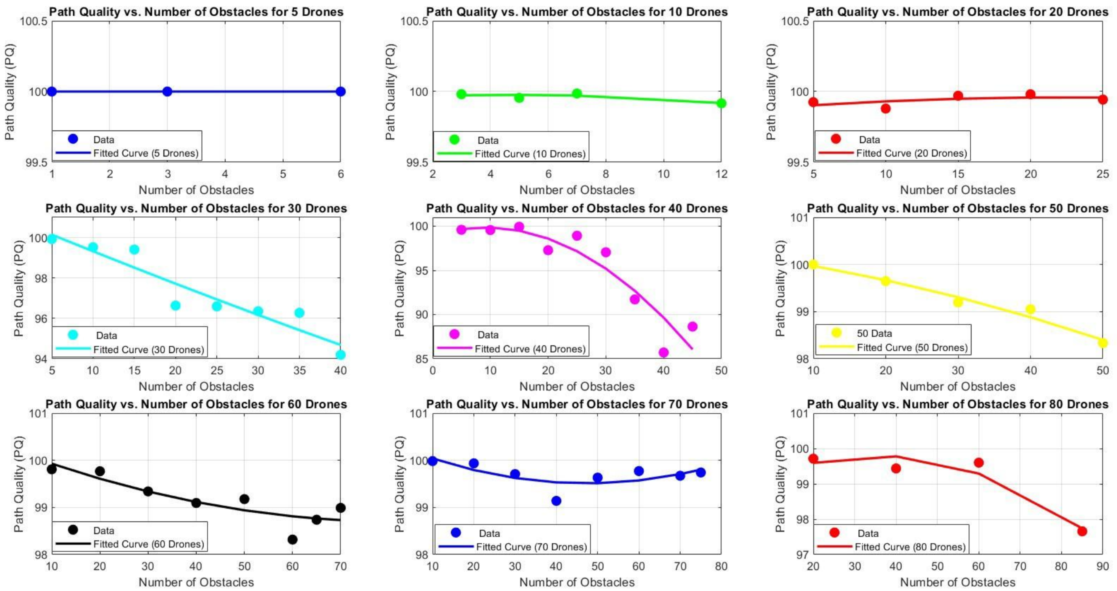
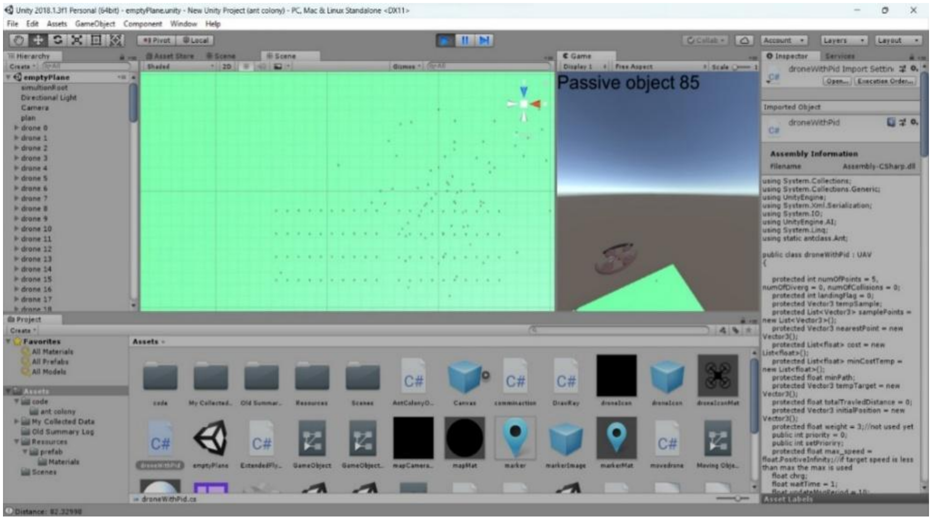


Figure 6. Path quality *vs.* the number of obstacles for different swarm sizes.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.Xml.Serialization;
using System.IO;
using UnityEngine.AI;
using System.Linq;
using static antclass.Ant;

public class droneWithPid : UAV
{

    protected int numOfPoints = 5,
numOfDiverg = 0, numOfCollisions = 0;
    protected int landingFlag = 0;
    protected Vector3 tempSample;
    protected List<Vector3> samplePoints =
new List<Vector3>();
    protected Vector3 nearestPoint = new
Vector3();
    protected List<float> cost = new
List<float>();
    protected List<float> minCostTemp =
new List<float>();
    protected float minPath;
    protected Vector3 tempTarget = new
Vector3();
    protected float totalTravledDistance = 0;
    protected Vector3 initialPosition = new
Vector3();
    protected float weight = 3;//not used yet
    public int priority = 0;
    public int setPrioriry;
    protected float max_speed =
float.PositiveInfinity;//if target speed is less
than max the max is used
    float chrg;
    float waitTime = 1;
    float updateMapPeriod = 10;
```

Figure 7. Eighty-UAV formation with 85 obstacles.

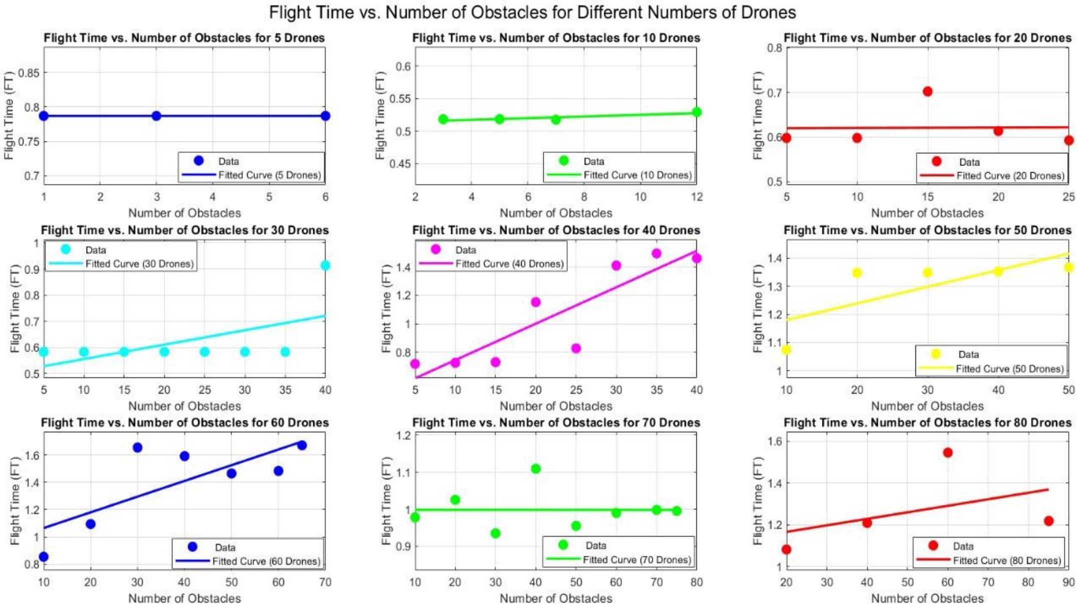


Figure 8. Swarm flight time *vs.* the number of obstacles for different swarm sizes.

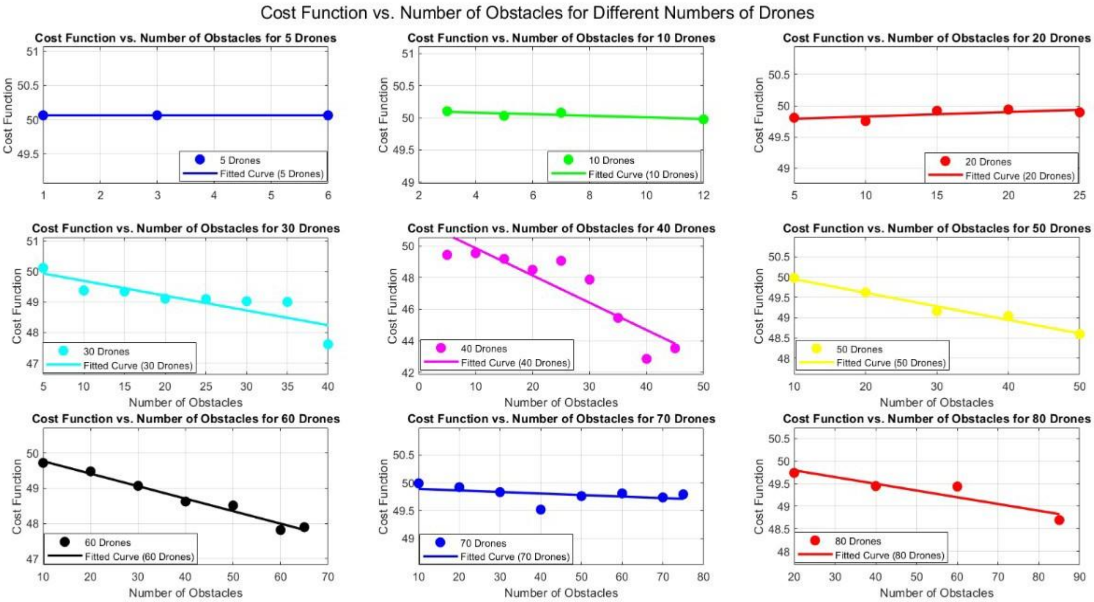


Figure 9. Cost function *vs.* the number of obstacles for different swarm sizes.

4.4 The Formation Threshold Effects

One of the important contributions of this study is the hybrid navigation approach, where application requirements are evaluated to prioritize following the optimal ACO path or maintaining a desired formation. In this sub-section, different thresholds (allowable change distance) are tested to evaluate the system's performance, where each case has a different formation with different distances between the drones within the formation and different distances between the start and destination points for each drone. All cases will be tested at a safe distance of 1 m.

- Path Quality: In the second option, with different threshold values, the system can manage the trade-off between maintaining formation and following the optimal path, resulting in optimal flight trajectories and minimal divergence. The system showed its ability to choose the nearest points to preserve the formation. As Figure 10 illustrates, the quality of the path is above 97% in all cases.
- Swarm Flight Time: Increasing the threshold allowed the drones more movement flexibility, reducing the time needed to complete the mission, as shown in Figure 11.

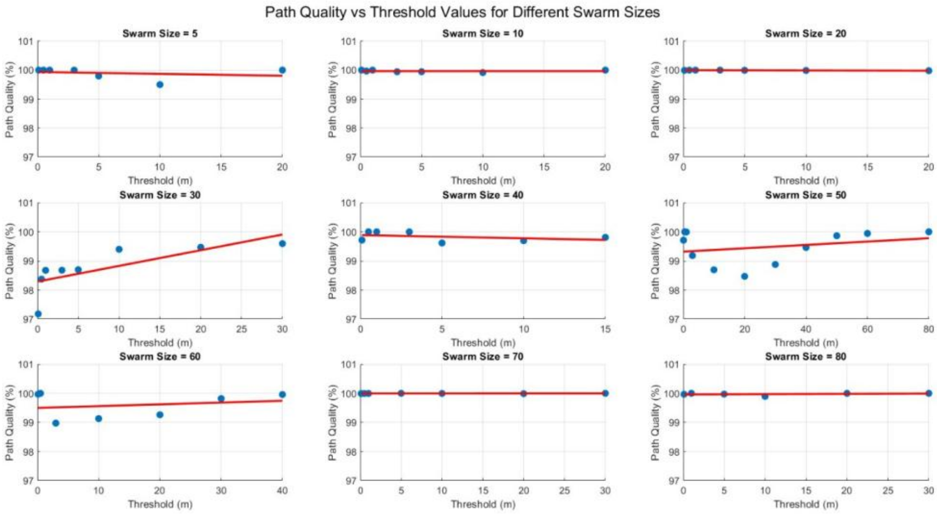


Figure 10. Path quality *vs.* the threshold values for different swarm sizes.

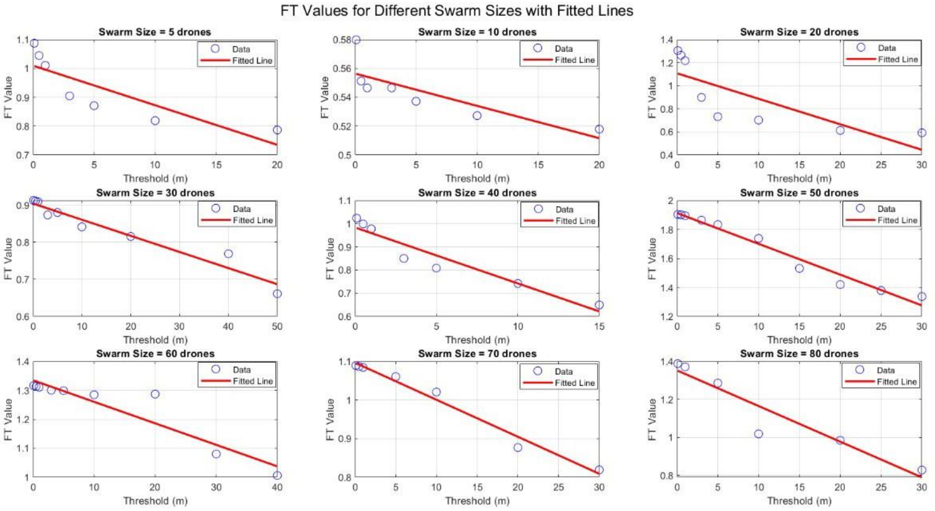


Figure 11. Swarm flight time *vs.* the threshold values for different swarm sizes.

- Formation Change: The formation change parameter evaluates the swarm's ability to maintain its desired formation during cooperative missions. The experiments demonstrated the success of the hybrid approach, as the formation change remained below 25% of the allowable change in all cases, as shown in Figure 12 and this percentage decreased when the threshold increased, but with different slopes, since each drone will generate a random point around its current position, which is directly related to its formation and optimal path and choose the nearest point that maintains its formation, saves the safe distance between the UAVs and avoids any potential collisions. However, in all cases, the system shows high adaptability with an acceptable formation change. It shows that the approach effectively conserves the formation during missions.

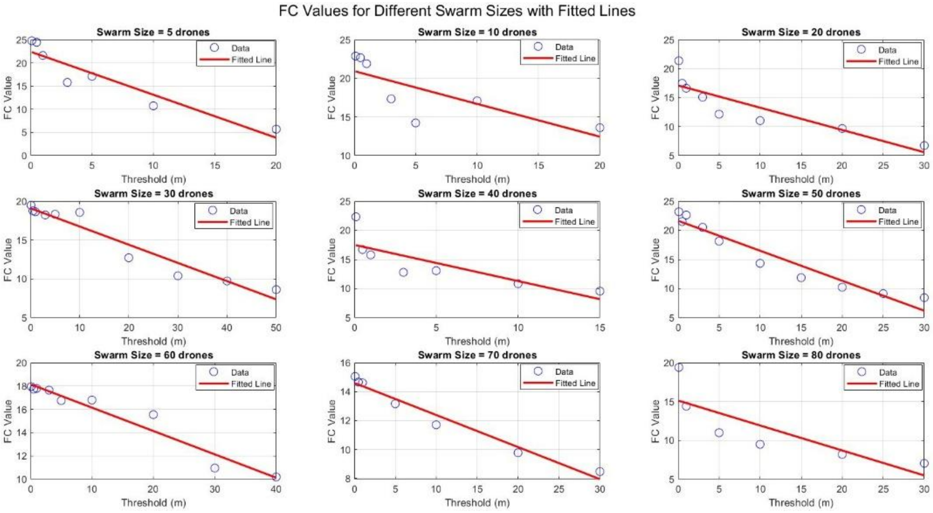


Figure 12. Formation change *vs.* the threshold values for different swarm sizes.

In summary, achieving robust solutions for complex tasks in dynamic and uncertain environments is a persistent challenge. In its integrated approach, the proposed system contributes to filling this gap by integrating ACO-based path planning, hybrid navigation and collision avoidance, enabling cooperative detection and avoidance in 3D dynamic environments with multiple objects, uncertainties and security restrictions. As demonstrated in the results, the system's performance is a direct consequence of this integration. ACO provides efficient path planning, while hybrid navigation and collision-avoidance algorithms work together to maintain formation and prevent collisions. The varying performance between swarms, where UAVs generate random points for collision avoidance, trading off mission objectives with safety directly related to these environments' dynamic and unpredictable nature is a key challenge identified in previous research. Although this variability is observed, the system consistently demonstrated high adaptability with acceptable formation changes, validating its robustness in complex scenarios.

4.5 Challenging Cases Evaluation

To further assess the system's performance, challenging cases were tested in which the swarm must preserve its formation with an allowable change distance of less than 0.1 m while flying in a dense-obstacle environment to assess how the system adapts to high levels of obstacle density while maintaining its formation. As shown in Table 2, cases with a threshold of 0.1 and many obstacles were tested when evaluating the hybrid navigation approach. This case's performance shows the hybrid approach's efficiency in achieving mission objectives while ensuring formation conservation.

Table 2. System's performance in challenging cases.

Number of Obstacles	**Number of Drones**	**AD**	**FT**	**MC (100%)**	**PQ (100%)**	**FC (100%)**	**Cost Function**
6	5	1	1.0969	100	97.7058	26.8118	34.3594
12	10	1	0.6097	100	99.7400	26.0080	35.0247
25	20	1.1	1.3182	100	99.1094	20.6193	35.8143
40	30	4.33	0.9135	100	94.1925	20.0458	34.3799
45	40	5.525	2.0233	100	94.5771	24.6229	33.0812
60	50	8.34	2.5419	100	97.6972	24.2275	33.1175
65	60	3.72	1.1032	100	98.9631	21.6300	35.1919
75	70	2.3571	1.5063	100	98.8808	31.3673	33.3299
85	80	4.19	1.8487	100	98.5522	28.7537	33.3524

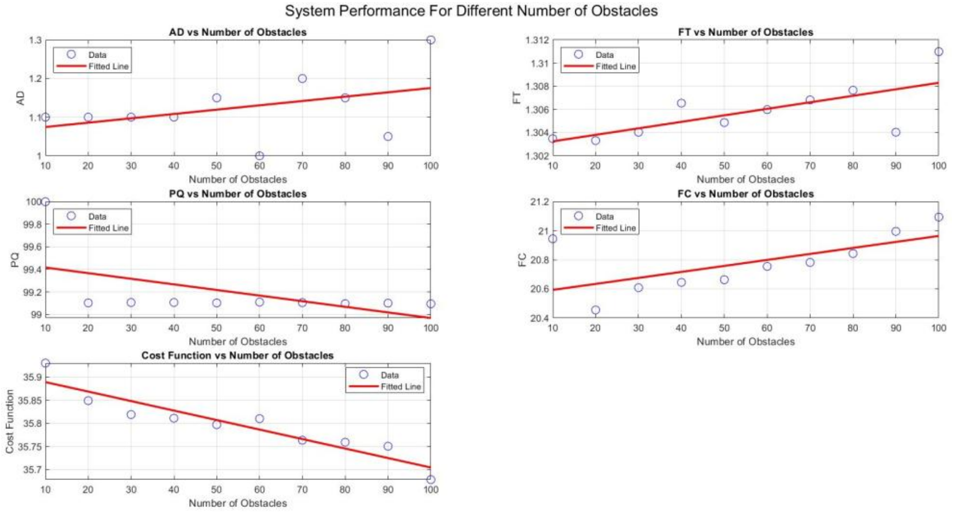


Figure 13. System performance for different numbers of obstacles.

To test the adaptability of the hybrid approach to the increased density of obstacles in the flight environment while maintaining the distance change to be less than 0.1 m, for a twenty-UAV swarm, the number of obstacles will start at 10 and then increase to 100, with a constant number of obstacles during each run session. As shown in Figure 13, increasing the number of obstacles will increase the possibility of collisions, since the operation space is so crowded, which increases the swarm flight time and the average number of divergences. With the increase in the number of obstacles, the UAVs need to increase the distance within the threshold to maintain their formation while saving their safe distance, which will normally affect path-quality and cost-function values. The tests are performed at a 1m safe distance.

5. Conclusion

This work presents an adaptable intelligent system for cooperative UAV swarm missions, integrating a path-planning algorithm based on the ACO algorithm, a collision-avoidance algorithm and a hybrid navigation system. The system was tested and evaluated in various scenarios, including different swarm sizes in dynamic 3D environments filled with moving and static obstacles while maintaining the desired formation. The simulation results demonstrate the system’s outstanding performance, achieving a path quality of around 97% in most cases and never dropping below 90%, even in challenging scenarios. This reflects the high efficiency of the ACO module in finding optimal paths and the system’s adaptability in consistently following them. The collision-avoidance module showed remarkable performance, ensuring that all missions remained collision-free, with a mission completeness rate of 100% in all testing scenarios. When the desired formation was necessary, the system showed its ability to maintain it even in dynamic environments within 30% of the allowable range in most cases. The system’s success lies in its cooperative approach, in which all the modules work together smoothly. This collaborative and intelligent system illustrates its potential for real-world applications in various cooperative UAV-swarm missions.

References

[1] S. Hayat et al., "Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint," IEEE Comm. Surveys and Tutorials, vol. 18, no. 4, pp. 2624–2661, 2016.

[2] M. Campion et al., "A Review and Future Directions of UAV Swarm Communication Architectures," Proc. of the 2018 IEEE Int. Conf. on Electro/Information Technology (EIT), pp. 0903–0908, 2018.

[3] R. Arnold, K. Carey, B. Abruzzo and C. Korpela, "What Is a Robot Swarm: A Definition for Swarming Robotics," Proc. of the 2019 IEEE 10th Annual Ubiquitous Computing, Electronics and Mobile Communication Conf. (UEMCON), pp. 0074–0081, New York, USA, 2019.

[4] M. Khelifi and I. Butun, "Swarm Unmanned Aerial Vehicles (SUAVs): A Comprehensive Analysis of Localization, Recent Aspects and Future Trends," J. of Sensors, vol. 2022, no. 1, p. 8600674, 2022.

[5] Q. Li et al., "A Review of Unmanned Aerial Vehicle Swarm Task Assignment," Proc. of the Int. Conf. on Guidance, Navigation and Control (ICGNC 2022), Springer, pp. 6469–6479, 2022.

[6] Y. Alqudsi, A. Kassem and G. El-Bayoumi, "A Robust Hybrid Control for Autonomous Flying Robots

in an Uncertain and Disturbed Environment," INCAS Bulletin, vol. 13, no. 2, pp. 187 – 204, 2021.

[7] S. A. H. Mohsan et al., "Unmanned Aerial Vehicles (UAVs): Practical Aspects, Applications, Open Challenges, Security Issues and Future Trends," Intelligent Service Robotics, vol. 16, no. 1, pp. 109–137, 2023.

[8] M. Abdelkader, S. Güler, H. Jaleel and J. S. Shamma, "Aerial Swarms: Recent Applications and Challenges," Current Robotics Reports, vol. 2, pp. 309–320, 2021.

[9] M. Cummings, "Operator Interaction with Centralized *versus* Decentralized UAV Architectures," Handbook of Unmanned Aerial Vehicles, pp. 977–992, DOI: 10.1007/978-90-481-9707-1_117, 2015.

[10] S. S. Ponda et al., "Cooperative Mission Planning for Multi-UAV Teams," Handbook of Unmanned Aerial Vehicles, vol. 2, pp. 1447–1490, DOI: 10.1007/978-90-481-9707-1_16, 2015.

[11] M.-H. Kim, H. Baik and S. Lee, "Response Threshold Model Based UAV Search Planning and Task Allocation," Journal of Intelligent and Robotic Systems, vol. 75, pp. 625–640, 2014.

[12] P. O. Pettersson and P. Doherty, "Probabilistic Roadmap Based Path Planning for an Autonomous Unmanned Helicopter," Journal of Intelligent and Fuzzy Systems, vol. 17, no. 4, pp. 395–405, 2006.

[13] L. De Filippis, G. Guglieri and F. Quagliotti, "Path Planning Strategies for UAVs in 3D Environments," Journal of Intelligent and Robotic Systems, vol. 65, pp. 247–264, 2012.

[14] O. Cetin, I. Zagli and G. Yilmaz, "Establishing Obstacle and Collision Free Communication Relay for UAVs with Artificial Potential Fields," J. of Intell. and Robotic Systems, vol. 69, pp. 361–372, 2013.

[15] S. Hacohen, S. Shoval and N. Shvalb, "Applying Probability Navigation Function in Dynamic Uncertain Environments," Robotics and Autonomous Systems, vol. 87, pp. 237–246, 2017.

[16] K. S. Camilus and V. Govindan, "A Review on Graph Based Segmentation," International Journal of Image, Graphics and Signal Processing, vol. 4, no. 5, p. 1, 2012.

[17] S. M. Persson and I. Sharf, "Sampling-based A* Algorithm for Robot Path-planning," The International Journal of Robotics Research, vol. 33, no. 13, pp. 1683–1708, 2014.

[18] H. Cartwright, "Swarm Intelligence by James Kennedy and Russell Ceberhart with Yuhui Shi. Morgan Kaufmann Publishers: San Francisco, 2001. £43.95.xxvii+512pp. ISBN: 1-55860-595-9," The Chemical Educator, vol. 7, pp. 123–124, 2002.

[19] A. Slowik and H. Kwasnicka, "Nature Inspired Methods and Their Industry Applications—Swarm Intelligence Algorithms," IEEE Trans. on Industrial Informatics, vol. 14, no. 3, pp. 1004–1015, 2017.

[20] R. D. Arnold and J. P. Wade, "A Definition of Systems Thinking: A Systems Approach," Procedia Computer Science, vol. 44, pp. 669–678, 2015.

[21] R. Austin, Unmanned Aircraft Systems: UAVs Design, Development and Deployment, John Wiley & Sons, vol. 54, ISBN: 978-0-470-05819-0, 2011.

[22] M. A. Akhloufi, S. Arola and A. Bonnet, "Drones Chasing Drones: Reinforcement Learning and Deep Search Area Proposal," Drones, vol. 3, no. 3, p. 58, 2019.

[23] R. J. Bachmann et al., "A biologically Inspired Micro-vehicle Capable of Aerial and Terrestrial Locomotion," Mechanism and Machine Theory, vol. 44, no. 3, pp. 513–526, 2009.

[24] M. Hassanalian et al., "A New Method for Design of Fixed Wing Micro Air Vehicle," Proc. of the Institution of Mechanical Engineers, Part G: J. of Aerospace Eng., vol. 229, no. 5, pp. 837–850, 2015.

[25] S. Roy, S. Biswas and S. S. Chaudhuri, "Nature-inspired Swarm Intelligence and Its Applications," Int. Journal of Modern Education and Computer Science, vol. 6, no. 12, p. 55, 2014.

[26] F. Glover, "Future Paths for Integer Programming and Links to Artificial Intelligence," Computers & Operations Research, vol. 13, no. 5, pp. 533–549, 1986.

[27] Y. Chen et al., "Delivery Path Planning of Heterogeneous Robot System under Road Network Constraints," Computers and Electrical Engineering, vol. 92, p. 107197, 2021.

[28] N. A. Kyriakakis et al., "Moving Peak Drone Search Problem: An Online Multi-Swarm Intelligence Approach for UAV Search Operations," Swarm and Evolutionary Comput., vol. 66, p. 100956, 2021.

[29] X. Yu, C. Li and J. Zhou, "A Constrained Differential Evolution Algorithm to Solve UAV Path Planning in Disaster Scenarios," Knowledge-based Systems, vol. 204, p. 106209, 2020.

[30] V. Gonzalez et al., "Coverage Mission for UAVs Using Differential Evolution and Fast Marching Square Methods," IEEE Aerospace and Electronic Systems Magazine, vol. 35, no. 2, pp. 18–29, 2020.

[31] C. Wu, X. Huang, Y. Luo and S. Leng, "An Improved Fast Convergent Artificial Bee Colony Algorithm for Unmanned Aerial Vehicle Path Planning in Battlefield Environment," Proc. of the 2020 IEEE 16th Int. Conf. on Control & Automation (ICCA), pp. 360–365, Singapore, 2020.

[32] X. Zhen et al., "Rotary Unmanned Aerial Vehicles Path Planning in Rough Terrain Based on Multi-objective Particle Swarm Optimization," J. of Sys. Eng. and Electr., vol. 31, no. 1, pp. 130–141, 2020.

[33] M. D. Phung and Q. P. Ha, "Safety-enhanced UAV Path Planning with Spherical Vector-based Particle Swarm Optimization," Applied Soft Computing, vol. 107, p. 107376, 2021.

[34] B. Tong et al., "A Path Planning Method for UAVs Based on Multi-objective Pigeon-inspired Optimisation and Differential Evolution," Int. J. of Bio-inspired Computation, vol. 17, no. 2, pp. 105–112, 2021.

[35] C. Qu et al., "A Novel Hybrid Grey Wolf Optimizer Algorithm for Unmanned Aerial Vehicle (UAV)

Path Planning," Knowledge-based Systems, vol. 194, p. 105530, 2020.

[36] Y. Alqudsi and M. Makaraci, "UAV Swarms: Research, Challenges and Future Directions," Journal of Engineering and Applied Science, vol. 72, no. 1, p. 12, 2025.

[37] Y. Alqudsi and M. Makaraci, "Exploring Advancements and Emerging Trends in Robotic Swarm Coordination and Control of Swarm Flying Robots: A Review," Proc. of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, vol. 239, no. 1, pp. 180–204, 2025.

[38] S. Alqefari and M. E. B. Menai, "Multi-UAV Task Assignment in Dynamic Environments: Current Trends and Future Directions," Drones, vol. 9, no. 1, p. 75, 2025.

[39] M. Dorigo, G. Di Caro and L. M. Gambardella, "Ant Algorithms for Discrete Optimization," Artificial Life, vol. 5, no. 2, pp. 137–172, 1999.

[40] D. Corne, M. Dorigo, F. Glover, D. Dasgupta, P. Moscato, R. Poli and K. V. Price, New Ideas in Optimization, ISBN: 0077095065, McGraw-Hill Ltd., UK, 1999.

[41] H.-B. Duan, "Ant Colony Algorithms: Theory and Applications," Chinese Science, 2005.

[42] B. H. Sababha et al., "Sampling-based Unmanned Aerial Vehicle Air Traffic Integration, Path Planning and Collision Avoidance," Int. Journal of Advanced Robotic Systems, vol. 19, no. 2, 2022.

[43] A. Al-Mousa, B. H. Sababha, N. Al-Madi, A. Barghouthi and R. Younisse, "Utsim: A Framework and Simulator for UAV Air Traffic Integration, Control and Communication," Int. Journal of Advanced Robotic Systems, vol. 16, no. 5, p. 1729881419870937, 2019.

ملخص البحث:

أصبحت الطّائرات غير المأهولة تلعب دوراً حاسماً في العديد من العمليات المتنوّعة، وبخاصّةٍ عندما يتعين الحفاظ على حياة البشر. ويُعدّ التّخطيط الفعال للمسار والتنسيق الذّاتي أمرين مهمين لأسراب الطّائرات غير المأهولة في المهمّات الدّيناميكية التّعاونية ثلاثية الأبعاد حيث يكون التّكيُّف في الزّمن الحقيقي أمراً أساسيّاً. ويتناول هذا البحث التّحدّي المتمثّل في تحسين عمليات الطّائرات غير المأهولة الّتي تعمل في أسراب عن طريق اقتراح نظام مبتكر هجين للمِلاحة الجوّية يرتكز إلى التّحسين القائم على الطّريقة المتّبعة في مستعمرة النّمل. ويوازن النّظام المقترح بين إيجاد المسار الأمثل والتّحكم الدّيناميكي بتشكيل سِرب الطّائرات بناءً على متطلبات المهمّة التي يقوم بها السّرب. وتتمثّل إحدى المساهمات الأساسية لهذا البحث في نظام المِلاحة الجوّية الهجين المستخدَم الّذي يُعطي الأولوية للتّشكيل المرغوب للسّرب أو طول المسار وزمن الطّيران عبر آليةٍ ذات عتبة، الأمر الّذي يمكّن من التّكيُّف في الزّمن الحقيقي للبيئات المتغيرة. كذلك يوفّر النّظام المقترح دالّةً شاملةً للتّكلفة تعمل على تقييم جودة المسار واستهلاك الوقت واستكمال المهمّة والانحراف عن المسار.

لقد أثبتت التّجارب أنّ النّظام المقترح يُنتج مساراتٍ ذات جودة عالية تصل إلى 97% في معظم الحالات ولا تهبط إلى ما دون 90% حتى في السيناريوهات الّتي تتّسم بالتّحدّيات. وتضمن وحدة تجنُّب الاصطدام استكمال المهمّات بنسبة 100%، سامحةً للطائرات دون طيار بالالتفاف حول العوائق مع الحفاظ على المسار الأمثل. ومن ناحية أخرى، تعمل آلية الحفاظ على التّشكيل بفعاليةٍ على المحافظة على التّشكيل المرغوب لسرب الطّائرات مع التّكيُّف مع العوائق، بحيث يبقى التّغيُّر في التّشكيل في حدود 30% من المدى المسموح به في معظم السيناريوهات، الأمر الّذي يدلّ على قدرة النّظام على الحفاظ على التّشكيل حتّى في البيئات الدّيناميكية. ويُعد هذا البحث إسهاماً في تقدُّم الذّكاء المتعلّق بأسراب الطّائرات غير المأهولة؛ فهو يمكّن من إنجاز العمليات –على نحو فعّال ومستقلّ- في البيئات المعقّدة ثلاثية الأبعاد للحصول على حُلولٍ تعاونية متنوّعة لإكمال المهمّات. وتفتح قابلية النّظام للاستجابة لمتطلبات تشكيلات أسراب الطّائرات بدون طيار الباب أمام إمكانياتٍ جديدةٍ للتطبيقات المرتبطة بأسراب الطّائرات غير المأهولة، مطوِّراً بذلك فعالية المِلاحة الجوّية ومحسّناً التّحكُّم بالتّشكيلات.

JORDANIAN ARABIC TO MODERN STANDARD ARABIC TRANSLATION USING A LARGE MODEL TUNED ON A PURPOSE-BUILT DATASET AND SYNTHETIC ERROR INJECTION

Gheith A. Abandah[1,2], Moath R. Khaleel[1], Iyad F. Jafar[1], Mohammad R. Abdel-Majeed[1], Yousef H. Hamdan[3], Ashraf E. Suyyagh[1], Asma A. Abdel-Karim[1] and Shorouq M. AlAwawdeh[1]

(Received: 2-Mar.-2025, Revised: 7-Jun.-2025, Accepted: 8-Jun.-2025)

ABSTRACT

This paper addresses the challenge of accurately translating Jordanian Arabic into Modern Standard Arabic (MSA) and correcting common linguistic errors. Although MSA is the formal standard for Arabic communication, the widespread use of local dialects in social media and everyday interactions often results in texts laden with spelling and grammatical issues. To overcome these challenges, we present an end-to-end system based on a newly constructed Jordanian Arabic dataset (JODA) comprising 59,135 sentences, as well as the Tashkeela dataset perturbed through synthetic error injection. We employ ByT5, a large pre-trained language model that processes text at the byte level, making it resilient to spelling variations and morphological complexities common in Arabic dialects. Our experimental results show that fine-tuning ByT5 on JODA and a 10% error-injected Tashkeela subset notably improves both BLEU score and character error rate (CER). Combining JODA with the synthetically modified Tashkeela data reduces the CER to 4.64% on the Test-200 test set and 1.65% on the TSMTS test set. Moreover, manual inspections reveal that the model produces correct or near-correct translations in most cases. Finally, we developed a custom smartphone keyboard and a web portal to demonstrate how the system can be made easily accessible to interested users, offering a practical solution for millions of Arabic speakers seeking to produce accurate, diacritized MSA text. This solution is currently limited to the Jordanian dialect; future work will focus on developing similar datasets and solutions for other Arabic dialects.

KEYWORDS

Jordanian Arabic, Modern Standard Arabic, Dialectal translation, Large language models, Synthetic error injection, Natural-language processing, ByT5.

1. INTRODUCTION

Arabic, as the official language of over 20 countries, exhibits a rich linguistic diversity shaped by various regional dialects [1]. In Jordan, everyday communication relies heavily on an informal local dialect distinct from *Modern Standard Arabic* (MSA). While MSA remains the formal standard for written communication in official contexts, many Jordanians encounter difficulties expressing themselves accurately, often producing texts riddled with lexical, morphological, grammatical, syntactic and spelling errors. The proliferation of social media has further amplified this issue, as informal dialects and spelling inconsistencies dominate many online platforms [2].

To address these challenges, modern natural-language processing (NLP) techniques offer promising solutions by leveraging powerful pre-trained large language models. These models have demonstrated remarkable success in understanding and generating text across different languages, including Arabic, when sufficiently trained on diverse and high-quality examples [3]. However, collecting large-scale datasets that reflect the intricacies of informal dialects and embedding them in a unified framework for effective NLP applications pose significant hurdles. Despite recent advancements, current solutions for

1. G. Abandah (Corresponding Author), M. Khaleel, I. Jafar, M. Abdel-Majeed, A. Suyyagh, A. Abdel-Karim and S. AlAwawdeh are with Computer Engineering Department, The University of Jordan, Amman, Jordan. Emails: abandah@ju.edu.jo, moathkhaleel@outlook.com, iyad.jafar@ju.edu.jo, M.Abdel-Majeed@ju.edu.jo, A.Suyyagh@ju.edu.jo, a.abdelkarim@ju.edu.jo and sh.alawawdeh@gmail.com

2. G. Abandah is currently on a sabbatical leave with Department of Computer Science and Engineering, American University of Sharjah, Sharjah, UAE. Email: gabandah@aus.edu

3. Y. Hamdan is with Department of Arabic Language and Literature, The University of Jordan, Amman, Jordan. Email: yousefhamdan4@yahoo.com

translating dialectal Arabic to MSA remain unsatisfactory in terms of accuracy and robustness. This is largely due to the low-resource nature of the problem, as most dialects lack extensive parallel corpora. The development of additional high-quality, dialect-specific resources is therefore essential to improve translation performance and to enable fine-tuning of large models for this challenging task.

In this work, we present an end-to-end system designed to translate Jordanian Arabic into MSA, correct common linguistic mistakes and provide optional diacritization (automatic restoration of missing short vowel marks). Our project involved collecting 59,135 Jordanian Arabic sentences, spanning various dialectal usages and error types, then pairing them with carefully proofread MSA renditions. This dataset was augmented with additional resources to address the scarcity of real-world error examples. By fine-tuning pre-trained large language models on these combined resources, we have created a robust system capable of significantly improving the quality of Jordanian Arabic texts. While this solution is currently limited to the Jordanian dialect, it can be extended to other Arabic dialects as similar datasets become available.

The key contributions of our research can be summarized as follows. First, we provide a new, purpose-built Jordanian Arabic dataset that captures authentic usage and errors, serving as a valuable resource for future NLP research in Arabic. Second, we introduce synthetic spelling errors into a well-known diacritized dataset, enabling the model to learn extensive error patterns beyond the scope of the Jordanian dialect alone. Third, we fine-tune and evaluate a large language model for the translation task, demonstrating its effectiveness in handling informal dialect and spelling issues. Finally, we make the resulting models available through user-friendly web and smartphone applications, allowing Jordanians to produce clear and accurate MSA texts.

After this introduction, Section 2 reviews some related previous work. The approach is outlined in Section 3, followed by the datasets in Section 4, which includes the Jordanian dialect dataset, the Tashkeela datasets with synthetic error injection and the test sets. Section 5 focuses on the models and experiments, describing the model tuning, optimization of synthetic error injection and training using the developed datasets. The results and discussion are presented in Section 6, encompassing a manual inspection of model predictions and a detailed analysis of the results. Finally, the paper concludes with insights, implications and future work in Section 7.

2. Literature Review

This review traces the evolution of machine translation, from rule-based methods to neural architectures, focusing on large language models (*e.g.*, GPT, BERT, T5 and ByT5) and highlighting their key features. Finally, it examines recent approaches for translating Arabic dialects into MSA.

2.1 Evolution of Machine-translation Approaches

Traditional language-translation methods, such as *rule-based machine translation* (RBMT), rely on comprehensive morphological, semantic and syntactic rules for both the source and target languages, requiring extensive expert input [4]. In contrast, *example-based machine translation* (EBMT) maps sentence examples from one language to another without requiring any handcrafted linguistic rules. However, its performance is heavily influenced by the quality of the example database [5]. *Statistical machine translation* (SMT), which was once dominant, integrates phrase, syntax and hierarchical models, but its complexity necessitates combining translation, language and sentence-reordering models [6]-[9]. Hybrid approaches that combine RBMT and SMT have also been explored [10].

Recently, *neural machine translation* (NMT) has become the standard, with widespread adoption by companies, like Google and Microsoft [6], [11]-[13]. NMT employs advanced models, like recurrent neural networks (RNNs), convolutional neural networks (CNNs), encoder-decoder stacks and transformers. With sufficient training data, these models can learn complex linguistic relationships and capture context and semantics from parallel data [12]-[13]. Popular NMT variants, such as bidirectional encoder representations from transformers (BERT) [14]-[15] and text-to-text transfer transformer (T5) [16], are widely used for natural-language processing. NMT systems, initially focused on language pairs, are now capable of translating across 200+ languages [17].

2.2 Large Language Models

Large language models (LLMs), such as generative pre-trained transformer (GPT) [18], BERT [19] and T5 [16] have significantly advanced NLP. These models are based on the *transformer* architecture, which uses self-attention mechanisms to process text in parallel rather than sequentially [20]. This parallel processing enables LLMs to better handle long-range dependencies and complex linguistic structures. Trained on large datasets, LLMs can perform various tasks, like text generation, translation, summarization and error correction, making them versatile tools for language applications. However, models like GPT and T5, which rely on token-based representations, may struggle with out-of-vocabulary words or small typographical errors.

ByT5 is a *token-free* variant of the T5 model that operates directly on byte-level inputs rather than relying on tokenized text [21]. It retains T5's core architecture, consisting of a heavy encoder and a lighter decoder, both built with multi-head self-attention mechanisms and feed-forward neural networks. The encoder converts raw byte sequences into continuous representations, effectively capturing semantic meaning even in the presence of spelling errors or non-standard formatting, while the decoder generates coherent output sequences from these representations. This byte-level processing eliminates the limitations of traditional tokenization, enabling ByT5 to handle diverse languages and character sets more flexibly.

We adopt ByT5 in our solution due to its demonstrated robustness against spelling variations, misspellings and unconventional text formats—characteristics that are prevalent in dialectal and informal Arabic. These strengths make it particularly well-suited for tasks, such as error correction, normalization and diacritization. ByT5 has also proven effective in Arabic NLP applications, including automatic text diacritization [22].

2.3 Recent Approaches to Translating Arabic Dialects

This sub-section reviews recent efforts in Arabic-dialect translation, arranged from broader to more closely related work.

Some studies have focused on translating Arabic dialects to or from English. Alzamzami and Saddik [23] proposed a transformer-based model for translating English tweets into four Arabic dialects. Nagoudi *et al*. [24] developed AraT5, a transformer model pre-trained on large-scale data and fine-tuned on several tasks, including Arabic dialect-to-English translation. AraT5 outperformed the more general multilingual mT5 model in these tasks.

Several other studies have targeted the translation of *multidialectal Arabic content* into MSA. Slim and Melouah [25] addressed the translation of three Maghrebi dialects into MSA using an incremental fine-tuning strategy on a transformer model to address the low-resource nature of dialectal Arabic. Baniata *et al*. [26] proposed integrating RNN-based part-of-speech tagging to enhance translation from Levantine and Maghrebi dialects into MSA, achieving a BLEU score of 43 for Levantine dialects. Alimi *et al*. [27] fine-tuned a variant of AraT5 for translating Levantine and Maghrebi dialects into MSA, reporting high BLEU scores of 43.38 and 64.99, respectively. Notably, both works on Levantine dialects include coverage of the Jordanian dialect.

There is also a line of research focusing on the translation of a *single dialect*, which aligns more closely with our work. Kchaou *et al*. [28], [29] applied data-augmentation techniques to Tunisian-dialect translation and demonstrated that a transformer model outperformed CNN and RNN baselines, achieving a BLEU score of 60. Faheem *et al*. [30] focused on translating the Egyptian dialect into MSA. Their model, trained on 40,000 supervised parallel sentences and supplemented with 35 million monolingual sentences in an unsupervised manner, achieved a BLEU score of 29.5.

Our approach aligns with Refs. [29]-[30] in targeting the translation of a single Arabic dialect into MSA and with Refs. [30], [27], [25], [24] in fine-tuning transformer-based models. However, we distinguish our work by adopting a pretrained, token-free transformer (ByT5), which we fine-tune using a parallel Jordanian-MSA dataset and stochastic error injection. To the best of our knowledge, this is the first work to fine-tune a transformer model specifically for translating not only Jordanian dialect, but also error-prone MSA text—including linguistic and spelling errors—into proper MSA.

3. Approach

Our research implements a comprehensive approach, shown in Figure 1, to accurately translate Jordanian Arabic into MSA, correct spelling mistakes and add diacritics. We began by collecting a dataset of 59,135 Jordanian Arabic sentences, encompassing a broad spectrum of language mistakes and dialectal variations. Working with Arabic-language specialists, we corrected mistakes, translated colloquial forms into MSA and thoroughly proofread all samples.

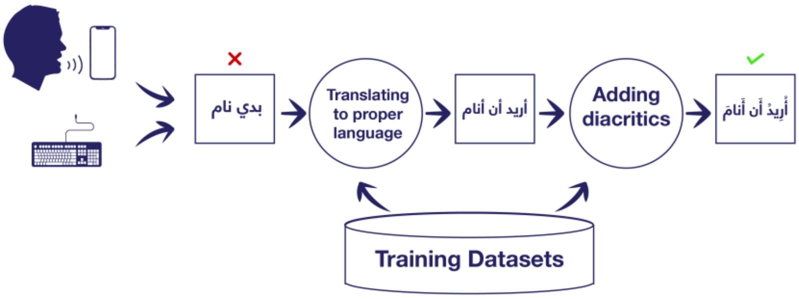


Figure 1. End-to-end approach for Jordanian Arabic to diacritized MSA conversion.

Building on this dataset, we further expanded it using the diacritized Tashkeela Classical Arabic dataset [31]. Synthetic spelling errors were introduced into Tashkeela via random error injection, enhancing the model’s capacity to handle real-world misspellings.

We fine-tuned pre-trained ByT5 models, leveraging their broad language understanding developed through training on large datasets. *Pre-trained* models like ByT5 are neural networks designed to learn general language representations, enabling them to understand and generate text effectively. *Fine-tuning* involves adapting these models to specific tasks by training them further on smaller, task-specific datasets. In our case, one model was fine-tuned to translate Jordanian Arabic into proper MSA, specializing in this linguistic transformation. Additionally, we explored another model inspired by Al-Rfooh *et al*. [22] to optionally add diacritics, though this lies outside the scope of this paper [32], [33].

Upon completion of training, the models exhibited strong performance in error correction, translation and diacritization. Finally, we integrated these trained models into both internet-based and smartphone applications, exploring open access for Jordanian users seeking reliable and accurate linguistic support. Despite its effectiveness, the approach faces limitations including the cost of developing high-quality parallel datasets and the computational intensity of training and deploying large models like ByT5, which constrains scalability and performance on resource-limited devices.

4. Datasets

This section describes the datasets used for training and evaluating our approach. Sub-section 4.1 details the newly developed Jordanian dialect dataset [34], Sub-section 4.2 introduces the Tashkeela-based datasets alongside synthetic error injection and Sub-section 4.3 outlines the test sets employed to measure model performance.

4.1 Jordanian Dialect Dataset

One key contribution of this research is the development of the *Jordanian dialect dataset* (JODA). This parallel dataset was constructed by collecting Arabic sentences that contain various linguistic mistakes or are in the informal Jordanian dialect. Each collected sentence was then paired with its corresponding correct MSA equivalent. The dataset draws from three primary sources to ensure diversity and authenticity (Figure 2). Approximately 72% of JODA comes from social-media platforms: YouTube, Facebook, Instagram and Twitter (X), covering various topics like economics, society and politics. Additionally, 22% are sentences selected from publicly available Arabic-dialect datasets: Dialectal Arabic tweets (DART) dataset [35] and the Shami dialect corpus (SDC) [36]. The remaining 6.6% of the dataset consists of transcriptions of eight short Jordanian movies capturing cultural and linguistic diversity.

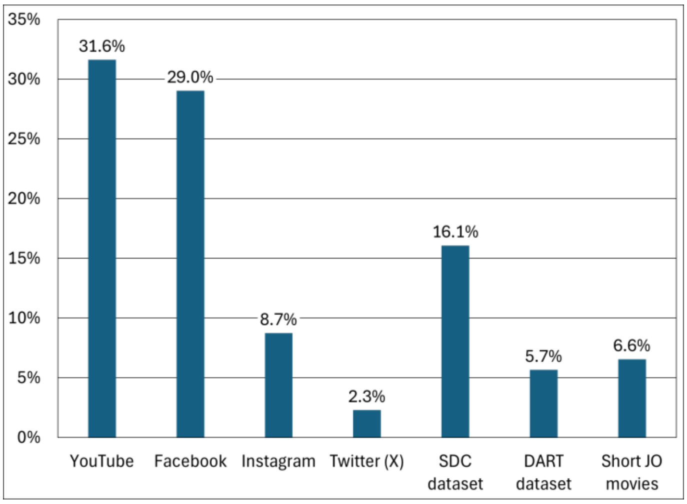
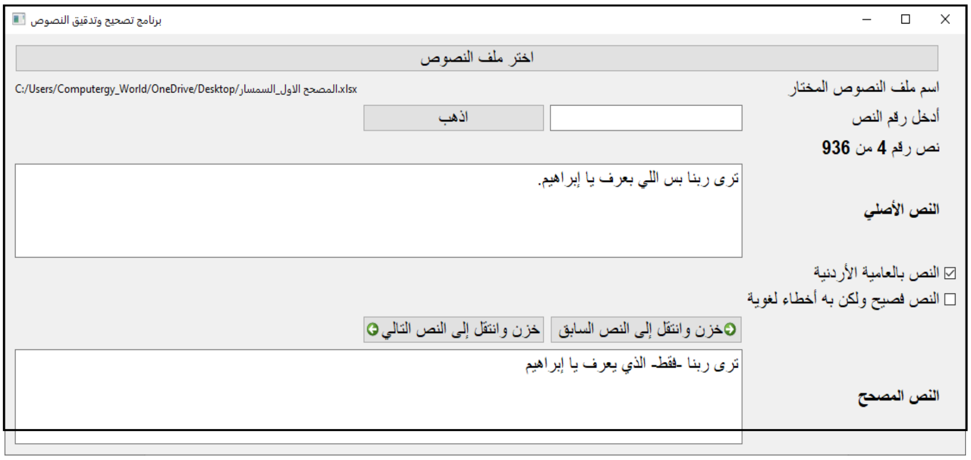


Figure 2. Composition of JODA dataset by sample source.

The collected samples from the various sources underwent extensive preprocessing, which included removing irrelevant elements, duplicates, emojis and unnecessary characters, as well as segmenting the text into meaningful sentences. Each sentence was manually reviewed to ensure proper segmentation and meaningful content, retaining only those in MSA containing mistakes or in the Jordanian dialect. The sentences range from 2 to 277 characters, reflecting natural-language usage. Arabic linguistic experts contributed to the development of this parallel dataset by providing either corrections for MSA sentences containing mistakes or translating Jordanian dialect sentences into MSA. For a broader linguistic perspective on this translation from Jordanian Arabic into MSA, interested readers are referred to [37].

While JODA was designed to be as representative as possible, some bias may exist. The Jordanian dialect varies by region, but most data likely reflect the central region, where most of the population resides. Northern and southern dialects may be underrepresented. Additionally, the heavy reliance on social-media content may skew the language toward younger, urban speakers. We also used curated datasets and film transcripts, which may not fully capture spontaneous speech. Despite these limitations, we made deliberate efforts to ensure diversity in topics, sources and linguistic styles across the dataset.

To expedite dataset corrections, we developed a custom PyQt-based GUI specifically tailored for Arabic-text processing. The tool is employed by both experts and auditors, who can selectively load dataset files, navigate individual sequences, classify entries and either provide or validate corrections. This interface was designed to accommodate right-to-left scripts and fully support Arabic display and parsing, ensuring minimal friction during annotation and review. Additionally, it offers streamlined functionality for saving changes, flagging problematic entries and maintaining detailed logs of edits. Figure 3 illustrates the tool's layout and features, highlighting its user-friendly design.


برنامج تصحيح وتدقيق النصوص
اختر ملف النصوص
اذهب
خزن وانتقل إلى النص السابق
خزن وانتقل إلى النص التالي

اسم ملف النصوص المختار C:/Users/Computergy_World/OneDrive/Desktop/المصحح الاول_السمسار.xlsx

أدخل رقم النص

نص رقم 4 من 936

النص الأصلي: ترى ربنا بس اللي بعرف يا إبراهيم.

☑ النص بالعامية الأردنية

☐ النص فصيح ولكن به أخطاء لغوية

النص المصحح: ترى ربنا -فقط- الذي يعرف يا إبراهيم

Figure 3. Correction and auditing tool.

The final version of the JODA dataset comprises 59,135 sentences, with 62.4% in the Jordanian dialect and 37.6% in MSA containing mistakes (Table 1). This version was randomly split into three sub-sets; 91.5% of the sentences were included in the training sub-set, while the remaining sentences were evenly divided between the validation and test sub-sets (2,500 sentences each).

Table 1. Distribution of the JODA dataset by sentence type and data split. The "Total" row and column show the number of sentences and their percentages relative to the entire dataset.

Sentence type	Training subset	Validation subset	Test subset	Total
Jordanian dialect	33,767	1,560	1,559	36,886 (62.4%)
MSA containing mistakes	20,368	940	941	22,249 (37.6%)
Total	54,135 (91.5%)	2,500 (4.2%)	2,500 (4.2%)	59,135 (100%)

During this split, stratification was applied to ensure representative sampling of the various sentence sources and types across the three sub-sets. Figure 4 shows the number of sentences in the three dataset sub-sets, categorized by sentence source and sentence type.

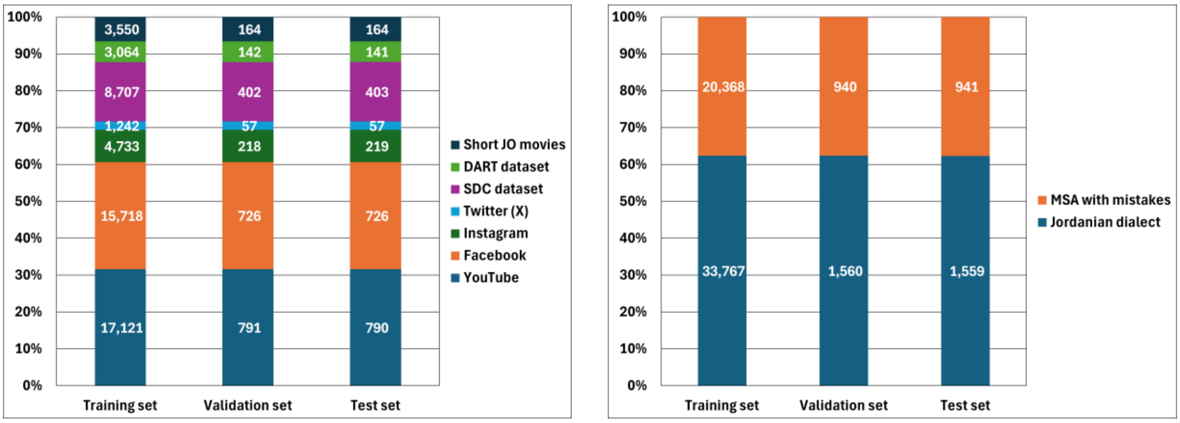


Figure 4. Stratified split of the JODA dataset by sentence source (left) and sentence type (right).

4.2 Tashkeela Datasets and Synthetic Error Injection

In addition to JODA, the proposed model was also trained using the Tashkeela Clean-50 and Clean-400 datasets, which primarily contain diacritized Classical Arabic text. The *Tashkeela Clean-50* dataset, developed by Fadel *et al*. [38], comprises 50,000 training sequences extracted from the original Tashkeela dataset [31]. These sequences were filtered to ensure a diacritic-to-character ratio of at least 80% and were processed using heuristics, such as diacritic correction, removal of English letters and isolation of numbers from words. Abdel-Karim and Abandah [39] expanded this dataset, creating the *Tashkeela Clean-400* dataset with 400,000 training sequences. Both datasets include, in addition to their respective training sets, the same validation sub-set of 2,500 sequences and the same test sub-set of 2,500 sequences. These datasets were truncated to a maximum sequence length of 512 bytes to maintain consistency with the JODA dataset.

These datasets were further processed into input-target pairs by introducing synthetic stochastic spelling errors [40]. Two methods were employed for error injection: directed error injection and general error injection. *Directed error injection* focuses on "soft spelling mistakes," which are common among Arabic speakers and learners due to the complexity of Arabic orthography. Following the approach of Abandah *et al*. [41], this method specifically targets frequent mistakes involving words with different forms of *hamza* (ء، أ، إ، آ، ؤ، ئ) and words ending with similarly pronounced letters (ه، ة، ت) and (و، وا). Errors were introduced based on their position within words, using three injection rates (2.5%, 10% and 40%) to evaluate their impact on model training. This method ensures that artificial errors closely resemble common real-world mistake patterns.

General error injection extends directed error injection by incorporating a broader range of spelling error patterns, including letter deletion, insertion, swapping and replacement. This approach introduces stochastic errors of selected probability, also evaluated at three injection rates. These errors simulate various mistake patterns found in Arabic text, allowing the model to learn corrections for a variety of mistake types. By combining directed and general error injection methods, the dataset is designed to improve the model's ability to correct both specific and general spelling mistakes. Table 2 provides the statistics for JODA, Tashkeela Clean-50 and Tashkeela Clean-400.

Table 2. Statistics of the datasets used.

Metric	**JODA**	**Tashkeela Clean-50**	**Tashkeela Clean-400**
Size (MB)	10.3	12.80	102.50
Number of sequences	59,135 pairs	50,000	400,000
Word count	1.14×10^6	1.62×10^6	12.95×10^6
Character count	5.92×10^6	7.34×10^6	58.67×10^6
Average number of words per sequence	9.6	32.40	32.36

4.3 Test Sets

To thoroughly evaluate the developed model's performance, we use three test sets. The first is the *JODA test subset* described above, which is critical for assessing performance and selecting optimal configurations. The second, *Test-200* [41], contains 200 sentences with "soft" spelling mistakes, averaging 6.5 mistakes per sentence and a 5%-character mistake rate. This set is particularly useful for fine-tuning the model when training on data with directed error injection.

We also developed a third test set, the *Tashkeela spelling mistakes test set* (TSMTS), derived from the 2,500 sequences of the Tashkeela test set. Each sequence in the Tashkeela test set serves as a target, paired with an input sequence generated by applying the general error injection described above to the original sequence. A character error rate of 5% was used to ensure that TSMTS mirrors the Test-200 set. This test set provides a benchmark for evaluating general error injection.

5. MODELS AND EXPERIMENTS

We selected ByT5 for its robustness in handling multilingual text and noisy inputs, operating at the byte level without tokenization. This language-agnostic approach ensures high flexibility across diverse languages and scripts [21]. ByT5's strengths include resilience to misspellings and compatibility with low-resource languages. For our experiments, we utilized the Small and Base model sizes due to their lower computational requirements. We did not use larger models, as the significantly higher computational cost was not justified by the relatively small performance gains reported in prior work [21]-[22]. Table 3 summarizes the architectures of both models.

Table 3. Architectures of the two ByT5 models explored.

Criterion	**Small**	**Base**
Number of parameters	300M	582M
Encoder/decoder layers	12 / 4	18 / 6
Feed forward dimension (d_{ff})	3,584	3,968
Model dimension (d_{model})	1,472	1,536

For evaluation, we used the BLEU and CER metrics. BLEU (*bilingual evaluation understudy*) measures the similarity between the model's output and reference translations by comparing overlapping n-grams, providing a score for translation quality. CER (*character error rate*) calculates the percentage of character-level errors, such as substitutions, insertions and deletions, in the model's output compared to the reference, offering insight into fine-grained accuracy.

The experiments were conducted on Google's Colab Pro Plus platform, utilizing TPU v2 units to accelerate the training process. The programming language used was Python 3.7.13, with TensorFlow 2.12.0 as the primary library.

The following sub-sections detail the experiments and results for tuning the ByT5 model, refining the error injection approach used in preparing the Tashkeela datasets and training the optimized model on a combined dataset of JODA and Tashkeela.

5.1 Tuning the ByT5 Model

The ByT5 model comes in multiple sizes and offers numerous hyperparameters that can be adjusted to improve performance, depending on the target task. In this work, we began by establishing a baseline model and then explored various hyperparameter configurations to arrive at a final tuned model. Table 4 summarizes the explored hyperparameter options and lists the values used in both the baseline and the tuned models. The following paragraphs describe the tuning experimental procedure and summarize the results.

Table 4. Explored ByT5 hyperparameters, evaluated options and the corresponding values for both the baseline and the tuned models.

Hyperparameter	Options	Baseline model	Tuned model
Model size	Small, Base	Small	Base
Batch size	128, 256, 512	256	128
Learning rate	0.0001, 0.003, 0.01	0.003	0.003
Optimizer	AdaFactor, Adam Weight Decay	AdaFactor	Adam Weight Decay

We fine-tuned the model using the JODA dataset, which includes the two implicit tasks: translating Jordanian Arabic into MSA and correcting linguistic mistakes. Our initial experiment assessed the baseline model's performance. Figure 5 shows the BLEU scores for both the training and validation sub-sets over successive training steps, where each step corresponds to a batch of a specified size (256 for the baseline model). During this experiment and others, we observed that the model exhibits overfitting, with the BLEU score on the training sub-set approaching 100 while the validation score plateaus at a lower level. To mitigate overfitting, we halted training when the validation score ceased to improve and adopted the model weights from the training step with the highest validation score. The baseline model achieves its highest BLEU score of 57.49 at the 3,000th training step, with a corresponding BLEU score of 56.07 on the JODA test sub-set.

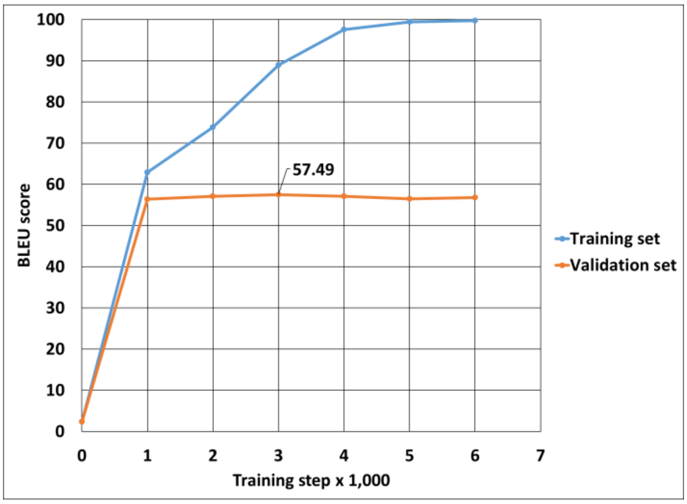


Figure 5. Training curves of the baseline model trained on JODA dataset.

In our fine-tuning experiments, we followed the methodology described in [42], which involves adjusting one hyperparameter at a time and comparing the resulting performance to the baseline. Although this "coordinate ascent" approach may overlook higher-order interactions between parameters (for instance, a different learning rate might produce better results with a larger model size), a full factorial design would be expensive, as it would require 2×3×3×2 = 36 experiments. Once the best individual hyperparameters were identified, we used those values to train the final model.

Table 5 provides the outcomes of the seven fine-tuning experiments involving the four hyperparameters. Each row presents the examined hyperparameter option, the training step where the validation score peaked and the corresponding BLEU scores for both the validation and test sub-sets. Based on these results, the optimal hyperparameters for the tuned model are those shown in Table 4.

Table 5. Results of fine-tuning the hyperparameters of the ByT5 model.

Hyperparameter	**Option**	**Best training step**	**BLEU score (validation)**	**BLEU score (test)**
Model size	Small (baseline)	3,000	57.49	56.07
	Base	7,000	59.01	**57.08**
Batch size	**128**	4,000	57.54	**56.43**
	256 (baseline)	3,000	57.49	56.07
	512	1,000	58.03	56.32
Learning rate	0.0001	20,000	56.53	55.38
	0.003 (baseline)	3,000	57.49	**56.07**
	0.01	9,000	55.13	53.90
Optimizer	AdaFactor (baseline)	3,000	57.49	56.07
	Adam Weight Decay	2,000	57.60	**56.29**

When trained on JODA, the tuned model achieves its highest BLEU score of 59.07 at the 3,000th training step on the validation sub-set, yielding a BLEU score of 57.77 on the test sub-set, which represents a 3% improvement over the baseline model.

5.2 Tuning Error Injection

The performance of a model trained with synthetic error injection is influenced by the chosen injection rate in [41]. This sub-section describes the experiments conducted to determine optimal rates and summarizes the results. In these experiments, we trained the tuned model on the Tashkeela datasets and evaluated it on the Test-200 or TSMTS test sub-sets. As in previous experiments, we stopped training once the validation score ceased to improve and adopted the model weights from the training step that produced the highest validation score for final evaluation.

5.2.1 Directed Error Injection

We explored three rates for directed error injection: 2.5%, 10% and 40%. In each experiment, the model was trained on a Tashkeela dataset with the specified rate of directed error injection, then evaluated on Test-200. We selected Test-200, because it contains common real-life spelling mistakes, like those introduced by the directed method.

Table 6 shows the results obtained using the Clean-50 dataset, where a 10% injection rate yielded the lowest CER on Test-200 (1.37%). Note that the CER on the validation sub-set increases with higher error rate in this sub-set. The table also reports results for training on the larger Clean-400 dataset at the same 10% rate, which further reduced the CER on Test-200 to 1.23%. This improvement demonstrates that a larger dataset provides the model with more examples of spelling variations, enhancing its ability to correct errors.

Table 6. Results of tuning directed error injection.

Dataset	**Error injection rate**	**Best training step**	**CER (validation sub-set)**	**CER (Test-200)**
Clean-50	2.5%	8,000	0.03%	2.26%
	10%	8,000	0.07%	**1.37%**
	40%	8,000	0.14%	1.53%
Clean-400	**10%**	13,000	0.04%	**1.23%**

5.2.2 General Error Injection

For general error injection, we similarly evaluated three rates: 2.5%, 10% and 40%. In each experiment, the model was trained on a Tashkeela dataset with the chosen rate of general error injection and tested on TSMTS. TSMTS was selected, because it contains synthetic spelling errors comparable to those produced by the general error injection method.

As shown in Table 7, using the Clean-50 dataset with a 10% injection rate resulted in the lowest CER on TSMTS (1.77%). When the model was trained on the larger Clean-400 dataset at the same 10% rate, the CER dropped further to 1.28%, indicating that a bigger training sub-set helps the model better generalize to diverse error patterns.

Table 7. Results of tuning general error injection.

Dataset	Error injection rate	Best training step	CER (validation sub-set)	CER (TSMTS)
Clean-50	2.5%	14,000	0.80%	2.09%
	10%	10,000	2.99%	**1.77%**
	40%	12,000	16.69%	2.78%
Clean-400	**10%**	14,000	2.20%	**1.28%**

Overall, these experiments confirm that a 10% error injection rate is most effective for both directed and general error injection methods. Furthermore, training on a larger dataset (Clean-400) yields better results, highlighting the importance of data size in improving the model's ability to correct spelling errors.

5.3 Training Using JODA and Tashkeela Datasets

To further improve the model's performance on both translating Jordanian Arabic into MSA and correcting linguistic mistakes, we explored training on a combined dataset. Specifically, we combined JODA with the 10% directed error-injected Clean-50 dataset and the 10% general error-injected Clean-50 dataset. As usual, this combined dataset was partitioned into training, validation and test sub-sets by merging the corresponding sub-sets from the three individual datasets.

Figure 6 illustrates the training curves for the tuned model on this combined dataset. The BLEU score for the training sub-set continued to improve with more training steps, whereas the validation score increased more slowly. Training was halted at Step 15,000 due to the slowing improvement on the validation sub-set and the widening gap between the training and validation scores. At this step, the validation BLEU score reached 87.57, which is considerably higher than the BLEU score of 59.07 achieved by training solely on the JODA dataset. This apparent discrepancy arises, because the validation sub-set in the single-dataset experiment contains only JODA sentences, which tend to be more challenging than the mixed-validation sub-set here. Indeed, when evaluated on the JODA test sub-set, this model achieves a BLEU score of only 57.39.

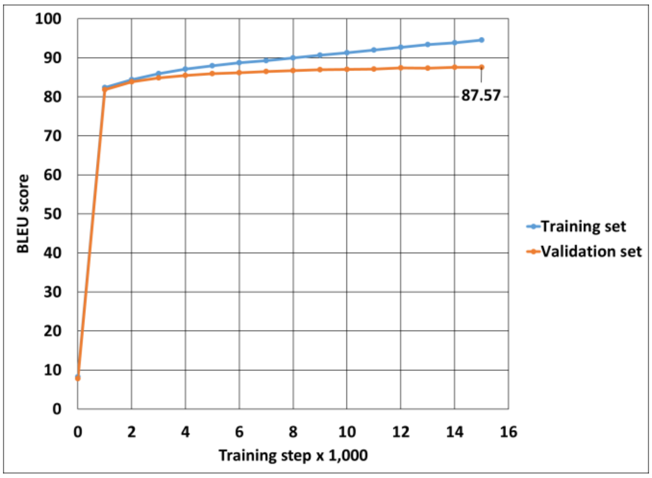


Figure 6. Training curves of the tuned model trained on JODA and Clean-50 datasets.

To assess whether the model could generalize beyond JODA, we compared the CER on the Test-200 and TSMTS test sub-sets between (1) the model trained on JODA only and (2) the model trained on the combined dataset. As shown in Figure 7, the combined-dataset model generalizes more effectively: the CER on Test-200 improves from 6.37% to 4.64% and on TSMTS from 11.95% to 1.65%. This result demonstrates the model's enhanced ability to correct common and general spelling mistakes.

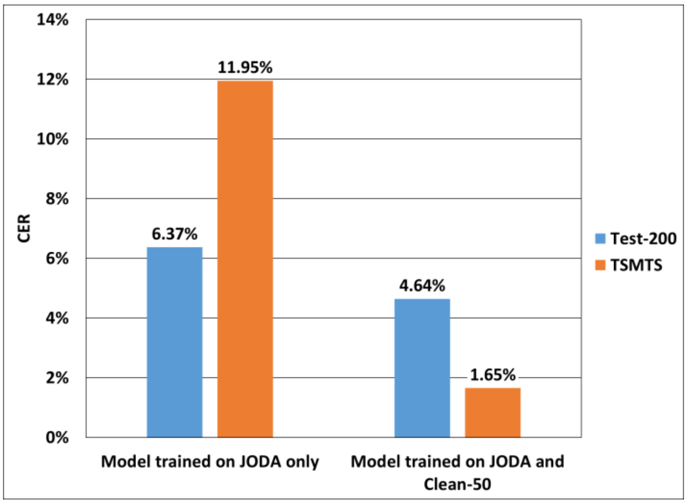


Figure 7. CER on two test sub-sets for the tuned model trained with two dataset configurations.

We also examined the model's performance when trained on a combined dataset consisting of JODA and the larger Tashkeela Clean-400 dataset. In this case, the model yielded a lower BLEU score of 53.24 on the JODA test sub-set, likely due to an imbalance between Jordanian dialect and MSA content in the larger dataset. Consequently, we adopted the model trained on the combined JODA and Clean-50 datasets.

6. Results and Discussion

Table 8 compares the three main models trained under different conditions to evaluate their performance in translating Jordanian Arabic into MSA and correcting linguistic mistakes. The baseline model, trained only on JODA, reaches a high BLEU 56.07, because many JODA references differ from the inputs by only minor spelling errors; n-gram overlap is therefore already near-saturated. However, CER exposes those spelling mistakes: the baseline scores 6.58% on Test-200 and 12.41% on TSMTS. Hyper-parameter tuning (still on JODA) nudges BLEU to 57.77 and trims CER to 6.37% and 11.95%. Adding the Clean-50 corpus introduces many perfectly spelled targets and forces the model to generalize beyond JODA. BLEU on the JODA test sub-set dips slightly to 57.39, but CER falls sharply to 4.64% on Test-200 and 1.65% on TSMTS. Thus, while BLEU shows only marginal gains, the steep CER reduction demonstrates that the final model corrects errors more aggressively and transfers this ability to unseen text, striking a practical balance between fluency (BLEU) and accuracy (CER).

Table 8. Comparison of the three main experiments on three test sub-sets.

Model	**Training time in hours**	**BLEU score (JODA test set)**	**CER (Test-200)**	**CER (TSMTS)**
Baseline model (trained on JODA)	1.5	56.07	6.58%	12.41%
Tuned model (trained on JODA)	2.1	57.77	6.37%	11.95%
Tuned model (trained on JODA + Clean-50)	10.3	57.39	4.64%	1.65%

Although large language models deliver impressive results, they often come with substantial computational costs. Table 8 lists the training times for the three models, showing that the tuned model employing the base ByT5 requires longer training than the baseline model, which uses the smaller ByT5 variant. Moreover, the final model trained on the combined larger dataset increases training time to around five times that of the tuned JODA-only model. In the prediction mode, the trained model translates a single Jordanian dialect sentence into MSA in approximately 1.5 seconds.

6.1 Comparison with Previous Work

Table 9 presents a comparative overview of recent efforts in translating Arabic dialects into MSA, highlighting the methods, datasets and BLEU scores reported for different dialects. Compared to previous studies, our work utilizes JODA—the largest Arabic mono-dialect dataset focused on Jordanian

Arabic—and achieves the highest BLEU score reported for Levantine dialects, demonstrating the effectiveness of our fine-tuned ByT5 model with stochastic error injection.

Table 9. Comparison with previous work in translating Arabic dialects into MSA.

Work	Method	Dataset/Size	BLEU score
Baniata *et al.* [26]	RNN with POS tagging	Multidialectal / 36K	43 for Levantine dialect
Alimi *et al.* [27]	Fine-tuning pretrained AraT5 model	Multidialectal / 69K	43.38 for Levantine dialect
Kchaou *et al.* [29]	Transformer with data augmentation	Tunisian dialect / 36K	60
Faheem *et al.* [30]	Pretraining followed by fine-tuning a transformer	Egyptian dialect / 40K	29.5
This work	Fine-tuning ByT5 and stochastic error injection	JODA / **59K**	**57.77**

6.2 Manual Inspection of the Model Predictions

Throughout this research, we primarily relied on BLEU and CER scores to evaluate translation and correction quality. While these metrics are generally appropriate, they can also underestimate the model's performance—especially given Arabic's linguistic richness, where a single sentence can be correctly translated into multiple valid forms. Consequently, the model may receive a penalty if it produces a legitimate translation that does not exactly match the target sequence.

Our main objective is to develop a model capable of translating Jordanian Arabic into MSA, rather than replicating a specific reference translation. To gain deeper insight into the model's real-world performance, we manually reviewed 100 randomly selected predictions from the JODA test sub-set, comparing them with the expert-provided target sentences. Table 10 provides sample inputs, target sentences and model outputs, categorized into three classes: (1) correct translations with contextual variations, (2) translations exhibiting minor differences and (3) translations with more substantial discrepancies, often reflecting synonym usage.

Table 10. Overview of 100 manually audited predictions.

	Prediction classification	Count	Example input sentence	Target sentence	Predicted sentence
1	Correct predictions	51	بامكاني انزل لمستواك و احكي بالطريقة الهمجية الي بتحكي فيها ،	بإمكاني أن أنزل إلى مستواك وأتكلم بالطريقة الهمجية التي تتكلمين بها	بإمكاني أن أنزل إلى مستواك وأتكلم بالطريقة الهمجية التي تتكلم بها
2	Small differences	40	المذيعه مش معاه ابدا ما خلتله مجال يحكي أسلوبها مزعج	المذيعة ليست معه أبدا، لم تترك له مجالا ليتكلم، أسلوبها مزعج	المذيعة ليست معه أبدا، لم تترك له مجالا ليتكلم بأسلوبها مزعج
3	Large differences	9	يا بنت الحلال مهو طول عمره هون شو عملك يعني	يا بنت الحلال، هو طوال عمره هنا، ماذا فعل لك إذا؟	يا بنت الحلال فهو طوال عمره هنا ماذا عمل لك إذا؟

Notably, the manual review revealed that the model's true performance exceeds the CER evaluation of 12.39%. When correct translations are not counted as errors, the CER declines to 5.56%. Although these findings are already encouraging, they further confirm that the model's practical performance is stronger than what traditional metrics alone may indicate.

Despite strong overall results, the final model shows two noteworthy limitations. First, ambiguous short sentences, where number or gender is underspecified, can lead to incorrect disambiguation in MSA. Second, because training data was restricted to Jordanian Arabic and MSA, the system struggles with inputs that mix codes (*e.g.* bilingual Arabic and English sentences). Addressing these issues will require additional annotated data and explicit modeling of ambiguity.

6.3 Evaluation on Additional Jordanian and Other Levantine-dialect Sentences

We conclude that our final model effectively translates Jordanian Arabic into MSA and corrects linguistic errors.

Table 11 provides examples of input sentences that were not part of the JODA dataset, alongside the model’s corresponding predictions and evaluations. In most instances, the model produces high-quality translations, occasionally exhibiting only minor errors.

Table 11. Sample input sentences, model predictions and evaluations.

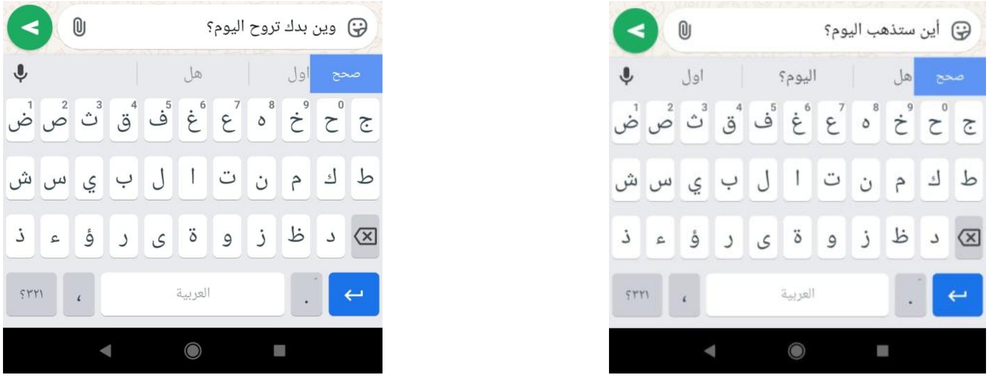
Input sentence	Predicted sentence	Evaluation
وين بدك تروح اليوم؟	أين ستذهب اليوم؟	Correct
شلونك يا زلمة؟	كيف حالك يا رجل؟	Correct
الجو دفا اليوم	الجو دفأ اليوم	Minor error
قديش الساعة هسا؟	كم الساعة الآن؟	Correct
امبارح كنت تعبان كتير	البارحة كنت متعبا كثيرا	Correct
اشتقتلك يخوي	اشتقت لك يا أخي	Correct
شو وراك يا زلمة	ما وراءك يا رجل؟	Correct

To probe generalizability beyond Jordanian Arabic, we manually assembled fifteen unseen sentences, five each in Palestinian, Syrian and Lebanese dialects and translated them with the final model. The model successfully rendered all sentences into grammatical MSA, confirming that its byte-level representations capture many shared Levantine structures. Accuracy, however, was lower than for Jordanian input: output fluency occasionally suffered from dialect-specific lexemes and translations of Lebanese examples that contained French loanwords (*e.g. ascenseur*, *parfum*). These observations suggest that while the system generalizes reasonably within the Levantine group, expanded training data would be needed for consistently high performance across all regional variants.

6.4 Accessing the Model *via* Smartphones and Web Portal

To provide the model’s Jordanian Arabic-to-MSA translation and Arabic error correction capabilities to end users, we developed a custom keyboard and a web-based portal. The model is hosted on a server and communicates with both the keyboard and web interface using the Flask framework. When users enter text and request a correction, the front end sends this text to the Flask API, which processes it through the trained model and returns the corrected output in real time. This setup ensures a responsive, lightweight user experience by offloading complex processing tasks to the server.

The custom keyboard, called *AI Board*, was developed using the open-source OpenBoard project [43] for Android and the KeyboardKit 7.9.8 package [44] for iOS. As shown in Figure 8, it features a dedicated “صحح” (Correct) button that translates or corrects any text entered *via* the Arabic keyboard or microphone, seamlessly converting Jordanian dialect into MSA.

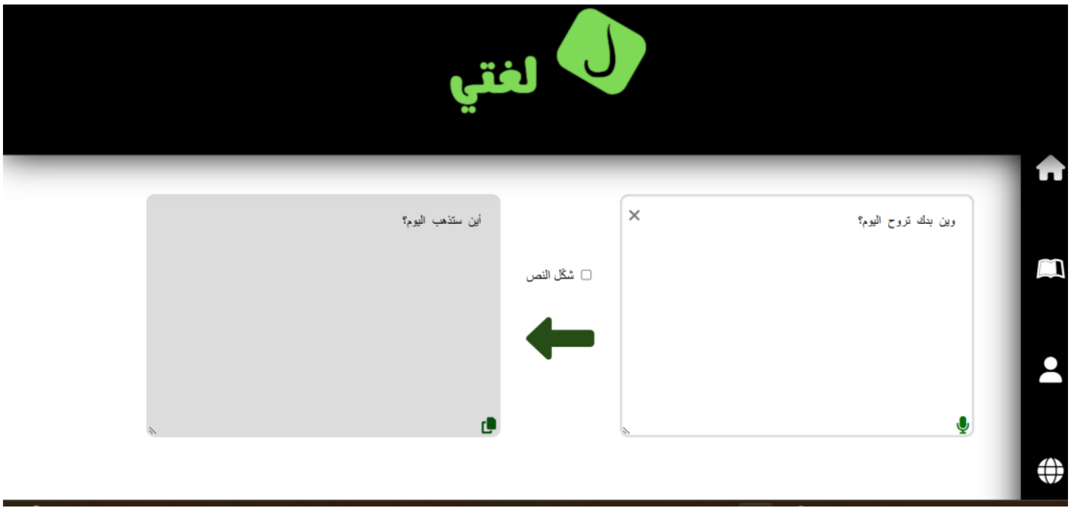


وين بدك تروح اليوم؟

أين ستذهب اليوم؟

Figure 8. The AI Board translating a Jordanian dialect sentence (left) into MSA (right).

We also built a web-based portal named *Loghati* (Arabic for “my language”) to offer open access to this solution. In addition to the translation feature shown in Figure 9, the portal provides references for Arabic grammar and spelling rules. It supports keyboard and microphone input, allows copying of translated text and is built using HTML, CSS, JavaScript, Bootstrap and React.js.



وين بدك تروح اليوم؟

□ شكّل النص

أين ستذهب اليوم؟

Figure 9. *Loghati* interface translating a sentence entered from the Jordanian dialect into MSA.

7. Conclusions

In this work, we presented an end-to-end system for translating Jordanian Arabic into MSA, correcting common linguistic errors and optionally adding diacritics. We began by collecting a large dataset of Jordanian dialect sentences (JODA), comprising diverse dialectal usages and error types. Each entry was carefully curated by Arabic-language experts, ensuring accurate MSA equivalents. To further enhance performance, we incorporated additional resources from Tashkeela, introducing synthetic spelling errors to increase the model's exposure to spelling mistake patterns and ability to correct Arabic text.

Our experiments employed the ByT5 architecture—well-suited for Arabic dialect processing due to its byte-level input handling—to achieve robust translation and correction. Through systematic fine-tuning of hyperparameters, we identified a tuned combination that improved BLEU scores on the JODA test subset by 3% over a baseline system. Furthermore, integrating the error-injected Tashkeela dataset enhanced the model's generalization, as evidenced by significant improvements in CER across various benchmark test sub-sets.

Beyond quantitative metrics, manual reviews revealed that the model's output often matched or closely approximated expert translations, underscoring its practical effectiveness. Finally, we made the resulting models accessible via a custom keyboard and a web portal, thus offering user-friendly solutions that expand the reach and impact of this research. These solutions will first be introduced in pilot scenarios to collect user feedback, enabling further refinement before a wider public launch.

Our approach, trained on JODA, the largest mono-dialect corpus, achieves the highest reported BLEU for Levantine dialects, outperforming prior Arabic-dialect-to-MSA systems. Nevertheless, it can mishandle number/gender ambiguities and code-mixed Arabic-English inputs, pointing to the need for richer data and explicit ambiguity modeling. Tests on other Levantine samples show reasonable cross-dialect transfer, but reduced accuracy with dialect-specific or French-derived terms, underscoring the need for further adaptation to other Levantine varieties.

One avenue for future research is to explore larger, more advanced ByT5 or similar transformer-based models. Increasing model parameters could enhance their capacity to capture a broader range of linguistic nuances, especially when trained on significantly expanded datasets.

While large models often produce superior results, they may be too resource-intensive for deployment on mobile devices with limited computational capabilities. A natural extension is to investigate smaller, more efficient architectures, employing techniques, like model distillation or quantization, to reduce size and inference time. This would facilitate on-device processing, ensuring offline usability and faster, more personalized performance.

Currently, we rely on two separate models whenever the corrected text also needs diacritization. However, modern-language models are powerful enough to handle multiple tasks within a single architecture—one task for correction only and another task for both correction and diacritization. This approach eliminates the need to chain two distinct models, which will reduce latency. Future work could integrate the developed translation capabilities into Arabic chatbots [45] to enable them to automatically understand and translate user inputs from dialectal Arabic into MSA, thereby enhancing their generality and linguistic accuracy.

ACKNOWLEDGEMENTS

The authors would like to thank the Jordanian Scientific Research and Innovation Support Fund for their support of this work. We are deeply grateful to data scientists Boshra Sadder, Raghda Hasan and Shorouq AlAwawdeh for their meticulous efforts in collecting and processing the JODA dataset. Additionally, we sincerely thank Arabic-language experts Dalal Alzuheiri, Hussein Adwan, Sondos Marian and Islam Eid for providing the translations of the collected Jordanian text into MSA. We also extend our thanks to application developers Abrar Samara, Eman AbuKhater and Maysa Al-Hwayan for their valuable contributions.

This work has been carried out during sabbatical leave granted to Gheith Abandah from the University of Jordan during the academic year 2024/2025.

REFERENCES

[1] G. Khan, M. P. Streck and J. C. Watson, The Semitic Languages: An International Handbook, vol. 36, Walter de Gruyter, 2011.

[2] G. Abandah, M. Khedher, W. Anati, A. Zghoul, S. Ababneh and M. S. Hattab, "The Arabic Language Status in the Jordanian Social Networking and Mobile Phone Communications," Proc. of the 7th Int'l Conf. on Information Technology (ICIT 2015), pp. 449–456, 2015.

[3] B. Min et al., "Recent Advances in Natural Language Processing via Large Pre-trained Language Models: A Survey," ACM Computing Surveys, vol. 56, no., pp. 1–40, 2023.

[4] S. Castilho et al., "Is Neural Machine Translation the New State of the Art?" The Prague Bulletin of Mathematical Linguistics, vol.108, no. 1, pp. 109–120, 2017.

[5] M. S. H. Ameur, F. Meziane and A. Guessoum, "Arabic Machine Translation: A Survey of the Latest Trends and Challenges," Computer Science Review, vol. 38, DOI: 10.1016/j.cosrev.2020.100305, 2020.

[6] H. Wang et al., "Progress in Machine Translation," Engineering, vol. 18, pp. 143–153, 2021.

[7] I. Rivera-Trigueros, "Machine Translation Systems and Quality Assessment: A Systematic Review," Language Resources and Evaluation, vol. 56, no. 2, pp. 593–619, 2022.

[8] G. Doğru, "Statistical Machine Translation Customization between Turkish and 11 Languages," TransLogos Translation Studies J., vol. 3, no. 1, pp. 98–121, 2020.

[9] I. T. Khemakhem, S. Jamoussi and A. B. Hamadou, "Improving English-Arabic Statistical Machine Translation with Morpho-syntactic and Semantic Word Class," Int'l J. of Intelligent Systems Technologies and Applications, vol. 19, no. 2, 172, 2020.

[10] C. España-Bonet and M. R. Costa-Jussà, "Hybrid Machine Translation Overview," Part of the Book Series: Theory and Applications of Natural Language Processing (NLP), pp. 1–24, Springer, 2016.

[11] J. Zakraoui, M. Saleh, S. Al-Maadeed and J. M. AlJa'am, "Evaluation of Arabic to English Machine Translation Systems," Proc. of the 2020 11th IEEE Int'l Conf. on Information and Communication Systems (ICICS), pp. 185–190, Irbid, Jordan, 2020.

[12] R. Dabre, C. Chu and A. Kunchukuttan, "A Survey of Multilingual Neural Machine Translation," arXiv, [Online], Available: https://arxiv.org/abs/1905.05395v1, 2019.

[13] S. C. Siu, "Revolutionizing Translation with AI: Unravelling Neural Machine Translation and Generative Pre-trained Large Language Models," SSRN Electr. J., DOI: 10.2139/ssrn.4499768, 2023.

[14] J. Guo, Z. Zhang, L. Xu, B. Chen and E. Chen, "Adaptive Adapters: An Efficient Way to Incorporate BERT into Neural Machine Translation," IEEE/ACM Trans. on Audio Speech and Language Processing, vol. 29, pp. 1740–1751, 2021.

[15] X. Wu, Y. Xia, J. Zhu, L. Wu, S. Xie and T. Qin, "A Study of BERT for Context-aware Neural Machine Translation," Machine Learning, vol. 111, no. 3, pp. 917–935, 2022.

[16] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," J. of Machine Learning Research, vol. 21, no. 140, pp. 1–67, 2020.

[17] M. R. Costa-Jussà et al., "Scaling Neural Machine Translation to 200 Languages," Nature, vol. 630, no. 8018, pp. 841–846, 2024.

[18] T. B. Brown et al., "Language Models are Few-shot Learners," Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901, 2020.

[19] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186, 2019.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention Is All You Need," Adv. Neural Inf. Process. Syst., vol. 30, pp. 5998–6008, 2017.

[21] L. Xue et al., "ByT5: Towards a Token-free Future with Pre-trained Byte-to-Byte Models," Trans. of the Association for Computational Linguistics, vol. 10, pp. 291–306, 2022.

[22] B. Al-Rfooh, G. Abandah and R. Al-Rfou, "Fine-Tashkeel: Fine-tuning Byte-level Models for Accurate Arabic Text Diacritization," Proc. of the 2023 IEEE Jordan Int'l Joint Conf. on Electrical Engineering and Information Technology (JEEIT), pp. 199–204, 2023.

[23] F. Alzamzami and A. E. Saddik, "OSN-MDAD: Machine Translation Dataset for Arabic Multi-dialectal Conversations on Online Social Media," arXiv: 2309.12137, 2023.

[24] E. Nagoudi, A. Elmadany and M. Abdul-Mageed, "AraT5: Text-to-Text Transformers for Arabic Language Generation," Proc. of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 628-647, 2022.

[25] A. Slim and A. Melouah, "Low Resource Arabic Dialects Transformer Neural Machine Translation Improvement through Incremental Transfer of Shared Linguistic Features," Arabian J. for Science and Engineering, vol. 49, no. 9, pp. 12393–12409, 2024.

[26] L. H. Baniata, S. Park and S. B. Park, "A Multitask-based Neural Machine Translation Model with Part-of-Speech Tags Integration for Arabic Dialects," Applied Sciences, vol. 8, no. 12, pp. 2502, 2018.

[27] T. Alimi, R. Boujebane, W. Derouich and L. Belguith, "Fine-Tuned Transformers for Translating Multi-dialect Texts to Modern Standard Arabic," Int'l J. of Cognitive and Language Sciences, vol. 18, no. 11, pp. 679-684, 2024.

[28] S. Kchaou, R. Boujelbane and L. Hadrich-Belguith, "Parallel Resources for Tunisian Arabic Dialect Translation," Proc. of the 5th Arabic Natural Language Processing Workshop, pp. 200–206, Barcelona, Spain, 2020.

[29] S. Kchaou, R. Boujelbane and L. Hadrich-Belguith, "Hybrid Pipeline for Building Arabic Tunisian Dialect-Standard Arabic Neural Machine Translation Model from Scratch," ACM Trans. on Asian and Low-Resource Language Information Processing, vol. 22, no. 3, pp. 1–21, 2022.

[30] M. A. Faheem, K. T. Wassif, H. Bayomi and S. M. Abdou, "Improving Neural Machine Translation for Low-resource Languages through Non-parallel Corpora: A Case Study of Egyptian Dialect to Modern Standard Arabic Translation," Scientific Reports, vol. 14, no. 1, 2024.

[31] T. Zerrouki and A. Balla, "Tashkeela: Novel Corpus of Arabic Vocalized Texts, Data for Auto Diacritization Systems," Data Brief, vol. 11, pp. 147–151, 2017.

[32] G. Abandah and A. Abdel-Karim, "Accurate and Fast Recurrent Neural Network Solution for the Automatic Diacritization of Arabic Text," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 2, pp. 103–121, 2020.

[33] R. Younis and G. Abandah, "Automatic Diacritization of Arabic Text and Poetry Using Pretrained Byte-to-Byte Language Models and Multiphase Training," Proc. of the 2025 1st Int'l Conf. on Computational Intelligence Approaches and Applications (ICCIAA), pp. 1-6, Amman, Jordan, 2025.

[34] G. Abandah, "Jordanian Dialect Dataset," GitHub, [online], Available: https://github.com/Gheith-Abandah/JODA, 2025.

[35] I. Alsarsou, E. Mohamed, R. Suwaileh and T. Elsayed, "DART: A Large Dataset of Dialectal Arabic Tweets," Proc. of the 11th Int'l Conf. on Language Resources and Evaluation (LREC-2018), Miyazaki, Japan, 2018.

[36] C. Qwaider, M. Saad, S. Chatzikyriakidis and S. Dobnik, "Shami: A Corpus of Levantine Arabic Dialects," Proc. of the 11th Int'l Conf. on Language Resources and Evaluation (LREC-2018), Miyazaki, Japan, 2018.

[37] D. Alzuheiri, H. Adwan, G. Abandah and Y. Hamdan, "Converting Colloquial to Classical Arabic as within the Project Developing Applications to Correct Jordanian Spoken Arabic to Proper Language Using Machine Learning Techniques," Int'l J. on Islamic Applications in Computer Science and Technology (IJASAT), vol. 12, no. 4, 2024.

[38] A. Fadel, O. Oueslati, H. Gahbiche and R. Shafik, "Neural Arabic Text Diacritization: State of the Art Results and a Novel Approach for Machine Translation," Proc. of the 6th Workshop on Asian Translation, pp. 215-225, Hong Kong, China, 2019.

[39] A. Abdel Karim and G. Abandah, "On the Training of Deep Neural Networks for Automatic Arabic-text Diacritization," Int'l J. of Advanced Computer Science and Applications, vol. 12, no. 8, 2021.

[40] M. Khaleel and G. Abandah, "Efficient Stochastic Error Injection for Optimizing Large Language Models in Arabic Spelling Correction," Proc. of the 2025 IEEE Int'l Conf. on New Trends in Computing Sciences (ICTCS), pp. 505-510, Amman, Jordan, 2025.

[41] G. Abandah, A. Suyyagh and M. Z. Khedher, "Correcting Arabic Soft Spelling Mistakes Using BiLSTM-based Machine Learning," Int'l J. of Advanced Computer Science and Applications, vol. 13, no. 5, 2022.

[42] P. Phakmongkol and P. Vateekul, "Enhance Text-to-Text Transfer Transformer with Generated Questions for Thai Question Answering," Applied Sciences, vol. 11, no. 21, 2021.

[43] OpenBoard Team, "OpenBoard," GitHub, [Online], Available: https://github.com/openboard-team/openboard, 2022.

[44] KeyboardKit, "KeyboardKit," [Online], Available: https://keyboardkit.com/, 2023.

[45] B. Sadder, R. Sadder, G. Abandah and I. Jafar, "Multi-domain Machine Learning Approach of Named Entity Recognition for Arabic Booking Chatbot Engines Using Pre-Trained Bidirectional Transformers," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 10, no. 1, pp. 1–16, 2024.

ملخص البحث:

تُعالج هذه الورقة التّحدّي المتمثل في إنتاج ترجمةٍ دقيقةٍ من العربية الأردنية إلى العربية الفصحى الحديثة (MSA) مع تصويب الأخطاء اللّغوية الشّائعة. وعلى الرّغم من أنّ العربية الفصحى الحديثة هي الصّيغة الرّسمية للتّواصل باللّغة العربية، فإنّ الانتشار الواسع للّهجات المحلّية في وسائل التّواصل الاجتماعي إلى جانب التّفاعلات اليومية نجم عنها انتشار نُصوص تُعاني من أخطاء في التّهجئة وعيوبٍ قواعدية.

وللتغلب على هذه التّحديات، نقدّم نظاماً مبتكراً مبنياً على مجموعة بياناتٍ بالعربية الأردنية (JODA) تتألف من (59135) جُملة، إلى جانب مجموعة البيانات (تشكيلة) المعدّلة من خلال حقن الأخطاء الصناعية. ونقوم بتوظيف نموذجً لغويٍ ضخم مُدرَّبٍ مُسبقاً يقبل نُصوص على مستوى البايْت تجعله قابلاً للتّكيُّف للتّبايُنات المتعلّقة بالتّهجئة في اللّهجات المحلّية المختلفة والتّعقيدات التّركيبية المختلفة في اللهجات العربية. ويظهر نظامنا من خلال التّجارب العملية تحسين معدّل الخطأ في النّصّ عبر تقليل ذلك المعدّل بفعل معايرة النّظام على مجموعات البيانات المذكورة، كما اتّضح أنّ النّظام المقترح يُنتج ترجماتٍ دقيقةً في معظم الحالات. وتبين أيضاً أنّ النّظام مَدار البحث يُشكّل حلّاً ناجعاً لملايين المتحدّثين بالعربية السّاعين للحصول على نصوصٍ دقيقة بالعربية الفصحى الحديثة.

والنّظام المقترح في وضعه الرّاهن مُقتصر على ترجمة النّصوص من العربية الأردنية، وتتّجه مساعي البحث المستقبلية إلى توسيع استخدامه ليغطّي التّرجمة من لهجاتٍ عربية أخرى.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 03, September 2025.

ADVANCED DEEP-LEARNING TECHNIQUES FOR IMPROVED CYBERBULLYING DETECTION IN ARABIC TWEETS

Marah Hawa, Thani Kmail and Ahmad Hasasneh

(Received: 1-Mar.-2025, Revised: 11-May-2025 and 15-Jun.-2025, Accepted: 19-Jun.-2025)

ABSTRACT

Cyberbullying has emerged as a pressing issue in the digital era, particularly within Arabic-speaking communities, where research remains limited. This study investigates the detection of Arabic cyberbullying on social media using both traditional machine learning (ML) and deep learning (DL) techniques. A publicly available dataset of Arabic tweets was used to train and evaluate several ML models (SVM, NB, LR and XGBoost), alongside a recurrent neural network (RNN). The results demonstrate that the RNN significantly outperforms classical ML models, highlighting the efficacy of DL in accurately identifying abusive content in Arabic text. These results emphasize the necessity of incorporating linguistically rich data and advanced neural architectures to improve cyberbullying-detection systems in low-resource languages such as Arabic.

KEYWORDS

Machine learning algorithms, Arabic tweets, Deep-learning techniques, Recurrent neural network, Cyberbullying.

1. INTRODUCTION

Cyberbullying involves the use of digital platforms—such as smartphones and social media—to inflict harm through behaviors like verbal abuse, offensive language and harassment. Its psychological impact can be profound, especially among teenagers, leading to issues, such as low self-esteem, anxiety and identity-related concerns. The problem has intensified globally with the growing popularity of platforms, like Twitter (now X), where anonymity enables harmful behavior without accountability [1].

Recent reports highlighted the scale of the issue: in 2024, 28% of adolescents experienced cyberbullying and over 42% of youth aged 13–24 years in the MENA region reported exposure to online abuse *via* popular apps like Instagram, TikTok and Twitter [2]–[5]. The ITU and Arab Social Media Observatory have similarly flagged cyberbullying as a major digital threat to the mental health of children and adolescents [6]-[7]. These findings point to an urgent need for scalable, data-driven solutions that go beyond manual moderation.

Despite growing efforts in English-language research, Arabic cyberbullying detection remains underexplored. The increasing use of Arabic on social media—especially Twitter—demands more targeted approaches, but the language's rich morphology, diverse dialects and limited annotated resources present ongoing challenges [8]. The situation was further exacerbated by the COVID-19 pandemic, which saw young users spending more time online and becoming more vulnerable to digital abuse [9].

To address these gaps, this study proposes a deep learning-based model for detecting cyberbullying in Arabic text. By combining three datasets representing different Arabic dialects into a single corpus and applying a Recurrent Neural Network (RNN)—a relatively underutilized method in this context—we achieve significant improvements in detection performance. Our work contributes to the development of more robust and linguistically aware systems for identifying abusive content in Arabic-language social media.

The rest of this paper is organized as follows: Section 2 reviews related work and the datasets used; Section 3 outlines the proposed methodology; Section 4 presents and analyzes the results; Section 5 offers a comprehensive discussion; and Section 6 concludes the paper.

M. Hawa, T. Kmail and A. Hasasneh (Corresponding Author) are with Department of Natural, Engineering and Technology Sciences, Faculty of Graduate Studies, Arab American University (AAUP), Ramallah, Palestine. Emails: m.hawa1@student.aaup.edu, t.kmial@student.aaup.edu and Ahmad.Hasasneh@aaup.edu

2. LITERATURE REVIEW

Natural-language processing (NLP) technologies have evolved substantially over the decades, becoming vital for enabling effective human-computer interaction [11]. Fundamentally, NLP transforms natural-language texts into machine-processable digital formats, enabling sophisticated tasks, such as machine translation and sentiment analysis [12]-[13]. The roots of NLP trace back to the 1950s with early systems, like the Georgetown-IBM translation experiment, which laid the groundwork for subsequent advances in AI-driven text understanding.

A critical initial step in NLP pipelines is text pre-processing, which ensures high-quality input data for improved model performance. Tokenization breaks text into meaningful units, such as sentences or words, facilitating downstream analysis. Techniques, like stemming, which reduce words to their root forms and stop-word removal, which excludes frequent, but semantically light words, are essential for reducing noise and dimensionality [14], [16]. Kanaan et al. [15] demonstrated that combining stemming with truncation, normalization and stop-word removal significantly boosts classification accuracy and F1-scores in document-classification tasks.

In the realm of machine learning (ML), classical algorithms such as Support Vector Machines (SVMs), Naive Bayes (NB), Logistic Regression (LR) and Extreme Gradient Boosting (XGBoost), have remained popular due to their efficiency and interpretability. These models have been applied extensively for Arabic-cyberbullying detection, yielding solid baseline results. For example, Hani et al. [23] reported over 89% accuracy using linear SVM with TF-IDF features on a small Arabic dataset, while Rashid et al. [24] and Moheb et al. [21] achieved accuracies up to 95% with NB classifiers. Logistic regression also performs competitively, with Rashid et al. [24] improving F1-scores through dataset balancing and feature engineering. XGBoost, a powerful ensemble method, showed promising results with 85% accuracy [24].

2.1 Classification Methods

Many researchers have collected data from popular social-media platforms, such as Twitter and Facebook, to study cyberbullying. For instance, Aladdin et al. [17] utilized the Twitter API to gather their dataset. Similarly, Haidar et al. [18]-[19] developed dedicated tools in Python and PHP to collect data from both Facebook and Twitter, storing it in a MongoDB database. Al-Harbi and colleagues [20] compiled a large dataset comprising 100,327 tweets and comments collected from Twitter, YouTube and Microsoft platforms. Meanwhile, Mohib et al. [21] gathered 25,000 tweets and comments from Twitter and YouTube using their respective APIs. Other studies employed tools, such as NLTK, for text analysis or platforms, like RStudio, for extracting tweets [22]-[23]. Although most of these datasets were primarily in English, some research focused on Arabic data collected from sources, including Twitter, Facebook and YouTube [22]. Most datasets were processed and manually annotated, while Arabic-cyberbullying datasets remain comparatively limited.

The literature highlights the significant role of machine-learning algorithms in addressing cyberbullying challenges by detecting harmful patterns and behaviors through classification and text analysis. Support Vector Machines (SVMs) have been widely used for text classification in Arabic-cyberbullying research. For example, Hani et al. [23] achieved over 89% accuracy using a linear SVM on a small dataset of 1.6K publications after extracting features with the term frequency-inverse document frequency (TF-IDF) method. The Naive Bayes (NB) classifier has also been extensively applied in Arabic-text analysis [12], [24]-[25]. Rashid et al. [24] employed NB with the bag of words model, reaching 87% accuracy and 35% recall, while Moheb et al. [21] reported up to 95% accuracy. Kanaan et al. [20] further demonstrated that NB attained 91% accuracy following demodulation and stop-word removal. Logistic regression (LR) is another common classification algorithm used in both binary and multi-class problems. Rashid et al. [24] used LR as a baseline model and, after balancing the dataset, improved the F1-score to 84% using TF-IDF features. Alfageh et al. [25] applied LR with TF-IDF, reporting results slightly lower by 1.8% compared to count vectorization. Lastly, the Extreme Gradient Boosting (XGBoost) algorithm has shown effectiveness in handling text data for cyberbullying detection, with Rashid et al. [24] reporting 85% accuracy using this approach.

2.2 Deep-learning Techniques

These methods have demonstrated impressive effectiveness in addressing the challenge of identifying cyberbullying in the Arabic language. For example, the researchers in [21], [25] developed a CNN-based model specifically tailored for this task. Their methodology involved four key steps: converting textual data into numerical representations, applying convolutional operations to extract significant features, reducing the convolution output to preserve only the most relevant information and finally feeding the processed data into a dense layer fully connected to all neurons in the network. This approach was tested on a dataset of 39,000 Arabic tweets collected *via* the Twitter API, achieving an impressive accuracy exceeding 95% without requiring manual intervention. Similarly, Banerjee et al. [26] extended the use of CNN to a larger dataset of 69,000 Arabic tweets, reporting an accuracy rate above 93%. In another study, Benaissa et al. [24] compared CNN with other deep-learning architectures, including Gated Recurrent Units (GRUs), Long Short-Term Memory (LSTM) and Bidirectional LSTM (BLSTM). Their analysis, conducted on a dataset of 32,000 Arabic comments from Aljazeera.net, showed that CNN outperformed the other models by a margin of one percent in the F1-score. Across the balanced dataset, these models collectively achieved an average F1-score of 84%. Further insights were provided by Srivastava et al. [27], who explored GRU, LSTM and BLSTM models for detecting objectionable content in online conversations. Their methodology incorporated rigorous data pre-processing steps, such as text cleaning, tokenization, stemming, lemmatization and stop-word removal prior to training the deep-learning algorithms. Among the models tested, BLSTM achieved the highest accuracy of 82.18%, followed closely by GRU (81.46%) and LSTM (80.86%). These results highlight the transformative potential of deep-learning techniques, particularly CNN, in enhancing the detection of cyberbullying within Arabic social media posts. Although these findings are promising, they also emphasize the need for continued research to further refine these models and effectively manage the growing volume and complexity of Arabic content on social-media platforms.

Building on the promising results of deep-learning techniques, such as CNN and RNN, in Arabic-cyberbullying detection, recent studies have explored hybrid and transformer-based approaches to further enhance performance. The study in [35] proposed a hybrid deep-learning model that combines LSTM networks with CNNs to detect cyberbullying in Arabic tweets. Their study focused on applying deep learning techniques to social-media data, specifically targeting the challenges of NLP. They demonstrated that their hybrid model outperformed several traditional ML algorithms, including SVM and NB, in terms of classification accuracy. While their contribution is significant, the study did not explicitly address dialectal variations within Arabic, nor did it elaborate on the size and linguistic diversity of the dataset used, which are important considerations in the context of Arabic social-media text. Abu Kwaik et al. [36] introduced an advanced methodology for identifying hate speech in Arabic tweets by integrating Recurrent Neural Network architectures—namely GRU and BiLSTM—with contextual word embeddings derived from AraBERT. Their experiments on dialectal Arabic-tweet datasets demonstrated strong discriminatory power, achieving an AUROC of approximately 0.84 in binary classification, 87.05% accuracy for the 2-class task, 78.99% for the 3-class task and 75.51% for the 6-class task. This study highlights the effectiveness of combining transformer-based embeddings with recurrent neural models when handling Arabic social-media content.

Building on these advances, a very recent study in [39] proposed state-of-the-art deep-learning techniques and provided comparative benchmarks closely aligned with the methodology of this research. The study applied a combination of CNN, RNN and transformer-based models to large-scale datasets of Arabic social-media content, emphasizing the importance of handling dialectal diversity and semantic nuances. Their results surpassed previous benchmarks, achieving improvements in both accuracy and F1 score metrics, demonstrating significant progress in the field between 2022 and 2024. Including such up-to-date research enhances the understanding of current capabilities and helps guide future work toward more robust cyberbullying-detection systems. Based on the previous studies referenced [13], [26], [23], [27], it has been observed that detecting bullying in the Arabic language remains a critical topic that requires significant attention in research. There is an urgent need for more studies on this topic. The existence of new technologies can help reduce the harmful impact of social media to prevent unwanted occurrences. Obeidat et al. [37] conducted a comparative study evaluating deep-learning models, such as RNN and CNN, against traditional machine-learning classifiers, like SVM and Random Forest for Arabic sentiment analysis on Twitter datasets. Their findings demonstrated that deep-learning approaches outperform traditional machine-learning methods in effectively handling the complexity and dialectal variations of Arabic social-media text. This is highly relevant to cyberbullying detection, which shares similar linguistic challenges. Our work extends these findings by applying RNN architectures on a larger, multi-dialectal dataset specifically focused on cyberbullying detection, further confirming the superior performance of deep-learning techniques over traditional models in Arabic NLP tasks. Earlier work by Al-Hassan and Al-Dossari [38] proposed one of the earliest benchmark datasets for Arabic-cyberbullying detection, compiling approximately 10,000 tweets labelled for offensive content. They evaluated both ML (Random Forest) and DL models (CNN, RNN), highlighting the promising performance of RNNs. Their dataset, however, is limited in scale and dialectal coverage. In contrast, our study utilizes a larger and more dialectally diverse dataset and focuses on standard RNN architectures, allowing for a more detailed exploration of their effectiveness in cyberbullying detection. Furthermore, other deep-learning models have also demonstrated promising results in Arabic-text classification tasks outside the cyberbullying domain. For instance, Jamaluddin et al. [43] proposed a multi-channel deep-learning model for Arabic news classification, emphasizing the importance of capturing semantic features through parallel architectures. Similarly, Al Qadi et al. [44] introduced a scalable shallow learning approach for tagging Arabic news articles, highlighting the benefits of lightweight models for Arabic NLP. These contributions further underline the growing applicability of both deep and shallow models across diverse Arabic-language NLP tasks.

While previous research demonstrates considerable progress using classical ML and deep learning for Arabic-cyberbullying detection, several gaps remain. Most studies rely on limited datasets with narrow dialectal coverage and modest sample sizes. The increasing linguistic complexity of Arabic social-media content necessitates larger, more diverse datasets and efficient deep-learning models. Our study addresses these gaps by utilizing extensive, dialect-rich datasets and focusing on RNN architectures that balance performance and complexity. This approach contributes to advancing robust cyberbullying detection in Arabic, complementing recent transformer-based innovations.

Therefore, a group of ML and deep-learning algorithms that were observed in the literature was chosen. Table 1 provides a summary of some of the literature on Arabic cyberbullying.

3. Materials and Methods

It is well known that the detection of a cyberbullying attack involves several steps, including data collection, visualization, pre-processing, feature extraction, model training and then model evaluation, as illustrated in Figure 1.

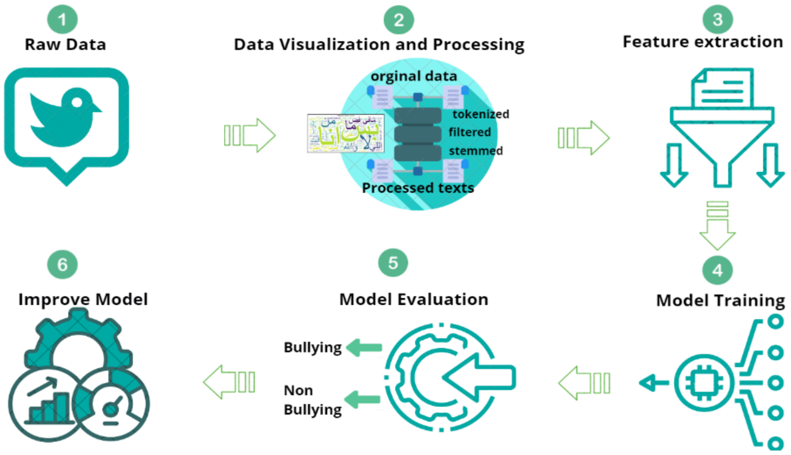


Figure 1. A general workflow of the proposed methodology.

3.1 Data Description

The data used in this research consists of public datasets published on Kaggle and divided into three separate and linguistically varied datasets, as shown in Table 2. The initial dataset, consisting of 5,846 Syrian/Lebanese political tweets, is included in the "Levantine Arab Hate Speech" dataset [42], which is divided into three groups: abusive tweets, hate-speech tweets and normal tweets. The second set, known as the "Arabic Sentiment Twitter Dataset Corpus" [43], consists of 56,795 Arabic tweets divided into two categories: positive and negative. The third group, "Arabic Dataset 1" [44], consists of a relatively small dataset of 1,100 tweets, divided into two categories using binary classification:

Table 1. Brief summary of the literature on Arabic cyberbullying.

Ref.	Classifier	Year	Dataset (Size)	Evaluation matric
[9]	XGBoost, NB.SVM, LR	2024	Twitter 9000 Tweets	Accuracy: 88%,78%, 84.4%, 83.95%
[14]	SVM	2021	Twitter API, (17.748 Tweets)	Accuracy: 85.49%
[15]	SVM, KNN, NB, RF	2020	X API, (4000 Tweets, Facebook2138Posts)	N/A
[24]	Deep Learning	2020	Aljazeera.net (32000 Comments)	Accuracy: 84%
[28]	NB	2023	YouTube Platform (4760 Comments)	Accuracy: 94%
[29]	SVM, NB	2024	Twitter and YouTube (30000 Tweets)	Accuracy: 95%, 70%
[30]	LSTM	2023	Twitter 10000 Tweets	Accuracy: 88%
[31]	MLP	2023	Twitter API 4140	Accuracy: 89%
[32]	LR, voting classifier, SVM	2024	Twitter 12000 Tweets	Accuracy: 65%, 71%, 98%
[33]	Codellama, DeepSeekCoder, Llama2	2025	10000Comments	Accuracy: 35%, 26%, 16.4%
[34]	AraBERT	2025	4240 Comments	N/A
[35]	Hybrid (CNN, LSTM)	2022	N/A	Accuracy: 97%
[36]	GRU and BiLSTM combined with contextual embeddings (AraBERT)	2023	N/A	Accuracy: 87.05% (2-class), 78.99% (3-class), 75.51% (6-class)
[39]	CNN, LSTM and BiLSTM	2025	50000 comments	Accuracy: 91%

negative speech (1) and positive or neutral speech (0). This data is characterized by the diversity of dialects used, including local dialects and classical Arabic, making it comprehensive and covering different linguistic styles in the Arab world. Tweets are categorized into two main categories: bullying, which contains offensive words or phrases and non-bullying, which does not. The final dataset comprises labels of 0 or 1 depending on whether the comment is bullying or not. Additionally, the data used is balanced, as shown in Figure 2.

Table 2. Data description.

Ref.	Group Name	No. of Tweets	Categories	Size - Notes
[42]	Arabic -Levantine Hate Speech	5846	Normal, Abusive, Hate	Syrian–Lebanese Politics
[43]	Arabic Twitter Sentiment Dataset	56795	Positive, Negative	Training 45275, Testing 11920
[44]	Arabic Dataset	1100	Negative, Positive	Relatively Small Data Size
Total		63741		

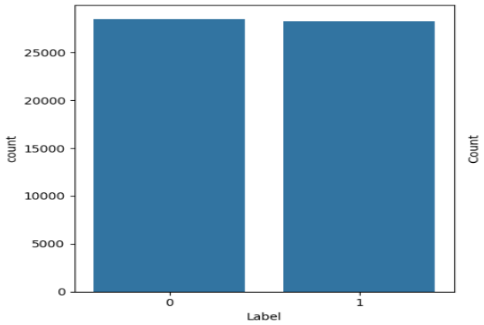


Figure 2. Distribution of positive and negative text, where (0 = Non-bullying, 1 = Bullying).

3.2 Data Visualization and Pre-processing

To know the most frequent words for bullying and non-bullying comments, this is expressed by displaying the word sizes, where large words are frequently repeated, as shown in Figure 3.

(a) (b)

Figure 3. Word cloud negative (a), Word cloud positive (b).

Figure 4 shows the representation of the most frequent words using the Count Vectorizer technique, where the frequency of words within texts is counted and converted into a numerical representation. The graph displays the twenty most frequently used words, ranked by frequency.

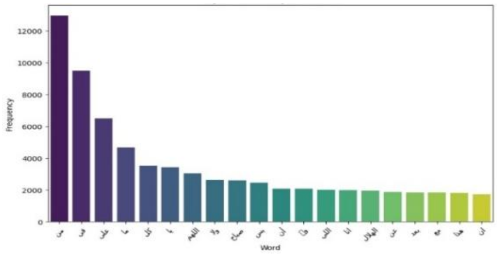


Figure 4. Count-vectorizer technique (top-20 most frequent words).

The first word is shown to have the highest frequency, occurring more than 250 times, followed by other words with decreasing frequencies. This representation is useful for understanding the distribution of words within the text data and discovering words that may be of high analytical interest in the context under study. The pre-processing stage is an important step in an ML technique, because it cleans and prepares the dataset, so that it can be used to train the model. In this study, the tweets are written in various dialects that differ from traditional Arabic. Therefore, we have used the NLP technique to address issues presented by comments on Twitter written in Arabic. This was applied in Figure 5.

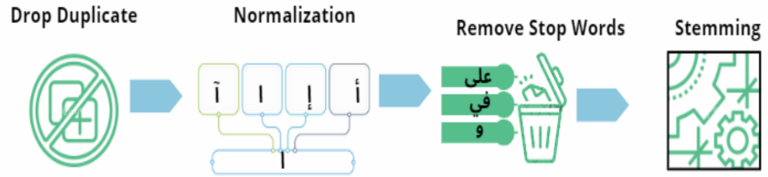
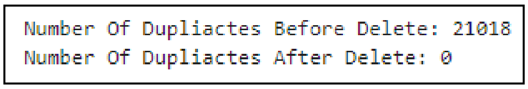


Figure 5. Data pre-processing main steps.

3.2.1 Removing Duplicates

There is a duplicate tweet; with bullying the duplicate count is 9896 and without bullying the duplicate count is 11122. So, by using the Python code, we remove these duplicates and they become zero duplicates, as shown in Figure 6.

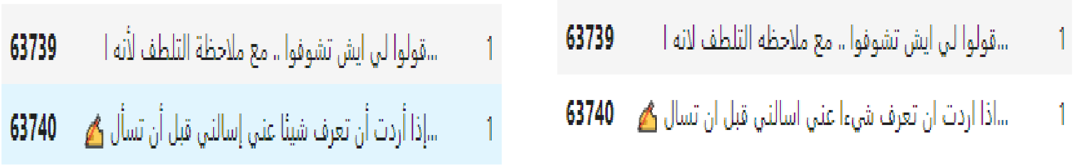


```
Number Of Dupliactes Before Delete: 21018
Number Of Dupliactes After Delete: 0
```

Figure 6. Remove duplicate.

3.2.2 Normalization

We applied the normalization to the dataset and converted it into a uniform text. The Python programming language implemented this process. It significantly contributes to improving the performance of models in ML tasks by reducing unnecessary linguistic variations. By converting texts into a standardized format, such as removing diacritics or similar characters, the model becomes better at understanding underlying patterns, which reduces noise in the data. This step leads to improved model accuracy and increased efficiency in handling unstructured and diverse texts, such as those found in cyberbullying tasks. In our study, we remove the English Letters, URLs, Hashtags, Special Characters and emojis. After applying the normalization process this led to the text being normalized and the result is shown here in Figure 7.



63739	...قولوا لي ايش تشوفوا .. مع ملاحظة التلطف لأنه ا	1
63740	...إذا أردت أن تعرف شيئا عني إسالني قبل أن تسأل ✍	1

(a)

63739	...قولوا لي ايش تشوفوا .. مع ملاحظه التلطف لانه ا	1
63740	...اذا اردت ان تعرف شيءا عني اسالني قبل ان تسال ✍	1

(b)

Figure 7. Text; before normalization (a); after normalization (b).

In Figure 7. (a), we see the tweets (“ إذا أردت أن تعرف شيئا همي إسألني قبل أن تسأل“), we see (“ي,ئ,أ”) converted into uniform text, as shown in Figure 7. (b). For example, (“أ,إ “) converted into (ا).

3.2.3 Removing Stop Words

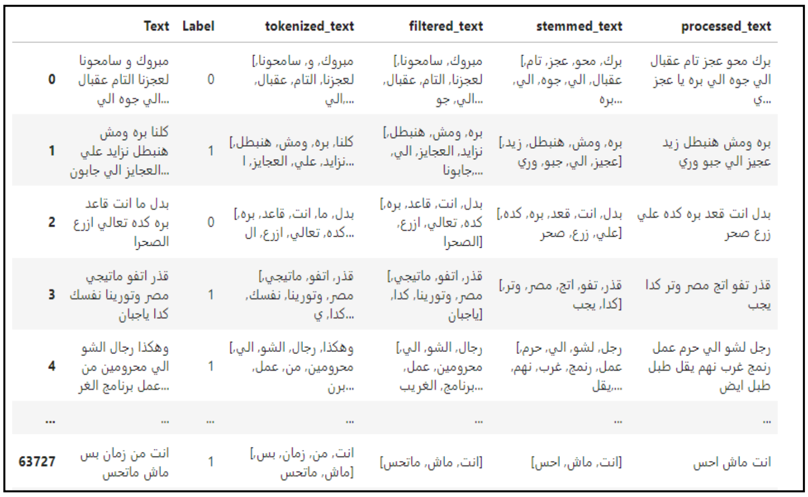
Stop words are meaningless in our study; we normalize text by removing the stop words. In applications, omitting standard words is a good way to implement and emphasize the most important words.

3.2.4 Tokenization

The texts were converted into words using natural-language units based on language rules defined by word boundaries. This step enabled the RNN model to treat each word as an independent unit within a sentence string, creating innovations in learning and processing context. When tokenization is carried out accurately, it makes it easier for the model to handle a wide variety of texts, such as those found in cyberbullying, which can include offensive words and complex phrases. Through good segmentation, the model can "understand" these offensive elements separately from other words, improving the accuracy of its predictions.

3.2.5 Stemming

In this step, words are reduced to their original roots by removing good suffixes or additions such as" ه" أو "ين" أو "ات". The goal of stemming is to reduce word diversity, helping the model understand that words derived from the same trigger have the same underlying meaning, such as "كتب" ,"كتابة" ,"كتاب" being reduced to "كتب". If stemming is applied effectively, it can improve the model's accuracy by reducing the variety of words associated with the same root. However, sometimes it can have a negative impact if it excessively reduces words, leading to weakened differentiation between important words. After we applied the pre-processing steps shown in Figure 5, Figure 8 shows a sample of the pre-processing phase.



	Text	Label	tokenized_text	filtered_text	stemmed_text	processed_text
0	مبروك و سامحونا لعجزنا التام عقبال الي جوه الي...	0	[مبروك, و, سامحونا, لعجزنا, التام, عقبال, الي,...	[مبروك, سامحونا, لعجزنا, التام, عقبال, الي, جو...	[برك, محو, عجز, تام, عقبال, الي, جوه, الي, بره...	برك محو عجز تام عقبال الي جوه الي بره يا عجز ي...
1	كلنا بره ومش هنبطل نزايد علي العجايز الي جابون...	1	[كلنا, بره, ومش, هنبطل, نزايد, علي, العجايز, ا...	[بره, ومش, هنبطل, نزايد, العجايز, الي, جابونا,...	[بره, ومش, هنبطل, زيد, عجيز, الي, جبو, وري]	بره ومش هنبطل زيد عجيز الي جبو وري
2	بدل ما انت قاعد بره كده تعالي ازرع الصحرا	0	[بدل, ما, انت, قاعد, بره, كده, تعالي, ازرع, ال...	[بدل, انت, قاعد, بره, كده, تعالي, ازرع, الصحرا]	[بدل, انت, قعد, بره, كده, علي, زرع, صحر]	بدل انت قعد بره كده علي زرع صحر
3	قذر اتفو ماتيجي مصر وتورينا نفسك كدا ياجبان	1	[قذر, اتفو, ماتيجي, مصر, وتورينا, نفسك, كدا, ي...	[قذر, اتفو, ماتيجي, مصر, وتورينا, كدا, ياجبان]	[قذر, تفو, اتج, مصر, وتر, كدا, يجب]	قذر تفو اتج مصر وتر كدا يجب
4	وهكذا رجال الشو الي محرومين من عمل برنامج الغر...	1	[وهكذا, رجال, الشو, الي, محرومين, من, عمل, برن...	[رجال, الشو, الي, محرومين, عمل, برنامج, الغريب...	[رجل, لشو, الي, حرم, عمل, رنمج, غرب, نهم, يقل,...	رجل لشو الي حرم عمل رنمج غرب نهم يقل طبل طبل ايض
...	...	...	...	...	...	...
63727	انت من زمان بس ماش ماتحس	1	[انت, من, زمان, بس, ماش, ماتحس]	[انت, ماش, ماتحس]	[انت, ماش, احس]	انت ماش احس

Figure 8. A sample of the pre-processed data.

As Figure 8 illustrates, there are six columns. The first and second columns represent the dataset before processing. Column 3 (“Tokenize Text”) shows how the text is tokenized into small words. Column 4 (“Filter Text”) displays the result after removing meaningless data using stop words. Column 5 (“Stem Text”) shows the text converted into its original form in Arabic and the final column presents the uniform data after pre-processing. The study then investigates the best model features that yielded the highest accuracy to identify the most effective ML algorithms for detecting cyberbullying in Arabic tweets using TF-IDF techniques. In this study, Twitter tweets are categorized into two groups: bullying and non-bullying. The TF-IDF feature-extraction method was employed to enhance the textual data representation by measuring the importance of terms within individual tweets relative to the entire dataset. Additionally, the study utilized n-grams to analyze the sequences of words rather than isolated terms, allowing the capture of contextual information. This significantly improved the model's understanding of language patterns associated with bullying behavior. This approach was essential for classifying tweets in a relevant and accurate manner.

3.3 Machine-learning Classification and Tuning

3.3.1 Support Vector Machine

In this study, the SVM algorithm was used as one of the basic ML techniques for tweet classification and cyberbullying analysis. SVM is an effective tool for handling high-dimensional text data and finding the best hyperplane between different categories, such as bullying-free tweets and bullying tweets. SVM has been applied to text features extracted using NLP techniques, such as converting text into numerical representation *via* TF-IDF vectorization. The algorithm has improved classification accuracy thanks to its ability to handle multi-dimensional text, especially in light of the diversity of dialects and linguistic patterns within the dataset [9].

3.3.2 Naïve Bayes

In this study, Arabic tweets were categorized into cyberbullying-related groups using the NB algorithm. As a result of its effectiveness and simplicity, NB was an appropriate option when handling huge and high-dimensional data. The algorithm's output also showed strong performance in rapidly and precisely gathering data, which aided in the efficient identification of cyberbullying in tweets [9].

3.3.3 Logistic Regression

In this study, Arabic tweets were analyzed using LR as a classification method and they were divided into two groups: cyberbullying and non-bullying. LR is a good choice for this kind of data, because it can handle binary problems well and has shown promise in identifying the correlation between textual characteristics and the degree of bullying in tweets. Obtaining precise and comprehensible classification models was also beneficial [31].

3.3.4 Extreme Gradient Boosting

XGBoost was employed in this study as a technique to classify Arabic tweets into two groups: those that involved cyberbullying and those that did not. XGBoost was selected because of its top-ranking performance and good accuracy in handling data with many different dimensions. Furthermore, the XGBoost method enhances performance by employing strategies, like regularization to lessen overfitting and enhance generalization [31].

3.3.5 Deep-learning Approach

In this research, RNNs were used to analyze Arabic tweets related to cyberbullying and categorize them into two classes: "bullying" and "non-bullying." This technique was selected due to its ability to recognize sequential patterns in text data, such as understanding context within a series of words. RNNs are particularly well-suited for tweet analysis, as they account for the chronological order of words and expressions, helping identify offensive messages influenced by contextual nuances.

The model was trained using Keras’s sequential interface, incorporating an embedding layer, followed

by a simple RNN layer and ending with a dense layer. The embedding layer transformed words into numerical representations, with an input dimension of 5,000 and an output dimension of 64. The sequence length was determined based on the maximum length of tweets in the dataset.

The training process was conducted using a set of pre-defined hyperparameters, with multiple tests performed to determine the optimal configuration. The model was initialized with random weights and the number of units in the output layer was tailored for binary classification, as the task requires categorizing tweets into two classes: "bullying" and "non-bullying."

The learning rate was optimized using the Adam optimizer, chosen for its efficiency in training deep-learning models. The batch size was set to 64, enabling the model to process a sufficient number of samples per iteration. The training spanned 27 epochs. The text data was also classified using a combination of ML and DL algorithms to enhance performance and identify the optimal model. The ML algorithms included SVC, LR and NB, while RNNs represented the deep-learning component. Texts were transformed into numerical representations using the TF-IDF technique and hyperparameters were fine-tuned using GridSearch to achieve the best possible performance. Below is a summary of the parameter settings used for each algorithm [24].

3.3.6 Hyper-parameter Fine-tuning and Evaluation Measures

Several algorithms were used with parameter adjustments to enhance performance. In SVC, the parameter *C* was set to control regularization, *max_iter* for the number of iterations, *length* for defining stopping criteria and *TF-IDF max_features* to specify the number of words considered in the TF-IDF representation. In LR, *C and solver* were configured to select the solution method, along with adjustments to *TF-IDF max_features* and *TF-IDF ngram_range* to define the range of words considered. In NB, the *alpha* parameter was used to regulate the influence of rare words and *TF-IDF ngram_range* was used to define the word range considered in the model. In the RNN model, the learning rate was determined using the Adam optimizer. The parameters *input_dim*, *output_dim* and *input_length* were set to properly format the text input, while *epochs* and *batch size* were selected for the training process. All models used TF-IDF to convert textual data into a numerical format and *GridSearch* was employed to determine the optimal values for each model’s parameters.

3.3.7 Model Generation and Evaluation

In this study, the Python, Sklearn and XGBoost libraries were used to develop four supervised ML models to classify the data. The results were evaluated using several performance metrics, including accuracy as given in Equation (1), precision as given in Equation (2), recall as given in Equation (3) and F1-score as given in Equation (4). These measures were calculated using the following equations [31], where TP is the true positives, TN is the true negatives, FP is the false positives and FN is the false negatives.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$F1 - Score = \frac{2*(precision*recall)}{(precision+recall)} \tag{4}$$

We also calculated the F1-score computed at the class level (Macro-F1) and at the sample level (Micro-F1). The Macro-F1 was calculated as the simple average of the F1 scores for each class and the Micro-F1 was calculated based on the confusion matrix, which takes into account all true positives, false positives, false negatives and true negatives across all classes, as follows:

$$Macro - F1 = \frac{F1\ class_0 + F1\ class_1}{2} \tag{5}$$

$$Micro - F1 = \frac{TP+TN+FP+FN}{TP} \tag{6}$$

4. RESULTS

Experiments were conducted to analyze the performance of models used in text classification, using ML and deep-learning algorithms with parameter adjustment to improve accuracy. The aim was to compare the effectiveness of the models and choose the most appropriate for the available data. The results are presented below.

4.1 Experiment Results Using Different Machine-learning Algorithms

The models were built and tested using a dataset collected and processed for this study. The collection contains 42,723 tweets after initial processing, obtained from Kaggle. This study used four ML algorithms: SVM, NB, LR and XGBoost. The performance of these models was evaluated using measures of accuracy, precision, recall and F1-score, as shown in Table 3.

Table 3. Experimental results of various machine learning methods.

ML	Feature Extraction	Accuracy	Precision		Recall	
SVM	TF - IDF	75%	0	76%	0	76%
			1	75%	1	75%
NB	TF - IDF	72%	0	73%	0	73%
			1	72%	1	72%
LR	TF - IDF	74%	0	76%	0	74%
			1	73%	1	75%
XGBoost	TF - IDF	74%	0	77%	0	70%
			1	71%	1	78%

Table 3 compares the performance of four models (SVM, NB, LR and XGBoost) in the cyberbullying-detection task using the TF-IDF feature-extraction method. Each model's performance was evaluated based on the mentioned metrics. The SVM model performed best, recording 75% accuracy, 76% precision, 76% recall and 76% F1-score. This makes it the most effective of all models, showing a good balance across all metrics. The NB model recorded 72% accuracy, 73% precision, 73% recall and 73% F1-score. Despite its weaker performance compared to SVM, it still offers acceptable results, particularly in recall. The LR model achieved 74% accuracy, 76% precision, 72% recall and 75% F1-score. LR performed close to SVM, but was lower in terms of recall and F1 score. The XGBoost model showed balanced performance, achieving 74% accuracy, 77% precision, 70% recall and 74% F1-score. XGBoost outperformed other models in terms of precision, with the highest score (77%), demonstrating its ability to make more accurate positive predictions. Although the initial results obtained using traditional ML algorithms were acceptable, they were not sufficient to meet the required objectives. Therefore, the accuracy and overall performance of the model were enhanced by applying deep-learning techniques using RNN.

4.2 Experiment Results Using RNN

To improve model performance and achieve better outcomes, we transitioned to using deep learning, with a focus on RNNs, to process the same large and complex dataset. During our experiments, neural networks demonstrated their ability to outperform traditional algorithms. In the first experiment, the model was trained for 20 epochs, resulting in an excellent accuracy of 96%. In the second experiment, we used 27 epochs and achieved an accuracy of 97%. These findings highlight the high proficiency of deep-learning techniques in extracting complex patterns from large datasets and underscore their significance as an effective approach to enhancing performance in this context. Table 4 presents the results of the experiment using RNNs.

Table 4. Experimental results of the deep-learning approach.

Experiments	Classifier	Accuracy	Precision		Recall		F1- Score
Experiment 1	RNN	96%	0	97%	0	94%	96%
			1	94%	1	97%	
Experiment 2	RNN	97%	0	97%	0	96%	97%
			1	96%	1	97%	

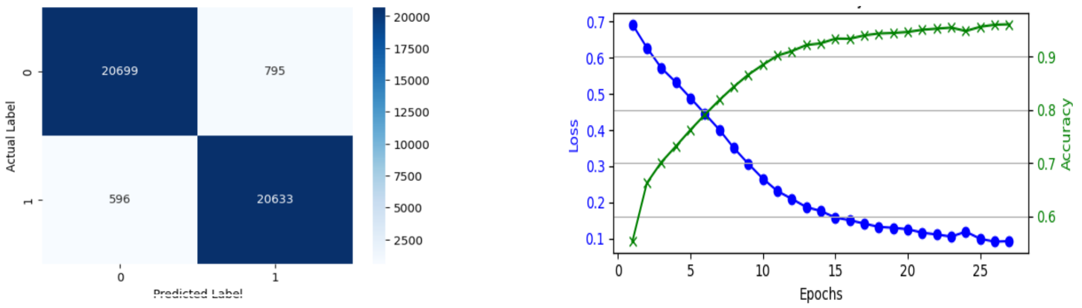
To ensure fair evaluation, we report precision, recall and F1-score separately for each class (0: non-bullying, 1: bullying). As shown in Experiment 2, Table 5, the model achieved high precision and recall for both classes (class 0: 97% recall, 96% F1-score; class 1: 96% precision, 97% F1-score), indicating balanced performance and minimal bias. In addition to per-class metrics, we computed the macro-averaged F1-score (97%) and micro-averaged F1-score (97%), confirming consistent performance across classes. We also include the confusion matrix to visualize the distribution of true positives, false positives, true negatives and false negatives, further supporting the reliability of our results.

Furthermore, the training process was stable, as evidenced by accuracy and loss curves, which show no signs of overfitting. These results highlight the model's robustness and its ability to distinguish between bullying and non-bullying instances effectively.

The attached diagrams illustrate the performance of the RNN model used in the experiment. Figure 9(a) shows the confusion matrix, which reflects the model's prediction accuracy, where the values in the cells indicate the number of correctly and incorrectly classified cases. For example, the model correctly classified 20,699 instances of the negative category (0) and 20,633 instances of the positive category (1), while misclassifications were limited to 795 and 596, respectively. These results indicate strong performance in data classification.

Figure 9(b) presents the loss and accuracy curve. This curve illustrates the relationship between the number of epochs and the corresponding values of loss and accuracy. The loss is shown to continuously decrease as the number of epochs increases, indicating the model's learning progress and improvement. Conversely, accuracy steadily increases to high levels, reflecting model stability and the ability to achieve accurate results over time.

Figure 10 displays the ROC Curve, which is used to evaluate model performance by comparing the True Positive Rate (TPR) with the False Positive Rate (FPR). The curve indicates that the model achieved an Area Under the Curve (AUC) of 97%, reflecting high effectiveness in distinguishing between categories. These results demonstrate the model's efficiency and its ability to process and classify data with high accuracy.



(a) (b)

Figure 9. (a) Confusion matrix and (b) Loss and accuracy curve for RNN.

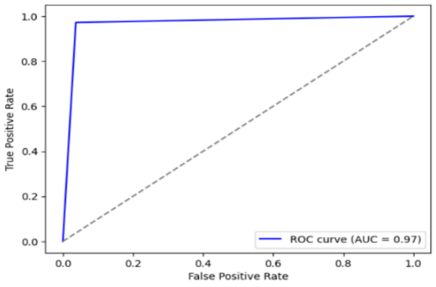


Figure 10. ROC curve for RNN results.

An important aspect of evaluating model performance involves analyzing false positives (FP) and false negatives (FN), as they directly impact the precision and recall scores, especially in sensitive tasks, like cyberbullying detection. As shown in the RNN confusion matrix (Figure 9A), the model misclassified 795 non-bullying tweets as bullying (false positives) and 596 bullying tweets as non-bullying (false negatives). While both types of errors are undesirable, false negatives are particularly critical in this context, as failing to identify a bullying instance could allow harmful content to persist unflagged. However, the low number of false negatives relative to the total sample suggests strong recall, particularly for the bullying class (97%). Similarly, the limited number of false positives supports the model's high precision (96%) in identifying actual bullying content without over-flagging benign posts. This balance between FP and FN reinforces the model's robustness and practical reliability in real-world applications.

5. DISCUSSION

To compare the proposed system with the latest available methods, we used a quantitative comparison between studies by selecting some recent studies that share three common aspects (language (Arabic), social media (Twitter) and data-collection source). However, the dataset used in those studies is different from the proposed one. In this regard, a comparison was made with three recent methods from 2023 [29]-[31], [35]-[36].

Table 5. Comparison between the proposed approach and state-of-the-art.

Approach	Feature Extraction	Classifier	Accuracy	Precision	Recall	F1-Score
[29]	TF- IDF, Wob	SVM, NB	95%,70%	92%	84%	88%
[30]	Non	LSTM	88%	88%	88%	88%
[31]	TF- IDF	MLP	89%	88%	90%	89%
[35]	Automatic feature extraction	hybrid LSTM-CNN	87.8%	N/A	83.6%	84.1%
[36]	AraBERT embeddings	GRU and BiLSTM with AraBERT embeddings	Accuracy: 87.05% (2-class), 78.99% (3-class), 75.51% (6-class)	N/A	N/A	N/A
[39]	standard text embeddings	CNN, LSTM and BiLSTM architectures.	91%	N/A	N/A	N/A
Proposed	TF- IDF	RNN	**97%**	**97%**	**97%**	**97%**

Based on the results shown in Table 5, previous studies utilizing traditional machine-learning techniques, such as SVM and NB, with feature-extraction methods, like TF-IDF and Bag of Words, have reported accuracies reaching up to 95%. However, these studies typically relied on smaller datasets, which may have contributed to inflated performance metrics due to reduced complexity. In contrast, our approach employed a standard RNN model trained on a large-scale (42,000 samples), multi-dialectal dataset, achieving an accuracy of 97%. This underscores the capacity of deep-learning models, particularly RNNs, to generalize effectively across more diverse and complex data, outperforming traditional algorithms when evaluated on a broader scale. Our findings are consistent with Obeidat et al. [37], who demonstrated that deep-learning models, such as RNNs, significantly outperformed traditional machine-learning approaches (e.g. SVM, Random Forest) in Arabic sentiment analysis on Twitter. This further supports the superiority of neural architectures in handling complex linguistic features in Arabic social-media content. The dataset referenced in [29]-[31], [35]-[36], [39] was used to evaluate the performance of our algorithms. In the broader context of Arabic-cyberbullying detection, our results extend prior literature by emphasizing the impact of both dataset size and dialectal diversity. For example, Al-Hassan and Al-Dossari [38] introduced one of the earliest benchmark datasets (~11K tweets) and reported an F1-score of 73% using CNN-LSTM models. Our study, leveraging a more comprehensive dataset, achieved significantly higher F1-scores using a simpler RNN architecture, highlighting the value of rich data over architectural complexity. Similarly, Al-Azani and El-Alfy [35] proposed a hybrid CNN-LSTM model that attained an F1-score of 84.1%. Despite their more intricate design, our RNN-based model achieved comparable or superior accuracy without relying on hybrid or ensemble methods, affirming that a well-optimized standard RNN can deliver state-of-the-art results when trained on appropriate data. Furthermore, Abu Kwaik et al. [36] combined GRU/BiLSTM models with AraBERT embeddings, reporting an AUROC of 0.84 and accuracies ranging from 75% to 87% across various classification tasks. Although their use of transformer-based contextual embeddings enhanced performance, our model demonstrated that even without such embeddings, classical RNNs can achieve competitive results, particularly when trained on diverse and large-scale datasets. In addition, the recent study by Alshahrani et al. [39] employed CNN, LSTM and BiLSTM architectures on a dataset of approximately 50,000 Arabic tweets, achieving an accuracy above 94%. However, their work did not focus on dialectal diversity or use RNNs. By contrast, our approach incorporated three distinct Arabic dialects and applied a standard RNN, achieving superior accuracy. This demonstrates that simpler architectures, when supported by carefully curated and dialect-diverse data, can outperform more complex models lacking linguistic variation. Collectively, these comparisons reinforce two critical conclusions of our study: (1) the effectiveness of deep learning in Arabic-cyberbullying detection is closely tied to dataset size and dialectal diversity and (2) standard RNN architectures remain a viable and efficient alternative to more complex hybrid or transformer-based models.

Although the size of the dataset was limited in the previous studies, applying the RNN algorithm yielded outstanding results. Regarding reference [29], we used their same dataset and applied our proposed RNN model to it. Our approach achieved an accuracy of 99.6%, compared to 95% reported in [29] using SVM. This confirms the superiority of our method, since the improvement was demonstrated on the same dataset under comparable conditions. Therefore, the performance gain is not only due to the size or structure of the dataset, but is directly related to the effectiveness of the proposed RNN-based architecture in capturing sequential patterns in Arabic text better than traditional classifiers, such as SVM. The comparison of the proposed approach with a closely related study is shown in Table 6.

Table 6. Comparison of the proposed approach with a closely related study [29].

Approach	**Classifier**	**Accuracy**	**Precision**	**Recall**	**F1- Score**
Proposed	RNN	**99.6%**	**99%**	**98%**	**98%**
[29]	SVM	95%	92%	84%	88%

While this study has demonstrated the effectiveness of the proposed algorithm in detecting cyberbullying in Arabic text, several limitations should be addressed in future work. Firstly, one challenge lies in the imbalance of the dataset, as the amount of cyberbullying content is often significantly lower than neutral or non-bullying content. This can affect the performance of the model and lead to a bias towards the majority class. In the future, techniques, such as data augmentation and oversampling, can be explored to balance the dataset and improve the detection accuracy. Furthermore, while our model achieved promising results, it may struggle to accurately interpret the context in longer and more complex sentences. In future studies, hybrid models combining RNNs with Transformers [45] could be explored to leverage the strengths of both approaches. Transformers, with their ability to capture long-range dependencies, could complement the sequential learning nature of RNNs, improving the overall model's understanding of the context. Moreover, challenges related to the diverse use of language and slang in cyberbullying cases, especially in Arabic, require further attention. Future research could focus on developing advanced pre-processing techniques and word embeddings to more effectively handle such linguistic variations. Finally, while this study provides valuable insights into cyberbullying detection using deep learning, future work should focus on overcoming these limitations through the integration of advanced techniques, such as hybrid models and better handling of data imbalance and contextual complexities.

6. CONCLUSIONS

Cyberbullying is becoming increasingly difficult to detect, as users can bully without being identified. Cyberbullying poses a threat to individuals and can lead to suicide or depression among victims, making its detection essential. Although there are many studies on this topic, most of them have focused on the English language, while there are only a few studies in Arabic. In the current study, we proposed and trained a different ML model to detect cyberbullying in Arabic comments of tweets from different dialects. This study achieved significant improvements in the performance of the proposed model using feature-extraction techniques. RNNs produced the best results when utilizing 27 echoes in perfect time.

REFERENCES

[1] W. N. H. Wan Ali, M. Mohd and F. Fauzi, "Cyberbullying Detection: An Overview," Proc. of the 2018 Cyber Resilience Conf. (CRC), pp. 1–6, DOI: 10.1109/CR.2018.8626869, Putrajaya, Malaysia, 2018.

[2] B. Srinandhini and J. I. Sheeba, "Online Social Network Bullying Detection Using Intelligence Techniques," Procedia Computer Science, vol. 45, pp. 485–492, DOI:10.1016/j.procs.2015.03.085, 2015.

[3] TechJury, "50 Alarming Cyberbullying Statistics to Know in 2024," [Online], Available: https://techjury.net/blog/cyberbullying-statistics/, Accessed: Jan. 2, 2025.

[4] Cyberbullying Research Center, "2023 Cyberbullying Data - Cyberbullying Research Center," [Online], Available: https://cyberbullying.org/2023-cyberbullying-data, Accessed: Aug. 27, 2024.

[5] Statista, "COVID-19 Vaccine: Adverse Events by Age and Gender in Spain," [Online], Available: https://www.statista.com/statistics/1220543/covid-19-vaccine-number-of-adverse-events-reported-by-age-and-gender-spain/, Accessed: May 10, 2025.

[6] UNICEF, "Search | UNICEF," [Online], Available: https://www.unicef.org/search?query=Statistic+cybeRbullying, Accessed: May 10, 2025.

[7] 7amleh, "7amleh - Annual Report 2023," [Online], Available: https://7amleh.org/annual23/eng/, Accessed: May 10, 2025.

[8] Ditch the Label, "Youth Charity | Mental Health, Bullying & Relationships," [Online], Available: https://www.ditchthelabel.org/cyber-bullying-statistics-what-they-tell-us, Accessed: Aug. 27, 2024.

[9] D. Musleh et al., "A Machine Learning Approach to Cyberbullying Detection in Arabic Tweets," Computers, Materials and Continua, vol. 80, no. 1, pp. 1033–1054, Jul. 2024.

[10] Statista, "Most Used Languages Online by Share of Websites 2024," [Online], Available: https://www.statista.com/statistics/262946/most-common-languages-on-the-internet/, Aug., 2024.

[11] A. Alqarni and A. Rahman, "Arabic Tweets-based Sentiment Analysis to Investigate the Impact of COVID-19 in KSA: A Deep Learning Approach," Big Data and Cognitive Computing, vol. 7, no. 1, p. 16, DOI: 10.3390/bdcc7010016, Jan. 2023.

[12] W. J. Hutchins, "The Georgetown-IBM Experiment Demonstrated in January 1954," Lecture Notes in Computer Science, vol. 3265, pp. 102–114, DOI: 10.1007/978-3-540-30194-3_12, 2004.

[13] A. Mandal, "Evolution of Machine Translation," Towards Data Science, [Online], Available: https://towardsdatascience.com/evolution-of-machine-translation-5524f1c88b25, Aug. 27, 2024.

[14] S. Almutiry, M. Abdel Fattah and S. Arabia-Almadinah Almunawarah, "Arabic CyberBullying Detection Using Arabic Sentiment Analysis," Egyptian Journal of Language Eng., vol. 8, no. 1, pp. 39–50, 2021.

[15] T. Kanan, A. Aldaaja and B. Hawashin, "Cyber-Bullying and Cyber-Harassment Detection Using Supervised Machine Learning Techniques in Arabic Social Media Contents," Journal of Internet Technology, vol. 21, no. 5, pp. 1409–1421, DOI: 10.3966/160792642020092105016, Sep. 2020.

[16] I. Abu El-Khair, "Effects of Stop Words Elimination on Arabic Information Retrieval," International Journal of Computing & Information Sciences, vol. 4, no. 3, pp. 119–133, 2006.

[17] M. A. Al-Ajlan and M. Ykhlef, "Deep Learning Algorithm for Cyberbullying Detection," Int. J. of Advanced Computer Science and Applications, vol. 9, no. 9, pp. 199-205, 2018.

[18] B. Haidar, M. Chamoun and A. Serhrouchni, "Arabic Cyberbullying Detection: Using Deep Learning," Proc. of the 2018 7th Int. Conf. on Computer and Communication Engineering (ICCCE), pp. 284–289, DOI: 10.1109/ICCCE.2018.8539303, Kuala Lumpur, Malaysia, Nov. 2018.

[19] B. Haidar, M. Chamoun and A. Serhrouchni, "A Multilingual System for Cyberbullying Detection: Arabic Content Detection Using Machine Learning," Advances in Science, Technology and Engineering Systems J., vol. 2, no. 6, pp. 275–284, DOI: 10.25046/AJ020634, 2017.

[20] B. Y. Alharbi et al., "Automatic Cyber Bullying Detection in Arabic Social Media," Int. J. of Engineering Research & Technology, vol. 12, pp. 2330–2335, 2019.

[21] D. Mouheb et al., "Detection of Arabic Cyberbullying on Social Networks Using Machine Learning," Proc. of the 2019 IEEE/ACS 16th Int. Conf. on Computer Systems and Applications (AICCSA), DOI: 10.1109/AICCSA47632.2019.9035276, Abu Dhabi, UAE, Nov. 2019.

[22] K. Reynolds et al., "Using Machine Learning to Detect Cyberbullying," Proc. of the 10th Int'l Conf. Mach. Learn. Appl. (ICMLA), vol. 2, pp. 241–244, DOI: 10.1109/ICMLA.2011.152, Honolulu, USA, 2011.

[23] J. Hani et al., "Social Media Cyberbullying Detection Using Machine Learning," Int. J. of Advanced Computer Science and Applications, vol. 10, no. 5, pp. 703–707, 2019.

[24] B. A. Rachid et al., "Classification of Cyberbullying Text in Arabic," Proc. of the IEEE Int. Joint Conf. on Neural Networks (IJCNN), DOI: 10.1109/IJCNN48605.2020.9206643, Glasgow, UK, Jul. 2020.

[25] T. D. Alsubait, "Comparison of Machine Learning Techniques for Cyberbullying Detection on YouTube Arabic Comments," Int. J. of Computer Science and Network Security, vol. 21, no. 1, pp. 1–5, 2021.

[26] V. Banerjee et al., "Detection of Cyberbullying Using Deep Neural Network," Proc. of the IEEE 2019 5th Int. Conf. on Advanced Computing & Communication Systems (ICACCS), pp. 604–607, DOI: 10.1109/ICACCS.2019.8728378, Coimbatore, India, Mar. 2019.

[27] C. Iwendi et al., "Cyberbullying Detection Solutions Based on Deep Learning Architectures," Multimedia Systems, vol. 29, no. 3, pp. 1839–1852, DOI: 10.1007/S00530-020-00701-5, Jun. 2023.

[28] D. A. Musleh et al., "Arabic Sentiment Analysis of YouTube Comments: NLP-based Machine Learning Approaches for Content Evaluation," Big Data and Cognitive Computing, vol. 7, no. 3, p. 127, Jul. 2023.

[29] K. T. Mursi et al., "ArCyb: A Robust Machine-learning Model for Arabic Cyberbullying Tweets in Saudi Arabia," Int. J. of Advanced Computer Science and Applications, vol. 14, no. 9, pp. 1059–1067, 2023.

[30] M. Alzaqebah et al., "Cyberbullying Detection Framework for Short and Imbalanced Arabic Datasets," J. of King Saud Uni. - Computer and Information Sciences, vol. 35, no. 8, p. 101652, Sep. 2023.

[31] A. M. Alduailaj and A. Belghith, "Detecting Arabic Cyberbullying Tweets Using Machine Learning," Machine Learning and Knowledge Extraction, vol. 5, no. 1, pp. 29–42, Jan. 2023.

[32] M. Khairy et al., "Comparative Performance of Ensemble Machine Learning for Arabic Cyberbullying and Offensive Language Detection," Language Resources and Evaluation, vol. 58, no. 2, pp. 695–712, DOI: 10.1007/S10579-023-09683-Y, Jun. 2024.

[33] A. H. Zahid et al., "Evaluation of Hate Speech Detection Using Large Language Models and Geographical Contextualization," arXiv, arXiv: 2502.19612, Feb. 2025.

[34] A. Charfi et al., "Hate Speech Detection with ADHAR: A Multi-dialectal Hate Speech Corpus in Arabic," Frontiers in Artificial Intelligence, vol. 7, p. 1391472, DOI: 10.3389/FRAI.2024.1391472, May 2024.

[35] A. Altayeva et al., "Hybrid Deep Learning Model for Cyberbullying Detection on Online Social Media Data," Int. J. of Computer Science, vol. 8, no. 3, Sep. 2022.

[36] A. Alhazmi et al., "Code-mixing unveiled: Enhancing the hate speech detection in Arabic dialect tweets using machine learning models," PLOS One, vol. 19, no. 7, p. e0305657, 2024.

[37] R. Obeidat et al., "Deep Learning *vs.* Traditional Machine Learning for Arabic Sentiment Analysis: A Comparative Study," Int. J. of Advanced Computer Science and Appl., vol. 12, no. 4, pp. 188–195, 2021.

[38] A. Al-Hassan and H. Al-Dossari, "A Benchmark Dataset for Arabic Cyberbullying Detection on Twitter: Design and Evaluation," Int. J. of Advanced Computer Science and Appl., vol. 11, no. 2, pp. 72–78, 2020.

[39] G. Jaradat et al., "Deep Learning Approaches for Detecting Cyberbullying on Social Media," J. of Computational and Cognitive Engineering, vol. 2025, no. 00, pp. 1–15, Mar. 2025.

[40] I. Jamaleddyn, R. El Ayachi and M. Biniz, "Novel Multi-channel Deep Learning Model for Arabic News Classification," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 10, no. 4, pp. 453–468, DOI: 10.5455/jjcit.71-1720086134, Dec. 2024.

[41] L. Al Qadi, H. El Rifai, S. Obaid and A. Elnagar, "A Scalable Shallow Learning Approach for Tagging Arabic News Articles," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 3, pp. 263–280, DOI: 10.5455/jjcit.71-1585409230, Sep. 2020.

[42] Haithem Hermessi, "Arabic Levantine Hate Speech Detection," [Online], Available: https://www.kaggle.com/datasets/haithemhermessi/arabic-levantine-hate-speech-detection, Jan. 2025.

[43] M. K. Saad, "Arabic Sentiment Twitter Corpus," [Online], Available: https://www.kaggle.com/datasets/mksaad/arabic-sentiment-twitter-corpus, Jan. 2025.

[44] A. Saleh, "Arabic Dataset1," [Online], Available: https://www.kaggle.com/datasets/ahmadsaleh2001/arabicdataset1, Jan. 2025.

[45] M. Tami et al., "Transformer-based Approach to Pathology Diagnosis Using Audio Spectrogram," Information, vol. 15, no. 5, p. 253, DOI: 10.3390/info15050253, 2024.

ملخص البحث:

لقد ظهر التّنمُّر السّيبراني كقضية مُلحّة في العصر الرّقمي، وبخاصة في المجتمعات المتحدّثة بالعربية حيث لا يزال البحث في هذا المجال محدوداً. وهذه الورقة تبحث في الكشف عن التّنمُّر السّيبراني بالعربية على وسائل التّواصل الاجتماعي باستخدام تقنيات التّعلُّم الآلي التّقليدية، وتقنيات التّعلُّم العميق. وقد جرى استخدام مجموعة بيانات لتغريدات بالعربية مُتاحة للعموم لتدريب وتقييم عدّة نماذج تعلُّم آلي، إلى جانب نموذج تعلُّم عميق قائم على شبكة عصبية (RNN).

وقد أثبتت النّتائج أنّ النّموذج القائم على الشّبكة العصبية تفوّق على النّماذج الأخرى المستندة إلى التّعلُّم الآلي، الأمر الّذي يُشير إلى فاعلية التّعلُّم العميق في التّحديد الدّقيق للمحتوى السّيّء في النّصوص المكتوبة باللُّغة العربية. وتؤكّد النّتائج ضرورة دمْج البيانات الغنية لغوياً والبِنى المتقدّمة المرتكزة إلى الشّبكات العصبية لتحسين عمل أنظمة الكشف عن التّنمُّر السّيبراني في اللُّغات المختلفة، وبخاصّةٍ العربية؛ لما تنطوي عليه من تعقيد وتنوُّع.

IMPROVING IOT SECURITY: THE IMPACT OF DIMENSIONALITY AND SIZE REDUCTION ON INTRUSION-DETECTION PERFORMANCE

Remah Younisse[1], Amal Saif[1], Nailah Al-Madi[1], Sufyan Almajali[1]
and Basel Mahafzah[2]

(Received: 16-Mar.-2025, Revised: 23-May-2025 and 24-Jun.-2025, Accepted: 27-Jun.-2025)

ABSTRACT

Intrusion detection in the Internet of Things (IoT) environments is essential to guarantee computer-network security. Machine-learning (ML) models are widely used to improve efficient detection systems. Meanwhile, with the increasing complexity and size of intrusion-detection data, analyzing vast datasets using ML models is becoming more challenging and demanding in terms of computational resources. Datasets related to IoT environments usually come in very large sizes. This study investigates the impact of dataset-reduction techniques on machine learning-based Intrusion Detection Systems (IDSs) regarding performance and efficiency. We propose a two-stage framework incorporating deep autoencoder-based feature reduction with stratified sampling to reduce the dimensionality and size of six publicly available IDS datasets, including BoT-IoT, CSE-CIC-IDS2018, and others. Multiple machine-learning models, such as Random Forest, XGBoost, K-Nearest Neighbors, SVM and AdaBoost, were evaluated using default parameters. Our results show that dataset reduction can decrease training time by up to 99% with minimal loss in F1-score, typically less than 1%. It is recognized that excessive size reduction can compromise detection accuracy for minority attack classes. However, employing a stratified sampling method can effectively maintain class distributions. The study highlights significant feature redundancy, particularly high correlation among features, across multiple IoT security-related datasets, motivating the use of dimensionality-reduction techniques. These findings support the feasibility of efficient, scalable IDS implementations for real-world environments, especially in resource-constrained or real-time settings. This work shows considerable redundancy in the datasets which questions the huge amount of these datasets, because, in many cases, the reduced datasets provide almost the same F1-score readings after data reduction. Rasing the alarm to notice the unnecessary massive amount of data used to build robust IDSs.

KEYWORDS

Dimensionality reduction, Data reduction, Autoencoders, Stratified sampling, Machine learning.

1. INTRODUCTION

Massive amounts of data are being generated due to digitization in different Internet of Things (IoT) domains, such as healthcare, vehicular networks [1]-[2], and Intrusion Detection Systems (IDSs) [3]. Two options are available for data reduction; reducing the number of features (feature reduction) or the number of tuples in the dataset (size reduction). Deep-learning (DL) techniques can deal with vast amounts of data. Still, DL only concerns some features in the data; thus, dimensionality reduction becomes an important step in best utilizing the resources [4]-[5].

Wearable devices, such as wearable healthcare devices, for example, generate a lot of features; it takes work to manage and store the generated data. It is hard to decide which features must be preserved for accurate diagnosis and which are not [6]. Due to the cost and computational resources needed to handle the enormous number of features, it becomes a challenge to reduce them without affecting the models' performance [7]. However, intrusion-detection datasets face unique issues. The extreme data imbalance is a major concern, where minority classes represent attack classes [8]. Hence, any reduction technique should consider the risk of eliminating them. Meanwhile, rapid learning and detection models are needed to enhance the detection process, because the sooner threats are detected, the less harmful the attacks are. Additionally, adversarial behaviors may intentionally mimic normal traffic, complicating feature learning. These challenges motivate the need for intelligent, attack-aware dataset-reduction strategies. Hence, the proposed approach in this study uses stratified sampling to maintain class balance and deep

[1] R. Younisse, A. Saif, N. Al-Madi, S. Almajali are with Department of Computer Science, Princess Sumaya University for Technology, Amman, Jordan. Emails: r.baniyounisse@psut.edu.jo, ama20219010@std.psut.edu.jo, n.madi@psut.edu.jo and s.almajali@psut.edu.jo

[2] Basel Mahafzah is with Computer Science Department, King Abdullah II School of Information Technology, The University of Jordan, Amman 11942, Jordan.

autoencoder-based feature extraction to preserve non-linear patterns and subtle feature dependencies critical for effective IDS performance.

Different dimensionality-reduction techniques could be used based on the data complexity, such as Principal Component Analysis (PCA), MDS, and Time-lagged independent component analysis (TICA) for linear manifolds, and Sketchmap, t-SNE, and deep methods for non-linear manifolds [9]. Principal Component Analysis (PCA) has been widely used in dimensionality reduction. It helps provide better data quality, improve classification, reduce the needed space and time, and remove irrelevant data [10].

At the same time, data reduction techniques are becoming popular and widely used for data visualization, simulation and analysis [11]. Stratified sampling is a famous method that divides data into similar groups known as strata [12]. Then, it selects a certain number of samples from each group, considering the data's distribution rate; any sample taken from the data should keep the same distribution in the original dataset. Stratified sampling was proven to be an efficient, unbiased sampling method and highly representative of the data being studied. The main drawback of stratified sampling is that it can only be applied when the data cannot be grouped in disjoint groups [13].

In recent years, many intrusion-detection datasets have been generated due to the rapid updates of the malware authors, and different attacks have been developed to maneuver different IDSs. It has been noticed that these datasets tend to be large, with millions of tuples and hundreds of features. Hence, different reduction techniques have to be studied and improved. The main motivation for this paper is to explore the value of using huge datasets to train machine learning (ML)-based IDSs and to assess the effect of reducing the size of the datasets used on these IDSs. Thus, this assessment work investigates the efficiency of different data reduction and feature-extraction techniques. Reducing the datasets' sizes will help improve the required ML-based IDS training time.

In the context of intrusion detection and dimensionality reduction, many works have focused on feature reduction techniques to speed up the ML models and enhance the outcomes of these models [14]-[15]. In comparison, the size-reduction aspect is not sufficient for the research work. One reason for this is the risk associated with removing potentially valuable information, primarily in class-imbalanced datasets where minority attack classes are already underrepresented. Unlike feature selection, which can often enhance generalization by removing noise and redundancy, size reduction, if not handled carefully, can negatively impact detection accuracy. Moreover, feature reduction methods compress dimensionality while keeping the overall event diversity. Our work aims to fill this gap by proposing a controlled size-reduction approach using stratified sampling, ensuring that data diversity and class proportions are preserved even in smaller training sets.The work in [16] explored how deep-learning models can be used as a feature-extraction tool aiming to remove redundant features from the dataset. The experiment was applied to an outdated balanced dataset with relatively small features. Meanwhile, information gain (IG-PCA) was also used as a dimensionality-reduction tool in [17]. In [18], two different feature-reduction methods were investigated with a more recent dataset than the dataset used in the previously mentioned works: the CISIDS2017 dataset.

This paper focuses on answering two questions. The first question is, "Is the large amount of data collected in IDS datasets needed to build robust IDS ML and AI systems?". The second question is, "What efficient reduction techniques can be used to reduce the size of IDS datasets, yet they can be used to build robust IDSs?".

Some works have focused on combining size and dimensionality-reduction techniques to extract the dataset's core value and enhance machine-learning (ML) model performance, but not in the context of intrusion-detection applications, such as the work in [19]. The primary contribution of this study is a practical framework for enhancing IDS performance through efficient data reduction rather than a novel detection algorithm. The proposed model combines deep feature extraction using autoencoders with stratified sampling to reduce the number of features and training samples without compromising classification performance. This two-stage reduction process significantly lowers computational costs and model complexity, making machine learning-based IDS solutions more scalable and suitable for real-time applications, especially in IoT scenarios. Experiments on six IDS-related datasets demonstrate that the proposed method preserves or even improves F1-scores while reducing training time by up to 99% in some cases. Therefore, this work's main contributions can be summarized as follows:

- We present a practical, two-stage dataset-reduction framework that combines stratified sampling with autoencoder-based feature selection to reduce both the size and dimensionality of IDS datasets. The first algorithm sorts the importance of the features in the dataset *via* an autoencoder. Then, the least important features are removed, followed by tuple reduction *via* stratified sampling. The second algorithm starts with stratified sampling, followed by feature ranking and selection.
- We empirically evaluate the trade-offs between different reduction percentages and their effects on training time and detection performance using multiple ML models across six public IDS datasets.
- We show that, when properly applied, dataset size and dimensionality reduction can achieve up to 99% decrease in training time with minimal performance loss (typically less than 1% drop in F1-score). The proposed reduction techniques prove that there is a notable degree of redundancy in the datasets. The huge amount of data should be questioned in these datasets, because, in many cases, the reduced datasets provide almost the same F1-score readings after data reduction. More attention should be paid to the unnecessary massive amount of data used to build robust IDSs.
- We provide a reproducible baseline for evaluating dataset-reduction strategies in IDSs, offering insights into scalability and efficiency for real-world deployment in resource-constrained environments.

The rest of this paper is organized as follows: Section 2 shows the related work. Preliminaries and methodology are presented in Section 3. Section 4 shows the results and assessments, and finally, the work is concluded in Section 5.

2. RELATED WORK

Data-reduction techniques are widely explored to address machine-learning datasets' growing complexity and size, specifically in intrusion detection systems (IDSs). These techniques typically fall into two categories: dimensionality reduction, which reduces the number of features (columns), and size reduction, which reduces the number of records (rows). This section critically investigates related works grouped by technique type and discusses their applicability to IDSs, mainly in IoT environments.

Linear techniques, such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), have been broadly used to project data into lower-dimensional spaces. PCA is widely employed due to its computational efficiency and ability to decorrelate features. PCA has shown considerable performance with high-dimensional datasets, such as medical imaging and network traffic [20]-[21]. However, PCA supposes linear relationships between the features, which may not hold in complex IDS datasets. At the same time, LDA is useful for maximizing class separability, but suffers from scalability issues in large-scale, high-dimensional environments. Recent work has focused on autoencoders and their variants, including Deep Sparse Autoencoders (DSAEs), to handle these restrictions for non-linear and data-driven feature extraction. Unlike PCA, autoencoders do not suppose linearity and can model complicated feature relations [22]. This capability of modeling complex relationships makes them specifically suitable for IDS datasets with complex patterns and correlations. For instance, [23] used autoencoders to improve classification accuracy through feature selection. However, their work focused on general accuracy rather than on IDS-specific issues, like class imbalance or real-time deployment. Recently, Nabi and Zhou [24] explored using PCA and random projection for dimensionality reduction in intrusion-detection schemes using the NSL-KDD dataset. Their results emphasized random projection's computational efficiency and accuracy benefits over PCA. In contrast, this study explores deep autoencoders as a non-linear and data-driven technique for feature extraction and links this with a structured dataset size-reduction pipeline. Moreover, this evaluation spans multiple recent IDS datasets, addressing generalization, dataset redundancy, and attack-class preservation. In contrast to earlier studies that used traditional datasets, like NSL-KDD or outdated benchmark sets [16]-[17], our work leverages recent and large-scale IDS datasets, such as CSE-CIC-IDS 2018 and BoT-IoT. When integrated with stratified sampling, we confirm that autoencoders can preserve detection performance even under significant feature reduction.

Stratified sampling is a widely utilized technique for reducing dataset size while keeping class distributions, which is critical in class-imbalanced IDS contexts. Multiple studies [25][26][27][28] have examined its effect on handling large-scale datasets. For example, [28] proposed an enhanced stratified sampling framework with over-sampling of minority classes using Gaussian noise and clustering of majority classes. However, these works frequently lacked comparative analysis of reduction order, sampling before *vs.* after feature reduction. Moreover, some prior works overlap in their discussion of stratified sampling without clearly distinguishing their contributions. We address this by systematically comparing each method's novelty and outcome: [25] applied stratified sampling in general big-data contexts, [26] optimized sampling with hash- based stratum construction, and [27] integrated stratified sampling with clustering for better illustration. Our method builds upon these by integrating sampling with deep feature selection, presenting a unified pipeline evaluated on multiple IDS datasets.

Recent developments have introduced scattering-based enhancements to graph neural networks for anomaly detection and feature learning. For instance, the STEG model [29] applies a wavelet-based scattering transform to edge features within an E-GraphSAGE architecture, significantly improving detection performance on network-intrusion datasets. STEG leverages multi-resolution edge encoding and node2vec embeddings to provide a fine-grained understanding of graph-structure anomalies, a strategy relevant to our anomaly-detection pipeline. In a related domain, the GeoScatt-GNN framework [30] combines geometric-scattering transforms with ANOVA-based statistical feature selection to predict Ames mutagenicity. While its application lies in bio-informatics, the architecture introduces a principled pipeline where meaningful features are extracted and filtered prior to GNN classification, highlighting the cross-domain effectiveness of scattering-transform approaches. Our work draws inspiration from these efforts, but focuses on reducing the dataset size, with a tailored architecture and feature-selection approach suited to network-level anomaly scenarios. We also emphasize the redundancy happening in the security-related dataset applied in the IoT environments.

A summary of the related works and methods is clarified in Table 1. Our approach closes this gap by employing a two-stage pipeline tested across six modern IDS datasets and comparing sampling-first *vs.* feature-first strategies. Additionally, we quantify training-time reduction and model resilience to aggressive reduction analysis, which previous studies often dismissed. The datasets related to security threats in IoT networks tend to be massive, hindering the detection models and requiring huge computational resources [31]. This work presents a methodology that can reduce dataset size while keeping the IDS performance high and accurate. Our work offers a more rigid, application-focused synthesis of dimensionality and size reduction in IDSs. It advances the field by addressing the interplay between reduction type and model performance using large-scale, recent IDS datasets. It also provides empirical proof across multiple classifiers and offers a reproducible framework for real-world deployment.

Table 1. Summary of data-reduction techniques in literature.

Technique	**Category**	**Dataset Used**	**Strengths**	**Limitations**
PCA [20]-[21], [24]	Dimensionality	NSL-KDD, CTG, DR	Fast, simple, linear separability	Fails on non-linear data
LDA [16]-[17]	Dimensionality	CTG, DR	Class separation-focused	Poor scalability
Deep Autoencoders [22]-[23]	Dimensionality	BoT-IoT, CSE- CIC-IDS2018	Handles non-linear features, scalable	Overfitting risk on small datasets
Stratified Sampling [25][26][27][28]	Size	KDD, CICIDS, financial	Maintains class balance	Requires stratification label
Sampling + Clustering [27]-[28]	Size	Big-data Clusters	Reduces outliers, enhances sample diversity	Adds clustering complexity
This Work	Size + Dimensionality	6 modern IDS datasets	Two-stage, flexible, efficient	Minor NB performance degradation noted

3. PRELIMINARIES AND METHODOLOGIES

This section introduces the datasets used in this study and the methodologies that are applied to reduce the size and dimensionality of the datasets. It also introduces the performance metrics that have been used to assess the efficiency of the methodologies used.

3.1 Datasets

Throughout this study, six intrusion-detection datasets were used: The Kitsune-ARP dataset [32], SNMP- MIB [33], the CSE-CIC-IDS2018 dataset [34], the BoTIoT dataset [35], the UNR-IDD dataset [36], and the credit-card fraud-detection dataset from [37]. The six datasets are all related to intrusion-detection applications, and they are collected from different hostile environments with different features and sizes.

What distinguishes the selected datasets in this study is that many datasets were recently collected from IoT environments. The datasets are challenging due to data imbalance, which is typical in intrusion-detection datasets in general. Meanwhile, many datasets are enormous, challenging for ML models, require a long time for training, and can result in very complex ML models. All non-binary datasets were transformed into binary datasets, such as the CSE-CIC-IDS2018 dataset.

A summary of the six datasets, including the size, dimension, and imbalance rate, is presented in Table.2. The datasets were renamed DS1-DS6 throughout this work, as shown in Table 2, to enhance the readability of the paper, especially the figures and tables.

Table 2. Summary of the datasets used in the study.

Dataset	Size (KB)	Records	Features	Imbalance Rate (%)
UNR-IDD (DS1)	267	2620	21	9.4
Kitsune-ARP (DS2)	15,300	15000	115	10
SNMP-MIB (DS3)	788	5000	34	10
CSE-CIC-IDS2018 (DS4)	315,233	1048576	80	50
BoTIoT (DS5)	620,600	2426574	24	27
Fraud detection (DS6)	100,500	248808	31	0.1724

3.2 The Techniques Used

We present here the main techniques used throughout this study. These techniques include sampling, dimensionality reduction, and ML techniques.

Sampling is selecting a representative set of items from a larger set. Sampling can be applied to select a specific number or percentage of samples. This work uses sampling with intrusion-detection datasets to select a certain percentage of the dataset to train the ML models, since many datasets are very large and contain hundreds of thousands of records, sometimes millions. Training machine-learning models with huge datasets requires high computational power and consumes time. The sampling process investigates the degree of redundancy existing in these datasets. When a half or a quarter of the data can be used to train the ML model and still give the same results as when the entire dataset was used, this can indicate that the dataset records include a noticeable degree of redundancy.

This work deploys stratified sampling to reduce the number of records in intrusion-detection datasets; meanwhile, it maintains the imbalance ratio. Since the data used is large and imbalanced, randomly selecting a small group from the data might alter the balance of the data; there is a more significant probability that a selected record belongs to the larger group. The datasets are also labeled, which makes stratified sampling a good choice for this presented work.

At the same time, dimensionality reduction is used for different purposes, such as having interpretable models or reducing the required computational time for ML-model training. PCA is commonly used for this task. The main difference between reducing the features using PCA and by autoencoders is that PCA can model linear structures. However, autoencoders do not assume linearity [18]. In [9], dimensionality-reduction techniques were divided into three main categories based on the data structure. Three main dimensionality-reduction methods are available in the literature: linear manifolds, non-linear manifolds and curved twisted manifolds.

Due to the high performance of autoencoders in reducing the intrusion detection features that outperform other methods, such as PCA and LDA [38]; dense autoencoders are used in the proposed methods in this paper. The main idea of the autoencoder is to have the ability to reconstruct the input after encoding it to a lower dimension. For dimensionality reduction, the most important part of the autoencoder is the latent space, the encoder's output, which has the most critical features of the input. Its size is a hyperparameter that can be controlled to define the desired number of features. In the proposed methods, features with the highest weights were selected after sorting all features based on their importance.

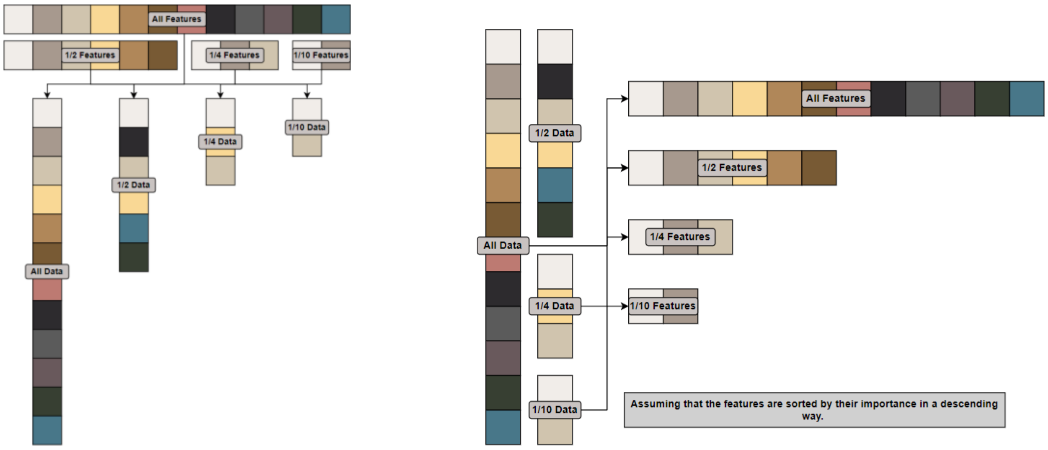
Eventually, the selected approaches to reduce the size and dimensionality of different intrusion-detection datasets are evaluated with nine different ML models. These ML models are the K-Nearest Neighbor algorithm (KNN), the Support Vector Machine Algorithm (SVM), Naive Bayes, linear regression, LDA, C5, XGBoost, Random Forest, and ADA. These ML models were selected throughout this study, because they are extensively used in the literature with similar datasets. The ML models were proven to be efficient and durable with tabular datasets. The random forest is a robust ensemble model that reduces overfitting and performs well on tabular data with noisy or redundant features. XGB is an ensemble model that proved its efficiency in many real problem-detection tasks. SVM model has a powerful feature which is called kernel trick that gives SVM the power to handle binary classification effectively. The KNN is a simple, non-parametric model that benefits from reduced feature spaces and works well for pattern recognition. AdaBoost is an ensemble technique that adapts to classification errors, making it more robust to decide on samples, which is very important with imbalanced datasets.

3.3 Proposed Methods

This work investigates how dataset size-reduction and feature-reduction methodologies affect machine-learning algorithms. The analysis is studied in the context of IDS systems and IoT environments. The method followed throughout the proposed work adapts two approaches clarified in Algorithms 1 and 2. In the first approach, data reduction is applied first, followed by size reduction, and then ML models are used with the data to build the IDS models. In the second approach, size reduction is applied before the feature-reduction step, and then ML models are used again to build the IDS models. Finally, the performance of the models built with the first approach is compared with the performance of those created with the second approach, as shown in Figure 1. The approach that produces better results is recommended for IDS datasets. The size-reduction method used throughout this study is the stratified sampling technique. Meanwhile, the feature reduction method used here is the dense autoencoder method.

Figure 1 illustrates the two-stage dataset-reduction strategies evaluated in this study. In the Feature Reduction First (FF) approach (Figure 2a), the full dataset is used to train an autoencoder, which ranks features based on their importance. The dimensionality of the dataset is then reduced by selecting the top-ranked features; 1/2, 1/4, or 1/10 of the full set. Finally, stratified sampling, 1/2, 1/4, or 1/10 of the full set, is applied on the reduced dataset to create reduced sub-sets for training, preserving class distribution. On the other hand, the Sampling First (SF) technique (Figure 2b) starts by applying stratified sampling directly to the full dataset, yielding sub-sets that are 1/2, 1/4, or 1/10 the dataset size. Each reduced sub-set is then passed to an autoencoder to perform feature reduction. The reduction at this stage is applied to extract 1/2, 1/4, 1/10, or full features. This order assumes that features are sorted in descending order of importance after training, as indicated in the figure. The main difference between the two methods is the timing of dimensionality reduction relative to volume reduction. FF (Feature reduction First) guarantees that the autoencoder is trained on the most complete data to capture more prosperous feature patterns. SF (Sampling First), meanwhile, reduces computational load earlier, but may lose important patterns due to early sub-sampling. This trade-off is critical when working with class-imbalanced and high-dimensional IDS datasets.

DS1, DS2, and DS3 were used through the investigation and steps mentioned in the previous paragraph. Meanwhile, DS4, DS5, and DS6 were used throughout the assessment process due to their large size where applying the reduction techniques is essential. During the assessment stage, we practically try to evaluate the performance of different ML models and the order of the reduction process. Time-analysis results, besides F-score measures, are recorded. In the second stage, we aim to prove the correctness of the conclusions made in the first stage. For example, time-reduction and close-to-perfect performance measures are used, despite using fewer data and features. In this study, we list only the F1-score as a suitable performance measure, which combines precision and recall. This allows us to evaluate the robustness of IDS performance across different levels of dataset reduction in a compact and interpretable way. Although additional metrics, such as recall, precision, and false-positive rate, were computed, their trends closely followed the F1-score. For clarity and space efficiency, only the F1-score is reported in the main-results tables, as it sufficiently describes the robustness of IDS performance under dataset reduction.



(a) Feature Reduction First (FF) Approach

(b) Size Reduction (SR) First Approach

Figure 1. The methodology followed to reduce the datasets.

Algorithm 1 Feature Extraction First (FF)

```
Require: Input dataset $DSi$
for each dataset $DSi$, $i \in [1,2,3]$ do
    for each $F$ in $\{1, 2, 4, 10\}$ do
        Apply feature extraction on $DSi$, extracting the most important $1/F$ features from $DSi$ and update $DSi$
        for each $F \in \{1, 2, 4,10\}$ do
            Apply stratified sampling on $DSi$ to extract $1/S$ of the data: $DSi \leftarrow DSi/S$
            Apply the machine learning methods to $DSi$
        end for
    end for
end for
```

Algorithm 2 Size Reduction First (SF)

```
Require: Input dataset $DSi$
for each dataset $DSi$, $I \in 1,2,3$ do
    for each $S$ in 1, {2, 4, 10 do}
        $DSi$ = StratifiedSampling($DSi$, $1/S$)
        for each $F \in \{ 1, 2, 4, 1\}0$ do
            $DSi$ = AutoencoderFeatureExtraction($DSi$, $1/F$ )
            Apply the machine learning methods to $DSi$
        end for
    end for
end for
```

We investigate how the datasets' size and feature reduction can affect the performance of nine different ML models. The models were trained with the data prior to reduction, then trained with the reduced datasets. Size and dimensionality were reduced in different scenarios, and then a comparison was held to assess the different reduction scenarios. The method used for data reduction is the Stratified-sampling process which reduces the data size and keeps the data distribution untouched.

The technique used for the feature-reduction process is ranking the importance of all features of the datasets using a dense autoencoder. Every dataset was used to train the autoencoder and then, the encoder was used to explore and rank the importance of all features based on their weights. The features were then sorted, and the less critical features were dropped from the dataset. Many scenarios were examined; a half of the features were selected, and one-fourth and one-tenth of the features were selected in other scenarios. Selecting-all-features scenarios were also analyzed.

The encoder architecture with the bottleneck consists of three dense layers with the LeakyReLU activation function and two batch-normalization layers, as shown in Figure 2. The autoencoder designed in this study follows a symmetrical architecture tailored for reconstructing input features while capturing meaningful representations in its bottleneck layer. The input-layer size corresponds directly to the number of features in each dataset. The encoder consists of two fully connected layers: the first layer expands the dimensionality to twice the input size and applies a LeakyReLU activation function, followed by batch normalization. At the same time, the second layer reduces the dimensionality back to the original feature size using the same activation and normalization setup. The bottleneck layer maintains this same dimensionality, serving as the latent representation of the input data without applying compression, allowing for feature-importance extraction. The decoder mirrors the encoder in structure, reconstructing the data through symmetric dense layers and concluding with a linear activation function in the output layer. The architecture was selected to balance expressive power and computational efficiency, particularly for high-dimensional, imbalanced intrusion-detection datasets where non-linear patterns and feature interactions are prevalent. The model was trained using the Adam optimizer with a learning rate of 0.001, a batch size of 16, and 100 epochs. The trained encoder was used to extract latent feature weights; all feature weights were reported without reduction at this level, which were subsequently ranked to identify the most important features. While this work focuses on autoencoder-based feature extraction, we acknowledge the importance of traditional methods, such as ANOVA and chi-square [39]. However, these classical approaches rely on assumptions of linearity and independence among features, which are often violated in intrusion detection scenarios. Autoencoders, by contrast, provide the flexibility to model complex, non-linear, and correlated feature interactions more effectively. However, although autoencoders offer powerful non-linear feature-extraction capabilities, they also introduce certain limitations. One concern is the risk of overfitting when training deep models on reduced datasets. They also require high computational capabilities when very large datasets are used. Hence, they should be used with caution to deliver accepted results while requiring minimal computational power.

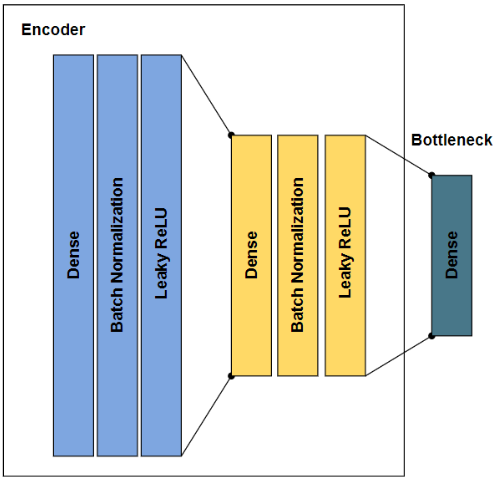


Figure 2. Encoder architecture.

Stratified sampling was used to reduce the size of the data. Every dataset was reduced to one half, one-fourth, and one-tenth; it was also analyzed without size reduction. The experiment goes through different steps, aiming to explore the efficiency of different reduction strategies. The whole data was analyzed with all features, a half of the features, one-fourth, and one-tenth using the nine ML models, which will be mentioned shortly. The exact process was repeated when one-fourth of the data was used, and one-tenth of the data was used. Reducing the data size followed by feature reduction is noted by (SF), which indicates "Sampling First " since the sampling method is applied to the data before the feature-reduction process. It is worth mentioning that when stratified sampling was applied before feature ranking, the importance ranks of some features changed due to the reduced dataset size. However, the most significant features showed minimal change in their ranking. During (FF) or "Feature First," the previously mentioned data-reduction process was applied, but feature reduction was applied first to the data, followed by the sampling step.

The used ML models are KNN, SVM, Naive Bayes, linear regression, LDA, C5, X-GBoost, Random Forest, and ADA. All models are evaluated with their default hyper-parameters as provided by sklearn Python libraries. Employing default parameters was to emphasize the practical applicability and effectiveness of the presented dataset-reduction techniques without requiring exhaustive hyper-parameter tuning. This setup demonstrates that meaningful improvements in computational efficiency and model performance can be achieved without additional optimization steps.

All datasets were normalized through a MinMaxScalar. During pre-processing, non-numerical features were dropped from the datasets, such as the Timestamp feature from the CSE-CIC-IDS2018 dataset and " Switch ID" and "Port Number" from DS1. Data pre-processing includes multiple steps, guaranteeing that the AI models will be fed with proper data values. During categorical feature encoding, all categorical features were encoded using One-Hot Encoding, transforming them into numerical formats suitable for machine-learning models. Moreover, features containing more than 50% of missing data were removed from the dataset. The remaining missing values were handled using mean imputation. Additionally, numerical features were scaled using Min-Max normalization, mapping feature values to a range between 0 and 1. This normalization improves model convergence and performance stability. Finally, stratified sampling was used explicitly to maintain the original class distribution, effectively managing dataset imbalance during data reduction. All datasets were divided into 64% for training and 36% for testing, while the 70/30 or 80/20 splits are widely used as standard practice. The slightly non-standard split in this study ensured that a representative portion of the minority class remained in the testing set, which is particularly important for performance evaluation on imbalanced datasets.

4. RESULTS AND ASSESSMENT

This section presents the results of the data-reduction techniques described in the previous section and investigates how combining different reduction techniques influences the ML models used. All the experiments were conducted using the Google Co-Lab platform based on Python 3. Google Co-Lab offers 12 GB RAM and 128 GB Disk. To rank feature importance, absolute weights from the first dense layer of the encoder were extracted. These weights reflect the strength of the connection between input features and their influence on the latent representation. We ranked in descending order based on the sum of absolute weights across all neurons in this layer. We then selected the top-k features: 1/2, 1/4, or 1/10 for further evaluation. Stratified sampling was applied using a class-wise sampling strategy to maintain class proportions. This was done *via* pandas.groupby(’class’).apply(lambda x: x.sample(frac=p)) in Python, where p is the target sampling fraction; 0.5, 0.25, 0.10. This method was used to generate progressively smaller, but balanced, datasets for training and testing. This step was either applied before or after feature selection based on the reduction strategy (SF or FF).

4.1 Machine-learning Model Results

To detail each model’s performance, the F-score metric is used to represent the results as values for all steps of the two approaches in Tables 3, 4, 5, 6, 7, and 8, because F-score is sufficient measure for imbalanced data. The numbers at the top of the columns represent the feature percentage and the size percentage; F-S "0.5–0.25," in Table 4, for example, denotes the ML models’ performance with a data sample retaining the top half of the features after ordering them according to their importance. If we have 20 features, for example, the top-10 features are used. Meanwhile, 0.25 means that one-fourth of the data tuples are used; for example, if we have 1000 tuples, 250 tuples are selected *via* stratified sampling and used through the training and testing processes. Features extraction precedes size reduction in this case where the “F” comes first. However, S-F "0.5–0.25," indicated using 50% of the tuples and 0.25 of the features where the size reduction precedes the feature extraction method.

In Table 3, all classifiers achieve a high F-score until the 0.1-1 reduction is applied, starting with size reduction. This is expected due to the small size and dimensions of DS1. However, when the reduction processes are swapped in Table 4, LR, SVM, and C4.5 can still produce high results. The conclusion that can be extracted from these results is that feature reduction first is better for small-sized and low-dimensional datasets. Moreover, RF, KNN, C4.5, and XGB are the best classifiers for DS2 based on the F-score when applying size reduction first, as shown in Table 5. XGB is the most stable classifier when feature reduction is applied first. At the same time, other models were unstable or could not achieve high F-scores in most data-reduction scenarios, as Table 6 demonstrates. As for the third dataset, KNN is the best classifier for size reduction first, as shown in Table 7 and XGB is the best for feature reduction first, as shown in Table 8. Most of the classifiers performed well, and it was the most suitable dataset for NB. In all Tables 3-8, we have colored the highest values in each column in yellow to highlight the best results for each division, also to highlight the best-performing models. Our analysis shows that dataset and feature-reduction strategies exhibit multiple levels of performance degradation. Decreasing the dataset or feature set to 1/2 or 1/4 generally resulted in a less than 2% drop in F1-score. More aggressive reduction to 1/10 greatly affected detection accuracy, particularly for complex datasets, like BoT-IoT and CSE-CIC-IDS2018. Notably, KNN and AdaBoost shared larger performance drops under 1/10 feature reduction, due to their sensitivity to input dimensionality. In contrast, ensemble tree- based models, such as XGBoost and Random Forest, showed higher resilience, maintaining performance even when trained on only 10% of features or samples. This indicates that the model's robustness to feature sparsity and sample diversity plays an essential function in mitigating the effects of reduction. These trade-offs emphasize the significance of choosing the proper model and reduction level based on the dataset's complexity and attack distribution.

Table 3. DS1 sampling first F1-score results.

S–F	1–1	1– 0.5	1– 0.25	1– 0.1	0.5– 1	0.5– 0.5	0.5– 0.25	0.5– 0.1	0.25– 1	0.25– 0.5	0.25– .025	0.25– 0.1	0.1–1	0.1– 0.5	0.1– 0.25
KNN	**1.000**	**1.000**	**1.000**	0.923	**1.000**	**1.000**	**1.000**	0.885	0.999	0.976	**1.000**	0.923	0.560	0.498	0.500
SVM	0.995	**1.000**	**1.000**	0.885	**1.000**	0.976	**1.000**	0.885	0.999	0.999	**1.000**	0.923	0.500	0.500	0.500
NB	0.991	0.993	0.998	0.508	0.986	0.993	0.995	0.514	0.974	0.983	0.986	0.982	0.543	0.535	0.529
LR	0.999	**1.000**	**1.000**	0.846	0.999	0.976	0.962	0.885	0.999	0.988	0.885	0.692	0.500	0.500	0.500
LDA	0.999	**1.000**	**1.000**	1.000	0.999	0.976	**1.000**	1.000	0.999	0.998	**1.000**	1.000	0.500	0.500	0.500
C4.5	**1.000**	**1.000**	**1.000**	0.962	**1.000**	**1.000**	**1.000**	0.962	1.000	**1.000**	**1.000**	1.000	0.500	0.500	0.500
XGB	**1.000**	**1.000**	**1.000**	0.962	**1.000**	**1.000**	**1.000**	0.962	1.000	**1.000**	**1.000**	1.000	0.500	0.500	0.500
RF	**1.000**	**1.000**	**1.000**	1.000	**1.000**	0.976	**1.000**	1.000	1.000	**1.000**	**1.000**	1.000	0.500	0.500	0.500
Ada	**1.000**	**1.000**	**1.000**	1.000	**1.000**	**1.000**	**1.000**	1.000	1.000	**1.000**	**1.000**	1.000	0.500	0.500	0.500

Table 4. DS1 feature extraction first F1-score results.

F-S	1–1	1– 0.5	1– 0.25	1– 0.1	0.5– 1	0.5– 0.5	0.5– 0.25	0.5– 0.1	0.25– 1	0.25– 0.5	0.25– .025	0.25– 0.1	0.1– 1	0.1– 0.5	0.1– 0.25	0.1– 0.1
KNN	**0.999**	**0.999**	**0.999**	**0.999**	**1.000**	0.990	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.990	**1.000**	0.333
SVM	0.995	0.995	0.995	0.995	0.998	0.992	0.995	0.973	**1.000**	0.984	0.798	0.471	0.809	0.781	0.491	**1.000**
NB	0.918	0.918	0.918	0.918	0.844	0.836	0.814	0.791	0.885	0.843	0.983	**1.000**	0.764	0.708	0.565	**1.000**
LR	**1.000**	**1.000**	**1.000**	**1.000**	0.668	0.544	0.470	0.468	0.465	0.470	0.459	0.471	0.465	0.467	0.491	0.333
LDA	**1.000**	**1.000**	**1.000**	**1.000**	0.791	0.749	0.723	0.851	0.803	0.779	0.459	**1.000**	0.465	0.466	0.491	0.333
C4.5	**1.000**	**1.000**	**1.000**	**1.000**	0.996	**1.000**	**1.000**	0.931	0.998	0.973	0.964	**1.000**	**1.000**	0.942	**1.000**	**1.000**
XGB	**1.000**	**1.000**	**1.000**	**1.000**	0.996	**1.000**	**1.000**	0.944	0.998	0.978	0.982	0.818	0.999	0.980	0.491	0.000
RF	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.959	0.999	**1.000**	0.964	0.884	0.999	0.960	0.824	0.333
Ada	**1.000**	**1.000**	**1.000**	**1.000**	0.999	0.997	**1.000**	0.959	0.999	**1.000**	**1.000**	0.884	0.960	0.969	0.491	0.333

Table 5. DS2 sampling first F1-score results.

S-F	1–1	1– 0.5	1– 0.25	1– 0.1	0.5– 1	0.5– 0.5	0.5– 0.25	0.5– 0.1	0.25– 1	0.25– 0.5	0.25– 0.25	0.25– 0.1	0.1– 1	0.1– 0.5	0.1– 0.25	0.1– 0.1
KNN	**0.999**	0.995	0.984	0.975	0.997	0.991	0.976	0.945	0.992	0.974	0.939	0.944	0.991	0.985	0.952	0.907
SVM	0.837	0.806	0.509	0.473	**0.839**	0.785	0.609	0.473	0.532	0.499	0.473	0.473	0.588	0.554	0.473	0.473
NB	0.250	0.253	0.249	0.255	0.233	0.234	0.232	0.240	0.235	0.240	0.169	0.179	0.412	0.417	**0.431**	0.407
LR	**0.882**	0.842	0.763	0.615	0.833	0.782	0.733	0.541	0.606	0.565	0.529	0.473	0.552	0.540	0.473	0.473
LDA	**0.978**	**0.978**	0.967	**0.990**	0.944	0.959	0.947	0.965	0.807	0.836	0.815	0.827	0.602	0.617	0.576	0.550
C4.5	**1.000**	**1.000**	0.998	0.985	0.998	0.995	0.992	0.995	**1.000**	0.995	0.994	**1.000**	0.995	0.977	0.955	0.946
XGB	**1.000**	0.998	0.998	0.995	**1.000**	0.998	0.996	0.995	**1.000**	0.999	0.994	0.995	0.998	0.995	0.975	0.985
RF	**1.000**	0.999	0.996	**1.000**	**1.000**	0.999	0.990	0.985	0.999	0.994	0.981	0.990	0.993	0.988	0.957	0.969
Ada	**0.999**	0.996	0.988	0.995	0.977	0.990	0.983	0.967	0.991	0.988	0.964	0.995	0.680	0.657	0.754	0.710

Table 6. DS2 feature extraction first F1-score results.

F-S	**1–1**	**1–0.5**	**1–0.25**	**1–0.1**	**0.5–1**	**0.5–0.5**	**0.5–0.25**	**0.5–0.1**	**0.25–1**	**0.25–0.5**	**0.25–0.25**	**0.25–0.1**	**0.1–1**	**0.1–0.5**	**0.1–0.25**	**0.1–0.1**
KNN	**0.999**	**0.999**	**0.999**	**0.999**	0.994	0.993	0.975	0.930	0.994	0.944	0.806	0.697	0.988	0.898	0.689	0.400
SVM	0.854	0.854	0.854	0.854	0.858	0.803	0.488	0.473	0.532	0.473	0.468	0.481	0.607	0.469	0.472	**1.000**
NB	0.255	0.255	0.255	0.255	0.231	0.238	0.247	0.259	0.232	0.235	0.243	0.184	0.408	0.429	0.382	**1.000**
LR	**0.899**	**0.899**	**0.899**	**0.899**	0.852	0.783	0.734	0.601	0.609	0.509	0.468	0.481	0.571	0.469	0.472	0.400
LDA	**0.987**	**0.987**	**0.987**	**0.987**	0.951	0.940	0.966	0.926	0.806	0.886	0.834	0.694	0.622	0.631	0.671	0.400
C4.5	**0.998**	**0.998**	**0.998**	**0.999**	0.996	0.994	0.996	0.985	0.996	0.986	0.993	**1.000**	0.985	0.955	0.689	0.400
XGB	**0.999**	**0.999**	**0.999**	**0.999**	**1.000**	0.994	0.998	0.990	**1.000**	**1.000**	**1.000**	**1.000**	0.995	0.964	0.817	**1.000**
RF	**0.999**	**0.999**	1.000	**0.999**	**1.000**	0.995	0.990	0.985	**1.000**	0.992	0.986	0.924	0.995	0.982	0.709	0.455
Ada	**0.997**	**0.997**	**0.997**	**0.997**	0.989	0.927	0.969	0.927	0.991	0.990	0.993	0.824	0.706	0.522	0.625	0.400

Table 7. DS3 sampling first F1-score results.

S-F	**1–1**	**1–0.5**	**1–0.25**	**1–0.1**	**0.5–1**	**0.5–0.5**	**0.5–0.25**	**0.5–0.1**	**0.25–1**	**0.25–0.5**	**0.25–0.25**	**0.25–0.1**	**0.1–1**	**0.1–0.5**	**0.1–0.25**	**0.1–0.1**
KNN	**1.000**	**1.000**	**1.000**	0.974	0.999	**1.000**	**1.000**	0.944	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.995	**1.000**
SVM	0.998	0.992	**1.000**	0.987	0.998	0.987	0.973	0.895	0.998	0.995	0.995	**1.000**	0.813	0.803	0.861	0.794
NB	0.912	0.923	0.922	0.874	0.852	0.866	0.853	0.807	0.880	0.883	0.860	0.856	0.738	0.757	0.751	0.763
LR	**1.000**	0.997	**1.000**	0.959	0.681	0.582	0.470	0.468	0.465	0.467	0.470	0.468	0.465	0.467	0.470	0.468
LDA	**1.000**	**1.000**	**1.000**	0.987	0.788	0.782	0.810	0.744	0.790	0.772	0.786	0.773	0.465	0.467	0.468	0.468
C4.5	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.949	0.973	**1.000**	0.998	0.965	0.987	**1.000**	0.997	0.994	0.881
XGB	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.949	0.973	**1.000**	**1.000**	0.965	0.959	**1.000**	0.997	0.994	0.916
RF	**1.000**	**1.000**	0.971	**1.000**	**1.000**	**1.000**	0.965	**1.000**	**1.000**	**1.000**	0.959	**1.000**	**1.000**	0.997	0.989	0.899
Ada	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.949	0.973	**1.000**	**1.000**	0.965	0.973	0.968	0.966	0.941	0.859

Table 8. DS3 feature extraction first F1-score results.

F-S	**1–1**	**1–0.5**	**1–0.25**	**1–0.1**	**0.5–1**	**0.5–0.5**	**0.5–0.25**	**0.5–0.1**	**0.25–1**	**0.25–0.5**	**0.25–0.25**	**0.25–0.1**	**0.1–1**	**0.1–0.5**	**0.1–0.25**	**0.1–0.1**
KNN	0.999	0.999	0.999	0.999	**1.000**	0.990	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.990	**1.000**	0.333
SVM	0.995	0.995	0.995	0.995	0.998	0.992	0.995	0.973	**1.000**	0.984	0.798	0.471	0.809	0.781	0.491	**1.000**
NB	0.918	0.918	0.918	0.918	0.844	0.836	0.814	0.791	0.885	0.843	0.983	**1.000**	0.764	0.708	0.565	**1.000**
LR	**1.000**	**1.000**	**1.000**	**1.000**	0.668	0.544	0.470	0.468	0.465	0.470	0.459	0.471	0.465	0.467	0.491	0.333
LDA	**1.000**	**1.000**	**1.000**	**1.000**	0.791	0.749	0.723	0.851	0.803	0.779	0.459	**1.000**	0.465	0.466	0.491	0.333
C4.5	**1.000**	**1.000**	**1.000**	**1.000**	0.996	**1.000**	**1.000**	0.931	0.998	0.973	0.964	**1.000**	**1.000**	0.942	**1.000**	**1.000**
XGB	**1.000**	**1.000**	**1.000**	**1.000**	0.996	**1.000**	**1.000**	0.944	0.998	0.978	0.982	0.818	0.999	0.980	0.491	0.000
RF	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.959	0.999	**1.000**	0.964	0.884	0.999	0.960	0.824	0.333
Ada	**1.000**	**1.000**	**1.000**	**1.000**	0.999	0.997	**1.000**	0.959	0.999	**1.000**	**1.000**	0.884	0.960	0.969	0.491	0.333

After training the different ML models with different portions from different datasets, the following notes should be considered from the tables:

- In all cases, data can be reduced by a half regarding both size and dimensionality, yet the ML models' performance remains the same.
- Applying the proper process to select part of the data to train the model can give the same results when all the data is used.
- The data-reduction techniques used throughout this work can enhance the required time to train and test the models.
- The data-reduction techniques used throughout this work can also produce less complicated models with the same efficiency.

4.2 Evaluating the Proposed Methods

The assessment step is presented and explored in this sub-section, where multiple data-reduction scenarios are being applied to three huge datasets. Feeding these datasets into the ML models requires very high computational resources. Additionally, time-demanding processes should be considered.

For the datasets DS4, DS5 and DS6, the reduction techniques were applied to investigate how the precision, recall, and F1-score were affected. The required training time to train all models is also measured. DS4 is a huge dataset in size and dimension; by extracting 0.001 of the size and a half of the features, all the classifiers still have a high performance of F-score, especially the KNN. Nevertheless, NB classifier behavior is sensitive to this level of reduction, as shown in Figure 4a. In other experiments, the NB was the worst when applied to a vast dataset with a small dimension, such as (DS4). The LDA performance with DS5 degrades, compared with its performance when DS4 was used, while other algorithms were robust to the reduction, as shown in Figure 4b. A moderate dimension and size dataset (DS6) was used to investigate the proposed approach; NB was the worst in comparison, even without reducing the data, while the other algorithms performed well. RF and XGB classifiers are the best for this data, as shown in Figure 4c. Every experiment held to reduce the size or the dimensionality of DS4, DS5, and DS6 datasets was repeated 10 times, and the ML model results were measured and averaged and then clarified in Figure 4. This step is necessary to examine the reduction techniques' effectiveness and confirm the derived conclusions.

The time required to train DS4 when all the data was used is 3750.40s. When the data was reduced to 0.05, 0.01, and 0.001, the required time to train all the models was reduced to 22.53s, 6.05s, and 3.74s, and when a half of the important features were selected from the 0.001 part of the data, the time was reduced to 3.2s. Yet, the ML classifiers still can detect anomaly behavior even when the dataset size is dramatically reduced (see Figure 4a).

As the experiment focuses on reducing the size of the datasets horizontally and vertically, this reduction is expected to affect the required time to train the ML models. DS4, Ds5, and DS6 are reduced in many ways to study how time is affected, and the time required to train all the mentioned ML models is reported. The time needed to train DS5 when all the data was used was 9779.81s, but when the size of the dataset was reduced to 0.01, the required time was 2.58s only, and the required time to train all the models was reduced to 1s when 0.001 of the dataset was used. Meanwhile, the ML models' performance measured in F-score are mostly close to 100% as shown in Figure 4b.

DS6 training time was 350.33s and reducing the size to the half made the training time become 118.4s. Reducing the features to the half made the training time become 221.41s, while combining both reductions made the time become 79.12s. ML models, such as KNN, SVM, XGB and RF, can still produce perfect results (see Figure 4c). Figure 3 lists the time required for training DS4, DS5 and DS6 and the required time when multiple reduction techniques were used. 0.5S means that a half the data was used, while 0.5F indicates the percentage of reduction applied to the features, where all the values in the figure are measured in seconds.

This reduction in computational time is due to the reduction in dataset rows and columns. The number of rows in each dataset is reduced *via* stratified sampling, while the number of columns is reduced via feature extraction carried out using the autoencoder model. Combining feature extraction with the size reduction process makes the dataset size shrink vertically and horizontally. The required processing time for ML models is a function of the number of rows and columns. Hence, if we can assume that the total computational time for these models is T = F (numOfRows,numOfCol,...), a function of the number of rows and the number of columns, then reducing the value of either numOfRows,numOfCol, or both will have a reducing impact on the required computational time.

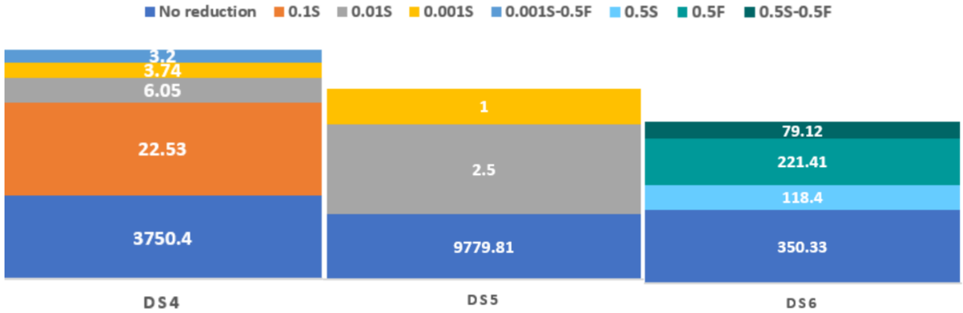
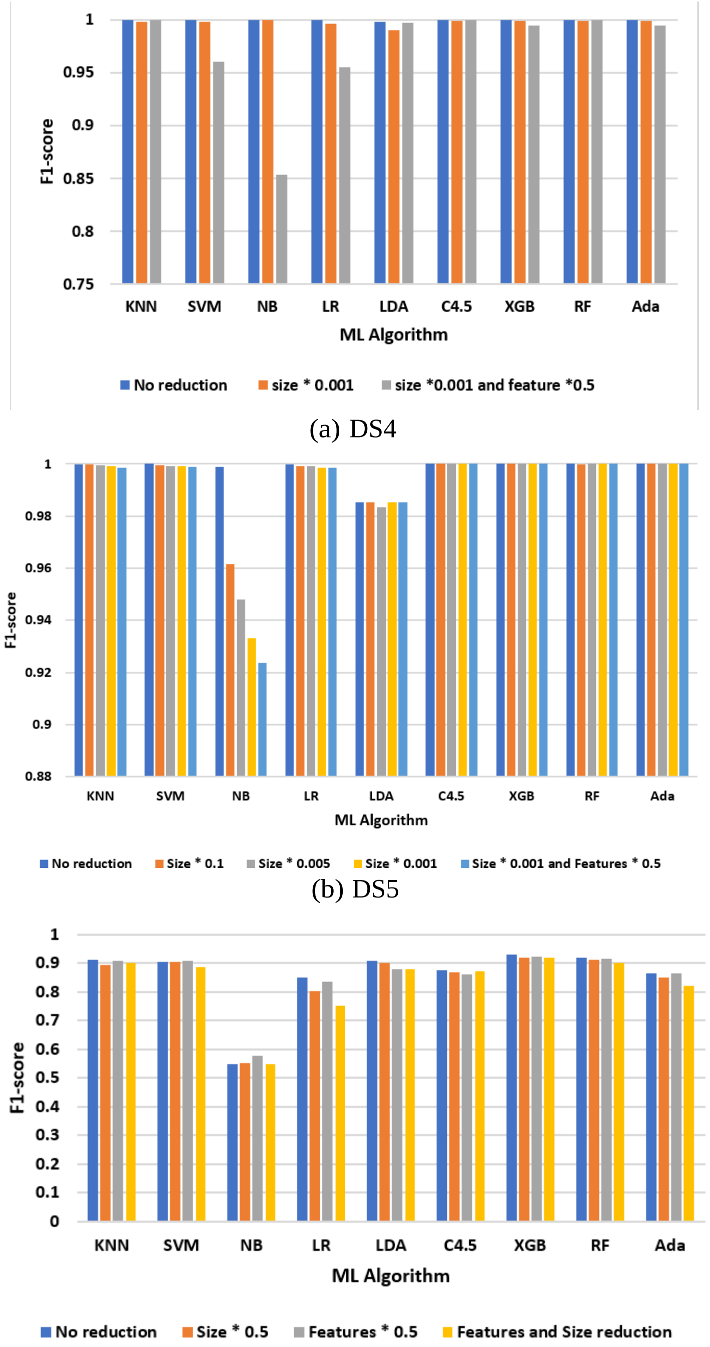


Figure 3. Time enhancement when large datasets were used.

4.3 Result Analysis and Recommendations

Simple reduction techniques, such as stratified sampling, can reduce the required time to build and train different ML models. Nevertheless, the performance of ML models is kept almost untouched. The huge amount of records stacked in different IDS datasets might be necessary, but not for IDS systems using ML models, such as those presented in this work. Some models can be less trusted, such as NB, and sometimes LR and LDA should be avoided, too. KNN, XGBoost, RF, and C-5 models are robust and can be trusted even when reduction methods are applied to the data.

When dealing with massive IDS datasets, reduction techniques, such as stratified sampling, and dimensionality-reduction techniques, such as autoencoders, are highly recommended to be used with the data to make it more usable. If the number of records in the dataset is small; i.e., < 20000, using the autoencoder first is highly recommended. For example, for a dataset similar to DS5, which is used here, reducing the data size first is recommended, since training the autoencoder and getting the results from the encoder will take a very long time.



(a) DS4

(b) DS5

(c) DS6

Figure 4. ML model performance when large datasets were used.

If the dataset is already small, but has a large number of features, like DS2, which has 115 features, extracting the important features first is preferred since the autoencoder accuracy will be better with more data tuples to train it. Extracting the most important features from the dataset might enhance the performance of some ML models, like NB and SVM, with the DS2 results above.

The amount of the reduction to the data; i.e., how much data should be used to train the model, is a subject of experience and the logic of trial and error. The reduction tools are available and should be used with wisdom. For example, DS6 was reduced to the half to make the training time more efficient. While DS5 was reduced to one-tenth, considering that DS5 is almost five times the size of DS6, DS6 is a very unbalanced dataset.

The answer to the first research question is that huge IDS datasets are not necessarily needed, because the study results show that the ML models can produce sufficient results in many reduction cases, especially when certain ML models are used, such as Random Forest and KNN. The answer to the second question is that size reduction, feature reduction, and combining both reduction techniques can be used to reduce the size of the datasets while keeping the ML-model results sufficient. Although the proposed method does not introduce a new detection algorithm, it handles a crucial operational challenge in IDSs: the need for scalable and efficient model training on large, high-dimensional datasets. The framework shows that significant computational gains can be achieved through structured dataset reduction, allowing faster deployment and real-time responsiveness without degrading detection performance. This contribution supports more practical and cost-effective implementation of IDSs in environments where computational resources and latency are constrained.

4.4 Scalability Considerations and Real-world Deployment

The suggested dataset-reduction framework is developed to be modular and scalable, allowing it to adapt to diverse deployment scenarios. In cloud or cluster-based environments, the autoencoder training process can be parallelized and accelerated using GPU hardware, making it feasible to extract feature importance from even larger IDS datasets, such as real-time streaming logs or full network captures. The feature-selection step, once learned, can be reused across multiple time windows or data batches with minimal retraining.

Our sampling-first (SF) pipeline offers a practical compromise for edge-computing environments whose computational resources of which are limited. Applying stratified sampling before dimensionality reduction minimizes resource usage and preserves class distribution. Additionally, autoencoder-based feature selection lowers memory requirements and latency for deployed ML models. Thus, the discussed reduction methods are sufficient for academic evaluation and functional for real-world IDS applications where scalability, model-retraining efficiency, and system throughput are key considerations.

4.5 Comparison with Other Works

Table 9 demonstrates a comparison between our work and recent works with similar contributions.

The comparison of our work with recent contributions emphasizes key dissimilarities in dataset selection, feature-reduction methodologies, machine-learning models, and overall effectiveness in cyber-threat detection. One of the main strengths of our technique is the use of multiple datasets, including Kitsune- ARP, SNMP-MIB, CSE-CIC-IDS2018, BoTIoT, UNR-IDD, and Credit Card Fraud, which provides a more comprehensive evaluation of cyber threats. This contrasts studies, such as Behiry and Aly (2024), which focus on certain datasets, like NSL-KDD, UNSW-NB15 and CICIDS2017. Using various datasets in our study improves the generalizability of the results, although it presents sophistication in formalizing feature-selection techniques.

The data-reduction strategy used in our study combines autoencoders with stratified sampling, setting it apart from the principal component analysis (PCA) and singular value decomposition (SVD) approaches used in other studies. Autoencoders allow for non-linear feature extraction, which provides more robust dimensionality reduction, unlike traditional methods that assume linear relationships between variables. Compared to the Coot Optimization Algorithm (COA) used by Vallabhaneni et al. [42], our approach fulfills similar feature-reduction effectiveness, but significantly reduces the computational cost. Combining autoencoders with stratified sampling ensures that essential features are retained while reducing redundancy, making our method accurate and efficient. Another distinguishing factor is using stratified sampling instead of synthetic oversampling methods, like SMOTE, which Behiry & Aly [40] utilized. While SMOTE artificially generates new samples, stratified sampling preserves the natural distribution of data, preserving class balance without introducing synthetic artifacts. This approach ensures that minority-class instances, crucial for fraud and intrusion detection, remain well-represented while reducing data size. By leveraging stratified sampling, our method enhances dataset efficiency without sacrificing classification performance.

The selection of machine-learning models further distinguishes our work from previous studies. Our evaluation encloses a diverse set of algorithms, including K-Nearest Neighbors (KNN), Support Vector Machines (SVMs), Naïve Bayes, Linear Discriminant Analysis (LDA), C5, XGBoost, Random Forest, and ADA, offering a comprehensive analysis of classification performance. In contrast, [40] and [42] mostly depend on deep-learning models, such as deep forward neural networks (DFNNs) and modified feedforward neural networks (FFNNs). While deep learning models perform well on high-dimensional data, they demand much more computational resources and training time. Our approach balances accuracy and computational efficiency by combining classical machine learning and ensemble methods, making it more suitable for real-time applications.

Table 9. Comparison of our work and recent works with similar contributions.

Criteria	**Our work**	**Behiry & Aly [40]**	**Hossain et al. [41]**	**Vallabhaneni et al. [42]**
Dataset Used	Kitsune-ARP, SNMP-MIB, CSE- CIC-IDS2018, BoTIoT, UNR-IDD, Credit Card Fraud	NSL-KDD, UNSW-NB15, CICIDS2017	Not specified (DDoS-related)	BotNet dataset
Dataset Size	Multiple large-scale datasets (ranging from 2,620 to 2,426,574 records)	175,466 samples (CICIDS2017)	Not provided	1,803,333 domain names
Feature-reduction Method	Autoencoders + Stratified Sampling	Singular Value De-composition (SVD) + PCA + KMC-IG	Hybrid Feature Selection	Coot Optimization Algorithm (COA)
Sampling Method	Stratified Sampling	SMOTE + ENN	Not specified	Not specified
Machine-learning Model	KNN, SVM, Naive Bayes, Linear Regression, LDA, C5, XGBoost, Random Forest, ADA	Deep Forward Neural Network (DFNN) + K-means Clustering (KMC)	Ensemble-based classifier	Modified Feed-forward Neural Network (FFNN)
Performance Metrics	Accuracy up to 99% (varies by dataset), F1-score analysis for different reduction strategies	Accuracy: 99.7%, F1-score: 98.8% (NSL-KDD)	Not specified	Accuracy: 97.56%, Precision: 96.76%
Computational Efficiency	Training time reduced significantly by applying size and feature reduction techniques	High efficiency due to hybrid feature selection	Not specified	Improved by using COA for feature selection
Real-time Applicability	Yes, reduces dataset size while maintaining accuracy for efficient IDS deployment	Yes, suitable for real-time WSN intrusion detection	Yes, aimed at robust DDoS mitigation	Yes, designed for Cybersecurity-attack prediction
Novelty	Combination of autoencoder-based feature selection and stratified sampling for dataset reduction	Hybrid feature reduction (SVD+PCA + KMC-IG) + deep learning	Hybrid feature selection + ensemble classification	COA-based feature selection with adaptive weight FFNN
Limitations	Some models (e.g., Naive Bayes) perform poorly on highly reduced datasets	Requires large labeled datasets	Requires further evaluation in real-world scenarios	Computational complexity in feature selection and training

The performance metrics indicate that our method achieves an accuracy of up to 99% across multiple datasets, comparable to the 99.7% accuracy reported in [40]. However, the key advantage of our approach lies in its computational efficiency. By reducing the dataset size while maintaining classification performance, our method enables faster training times, making it highly scalable for real-time intrusion-detection systems. In contrast, with a computationally expensive feature selection process [42], it achieved a slightly lower accuracy of 97.56%. Using autoencoder-based feature-selection in our work ensures optimal feature retention with minimal processing overhead, achieving a balance between performance and efficiency.

Real-time applicability is a critical aspect of intrusion-detection systems. Our study prioritizes this using efficient data-reduction techniques and lightweight machine-learning models. While [40] and [41] argue real-time relevancy, their studies lack detailed evaluations of computational efficiency. Our work explicitly shows that dataset-size reduction leads to significantly lower training times, confirming that the model remains deployable in practical cyber-security environments. The novelty of our work lies in the hybrid combination of autoencoder-based feature selection with stratified sampling, which optimizes both dataset size and model performance. Unlike previous studies that rely only on statistical reduction techniques or heuristic optimization, our approach integrates deep feature extraction and data-selection strategies. This hybrid approach results in an efficient intrusion-detection system capable of handling large-scale datasets while maintaining high detection accuracy.

Despite the benefits, there are areas for additional improvement. Some models, such as Naïve Bayes, exhibit performance degradation when involved with highly-reduced datasets, suggesting that feature-selection techniques could be further purified to improve compatibility with a more expansive range of classifiers. Additionally, estimating the trade-off between dataset reduction and accuracy loss under extreme conditions would provide further insights into the scalability of our approach. Expanding the study to real-world cyber-security attack scenarios would further validate its functional applicability.

5. Conclusion and Future Work

This study presents and tests two methods to reduce the amount of data used to train and test IDSs. The first method depends on reducing the size of the datasets with very large tuples, followed by feature selection to improve the ML model's performance. The second method, which is more practical with relatively small datasets, aimed to select the most important features first and then reduce the number of used tuples; this method guarantees the selection of better features and also improves the ML-model performance. This emphasizes the redundancy happening in some datasets related to security attacks in IoT datasets, especially simulated datasets.

This study shows that careful dataset size and feature-dimensionality reduction can lower computational costs while maintaining equivalent intrusion-detection performance. Specifically, using only 25% of the original data or feature set resulted in a less than 2% reduction in F-score for most models and datasets. Even with a large reduction to 10%, the average F1-score declined by only 4%–6%, with ensemble models, such as XGBoost and Random Forest, showing more resilience compared to other simple classifiers, like KNN. The reduction framework is computationally efficient and robust across various IDS scenarios. Meanwhile, data-reduction processes should be taken with caution, because random or extreme data reduction might cause the models to produce unacceptable results, as seen in many scenarios throughout this study.

In the future, we plan to repeat the experiment with multi-class labeled datasets and check how the proposed reduction techniques would affect the ML models. We also wish to investigate and compare other multiple reduction techniques. Our plan also includes applying the reduction techniques to different convolutional neural-network architectures and employing XAI tools to explore the reasons behind feature-ranking results. It is also necessary to have methods to evaluate the redundancy level in a dataset to estimate the possible efficient reduction percentages that can be applied to the data.

References

[1] M. B. Younes and A. Boukerche, "A Performance Evaluation of a Context-aware Path Recommendation Protocol for Vehicular Ad-hoc Networks," Proc. of the 2013 IEEE Global Communications Conf. (GLOBE- COM), IEEE, pp. 516–521, Atlanta, USA, 2013.

[2] M. B. Younes, G. R. Alonso and A. Boukerche, "A Distributed Infrastructure-based Congestion Avoidance Protocol for Vehicular Ad Hoc Networks," Proc. of the 2012 IEEE Global Communications Conf. (GLOBECOM), pp. 73–78, Anaheim, USA, 2012.

[3] J. Al-Sawwa, M. Almseidin, M. Alkasassbeh, K. Alemerien and R. Younisse, "Spark-based Multi-verse Optimizer as Wrapper Features Selection Algorithm for Phishing Attack Challenge," Cluster Computing, vol. 27, no. 5, pp. 5799–5814, 2024.

[4] L. A. C. Ahakonye et al., "SCADA Intrusion Detection Scheme Exploiting the Fusion of Modified Decision Tree and Chi-square Feature Selection," Internet of Things, vol. 21, p. 100676, 2023.

[5] Y. Han, Y. Zhang and J. Wang, "Semantic-driven Dimension Reduction for Wireless Internet of Things," Internet of Things, vol. 25, p. 101138, 2024.

[6] F. Ali et al., "An Intelligent Healthcare Monitoring Framework Using Wearable Sensors and Social Networking Data," Future Generation Computer Systems, vol. 114, pp. 23–43, 2021.

[7] A. Shiravani, M. H. Sadreddini and H. N. Nahook, "Network Intrusion Detection Using Data Dimensions Reduction Techniques," Journal of Big Data, vol. 10, no. 1, p. 27, 2023.

[8] R. Younisse and M. AlKasassbeh, "SGID: A Semi-synthetic Dataset for Injection Attacks in Smart Grid Systems," Proc. of the 2024 15th IEEE Int. Conf. on Information and Communication Systems (ICICS), pp. 1–4, Irbid, Jordan, 2024.

[9] A. Glielmo, B. E. Husic, A. Rodriguez, C. Clementi, F. Noé and A. Laio, "Unsupervised Learning Methods for Molecular Simulation Data," Chemical Reviews, vol. 121, no. 16, pp. 9722–9758, 2021.

[10] B. M. S. Hasan and A. M. Abdulazeez, "A Review of Principal Component Analysis Algorithm for Dimensionality Reduction," Journal of Soft Computing and Data Mining, vol. 2, no. 1, pp. 20–30, 2021.

[11] S. Li, N. Marsaglia et al., "Data Reduction Techniques for Simulation, Visualization and Data Analysis," Computer Graphics Forum, vol. 37, pp. 422–447, Wiley Online Library, 2018.

[12] M. Dumelle, T. Kincaid, A. R. Olsen and M. Weber, "Spsurvey: Spatial sampling design and analysis in R," Journal of Statistical Software, vol. 105, no. 3, pp. 1–29, 2023.

[13] G. Sharma, "Pros and Cons of Different Sampling Techniques," International Journal of Applied Research, vol. 3, no. 7, pp. 749–752, 2017.

[14] Z. Ashi, L. Aburashed, M. Al-Qudah and A. Qusef, "Network Intrusion Detection Systems Using Supervised Machine Learning Classification and Dimensionality Reduction Techniques: A Systematic Review," Jordanian J. of Computers and Inform. Technol. (JJCIT), vol. 7, no. 4, pp. 373 – 390, 2021.

[15] N. Saran and N. Kesswani, "Intrusion Detection System for Internet of Medical Things Using GRU with Attention Mechanism-based Hybrid Deep Learning," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 11, no. 2, pp. 136-150, 2015.

[16] Y. Xiao, C. Xing, T. Zhang and Z. Zhao, "An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks," IEEE Access, vol. 7, pp. 42210–42219, 2019.

[17] F. Salo, A. B. Nassif and A. Essex, "Dimensionality Reduction with IG-PCA and Ensemble Classifier for Network Intrusion Detection," Computer Networks, vol. 148, pp. 164–175, 2019.

[18] R. Abdulhammed et al., "Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection," Electronics, vol. 8, no. 3, p. 322, 2019.

[19] S. Ryu et al., "Convolutional Autoencoder Based Feature Extraction and Clustering for Customer Load Analysis," IEEE Trans. on Power Systems, vol. 35, no. 2, pp. 1048–1060, 2019.

[20] G. T. Reddy et al., "Analysis of Dimensionality Reduction Techniques on Big Data," IEEE Access, vol. 8, pp. 54776–54788, 2020.

[21] K. K. Pandey and D. Shukla, "Stratified Linear Systematic Sampling Based Clustering Approach for Detection of Financial Risk Group by Mining of Big Data," Int. J. of System Assurance Engineering and Management, vol. 13, pp. 1239–1253, 2021.

[22] K. Zhang et al., "History Matching of Naturally Fractured Reservoirs Using a Deep Sparse Autoencoder," SPE Journal, vol. 26, no. 4, pp. 1700– 1721, 2021.

[23] B. Manjunatha et al., "A Network Intrusion Detection Framework on Sparse Deep Denoising Autoencoder for Dimensionality Reduction," Soft Computing, vol. 28, no. 5, pp. 4503–4517, 2024.

[24] F. Nabi and X. Zhou, "Enhancing Intrusion Detection Systems through Dimensionality Reduction: A Comparative Study of Machine Learning Techniques for Cyber Security," Cyber Security and Applications, vol. 2, p. 100033, 2024.

[25] K. K. Pandey and D. Shukla, "Stratified Sampling-based Data Reduction and Categorization Model for Big Data Mining," Proc. of Communication and Intelligent Systems (ICCIS 2019), pp. 107–122, Springer, 2020.

[26] X. Zhao, J. Liang and C. Dang, "A Stratified Sampling Based Clustering Algorithm for Large-scale Data," Knowledge-based Systems, vol. 163, pp. 416–428, 2019.

[27] Y. Yang, J. Cai, H. Yang, Y. Li and X. Zhao, "ISBFK-means: A New Clustering Algorithm Based on Influence Space," Expert Systems with Applications, vol. 201, p. 117018, 2022.

[28] L. Cao and H. Shen, "CSS: Handling Imbalanced Data by Improved Clustering with Stratified Sampling," Concurrency and Computation: Practice and Experience, vol. 34, no. 2, p. e6071, 2022.

[29] A. Zoubir and B. Missaoui, "Graph Neural Networks with Scattering Transform for Network Anomaly Detection," Engineering Applications of Artificial Intelligence, vol. 150, p. 110546, 2025.

[30] A. Zoubir and B. Missaoui, "GeoScatt-GNN: A Geometric Scattering Transform-based Graph Neural Network Model for Ames Mutagenicity Prediction," arXiv preprint, arXiv: 2411.15331, 2024.

[31] M. Alqarqaz, M. Bani Younes and R. Qaddoura, "An Object Classification Approach for Autonomous Vehicles Using Machine Learning Techniques," World Electric Vehicle J., vol. 14, no. 2, p. 41, 2023.

[32] Y. Mirsky, T. Doitshman, Y. Elovici and A. Shabtai, "Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection," arXiv preprint, arXiv: 1802.09089, 2018.

[33] M. Al-Kasassbeh et al., "Towards Generating Realistic SNMP-MIB Dataset for Network Anomaly Detection," Int. J. of Computer Science and Information Security, vol. 14, no. 9, p. 1162, 2016.

[34] UNB, "CSE-CIC-IDS2018 on AWS," [Online], Available: http://www.unb.ca/cic/datasets/ids-2018.html, Accessed on Apr. 25, 2023, 2018.

[35] N. Koroniotis et al., "Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset," Future Generation Computer Systems, vol. 100, pp. 779–796, 2019.

[36] T. Das et al., "UNR-IDD: Intrusion Detection Dataset Using Network Port Statistics," Proc. of the 2023 IEEE 20th Consumer Comm. & Networking Conf. (CCNC), pp. 497–500, Las Vegas, USA, 2023.

[37] Kaggle, "Credit Card Fraud Detection," [Online], Available: www.kaggle.com/datasets/mlg-ulb/creditcardfraud, Accessed: June 1, 2023.

[38] Y. N. Kunang et al., "Automatic Features Extraction Using Autoencoder in Intrusion Detection System," Proc. of the 2018 Int. Conf. on Electrical Engineering and Computer Science (ICECOS), pp. 219–224, Pangkal, Indonesia, 2018.

[39] Z. Salah et al., "Optimizing Intrusion Detection in 5G Networks Using Dimensionality Reduction Techniques," Int. J. of Electrical & Computer Engineering, vol. 14, no. 5, pp. 2088-8708, 2024.

[40] M. H. Behiry and M. Aly, "Cyberattack Detection in Wireless Sensor Networks Using a Hybrid Feature Reduction Technique with AI and Machine Learning Methods," J. of Big Data, vol. 11, no. 1, 2024.

[41] M. A. Hossain and M. S. Islam, "Enhancing DDoS Attack Detection with Hybrid Feature Selection and Ensemble-based Classifier: A Promising Solution for Robust Cybersecurity," Measurement: Sensors, vol. 32, p. 101037, 2024.

[42] R. Vallabhaneni et al., "Feature Selection Using COA with Modified Feedforward Neural Network for Prediction of Attacks in Cyber-security," Proc. of ICDCOT, pp. 1–6, DOI: 10.1109/ICDCOT61034, 2024.10516044, 2024.

ملخص البحث:

يُعدّ كشف الاختراقات في بيئات إنترنت الأشياء أمراً أساسياً لضمان أمان شبكات الحاسوب. وتستخدم نماذج التّعلُّم الآلي على نطاقٍ واسعٍ لتحسين أنظمةٍ فعّالة للكشف عن الاختراقات. ومع التّزايد السّريع في تعقيد وحجم البيًانات في أنظمة الكشف عن الاختراقات، فإنّ تحليل مجموعات بياناتٍ ضخمة باستخدام نماذج التّعلُّم الآلي باتَ ينطوي على المزيد من التّحدّيات والمتطلّبات المتعلّقة بمصادر الحوسبة. وتأتي مجموعات البيانات المتعلّقة ببيئات إنترنت الأشياء بأحجام ضخمة. وتبحث هذه الدّراسة في أثر تقنيات تقليل البيانات في مجموعات البيانات في فاعلية وأداء أنظمة الكشف عن الاختراقات.

نقترح في هذه الورقة إطار عملٍ ثنائي المراحل يدمج بين تقليل السِّمات وتقليل الأبعاد والحجم في مجموعات البيانات المتعلّقة ببيئات إنترنت الأشياء، وذلك على ستّ مجموعات بياناتٍ مُتاحة للعموم. وتمّ تقييم أداء عددٍ من نماذج التّعلُّم الآلي على مجموعات البيانات المدروسة. وتوضّح النّتائج الّتي حصلنا عليها أنّ تقليل البيانات من شأنه أن يؤدّي إلى تقليل زمن التّدريب بما يصل إلى 99% مع فقدانٍ هامشي في مؤشّرات الأداء لا يتجاوز 1%، وقد تبين أنّ التّقليل الزّائد للبيانات قد يؤثر سلباً في دقّة الكشف. وتسلط الدّراسة الضّوء على فوائد تقليل البيانات في مجموعات بيانات إنترنت الأشياء، وتدعم نّتائج الدّراسة جدوى تطبيقات الكشف الفعّال عن الاختراقات لبيئات العالم الحقيقي، وخاصّةً في الأوضاع الّتي تتّسم بمحدودية الموارد أو الّتي تتعلّق بالزّمن الحقيقي.

ENHANCING FEW-SHOT LEARNING PERFORMANCE WITH BOOSTING ON TRANSFORMERS: EXPERIMENTS ON SENTIMENT ANALYSIS TASKS

Lenh Phan Cong Pham and Huan Thai Phong

(Received: 23-Dec.-2024, Revised: 11-Jun.-2025, Accepted: 4-Jul.-2025)

ABSTRACT

This study addresses challenges in sentiment analysis for low-resource educational contexts by proposing a framework that integrates Few-Shot Learning (FSL) with Transformer-based ensemble models and boosting techniques. Sentiment analysis of student feedback is crucial for improving teaching quality, yet traditional methods struggle with data scarcity and computational inefficiency. The proposed framework leverages self-attention mechanisms in Transformers and combines models through Gradient Boosting to enhance performance and generalization with minimal labeled data. Evaluated on the UIT-VSFC dataset, comprising Vietnamese student feedback, the framework achieved superior F1-scores in sentiment and topic-classification tasks, outperforming individual models. Results demonstrate the potential of the proposed framework for extracting actionable insights to enhance educational experiences. Despite its effectiveness, the approach faces limitations, such as reliance on pre-trained models and computational complexity. Future work could optimize lightweight models and explore applications in other domains, like healthcare and finance.

KEYWORDS

Few-shot learning, Boosting, Transformer models, Sentiment analysis.

1. INTRODUCTION

In natural language processing (NLP), sentiment analysis, also referred to as opinion mining, is a method used for evaluating the emotional state of a given text [1]. This technique has become a valuable tool for extracting user opinions from product and service reviews, providing businesses with actionable insights to improve their offerings [2]. Student feedback is essential for assessing learning-management systems, instructional strategies and course material in the educational setting [3]. To facilitate efficient analysis, these feedback responses, which are frequently in the form of text, need to be pre-processed using NLP techniques as feature extraction and selection [4].

The initial step in sentiment analysis involves labeling text with emotional categories, like positive, negative, or neutral, reflecting students' feelings about the courses and services provided [5]. However, the manual annotation process can be time-consuming and require substantial resources, as well as an understanding of educational content. This challenge has been addressed through automated methods powered by AI and machine learning [6]. With its ability to process and analyze vast amounts of student input, artificial intelligence (AI) greatly improves the precision and effectiveness of sentiment categorization [7]. Even when feedback is unlabeled, machine learning, deep learning and transformer models are very good at using attention processes to identify students' feelings [8].

In the age of online and blended learning, where emotional cues may be harder to discern, leveraging sentiment-analysis tools becomes essential for extracting meaningful insights from textual data [9]. Furthermore, various machine-learning algorithms, such as Naive Bayes, Support Vector Machines (SVMs) and lexicon-based methods, have been used to analyze sentiments in student feedback, demonstrating their effectiveness in processing and interpreting these responses [10]–[12]. With these advancements, sentiment analysis not only contributes to enhancing teaching quality, but also provides valuable insights into the experiences and perspectives of students in the educational process.

Traditional supervised-learning approaches have been extensively applied in sentiment analysis, yet they are constrained by inherent limitations. One major challenge arises in scenarios with limited labeled training data, where traditional machine-learning models often suffer from overfitting, rendering them unable to generalize effectively to unseen data [13]. This limitation is particularly problematic in

P. C. P. Lenh (Corresponding Author) and T. P. Huan are with Faculty of Artificial Intelligence, FPT University, Can Tho, Vietnam. Emails: Lenhppcce180059@fpt.edu.vn and huantpce180685@fpt.edu.vn

sentiment analysis, where diverse and complex text patterns demand robust generalization. Moreover, while humans can intuitively generalize concepts with minimal exposure or partial information, machine-learning models struggle to replicate this ability [14]. As a result, traditional methods falter in low-data settings, leaving critical gaps in performance and scalability.

Previously, sentiment analysis has depended on supervised techniques that handle issues, like lexical variety and long-distance interdependence, present in textual data. To capture these relationships, sequence models such as RNNs and LSTM networks, have been frequently employed. While these models can encode complex relationships within text, their serialized processing makes them computationally inefficient and limits their scalability, especially in real-world applications. Through the application of parallelized processing, Transformer models, on the other hand, have transformed sentiment analysis and greatly increased computational effectiveness while maintaining the capacity to identify long-distance relationships. Their self-attention mechanisms allow for a more comprehensive understanding of text structure and semantics, making them well-suited for sentiment analysis. However, these models often require large amounts of labeled data to perform effectively, which poses a challenge in resource-constrained environments.

To address these challenges of data scarcity and computational inefficiency, Few-Shot Learning (FSL) has emerged as a promising solution. FSL enables models to generalize effectively from only a few labeled examples, mimicking human-like learning. However, traditional supervised methods still face limitations in terms of overfitting and dependency on large datasets. To overcome these issues, integrating ensemble learning with Transformer architecture and FSL offers a novel approach. By combining multiple Transformer models trained with few-shot data, ensemble learning can improve generalization and robustness, mitigating the risks of overfitting. The hybrid approach leverages the computational efficiency of Transformers, the contextual power of self-attention mechanisms and the scalability of FSL, offering a more effective and resource-efficient framework for sentiment analysis in real-world applications.

While traditional sentiment-analysis approaches have demonstrated strong performance on large-scale datasets, their applicability is limited in low-resource educational environments, where collecting and annotating large volumes of labeled data are often impractical due to time, budgetary and expertise constraints. Deep learning and transformer-based techniques have achieved promising results in educational contexts, such as analyzing course feedback or evaluating learning-management systems [60–62]. However, these approaches are highly dependent on the availability of comprehensively labeled datasets, which poses a significant barrier in many real-world educational scenarios, particularly in under-resourced institutions or less-documented languages. Moreover, existing research has paid limited attention to the use of boosting strategies for ensembling Transformer-based models in educational sentiment analysis. Most prior studies, such as [63] and [64], have focused on combining traditional deep-learning models and basic machine-learning techniques rather than leveraging the potential diversity and complementary strengths of multiple Transformer architectures. This reflects a research gap in exploring ensemble-learning techniques, particularly boosting, in conjunction with modern pre-trained language models for low-resource educational contexts.

To address the critical challenge of data scarcity in analyzing student feedback, particularly for under-resourced languages, like Vietnamese, within educational settings, this paper proposes a novel approach. We investigate the synergistic integration of Few-Shot Learning (FSL) with boosting-enhanced Transformer-based ensemble models. While FSL addresses the limited data and Transformers offer powerful text representation, the strategic application of boosting techniques over an ensemble of such FSL-trained Transformers is a relatively unexplored configuration aimed at maximizing performance and robustness specifically for this low-resource niche.

The purpose of the research includes:

- To rigorously assess the viability and effectiveness of integrating FSL with boosted Transformer ensembles for sentiment analysis specifically on scarce Vietnamese student-feedback data, thereby demonstrating a practical solution for low-resource educational contexts.
- To explore and apply boosting methods to combine model predictions and evaluate the effectiveness of ensemble techniques in improving accuracy and prediction performance for sentiment and topic

classification tasks.

- To develop and provide a high-performance model for student sentiment analysis, particularly suited for small datasets, to support research and enhance sentiment-analysis methods in the educational context.
- To evaluate the proposed model on an additional sentiment-analysis dataset from a different domain to ensure the model's robustness and generalizability across various contexts, thereby supporting its applicability in broader sentiment-analysis tasks beyond the educational setting.

2. RELATED WORK

2.1 Contrastive Learning in Sentiment Analysis

The primary objective of contrastive learning (CL), a self-supervised machine-learning technique, is to develop representations through the comparison of various data samples. More specifically, CL learns to push negative pairings farther apart and bring positive pairs closer together in the representation space. In order to decrease dimensionality and enhance classification and recognition performance, CL was presented as a technique that involves learning an invariant mapping [15]. With a momentum encoder that continuously updates negative samples, it was shown how important the quantity of negative samples is to improving representation learning [16]. Constructing effective positive pairs was highlighted as a critical factor in learning high-quality representations in CL [17].

Contrastive learning has shown itself to be an effective technique in sentiment-analysis applications. Supervised CL has been directly used in a number of research studies [18]-[20] to align sentiment representations with corresponding sentiment labels in order to develop fine-grained sentiment representations. In order to promote more efficient sentiment-analysis learning, supervised CL creates positive pairings based on labels, where samples with the same label are regarded as positive pairs and samples with different labels are regarded as negative pairs [21]. Additionally, to improve the accuracy and resilience of sentiment-analysis models, multi-aspect samples for CL were created using an in-domain generator and a cross-channel data-augmentation technique [22]. In order to enhance sentiment-analysis performance, cross-lingual contrastive learning also employed token-level and sentence-level data-augmentation techniques in addition to sentiment identifying [23].

2.2 Boosting

Boosting is a method of machine learning that combines weak learners in an ensemble style to turn them into a strong classifier. Its main goal is to minimize bias, which aids in the improvement of highly biased models. Combining the outcomes of each iteration using a weighted vote for classification or a weighted sum for regression yields the final output of boosting [24].

2.2.1 AdaBoost

Adaptive boosting is a powerful boosting algorithm introduced by [25], designed to combine weak learners, typically decision stumps (decision trees with a single split), into a strong classifier. It is widely regarded as one of the most robust machine-learning algorithms, with AdaBoost.M1 being a notable implementation for binary-classification tasks [26]. AdaBoost requires little hyper-parameter tuning and is simple to deploy [27]. To create the strong classifier, the several base learners are added one after the other and weighted [28]. The learning process involves iteratively training base classifiers, updating sample weights based on their classification performance and prioritizing misclassified samples in subsequent iterations. Initially, all samples are assigned equal weights:

$$\mathrm{D}_1(i) = \frac{1}{m}, \qquad i = 1,2,\dots,m.$$

The weights are then updated after each iteration using the formula:

$$\mathrm{D}_{t+1}(i) = \frac{\mathrm{D}_t(i)}{Z_t}\exp(-\alpha_t \mathrm{y}_i \mathrm{h}_t(\mathrm{x}_i).$$

Here, the importance of each base classifier is quantified as:

$$\alpha_t = \frac{1}{2}\ln(\frac{1-\epsilon_t}{\epsilon_t})$$

where ϵ_t is the error rate of the base classifier. After $\mathrm{T}_{iterations}$, the final strong classifier is computed as:

$$\mathrm{H(x)} = \mathrm{sign}(\sum_{t=1}^{T} \alpha_t \mathrm{h}_t(\mathrm{x})$$

This approach ensures a weighted combination of base classifiers to optimize performance. AdaBoost's adaptability and sequential focus on hard-to-classify samples make it highly effective for diverse applications.

2.2.2 Gradient Boosting

A popular machine-learning technique, called gradient boosting, iteratively combines weaker base learners, usually decision trees, to create a powerful prediction model. Because it uses decision trees as essential building elements, it is frequently referred to as Gradient Boosted Decision Tree (GBDT). [29] was the first reference to describe the concept, demonstrating that boosting can be seen as an optimization problem that aims to achieve a certain loss function.

An advanced version of this approach was later developed [30], focusing on sequentially training models to construct a robust ensemble classifier. Unlike other boosting methods, the key idea in Gradient Boosting is to design base learners that align with the negative gradient of the loss function for the overall ensemble [31].

For a given training dataset $S = \{(\mathrm{x}_i, y_i\}_{i=1}^{N}$, the goal of Gradient Boosting is to approximate a function $\mathrm{F}^*(\mathrm{x})$ that predicts the response variable y based on input features x, by minimizing a pre-defined loss function L(y, F(x)). This approximation is achieved iteratively by creating an additive model expressed as:

$$\mathrm{F_m(x)} = \mathrm{F_{m-1}(x)} + \rho_\mathrm{m}\mathrm{h_m(x)}$$

Here:

- $\mathrm{F_m(x)}$: The prediction at iteration m.
- $\mathrm{F_{m-1}(x)}$: The prediction from the previous iteration.
- $\rho\mathrm{m}$: The weight of the mth learner.
- $\mathrm{h_m(x)}$: The mth base learner, typically a decision tree.

The initial model, $\mathrm{F_0(x)}$, is determined by minimizing the loss across all samples:

$$F_0(x) = \arg\min_{\alpha} \sum_{i=1}^{N} L(y_i, \alpha)$$

In subsequent iterations, each new learner $h_m(x)$ is trained to minimize the error of the current model:

$$h_m(x) = \arg\min_{h} \sum_{i=1}^{N} L\big(y_i, F_{m-1}(x_i) + \rho h(x_i)\big)$$

A critical aspect of this process involves computing pseudo residuals, which represent the gradients of the loss function with respect to the model's predictions. These are calculated as:

$$r_{mi} = \left[\frac{\partial L\big(y_i, F(x)\big)}{\partial F(x)}\right] \quad F(x) = F_{m-1}(x)$$

The optimal weight ρ_m is subsequently obtained through a line-search procedure.

To mitigate overfitting, the algorithm applies shrinkage, scaling the contribution of each step by a learning rate y (commonly set to 0.1):

$$F_m(x) = F_{m-1}(x) + v\rho_m h_m(x)$$

Gradient boosting stands out for its ability to uncover intricate patterns in data by systematically addressing errors in previous iterations. However, it is susceptible to overfitting, especially with noisy datasets, if regularization techniques are not adequately employed [31 - 32]. Despite this, it remains a powerful choice, particularly for small datasets [33].

2.2.3 XGBoost

Extreme Gradient Boosting, or XGBoost, is a decision tree-based ensemble technique that uses the gradient-boosting framework and is incredibly effective and scalable. Because of its excellent accuracy in both classification and regression tasks, it has become more well-known. After winning many Kaggle tournaments, XGBoost has emerged as a major force in machine learning in recent years. Originally developed by [34], XGBoost introduces several enhancements over traditional gradient-boosting algorithms. A key feature of XGBoost is the incorporation of a regularization term in its loss function, which helps prevent overfitting [35].

The regularized loss function used in XGBoost is defined as:

$$L_M(F(x_i)) = \sum_{i=1}^{n} L(y_i, F(x_i)) + \sum_{m=1}^{M} \Omega(h_m)$$

where $L(y_i, F(x_i))$ measures the error between the predicted and actual values and $\Omega(h_m)$ represents the regularization term. The regularization term is expressed as:

$$\Omega(h) = \gamma^T + \frac{1}{2}\lambda|\omega|^2$$

In this expression, ϒ regulates the complexity of the trees, T is the number of tree leaves, λ serves as a penalty parameter and ω corresponds to the outputs from the leaf nodes.

Unlike standard gradient boosting, which uses first-order derivatives, XGBoost improves upon this by using a second-order Taylor approximation to optimize the loss function more effectively. The revised form of the loss function is:

$$L_M \approx \sum_{i=1}^{n} \left[g_i f_m(x_i) + \frac{1}{2} h_i f_m^2(x_i) \right] + \Omega(h_m)$$

where g_i and h_i represent the first and second derivatives of the loss function, respectively. The total loss is computed by summing the contributions from each leaf node, as described by:

$$L_M = \sum_{j=1}^{T} \sum_{i \in I_j} g_i \omega_j + \frac{1}{2} \sum_{i \in I_j} h_i + \lambda \omega_j^2 + \gamma T$$

The objective function is approximated quadratically as a result of this modification to the optimization process. Furthermore, according to [36], the regularization term makes sure that XGBoost is immune to overfitting. In order to prevent overfitting, XGBoost uses parameters, like tree depth, learning rate and sub-sampling, just like conventional gradient boosting.

One of the key advantages of XGBoost is its ability to handle minimal feature engineering, including dealing with missing values, data normalization and feature scaling. Furthermore, XGBoost can output feature importance, making it easier to understand the significance of different input features and perform feature selection. It can handle big datasets effectively, is quicker than the majority of machine-learning algorithms and frequently performs better than other models. This has made XGBoost a popular choice, particularly in Kaggle competitions. However, a disadvantage is that it has many hyper-parameters, which can make the model-tuning process quite complex [37]-[38].

2.3 Base Transformer Models for Ensemble Learning Boosting

The Transformer, introduced by [39], was designed to overcome the limitations of RNNs and traditional encoder-decoder architectures. By replacing RNNs with attention mechanisms, it enables efficient long-term memory handling. With feed-forward layers, residual connections and normalization layers combined with multi-head attention layers, the model concentrates on every token from the past. With attention weights derived from the encoder hidden states (K) and decoder state (Q), the attention mechanism aids the model in focusing on pertinent information depending on the current input. These weights are generated by an alignment function and distribution function, such as SoftMax, to enhance processing efficiency. Self-attention further enables the model to link positions within a single sequence to form comprehensive representations. Table 1 summarizes the transformer models experimented with in this study.

Table 1. Base models for boosting in transformer-based architectures.

Type	**Model**	**Supported Language**	**Training Data Source**	**Base Model**	**Highlights**	**Citation**
Mono-lingual	PhoBERT	Vietnamese	20GB pre-training dataset, including: (i) Vietnamese Wikipedia (~1GB); (ii) Vietnamese news dataset (~19GB)	RoBERTa	Uses syllable-level tokenizer, trained on a large Vietnamese dataset with fastBPE.	[40]
Mono-lingual	viBERT	Vietnamese	0GB Vietnamese news datasets (vnexpress.net, dantri.com.vn, baomoi.com, zingnews.vn, vitalk.vn, …etc.)	BERT	Improved performance on Vietnamese text processing tasks due to training on Vietnamese-specific data and pre-training techniques.	[41]
Mono-lingual	BARTpho	Vietnamese	The training data is an undivided variant of the PhoBERT pre-training corpus (about 4 billion syllable tokens)	BART	Combines Transformer structure with BERT, using a large Vietnamese dataset to enhance text generation and summarization quality.	[42]
Mono-lingual	ViT5	Vietnamese	- CC100 Dataset: Total size 138GB of raw text. - Data split: - 69GB short sentences for 256-length model. - 71GB long sentences for 1024-length model	T5	ViT5 applies Transformer-based Encoder-Decoder architecture, with two versions: Base (310M parameters) and Large (866M parameters). The model uses 36K sub-words generated by SentencePiece and trained with span-corruption self-supervision (15% rate).	[43]
Multi-lingual	XLM-RoBERTa-Base	100 languages	CommonCrawl, Wikipedia	RoBERTa	Trained on 100 languages. Uses Masked Language Modeling (MLM) objective. Vocabulary size = 250K, using SentencePiece. Training data from CommonCrawl and Wikipedia, with improved support for low-resource languages.	[44]
Multi-lingual	BERT	English	Wikipedia (2.5 billion words), BooksCorpus (800 million words)	Transformer	Trained using two unsupervised tasks: Masked LM and Next Sentence Prediction, utilizing a bidirectional Transformer architecture.	[45]
Multi-lingual	mT5	Over 100 languages, including Vietnamese	mC4 dataset (Massive Multi-lingual Crawled Corpus) collected from billions of web pages	T5	Multilingual pretraining, supports numerous languages using the T5 architecture.	[46]

In the context of this research, various Transformer-based models serve as the base models for the boosting methods explored. These models, which include both mono-lingual and multi-lingual variants, are pre-trained on large, domain-specific datasets and exhibit remarkable performance in natural-language processing tasks. Table 1 summarizes these base models, their training data sources and key highlights, showing how they contribute to enhancing model performance through boosting techniques.

2.4 Few-shot Learning Using Contrastive Learning

Few-shot learning (FSL) presents a significant challenge, as it requires models to adapt and generalize effectively with only a limited amount of data. Contrastive learning, a self-supervised method, has proven to be highly effective in addressing this challenge by learning meaningful and discriminative feature representations. By emphasizing similarities and differences among data points, contrastive learning aligns well with the objectives of FSL, where the focus is on distinguishing between unseen classes using minimal training data.

Contrastive-learning methods for FSL are often based on principles, such as noise contrastive estimation (NCE) [47]-[48] or N-pair losses [49], which facilitate the learning of robust feature spaces. For instance, SimCLR [17] employs data augmentation and non-linear transformations to train encoders that pull embeddings of similar data points closer together while pushing apart embeddings of dissimilar ones. Additionally, supervised contrastive learning [21] extends this framework to leverage labeled data, which is particularly useful in FSL scenarios where labeled support sets are small, but crucial.

In the context of FSL, contrastive learning enhances the effectiveness of models by improving the quality of representations derived from the support set (training examples). Key methods include:

- Instance-based Representations: Non-parametric softmax classifiers, such as those introduced in [50], focus on maximizing the separation between instance-level feature embeddings, helping models better distinguish between novel classes in FSL tasks.
- Multi-view Learning: Techniques like Time-Contrastive Networks (TCNs) [51] make use of multi-view data, aligning positive pairs (e.g. related samples, such as video frames) while separating negative pairs. In FSL, this can help bridge gaps between the limited support and query sets.
- Maximizing Information Representation: Methods, such as Deep InfoMax [52] among others [53], aim to maximize mutual information either within input-output pairs or across views of the same data. These methods ensure robust and meaningful feature extraction, improving FSL task performance.

Contrastive learning naturally integrates with metric-based FSL approaches, such as Prototypical Networks [54] and Siamese Networks [55], which rely on embedding distances. Discriminative representations learned through contrastive losses can significantly enhance the performance of these methods. Moreover, episodic training, commonly used in FSL, complements contrastive learning by structuring tasks to mimic real-world applications.

By leveraging contrastive learning, FSL models are better equipped to generalize from minimal data, offering a robust pathway for improving performance on tasks with scarce training resources. This combination demonstrates substantial potential to advance the effectiveness of few-shot learning in various domains.

3. METHODOLOGY

3.1 Dataset

3.1.1 Vietnamese Student Feedback

The dataset used in this study is the UIT-VSFC corpus, which consists of student feedback collected from a Vietnamese university. The dataset comprises 16,175 feedback sentences annotated with three sentiment categories: negative (0), neutral (1) and positive (2). Additionally, the dataset includes classifications for four main topics: Lecturer (0), Curriculum (1), Facility (2) and Others (3). Feedback was gathered between 2013 and 2016 through an automated survey system at the end of each semester. The surveys employed a 5-point Likert scale to assess pre-defined criteria and open-ended questions to gather more detailed feedback.

A key strength of this dataset is its reliability, demonstrated by an inter-annotator agreement score of 91%, which reflects a high level of consistency in sentiment labeling [56]. To evaluate few-shot learning scenarios, sub-sets of the training data were constructed with limited labeled samples per class. This setup ensured that the models were trained and tested under minimal data conditions, providing a robust assessment of their generalization capabilities with few-shot learning. Table 2 presents some examples from the dataset.

Table 3 presents the distribution of sentiment and topic categories. The dataset is highly imbalanced, with positive and negative sentiments each accounting for nearly 50%, while neutral feedback represents only 4.32%. In terms of topic labels, the majority of the feedback pertains to the Lecturer category (71.76%), followed by Curriculum (18.79%), indicating that students tend to comment most frequently on teaching-related aspects.

Furthermore, a linguistic analysis of the dataset reveals that student feedback tends to be concise: over 83% of the sentences contain 15 words or fewer. As shown in Table 4, negative sentences are generally longer than positive or neutral ones, likely because they often include justifications or suggestions for improvement. Table 5 displays the length distribution by topic, where feedback related to Lecturer, Curriculum and Facility frequently involves more detailed expressions (i.e., more than five words), reflecting students' emphasis on those aspects.

Table 2. Examples of the UIT-VSFC dataset.

No.	Sentence	Sentiment	Topic
1	Giảng dạy nhiệt tình, liên hệ thực tế khá nhiều, tương tác với sinh viên tương đối tốt. (Enthusiastic teaching, incorporating a lot of real-life examples and relatively good interaction with students.)	Positive (2)	Lecturer (0)
2	Tính thực tế cũng cao so với việc thi lý thuyết lấy điểm. (It is also more practical compared to taking theoretical exams for	Positive (2)	Curriculum (1)
3	Phòng máy cũ, nhưng nhìn chung thì không có ảnh hưởng gì vì thầy dạy rất nhiệt tình. (The computer lab is outdated, but overall, it doesn't affect much, because the teacher is very enthusiastic.)	Neutral (1)	Facility (2)
4	Học thì quá ít nhưng khi thi thì quá nhiều yêu cầu viết code trong đề thi thì sao mà sinh viên có thể làm được. (The amount of learning is too little, but the exam demands too much coding. How can students possibly handle it?)	Negative (0)	Others (3)

Table 3. Distribution of sentiment and topic labels in the UIT-VSFC corpus (%).

Topic	Positive (%)	Negative (%)	Neutral (%)	Total (%)
Lecturer	33.57	25.38	1.81	71.76
Curriculum	3.40	14.39	1.00	18.79
Facility	0.11	4.21	0.08	4.4
Others	1.61	2.01	1.43	5.04
Total	**49.69**	**45.99**	**4.32**	**100**

Table 4. Distribution of sentences by sentiment and sentence length (%).

Length (words)	Positive (%)	Negative (%)	Neutral (%)	Total (%)
1–5	17.26	9.75	2.31	29.32
6–10	21.00	15.34	1.17	37.55
11–15	7.19	8.59	0.51	16.29
16–20	2.37	5.17	0.15	7.69
21–25	1.06	2.85	0.07	3.98
26–30	0.37	1.72	0.07	2.16
>30	0.40	2.57	0.04	3.01
Total	**49.65**	**45.99**	**4.32**	**100**

Table 5. Sentence-length distribution by topic (%).

Length (words)	**Lecturer (%)**	**Curriculum (%)**	**Facility (%)**	**Others (%)**	**Total (%)**
1–5	20.80	3.61	2.63	2.28	29.32
6–10	27.84	6.69	1.94	1.08	37.55
11–15	11.93	2.61	0.84	0.91	16.29
16–20	5.44	1.35	0.46	0.44	7.69
21–25	2.96	0.62	0.25	0.15	3.98
26–30	1.56	0.32	0.19	0.09	2.16
>30	1.13	0.59	0.10	1.19	3.01

3.1.2 Customer Product Reviews Dataset

To further evaluate model generalization, particularly for few-shot learning tasks across different domains, we utilized the "Vietnamese Sentiment Analyst" dataset, herein referred to as Customer Product Reviews. This corpus contains 31,460 Vietnamese customer reviews focused on various products. Each review is labeled with one of three sentiment polarities: positive, negative, or neutral. Table 6 presents some examples from the dataset.

Table 7 details the distribution of sentiment labels and sentence lengths within this dataset. Overall, positive sentiment is predominant (63.87%, N=20,093). In terms of sentence length, reviews are generally concise, with the highest concentration of positive reviews in the 1-5 word (20.84% of total dataset) and 6-10 word (21.14%) brackets.

Table 6. Examples of the customer product reviews dataset.

No.	**Sentence**	**Sentiment**
1	Chất lượng sản phẩm đúng như hình. Đóng gói sản phẩm tạm được. (The product quality is just like in the pictures. The packaging is acceptable.)	Positive (2)
2	Cơ mà tôi mua hôm nay, ngày mai shop làm flash sale là sao. (But I bought it today and now the shop is doing a flash sale tomorrow — what's that about?)	Neutral (1)
3	Có giống hình nhưng vải rất mỏng không đúng như trong hình. Giá tiền tương đương với sản phẩm. (It looks like the picture, but the fabric is very thin and not as shown. The price is equivalent to the product.)	Negative (0)

Table 7. Distribution of sentiment labels by review length.

Length (words)	**Positive (%)**	**Negative (%)**	**Neutral (%)**
1–5	20.84	6.61	5.18
6–10	21.14	7.47	5.36
11–15	9.46	3.53	2.4
16–20	4.96	1.71	1.06
21–25	2.83	0.79	0.52
26–30	1.96	0.48	0.2
>30	2.68	0.62	0.21
Total	63.87	21.2	14.93

3.2 Model Evaluation Metrics

These metrics are typically calculated using weighted averages to better reflect performance, especially in imbalanced datasets.

Precision measures the ratio of correctly predicted positive instances to all predicted positive instances. It is crucial in problems where false positives have high costs. Precision ranges from 0 to 1 and can be calculated as a weighted average, considering class sample sizes.

$$Precision = \frac{True\ positives}{True\ positives + False\ positives}$$

Recall measures the model's ability to detect actual positive instances. It is important in problems where missing positive cases can have severe consequences. Like Precision, Recall ranges from 0 to 1 and can be computed as a weighted average.

$$Recall = \frac{True\ positives}{True\ positives + False\ negatives}$$

F1-score combines Precision and Recall to give a comprehensive performance measure, especially useful in imbalanced datasets. It ranges from 0 to 1, with higher values indicating a better balance between Precision and Recall. When calculated as a weighted average, it reflects the model's overall performance across all classes.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

3.3 Software and Hardware

For the proposed research, Python was used as the programming language within the Google Colab runtime environment, which provides access to powerful hardware acceleration through GPUs. Specifically, the NVIDIA Tesla T4 GPU was utilized, equipped with 2560 CUDA cores designed to support deep-learning tasks. These cores, along with specialized Tensor Cores, allow for efficient execution of matrix-heavy operations commonly used in neural-network models. The environment ran on a CPU with an Intel (R) Core (TM) i3-4005U Processor at 1.70 GHz, paired with 4 GB of RAM.

To clarify the computational cost, Table 8 presents the number of trainable parameters and the approximate model size (in MB) for each transformer-based model evaluated in this study. Models with a higher number of parameters and larger memory footprints—such as mBART Large EN-RO (610M parameters, ~2.3GB) or mT5 Base (390M parameters, ~1.5GB)—require significantly more GPU memory, training time and processing power for both fine-tuning and inference. In contrast, smaller models, like ViBERT and PhoBERT, are comparatively lightweight and faster to train, making them more suitable for environments with limited computational resources. Table 8 presents the number of parameters and the sizes of the transformer models used in this study.

Table 8. Trainable parameters and approximate model sizes of pretrained transformer models.

Model	**Trainable Parameters**	**Model Size (MB)**
PhoBERT	134,998,272	514.98
ViBERT	115,354,368	440.04
XLM-RoBERTa Base	278,043,648	1,060.65
BERT Base Uncased	109,482,240	417.64
mT5 Base	390,315,264	1,488.93
BERT Base Multilingual Cased	177,853,440	678.46
mBART Large EN-RO	610,851,840	2,330.21
BARTpho-syllable	395,814,912	1,509.91
ViT5 Base	225,950,976	861.93

3.4 Experimental Framework

Few-shot Learning was implemented with varying levels of data availability (N = 1, 5 and 20) to evaluate the performance of several transformer-based models on limited labeled data. The models included PhoBERT, ViBERT, XLM-RoBERTa, mT5, multi-lingual BERT, base BERT, MBart, BARTpho and

ViT5. Each model was fine-tuned using a contrastive learning approach and their performances were evaluated using the F1-score. In addition to transformer-based models, the study also conducted experiments with several classical machine-learning architectures, including RNN, GRU and LSTM, to serve as comparative baselines. This inclusion provides a broader perspective on the effectiveness of modern pre-trained models under low-resource conditions.

For the ensemble-learning stage, our primary selection criterion was individual model performance. Consequently, the top three models demonstrating the highest average F1-scores were chosen as base learners. To validate this selection, we conducted pairwise statistical significance tests (paired t-tests), which confirmed that these models belonged to a top-performing tier, showing statistically significant improvements over most other models. This approach ensures that the components of our ensemble are strong and reliable individual predictors.

To further improve prediction accuracy, a supervised ensemble strategy based on boosting was applied. Instead of using simple combination methods, such as majority voting or averaging, the outputs from the top-three transformer models served as input features for three ensemble learners: AdaBoost, Gradient Boosting and XGBoost. These ensemble models were trained to learn from the prediction patterns of the base models, functioning as meta-learners that integrate their outputs into a final decision. This method is analogous to a stacking framework, where boosting algorithms iteratively focus on samples that are harder to classify, thereby refining predictions and enhancing overall generalization performance. Detailed descriptions of the proposed method and framework are presented in Figure 1.

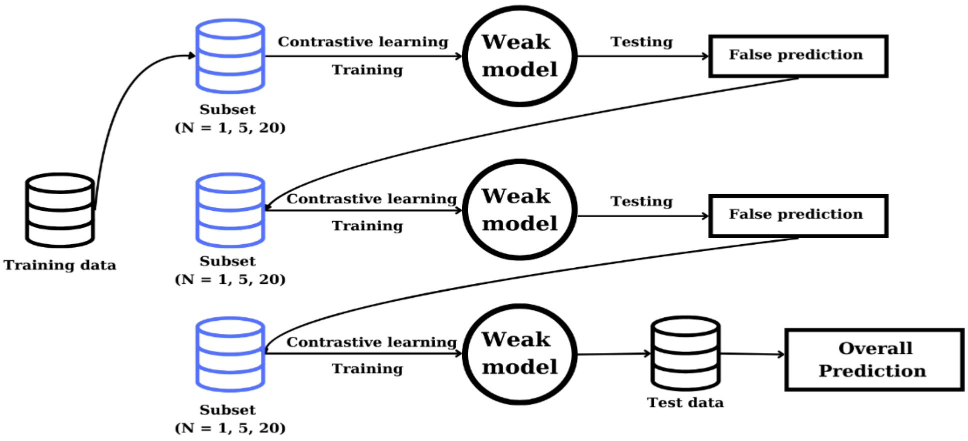


Figure 1. Flow diagram of proposed methodology. The framework trains weak models on data subsets (N = 1, 5, 20) using contrastive learning. False predictions are identified during testing and outputs are combined to produce the final overall prediction on test data [56]–[57].

3.5 Hyper-parameter Tuning

Bayesian optimization is a powerful and efficient method for hyper-parameter tuning, especially in complex machine-learning models where traditional techniques, such as Grid Search and Random Search, fall short due to their inefficiency or lack of strategic sampling. By modeling the objective function using a probabilistic surrogate model, Bayesian optimization intelligently selects the next sampling point based on past evaluations, effectively balancing exploration and exploitation. This approach is particularly suitable for combinatorial optimization problems where gradient-based methods are not applicable. Bayesian optimization is the top choice for optimizing objective functions [57-59]. In this study, Bayesian optimization is employed to tune hyper-parameters for boosting algorithms, including AdaBoost, Gradient Boosting and XGBoost. Examples of optimized parameters include the learning rate, number of estimators, maximum tree depth, …etc.

Tables 9, 10 and 11 present the hyper-parameters of the boosting models—AdaBoost, Gradient Boosting and XGBoost—that were optimized using Bayesian optimization. These tables detail the specific parameters selected for tuning, such as learning rate, number of estimators and maximum depth, among others, which play a crucial role in controlling model complexity, convergence behaviour and overall predictive performance.

Table 9. Optimized hyper-parameters using Bayesian optimization for AdaBoost across datasets and N-shot settings.

Dataset	N-shot	Learning Rate	N estimators
UIT-VSFC (Sentiment)	N=1	0.010	820
	N=5	0.650	29
	N=20	0.279	884
UIT-VSFC (Topic)	N=1	0.159	920
	N=5	0.677	1000
	N=20	0.558	180
Customer Product Reviews	N=1	0.820	884
	N=5	0.128	730
	N=20	0.159	920

Table 10. Optimized hyper-parameters using Bayesian optimization for XGBoost across datasets and N-shot settings.

Dataset	N-shot	Column Subsample	Learning Rate	Max. Depth	No. of Estimators	L1 Regularization	L2 Regularization	Subsample Ratio
UIT-VSFC (Sentiment)	N=1	0.300	0.010	11	506	0.703	0.955	1.000
	N=5	0.680	0.229	7	854	0.324	0.051	0.785
	N=20	0.969	0.108	11	474	0.381	0.211	0.500
UIT-VSFC (Topic)	N=1	1.000	0.168	12	1000	1.000	0.000	0.873
	N=5	0.300	0.062	5	1000	0.000	1.000	1.000
	N=20	0.611	0.228	4	490	0.188	0.454	0.578
Customer Product Reviews	N=1	1	0.027	3	100	1	0	1
	N=5	0.969	0.108	11	474	0.381	0.211	0.5
	N=20	1	0.025	9	551	1	0.549	0.519

Table 11. Optimized hyper-parameters using Bayesian optimization for Gradient Boosting across datasets and N-shot settings.

Dataset	N-shot	Learning Rate	Maximum Depth	Minimum Samples per Leaf	Minimum Samples to Split	Number of Estimators	Subsample Ratio
UIT-VSFC (Sentiment)	N=1	0.082	10	4	9	633	0.797
	N=5	0.072	11	2	8	812	0.504
	N=20	0.279	7	10	2	173	0.597
UIT-VSFC (Topic)	N=1	0.029	3	1	2	337	0.913
	N=5	0.013	8	2	4	600	0.900
	N=20	0.170	12	10	2	100	0.774
Customer Product Reviews	N=1	0.258	9	9	5	443	0.606
	N=5	0.298	10	9	9	195	0.520
	N=20	0.146	11	2	7	608	0.531

The hyper-parameters optimized in this study critically influence the balance between model bias and variance, as well as training efficiency. Learning rate determines the step size during model updates, affecting convergence speed and overfitting risk. Number of estimators specifies how many weak learners (trees) are combined, impacting the model's capacity and complexity.

For XGBoost, additional parameters, such as column sub-sample ratio, control the fraction of features used per tree to prevent overfitting. Maximum tree depth limits the complexity of individual trees. L1 (reg_alpha) and L2 (reg_lambda) regularization terms penalize model complexity to enhance robustness, while sub-sample ratio governs the portion of training data sampled per tree, reducing variance.

In Gradient Boosting, besides learning rate and number of estimators, the minimum samples per leaf and minimum samples to split parameters regulate tree growth by specifying thresholds for leaf-node formation and internal-node splitting, further preventing overfitting.

3.6 Statistical Significance Testing and Confidence Intervals

A paired t-test is used to determine whether the difference in performance between models is statistically significant. Instead of using k-fold cross-validation, the models are run multiple times with different random initializations to generate sets of performance results. For each run, the performance difference between two models A and B is calculated as:

$$d_i = acc_i(A) - acc_i(B)$$

From these differences, the sample mean is computed as:

$$m = \frac{1}{N}\sum_{n=1}^{N} \mathrm{diff}_n$$

and the sample standard deviation is:

$$sd = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N} (\mathrm{diff}_n - m)^2}$$

The t-statistics are then calculated as:

$$t = \frac{m\sqrt{N}}{sd}$$

Finally, the t-value is compared against the critical value from the t-distribution with N−1 degrees of freedom to test the null hypothesis. If the p-value is less than 0.05 ($p<0.05$), it can be concluded that the difference between the two models is statistically significant. Using the paired t-test thus helps strengthen the reliability of selecting more effective models.

Besides the paired t-test, the 95% Confidence Interval (CI) is used to provide a range within which the true performance metric of each model is likely to fall with 95% certainty. Each model is run 5 times with different random seeds to capture the variability caused by random initialization. Reporting the mean performance along with the 95% CI reflects the stability and reliability of the models.

This approach allows for a more comprehensive evaluation by quantifying the uncertainty around the average performance, ensuring that model comparison and selection consider not only the mean accuracy, but also the consistency across multiple runs.

4. RESULTS

4.1 Few-shot Learning Experiments on Transformer Models

The experimental results of transformer models are presented on the dataset for two tasks: sentiment classification and topic classification. Additionally, experiments were conducted on sentiment analysis using the customer product reviews dataset. Each model is evaluated on the same training dataset with setups of N = 1, N = 5 and N = 20. The training environment and hyper-parameters are identical across all models. The reports highlight the precision, recall and F1-score achieved by each model, specifying which transformers perform well in 1-shot learning (N = 1), few-shot learning (N = 5) and scenarios with a significant amount of data.

Table 12 shows the experimental results on the sentiment-analysis task, with XLM-RoBERTa outperforming other models and achieving the highest F1-scores. This model demonstrates the best performance in precision, recall and F1-score, making it the most effective model for sentiment analysis. Other models, such as BARTpho and BERT multi-lingual, also show strong results.

Table 13 shows the experimental results on the topic-classification task. The highest F1-score for N = 20 is 0.817, achieved by XLM-RoBERTa. PhoBERT and BARTpho also show strong performance, but XLM-RoBERTa leads in this setup. Table 14 presents the experimental results on the customer product reviews dataset. The highest F1-score for N = 20 is 0.744, achieved by mT5. ViBERT and ViT5 also show strong performance.

Notably, the confidence intervals (CIs) among transformer-based models show minimal variation, with differences generally remaining below 0.02. This indicates consistent and stable performance across different runs. In contrast, traditional models, such as LSTM, RNN and GRU, exhibit greater fluctuations in their CI values, reflecting less stability and higher variability in performance.

Table 12. The experimental results of transformer models for sentiment analysis.

Model	**N = 1**			**N = 5**			**N = 20**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
RNN	0.449± 0.0375	0.251± 0.0451	0.322± 0.0396	0.520± 0.0296	0.387± 0.0416	0.444± 0.0312	0.645± 0.0261	0.502± 0.0421	0.565± 0.0364
GRU	0.369± 0.0223	0.287± 0.0322	0.323± 0.0268	0.552± 0.0575	0.477± 0.0428	0.512± 0.0443	0.654± 0.0370	0.591± 0.0503	0.621± 0.0449
LSTM	0.381± 0.0122	0.381± 0.0320	0.381± 0.0289	0.626± 0.0366	0.504± 0.0198	0.558± 0.0217	0.657± 0.0366	0.586± 0.0310	0.619± 0.0343
PhoBERT	0.610± 0.0081	0.591± 0.0098	0.596± 0.0079	0.759± 0.0048	0.708± 0.0053	0.733± 0.0036	0.846± 0.0055	0.812± 0.0045	0.829± 0.0049
ViBERT	0.549± 0.0121	0.278± 0.0106	0.369± 0.0088	0.580± 0.0083	0.499± 0.0036	0.536± 0.0076	0.723± 0.0083	0.608± 0.0076	0.661± 0.0077
XLM-RoBERTa	0.603± 0.0075	0.470± 0.0089	0.528± 0.0077	0.720± 0.0040	0.625± 0.0066	0.669± 0.0058	0.843± 0.0081	0.834± 0.0075	***0.838*± *0.0075***
BERT base	0.597± 0.0098	0.527± 0.0032	0.560± 0.0038	0.692± 0.0020	0.460± 0.0088	0.553± 0.0033	0.672± 0.0038	0.630± 0.0081	0.650± 0.0076
mT5	0.606± 0.0072	0.471± 0.0025	0.530± 0.0057	0.769± 0.0047	0.653± 0.0027	0.653± 0.0046	0.779± 0.0096	0.692± 0.0052	0.721± 0.0080
BERT multilingual	0.656± 0.0125	0.655± 0.0098	***0.655*± *0.0101***	0.748± 0.0186	0.672± 0.0143	0.672± 0.0153	0.801± 0.0142	0.743± 0.0096	0.765± 0.0138
MBart	0.582± 0.0069	0.525± 0.0052	0.552± 0.0057	0.685± 0.0091	0.638± 0.0093	0.661± 0.0090	0.811± 0.0076	0.793± 0.0096	0.801± 0.0082
BARTpho	0.608± 0.0093	0.533± 0.0082	0.568± 0.0081	0.764± 0.0091	0.712± 0.0087	***0.737*± *0.0090***	0.843± 0.0064	0.780± 0.0097	0.806± 0.0084
ViT5	0.594± 0.0188	0.590± 0.0157	0.592± 0.0165	0.745± 0.0109	0.611± 0.0146	0.671± 0.0138	0.825± 0.0070	0.742± 0.0051	0.771± 0.0069

4.2 Pairwise Statistical Significance Testing Using Paired T-test

After training and evaluating all models on two primary tasks, sentiment analysis and topic classification, additional experiments were also conducted on sentiment analysis using the customer product reviews dataset. The three models with the highest F1-scores were selected to undergo paired t-test evaluation against each of the remaining models. The objective was to assess whether the performance differences between models are statistically significant.

Each model was run five times with different random seeds to capture variation introduced by random initialization. The performance differences (in terms of F1-score) between each model pair were calculated and a paired t-test was conducted using a significance threshold of $p<0.05$. The results show that the top three models consistently outperformed most other models with statistically significant differences, confirming their superiority in a reliable manner. Notably, the model with the lowest average performance still achieved statistically significant results ($p < 0.05$) in two comparisons, indicating that it also qualifies for inclusion in the ensemble model.

Table 13. The experimental results of transformer models for topic analysis.

Model	**N = 1**			**N = 5**			**N = 20**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
RNN	0.540± 0.0411	0.197± 0.0325	0.289± 0.0336	0.599± 0.0233	0.297± 0.0341	0.397± 0.0302	0.624± 0.0265	0.388± 0.0372	0.478± 0.0298
GRU	0.481± 0.0231	0.237± 0.0421	0.318± 0.0403	0.633± 0.0158	0.356± 0.0229	0.456± 0.0196	0.644± 0.0331	0.669± 0.0253	0.656± 0.0268
LSTM	0.491± 0.0321	0.229± 0.0210	0.312± 0.0298	0.649± 0.0254	0.323± 0.0187	0.431± 0.0203	0.524± 0.0135	0.715± 0.0201	0.605± 0.0184
PhoBERT	0.708± 0.0101	0.647± 0.0128	0.676± 0.0120	0.762± 0.0063	0.667± 0.0098	0.711± 0.0088	0.821± 0.0063	0.767± 0.0041	0.791± 0.0055
ViBERT	0.679± 0.0156	0.214± 0.0203	0.325± 0.0139	0.708± 0.0109	0.534± 0.0056	0.609± 0.0063	0.774± 0.0182	0.682± 0.0099	0.725± 0.0103
XLM-RoBERTa	0.639± 0.0095	0.646± 0.0127	0.642± 0.0110	0.741± 0.0063	0.630± 0.0036	0.681± 0.0054	0.841± 0.0096	0.795± 0.0082	***0.817*± *0.0079***
BERT base	0.588± 0.0153	0.278± 0.0102	0.378± 0.0115	0.691± 0.0118	0.497± 0.0064	0.578± 0.0082	0.754± 0.0053	0.644± 0.0089	0.695± 0.0076
mT5	0.672± 0.0089	0.448± 0.0056	0.538± 0.0076	0.734± 0.0038	0.451± 0.0025	0.559± 0.0030	0.836± 0.0056	0.719± 0.0089	0.773± 0.0088
BERT multilingual	0.696± 0.0145	0.696± 0.0096	0.696± 0.0135	0.790± 0.0202	0.594± 0.0158	0.678± 0.0166	0.820± 0.0083	0.719± 0.0103	0.766± 0.0096
MBart	0.642± 0.0080	0.547± 0.0088	0.591± 0.0082	0.823± 0.0093	0.738± 0.0066	***0.778*± *0.0083***	0.846± 0.0103	0.768± 0.0152	0.805± 0.0109
BARTpho	0.692± 0.0132	0.419± 0.0122	0.522± 0.0126	0.783± 0.0102	0.661± 0.0123	0.744± 0.099	0.850± 0.0101	0.763± 0.0095	0.804± 0.0097
ViT5	0.736± 0.0052	0.684± 0.0085	***0.709*± *0.0063***	0.786± 0.0102	0.660± 0.0092	0.741± 0.0091	0.846± 0.0064	0.780± 0.0092	0.812± 0.0066

Table 14. The experimental results of transformer models for customer product reviews dataset.

Model	**N = 1**			**N = 5**			**N = 20**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
RNN	0.305± 0.0482	0.321± 0.0554	0.313± 0.0501	0.462± 0.0363	0.453± 0.0382	0.457± 0.0351	0.515± 0.0334	0.496± 0.0312	0.503± 0.0305
GRU	0.324± 0.0505	0.343± 0.0578	0.332± 0.0524	0.481± 0.0381	0.472± 0.0403	0.475± 0.0372	0.533± 0.0352	0.514± 0.0331	0.521± 0.0323
LSTM	0.342± 0.0521	0.361± 0.0595	0.350± 0.0543	0.503± 0.0402	0.491± 0.0425	0.494± 0.0391	0.552± 0.0373	0.535± 0.0354	0.543± 0.0342
PhoBERT	0.456± 0.0121	0.484± 0.0142	0.470± 0.0135	0.623± 0.0083	0.616± 0.0102	0.619± 0.0091	0.701± 0.0072	0.679± 0.0064	0.690± 0.0068
ViBERT	0.469± 0.0163	0.484± 0.0211	0.476± 0.0184	0.685± 0.0119	0.680± 0.0098	0.682± 0.0105	0.729± 0.0121	0.729± 0.0103	0.729± 0.0112
XLM-RoBERTa	0.397± 0.0112	0.535± 0.0135	0.456± 0.0121	0.694± 0.0091	0.643± 0.0103	0.668± 0.0095	0.725± 0.0087	0.677± 0.0079	0.700± 0.0081
BERT base	0.471± 0.0185	0.516± 0.0199	***0.492± 0.0191***	0.620± 0.0131	0.622± 0.0124	0.621± 0.0128	0.679± 0.0093	0.670± 0.0108	0.674± 0.0099
mT5	0.457± 0.0138	0.508± 0.0145	0.481± 0.0141	0.699± 0.0095	0.676± 0.0115	***0.687± 0.0101***	0.748± 0.0086	0.741± 0.0094	***0.744± 0.0090***
BERT multilingual	0.451± 0.0152	0.427± 0.0148	0.439± 0.0149	0.685± 0.0122	0.632± 0.0138	0.657± 0.0129	0.728± 0.0098	0.697± 0.0113	0.712± 0.0104
MBart	0.437± 0.0115	0.456± 0.0128	0.446± 0.0119	0.658± 0.0081	0.628± 0.0094	0.643± 0.0094	0.756± 0.0079	0.677± 0.0091	0.714± 0.0084
BARTpho	0.444± 0.0141	0.441± 0.0153	0.442± 0.0148	0.669± 0.0112	0.632± 0.0109	0.650± 0.0110	0.760± 0.0081	0.709± 0.0092	0.734± 0.0087
ViT5	0.483± 0.0102	0.485± 0.0115	0.484± 0.0108	0.653± 0.0092	0.669± 0.0105	0.661± 0.0097	0.726± 0.0074	0.734± 0.0082	0.730± 0.0078

This evaluation approach, based on paired t-tests, ensures that model selection is not solely based on average performance, but also considers stability and statistical significance across multiple runs, thereby enhancing the robustness and reliability of the final model-selection process. The results of the paired t-tests are reported in Tables from 15 to 23.

Note on statistical-significance levels: (*: p < 0.05), (**: p < 0.01) and (***: p < 0.001).

Table 15. Pairwise statistical significance testing using paired t-test on sentiment-analysis task (N = 1).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
PhoBERT		***	***	0.0245	***	***	***	***	0.0377
BERT multilingual	***	***	***	***	***		***	***	***
ViT5	0.0377	***	***	***	***	***	***	***	

Table 16. Pairwise statistical significance testing using paired t-test on sentiment-analysis task (N = 5).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
PhoBERT		***	***	***	***	0.0108	***	***	***
BERT multilingual	0.0108	***	***	***	***		***	***	***
BARTpho	***	***	***	***	**	***	***0.0518***		***

Table 17. Pairwise statistical significance testing using paired t-test on sentiment-analysis task (N = 20).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
XLM-RoBERTa	***	***		***	***	***	***	***	***
PhoBERT		***	***	***	***	***	***	***	***
BARTpho	***	***	***	***	***	***	***		***

Table 18. Pairwise statistical significance testing using paired t-test on topic-classification task (N = 1).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
PhoBERT		***		***	***	***	***	***	***
BERT multilingual	**	***	***	***	***		***	***	0.0249
ViT5	***	***	***	***	***	0.0249	***	***	

Table 19. Pairwise statistical significance testing using paired t-test on topic-classification task (N = 5).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
MBart	***	***	***	***	***	***		***0.6952***	***0.6951***
BARTpho	**	***	***	***	***	***	***0.6952***		**
ViT5	0.0730	***	***	***	***	***	***	**	

Table 20. Pairwise statistical significance testing using paired t-test on topic-classification Task (N = 20).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
XLM-RoBERTa	***	***		***	***	***	***	***	**
MBart	***	***	***	***	***	***		***0.3903***	0.0479
ViT5	***	***	**	***	***	***	0.0479	***	

Table 21. Pairwise statistical significance testing using paired t-test on customer product reviews dataset (N = 1).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
ViBERT	0.0322		***	0.0359	***	***	***	**	**
BERT base	***	0.0359	***		**	***	***	***	**
mT5	***	***	**	**		***	***	**	

Table 22. Pairwise statistical significance testing using paired t-test on customer product reviews dataset (N = 5).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
mT5	***	**	0.0122	***		***	***	***	**
ViBERT	***		***	***	**	***	***	**	***
XLM-RoBERTa	***	***		***	0.0122	***	**	***	***

Table 23. Pairwise statistical significance testing using paired t-test on customer product reviews dataset (N = 20).

	PhoBERT	ViBERT	XLM-RoBERTa	BERT base	mT5	BERT multilingual	MBart	BARTpho	ViT5
mT5	***	***	***	***		***	***	***	***0.5856***
BARTpho	***	***	***	***	***	**	***		0.0152
ViT5	***	***	***	***	***0.5856***	***	***	0.0152	

4.3 Experiments on Boosting Models with Transformers

Based on the few-shot learning experiments with transformers, the study conducted boosting experiments using the best-performing models. Specifically, the three models with the highest F1-scores were selected as base models for three boosting methods. Table 24 and Table 25 present the experimental results for two tasks: sentiment analysis and topic classification. Table 26 presents the experimental results on the customer product reviews dataset. The results indicate that Gradient Boosting achieved the best performance across all tasks and base models. With N=20, Gradient Boosting reached an F1-score of 0.836 on the sentiment-analysis task and 0.824 on the topic-classification task. However, the performance of the other two methods was also very promising.

Table 24. Experimental results of boosting on the sentiment-analysis task.

N	**Base model**	**AdaBoost**			**Gradient Boosting**			**XGBoost**		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
1	PhoBERT + BERT multilingual + ViT5	0.639	0.670	0.648	0.665	0.675	***0.661***	0.638	0.671	0.653
5	PhoBERT + BERT multilingual+ BARTpho	0.754	0.785	0.765	0.792	0.796	***0.776***	0.772	0.796	0.774
20	XLM-RoBERTa +BERT multilingual+ BARTpho	0.798	0.841	0.819	0.837	0.853	***0.836***	0.833	0.849	***0.836***

Table 25. Experimental results of boosting on the topic-classification task.

N	**Base model**	**AdaBoost**			**Gradient Boosting**			**XGBoost**		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
1	PhoBERT + BERT multilingual + ViT5	0.732	0.754	0.709	0.723	0.758	***0.725***	0.717	0.748	0.723
5	MBart + BARTpho + ViT5	0.799	0.803	0.735	0.811	0.812	***0.804***	0.789	0.804	0.790
20	XLM-RoBERTa+ Bart + ViT5	0.826	0.834	0.817	0.832	0.819	***0.824***	0.795	0.829	0.811

Table 26. Experimental results of boosting on the customer product reviews dataset.

N	Base model	AdaBoost			Gradient Boosting			XGBoost		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
1	ViBERT + BERT base + mT5	0.532	0.556	0.544	0.536	0.573	***0.554***	0.530	0.555	0.542
5	mT5 + ViBERT+ XLM-RoBERTa	0.665	0.685	0.675	0.709	0.706	***0.707***	0.694	0.703	0.698
20	mT5 + BARTpho + ViT5	0.740	0.750	0.745	0.749	0.761	***0.755***	0.751	0.753	0.752

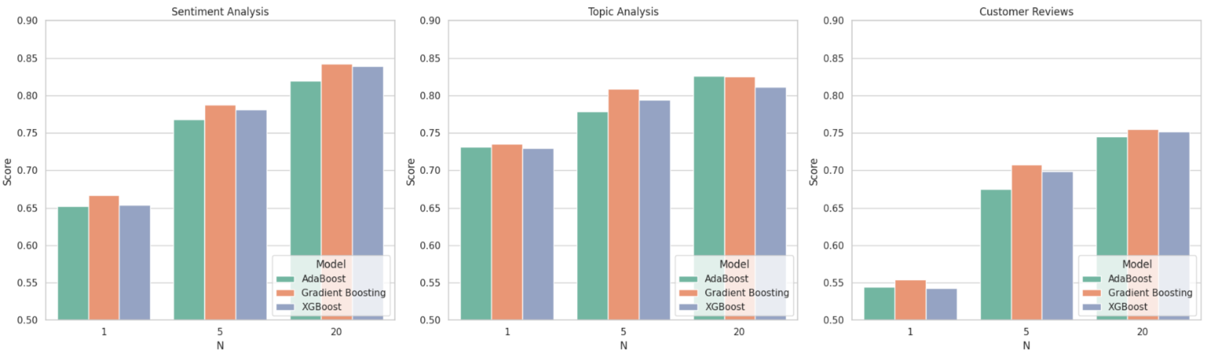


Figure 2. Comparison of F1-scores of boosting algorithms (AdaBoost, Gradient boosting, XGBoost) on two tasks: sentiment analysis and topic analysis, using different combined models.

5. Conclusions

The findings of this study have far-reaching implications that contribute to yet another theoretical and practical advancement in sentiment analysis, particularly in low-resource educational environments. To mitigate challenges, such as limited data and computational inefficiency, the proposed study introduces a novel framework that combines Few-Shot Learning (FSL) and Transformer-based ensemble models with boosting approaches.

By drawing on the strengths of both Transformer models using self-attention to learn patterns from rich data and adapting the FSL setting, this paper then introduces a hybrid methodology that addresses the shortcomings of traditional supervised approaches in low-data scenarios. Moreover, it presents the role of boosting techniques, such as Gradient boosting and XGBoost, and their capabilities in classifying the sentiments, which may set a pathway for forthcoming research on ensemble learning for NLP tasks.

On the practical side, the framework presented in this research will serve as a basis for providing actionable knowledge to educational institutes to better analyze students' feedback, hence improving their learning experience and the quality of teaching. The scalability of the method makes it relevant for a wide range of fields that experience a scarcity of labeled data. Furthermore, its efficient use of resources demonstrates its practicality for translating to practice, even in settings where computational power is limited. Although the model demonstrates effectiveness in sentiment-analysis tasks with limited training resources, particularly in educational feedback systems, this study acknowledges the ethical aspects associated with its real-world deployment. Fairness is a key concern when sentiment models are trained on imbalanced datasets in terms of class distribution, dialectal expressions and stylistic variations, which often predominantly reflect students' perspectives. This may result in systematic bias against certain groups.

Bias during evaluation and sentiment classification may lead the model to misinterpret students' feedback, especially when cultural context or specific expression styles are not accurately captured in the training data. For instance, negative feedback expressed politely or formally may be misclassified as neutral or even positive. This misunderstanding can delay necessary interventions by model users when addressing customer requests or student concerns. Another issue to consider is the impact of misclassification, which can lead to incorrect conclusions in both educational and customer-service evaluations. If negative feedback is misinterpreted as positive, educational administrators or customer-service staff may overlook significant issues raised by students or customers, potentially affecting the overall learning or service experience. To mitigate these risks, future research and deployments should apply fairness-aware training methods, such as data rebalancing and debiasing techniques, utilize more diverse datasets to increase representativeness and integrate human oversight during the result-validation process.

Despite the promising results, this study has several limitations that provide clear avenues for future research. First, our framework's effectiveness is contingent on the availability of high-quality pre-trained Transformer models. Consequently, its application may be challenging for low-resource languages or specialized domains that lack representative pre-training corpora. Second, the use of ensemble and boosting techniques, while improving performance, introduces additional computational complexity, which might be a barrier for organizations with limited resources. A third limitation lies in our ensemble selection logic. In this study, base models were chosen primarily based on their individual performance. While this ensures strong components, it does not explicitly guarantee model diversity, a critical factor for robust ensembling. Finally, as the evaluation was conducted on a single dataset (UIT-VSFC), the generalizability of our findings needs further validation on other datasets and across different domains.

Building on these limitations, future work can proceed in several promising directions. To address generalizability, the framework should be evaluated across diverse domains, such as healthcare or finance, and on datasets in other languages. To enhance the ensemble methodology, future research should explore more sophisticated, diversity-aware selection strategies that co-optimize for both model performance and diversity; for instance, by analyzing prediction correlations. Furthermore, performance in extremely low-data environments could be improved by optimizing Transformer architectures for lightweight deployments and leveraging advanced data-augmentation strategies. Finally, integrating human-in-the-loop feedback systems could improve model adaptability in ambiguous cases, making the framework more practical for real-world deployment. This research underscores the transformative potential of advanced NLP techniques in enhancing sentiment analysis, offering a valuable framework for addressing challenges in resource-constrained scenarios.

REFERENCES

[1] M. Bansal, S. Verma, K. Vig and K. Kakran, "Opinion Mining from Student Feedback Data Using Supervised Learning Algorithms," Lecture Notes in Networks and Systems, vol. 514, pp. 1–15, 2022.

[2] A. Ligthart, C. Catal and B. Tekinerdogan, "Systematic Reviews in Sentiment Analysis: A Tertiary Study," Artificial Intelligence Review, vol. 54, no. 7, pp. 4997–5053, 2021.

[3] A. I. M. Elfeky et al., "Advance Organizers in Flipped Classroom *via* e-Learning Management System and the Promotion of Integrated Science Process Skills," Thinking Skills and Creativity, vol. 35, 2020.

[4] H. Zhao et al., "A Machine Learning-based Sentiment Analysis of Online Product Reviews with a Novel Term Weighting and Feature Selection Approach," Inf. Process. Manag., vol. 58, no. 5, pp. 1–15, 2021.

[5] Y. Zhang, J. Wang and X. Zhang, "Conciseness is Better: Recurrent Attention LSTM Model for Document-level Sentiment Analysis," Neurocomputing, vol. 462, pp. 1–12, 2021.

[6] Z. Liu et al., "Temporal Emotion-aspect Modeling for Discovering What Students are Concerned about in Online Course Forums," Interactive Learning Environments, vol. 27, no. 5–6, pp. 1–15, 2019.

[7] J. J. Zhu et al., "Online Critical Review Classification in Response Strategy and Service Provider Rating: Algorithms from Heuristic Processing, Sentiment Analysis to Deep Learning," Journal of Business Research, vol. 129, pp. 1–12, DOI: 10.1016/j.jbusres.2020.11.007, 2021.

[8] F. A. Acheampong et al., "Transformer Models for Text-based Emotion Detection: A Review of BERT-based Approaches," Artificial Intelligence Review, vol. 54, no. 8, pp. 1–41, 2021.

[9] C. Dervenis, P. Fitsilis and O. Iatrellis, "A Review of Research on Teacher Competencies in Higher Education," Quality Assurance in Education, vol. 30, no. 2, pp. 1–15, 2022.

[10] M. Y. Salmony et al., "Leveraging Attention Layer in Improving Deep Learning Models' Performance for Sentiment Analysis," Int. J. of Information Technology (Singapore), vol. 15, no. 1, pp. 1–10, 2023.

[11] F. Sebastiani, "Machine Learning in Automated Text Categorization," ACM Computing Surveys, vol. 34, no. 1, pp. 1-47, DOI: 10.1145/505282.505283, 2002.

[12] B. Pang, L. Lee and S. Vaithyanathan, "Thumbs up? Sentiment Classification Using Machine Learning Techniques," Proc. of the 2002 Conf. on Empirical Methods in Natural Language Processing (EMNLP 2002), pp. 79-86, DOI: 10.3115/1118693.1118704, 2002.

[13] N. Dong and E. P. Xing, "Few-shot Semantic Segmentation with Prototype Learning," Proc. of Brit. Mach. Vis. Conf. (BMVC 2018), [Online], Available: http://bmvc2018.org/contents/papers/0255.pdf.

[14] W. Li et al., "Revisiting Local Descriptor Based Image-to-class Measure for Few-shot Learning," Proc.

IEEE Conf. Comput. Vis. Pattern Recognit., pp. 7260-7268, DOI: 10.1109/CVPR.2019.00743, 2019.

[15] R. Hadsell et al., "Dimensionality Reduction by Learning an Invariant Mapping," Proc. IEEE Conf. Comput. Vis. Pattern Recognit., vol. 2, pp. 1735-1742, DOI: 10.1109/CVPR.2006.100, 2006.

[16] K. He et al., "Momentum Contrast for Unsupervised Visual Representation Learning," Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 9726-9735, DOI: 10.1109/CVPR42600.2020.00975, 2020.

[17] T. Chen et al., "A Simple Framework for Contrastive Learning of Visual Representations," Proc. of the 37th Int. Conf. on Machine Learning (ICML 2020), pp. 1597-1607, Vienna, Austria, 2020.

[18] C. Li et al., "SentiPrompt: Sentiment Knowledge Enhanced Prompt-tuning for Aspect-based Sentiment Analysis," arXiv preprint, arXiv: 2109.08306, 2021.

[19] B. Liang et al., "Enhancing Aspect-based Sentiment Analysis with Supervised Contrastive Learning," Proc. Int. Conf. Inf. Knowl. Manage., pp. 3242-3247, DOI: 10.1145/3459637.3482096, 2021.

[20] J. J. Peper and L. Wang, "Generative Aspect-based Sentiment Analysis with Contrastive Learning and Expressive Structure," Proc. of Findings Assoc. Comput. Linguistics: EMNLP 2022, pp. 6086-6099, DOI: 10.18653/v1/2022.findings-emnlp.451, 2022.

[21] P. Khosla et al., "Supervised Contrastive Learning," Adv. Neural Inf. Process. Syst., vol. 33, pp. 18661-18673, 2020.

[22] Y. Wang, J. Wang, Z. Cao and A. Barati Farimani, "Molecular Contrastive Learning of Representations *via* Graph Neural Networks," Nature Machine Intelligent, vol. 4, no. 3, pp. 279-287, 2022.

[23] Z. Lin et al., "Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning," Proc. ACM Web Conf., pp. 2320-2329, DOI: 10.1145/3485447.3512104, 2022.

[24] J. Elith, J. R. Leathwick and T. Hastie, "A Working Guide to Boosted Regression Trees," Journal of Animal Ecology, vol. 77, no. 4, pp. 802-813, DOI: 10.1111/j.1365-2656.2008.01390.x, 2008.

[25] R. E. Schapire, "A Short Introduction to Boosting," Journal of the Japanese Society for Artificial Intelligence, vol. 14, no. 5, pp. 771-780, DOI: 10.1.1.112.5912, 2009.

[26] M. Kuhn and K. Johnson, Applied Predictive Modeling, New York, NY: Springer, DOI: 10.1007/978-1-4614-6849-3, 2013.

[27] P. Wu and H. Zhao, "Some Analysis and Research of the AdaBoost Algorithm," Communications in Computer and Information Science, vol. 134, pp. 1-8, DOI: 10.1007/978-3-642-18129-0_1, 2011.

[28] F. Wang et al., "Feature Learning Viewpoint of AdaBoost and a New Algorithm," IEEE Access, vol. 7, pp. 149890-149899, DOI: 10.1109/ACCESS.2019.2947359, 2019.

[29] L. Breiman, "Arcing Classifiers," Annals of Statistics, vol. 26, no. 3, pp. 801-849, 1998.

[30] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," Annals of Statistics, vol. 29, no. 5, pp. 1189-1232, DOI: 10.1214/aos/1013203451, 2001.

[31] A. Natekin and A. Knoll, "Gradient Boosting Machines, a Tutorial," Frontiers in Neurorobotics, vol. 7, DOI: 10.3389/fnbot.2013.00021, Dec. 2013.

[32] B. Zhang et al., "Health Data Driven on Continuous Blood Pressure Prediction Based on Gradient Boosting Decision Tree Algorithm," IEEE Access, vol. 7, pp. 32423-32433, 2019.

[33] J. Jiang et al., "Boosting Tree-assisted Multitask Deep Learning for Small Scientific Datasets," Journal of Chemical Information and Modeling, vol. 60, no. 3, pp. 1235-1244, 2020.

[34] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 785-794, DOI: 10.1145/2939672.2939785, 2016.

[35] Y. Li and W. Chen, "A Comparative Performance Assessment of Ensemble Learning for Credit Scoring," Mathematics, vol. 8, no. 10, p. 1756, DOI: 10.3390/math8101756, 2020.

[36] W. Liang et al., "Predicting Hard Rock Pillar Stability Using GBDT, XGBoost and LightGBM Algorithms," Mathematics, vol. 8, no. 5, p. 765, DOI: 10.3390/MATH8050765, 2020.

[37] J. Nobre et al., "Combining Principal Component Analysis, Discrete Wavelet Transform and XGBoost to Trade in the Financial Markets," Expert Systems with Applications, vol. 125, pp. 19-33, 2019.

[38] B. Zhang, Y. Zhang and X. Jiang, "Feature Selection for Global Tropospheric Ozone Prediction Based on the BO-XGBoost-RFE Algorithm," Scientific Reports, vol. 12, no. 1, 2022.

[39] A. Vaswani et al., "Attention Is All You Need," Advances in Neural Inf. Proces. Syst., vol. 30, 2017.

[40] D. Q. Nguyen and A. T. Nguyen, "PhoBERT: Pre-trained Language Models for Vietnamese," Proc. of Findings Assoc. Comput. Linguistics: EMNLP 2020, pp. 1037-1042, DOI: 10.18653/v1/2020.findings-emnlp.92, 2020.

[41] T. O. Tran and P. Le Hong, "Improving Sequence Tagging for Vietnamese Text Using Transformer-based Neural Models," Proc. of the 34th Pacific Asia Conf. Lang., Inf. Comput., pp. 13-20, 2020.

[42] N. L. Tran, D. M. Le and D. Q. Nguyen, "BARTpho: Pre-trained Sequence-to-sequence Models for Vietnamese," Proc. Interspeech, pp. 4895-4899, DOI: 10.21437/Interspeech.2022-10177, 2022.

[43] L. Phan et al., "ViT5: Pre-trained Text-to-text Transformer for Vietnamese Language Generation," Proc. NAACL-HLT Student Res. Workshop, pp. 128-135, DOI: 10.18653/v1/2022.naacl-srw.18, 2022.

[44] A. Conneau et al., "Unsupervised Cross-lingual Representation Learning at Scale," Proc. Annu. Meet. Assoc. Comput. Linguistics, pp. 8440-8451, DOI: 10.18653/v1/2020.acl-main.747, 2020.

[45] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proc. NAACL-HLT, pp. 4171-4186, [Online], Available: https://aclanthology.org/N19-1423.pdf, 2019.

[46] L. Xue et al., "mT5: A Massively Multilingual Pre-trained Text-to-text Transformer," Proc. NAACL-HLT, pp. 483-498, DOI: 10.18653/v1/2021.naacl-main.41, 2021.

[47] M. Gutmann and A. Hyvärinen, "Noise-contrastive Estimation: A New Estimation Principle for Unnormalized Statistical Models," Journal of Machine Learning Research, vol. 9, pp. 297-304, 2010.

[48] A. Mnih and K. Kavukcuoglu, "Learning Word Embeddings Efficiently with Noise-contrastive Estimation," Advances in Neural Information Processing Systems, vol. 26, pp. 1-9, 2013.

[49] K. Sohn, "Improved Deep Metric Learning with Multi-class N-pair Loss Objective," Advances in Neural Information Processing Systems (NIPS 2026), vol. 29, 2016.

[50] Z. Wu et al., "Unsupervised Feature Learning *via* Non-parametric Instance Discrimination," Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 3733-3742, DOI: 10.1109/CVPR.2018.00393, 2018.

[51] P. Sermanet et al., "Time-contrastive Networks: Self-supervised Learning from Multi-view Observation," Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, pp. 486-493, 2017.

[52] R. D. Hjelm et al., "Learning Deep Representations by Mutual Information Estimation and Maximization," arXiv preprint, arXiv: 1808.06670, 2019.

[53] Y. Tian, D. Krishnan and P. Isola, "Contrastive Multiview Coding," Lecture Notes in Computer Science, vol. 12356, pp. 776-794, DOI: 10.1007/978-3-030-58621-8_45, 2020.

[54] J. Snell, K. Swersky and R. Zemel, "Prototypical Networks for Few-shot Learning," Advances in Neural Information Processing Systems, vol. 30, 2017.

[55] E. van der Spoel et al., "Siamese Neural Networks for One-shot Image Recognition," Proc. of the 32nd Int. Conf. on Machine Learning, Lille, France, 2015.

[56] K. V. Nguyen et al., "UIT-VSFC: Vietnamese Students' Feedback Corpus for Sentiment Analysis," Proc. Int. Conf. Knowl. Syst. Eng., pp. 115-120, DOI: 10.1109/KSE.2018.8573337, 2018.

[57] Z. Ghahramani, "Probabilistic Machine Learning and Artificial Intelligence," Nature, vol. 521, no. 7553, pp. 452-459, DOI: 10.1038/nature14541, 2015.

[58] J. Snoek, H. Larochelle and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," Advances in Neural Information Processing Systems, vol. 25, 2012.

[59] Y. Xia et al., "A Boosted Decision Tree Approach Using Bayesian Hyper-parameter Optimization for Credit Scoring," Expert Systems With Applications, vol. 78, pp. 225-241, 2017.

[60] S. Tuan et al., "On Students' Sentiment Prediction Based on Deep Learning: Applied Information Literacy," SN Computer Science, vol. 5, no. 6, p. 928, 2024.

[61] R. Ahuja and S. C. Sharma, "Student Opinion Mining About Instructor Using Optimized Ensemble Machine Learning Model and Feature Fusion," SN Computer Science, vol. 5, no. 6, p. 672, 2024.

[62] D. V. Thin, D. N. Hao and N. L. Nguyen, "A Study of Vietnamese Sentiment Classification with Ensemble Pre-trained Language Models," Vietnam J. of Comp. Science, vol. 11, no. 2, pp. 137-165, 2023.

[63] X. Zhu, S. Wang, J. Lu, Y. Hao, H. Liu and X. He, "Boosting Few-shot Learning *via* Attentive Feature Regularization," arXiv preprint arXiv: 2403.17025, 2024.

[64] C. Huertas, "Gradient Boosting Trees and Large Language Models for Tabular Data Few-Shot Learning," Proc. Conf. on Computer Science and Information Systems, [Online], Available: https://www.semanticscholar.org/paper/273877899/paper/273877899, 2024.

ملخص البحث:

تتناول هذه الورقة التّحديات المرتبطة بتحليل المشاعر في السّياقات التّعليمية عبر اقتراح إطار عمل يجمع بين التّعليم قصير المدى والنّماذج القائمة على المحوّلات وتقنيات التّعزيز. حيث أنّ تحليل المشاعر يعد أمراً حاسماً لتحسين جودة التّعليم، ومن أبرز التّحديات شّح البيانات وضعف فاعلية الحوسبة. لذلك يعمل إطار العمل المقترح على تعزيز آليات الانتباه الذّاتي في المحوّلات، ويجمع بين النّماذج من خلال تقنيات التّعزيز لتحسين الأداء وإمكانية التّعميم بأقلّ قدْر من البيانات الموسومة. وجرى تطبيق إطار العمل المقترح على مجموعة بياناتٍ تحوي التّغذية الرّاجعة من الطّلبة باللّغة الفيتنامية، وحقّق نتائج مميزة في مهمّات تحليل المشاعر وتصنيف العناوين مقارنةً مع ما حققته النّماذج منفردة. وبالرغم من أنّ إطار العمل المقترح يمتلك الإمكانية لتحسين الخبرات التّعليمية إلّا إنّه يواجه بعض المحددات، مثل اعتماده على نماذج مدّربة مسبقاً وعلى تعقيد الحوسبة.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 03, September 2025.

FROM SURVEYS TO SENTIMENT: A REVIEW OF PATIENT FEEDBACK COLLECTION AND ANALYSIS METHODS

Ayushi Gupta, Anamika Gupta, Dhruv Bansal and Khushi

(Received: 15-May-2025, Revised: 21-Jul.-2025, Accepted: 4-Aug.-2025)

ABSTRACT

Patient feedback plays a crucial role in improving the quality, responsiveness and patient-centric approach of healthcare services. This paper presents a comprehensive review of both traditional and digital methods used to collect patient feedback, emphasizing their value in improving healthcare delivery, examines the tools and channels used, including surveys, interviews and multi-channel digital platforms. The review further explores sentiment-analysis techniques applied to patient feedback, focusing on how machine learning, deep learning and large language models are used to interpret and categorize unstructured text. The recent literature is systematically analyzed, with comparative tables that highlight feature-extraction methods, classification algorithms and performance metrics reported in various studies. Additionally, the paper addresses key challenges in feedback collection and sentiment analysis. Future research directions are proposed, such as automating feedback systems and incorporating patient perspectives into quality-improvement frameworks. This review is intended to assist Healthcare IT Professionals and medical Data Scientists who deal with healthcare delivery and computational analysis, whose target is to extract actionable insights from patient feedback using modern AI techniques.

KEYWORDS

Patient feedback, Sentiment analysis, Lexicon, Machine learning, Deep learning, Generative AI.

1. INTRODUCTION

Patient satisfaction is crucial for measuring the quality of healthcare services. It reflects how effective clinical care is and the broader experience of patients within the healthcare system. However, patient experiences are influenced by many different things, such as a person’s age, gender, education level and health condition. Traditionally, patient experience was viewed as a set of interactions that shape a patient’s point of view regarding care. Over time, in modern healthcare systems, the concept also includes the experiences of healthcare workers, families and the wider community. In [1], the authors stated that every interaction of a patient with healthcare-system matters, the values and behavior of the healthcare organization affect the care received by a patient, each patient’s personal feelings and background shape their views and patient experience changes throughout the entire treatment process. The authors highlighted the fact that the way healthcare workers feel and what they go through also affect the care they give to patients. The authors of [2] exhaustively reviewed 60 research papers from 1969 to 2019 to understand the factors that shape patient experiences and concluded that patient satisfaction is a complex topic and must be researched further to understand how thoughts and feelings of a patient affect his/her satisfaction. The authors of [3] developed a theory - Clinical Performance Feedback Intervention Theory (CP-FIT) to explain how patient feedback works and what makes it successful. The authors found that the feedback process involves goal setting, data collection, feedback delivery, interpretation, acceptance and behavior change. They identified 42 high-confidence factors that influence the success of feedback and concluded that feedback is most effective when it aligns with the values of healthcare professionals and results in clear and easy to implement improvements.

Feedback plays an important role in the growth and improvement of an organization. Taking feedback on a regular basis encourages an individual or an organization to engage in a culture of continuous learning and personal development. In the context of medicine, understanding patient feedback is crucial for enhancing healthcare services, as it provides insights into patient experiences and identifies

A. Gupta, Anamika Gupta (Corresponding Author), D. Bansal and Khushi are with Faculty of Computer Science, Shaheed Sukhdev College of Business Studies, Delhi, India. Emails: ayushigupta@sscbs.du.ac.in, anamikargupta@sscbsdu.ac.in, dhruv.23520@sscbs.du.ac.in and khushi.23531@sscbs.du.ac.in

areas for improvement.

Without any feedback mechanism, the quality of healthcare cannot be measured. Unstructured patient feedback full of useful information (from social media and online platforms) is growing quickly. However, it is not being used as much as it could be to improve healthcare services. Manually analyzing such large-scale data is not feasible due to time and resource constraints. The authors of [4] reviewed 19 studies that utilized natural language processing and machine-learning techniques for sentiment analysis and classification of patient feedback collected through surveys as well as social media. The selected studies employed supervised, unsupervised and semi-supervised learning methods that could categorize feedback into positive, negative or neutral sentiment and can be used for processing millions of such responses.

Figure 1 illustrates a structured workflow, used by various researchers, for classifying patient feedback into sentiments, incorporating both human annotation and artificial intelligence. AI mainly comprises of Natural Language Processing (NLP), Machine Learning (ML) and Deep Learning (DL) techniques. Initially, feedback of patients is collected through various mechanisms and stored in a database which follows pre-processing with several techniques, like Tokenization, Stemming, Lemmatization, Lowercasing …etc. to standardize the textual data. The standardized and processed textual data then undergoes two major pipelines, so that labels or sentiments can be generated for the data:

1) Traditional Machine Learning algorithms: Supervised, unsupervised, semi-supervised.
2) Large Language Models directly convert textual data and generate sentiment labels efficiently.

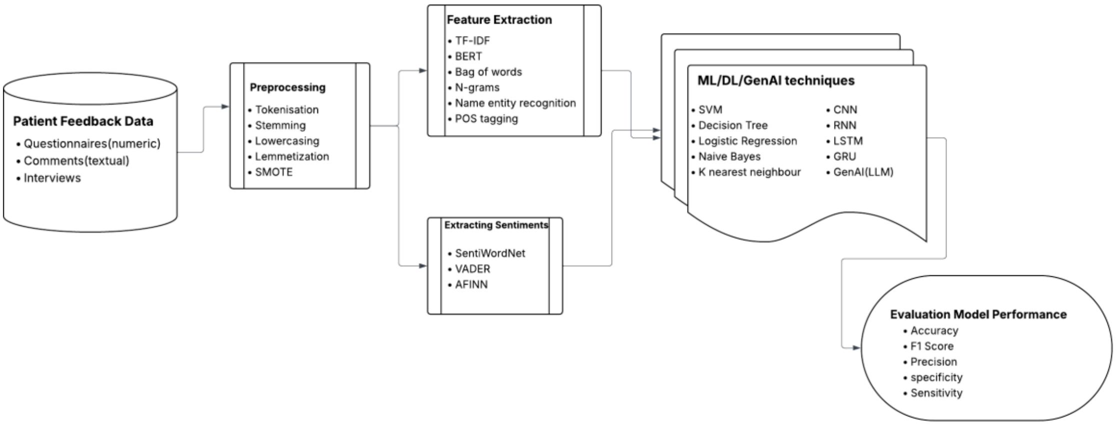


Figure 1. Methodology of sentiment analysis.

The labels are then manually checked for a sub-set of data by annotators ensuring consistency *via* Inter Annotator Agreement (IAA). When humans label data (e.g. tagging a comment as "positive", "neutral" or "negative"), their decisions can differ due to personal interpretation. IAA measures how consistently multiple human labelers agree when labeling or classifying data. The final human check ensures accurate sentiment analysis.

In this paper, our aim is to study the research space of sentiment classification in patient feedback. The initial focus is on the data-collection methods used by various researchers, followed by an analysis of the methods used for sentiment classification. Reliability and performance of sentiment-classification methods depend on the quality, accuracy and format of the collected feedback. Thus, it is crucial to study the data-collection mechanisms of the patient feedback. Various forms of inputs, such as surveys, interviews, questionnaires, and social-media content, yield different data types which will require different preprocessing and modeling strategies.

The Scopus database is chosen for literature reviews. The keywords "Patient Feedback" and ("Sentiment Analysis or Natural Language Processing or Machine Learning") are used. The documents are filtered from the last five years (2019-2024), including some studies from 2025 to focus on recent publications that reflect the latest advances and developments in this area. In this review are high-citation research papers related to feedback data-collection mechanism and sentiment-classification strategies.

Based on the motivation and scope of this review, the following research questions (RQs) are addressed.

1) **RQ1:** What are the current methods used for collecting patient feedback?
2) **RQ2:** How is sentiment analysis applied to patient feedback and what AI techniques (ML, DL, LLMs) are commonly used?
3) **RQ3:** What practical challenges arise when collecting and analyzing patient feedback, particularly at scale?

To address the above-mentioned RQs, various sections have been introduced. Section 2 details various methods that have been employed for collection and analysis of patient feedback without employing any AI techniques. Further, Section 3 provides a brief overview of how sentiment is analyzed using various ML and DL techniques and how generative AI is now being used for the same. This is followed by Section 4, which provides a review of recent studies that have performed sentiment analysis on patient feedback data. Moreover, the challenges associated with the collection and analysis of patient feedback are presented in Section 5. Lastly, Section 6 concludes the study along with future scope. This review is mainly for health-informatics researchers and IT professionals who want to develop or improve systems that can automatically analyze patient feedback. The goal is to help create tools that make it easier for healthcare teams to understand overall patient satisfaction and find areas that need improvement without reading thousands of comments manually. In addition, feedback-collection methods will help healthcare administrators and practitioners who need to implement them.

2. Understanding and Collecting Patient Feedback

This section addresses RQ1 by discussing methods for understanding and collecting patient feedback. Recent research has explored various methods for collecting, analyzing and utilizing patient feedback effectively. Some of the recent studies that focus on data collection and highlight the challenges faced during the process are mentioned in this section. In [5], the authors explored different ways to collect patient feedback and followed a participatory research approach involving patients, general practitioners (GPs), medical receptionists and an advisory group. Semi-structured interviews were conducted, where a set of open-ended questions were prepared. The interviews were analyzed using Thematic Analysis, in which the responses were categorized by attaching keywords to them. The software that was used was MAXQDA software (version 2022). It was concluded that real-time feedback is the most effective way to capture patient experiences. Also, rather than continuous collection, periodic feedback was found to be more practical and manageable.

In study [6], the authors focus on whether collecting data in real time at multiple stages of hospitalization can identify areas for improvement more effectively than traditional satisfaction surveys. This research was carried out in the Orthopedics Department of an Italian university hospital. Patients were given two different paper-based questionnaires at two time points: at hospital admission and at discharge. The data collected covered four key categories - Patient-Reported Outcomes (PROs) to measure self-rated health, Patient-Reported Experiences (PREs) to evaluate the quality of care and efficiency of services, Patient-Reported Preferences (PRPs) to capture other aspects of care that patients value and Emotional State Tracking to measure patient emotions at different stages. The authors observed that capturing patient experiences at multiple points in the hospital journey provided better insights than a single post-discharge survey.

In [7], the authors studied a digital patient feedback platform Hospitalidee, where patients may post positive or negative feedback about hospitals that have partnered with the platform. They selected all the negative feedback from the platform for a single hospital called OSTI. A two-step analysis of 134 negative feedback comments was performed to reveal common themes in patient complaints. Firstly, complaints were classified into four categories based on the service provided. Further, complaints were classified according to departments in order to target the process of quality improvement to the areas where most needed. This was followed by thematic analysis of the feedback comments in order to identify important themes. The study concluded with the statement that digital patient-feedback platforms should be actively integrated into hospital decision-making processes.

In [8], the authors explored current practices of collecting feedback and utilizing it. The authors conducted semi-structured interviews with nine participants from three different hospitals. Four types of methods were identified to collect feedback, which are given in Table 1. The challenges faced during the process are also mentioned.

Table 1. Different methods of feedback collection [8].

Methods	**Description**	**Challenges**
Structured, Official Feedback	Standardized surveys distributed through web-based platforms, paper forms or automated systems.	Response rates are low. Feedback delayed post discharge. Limited depth due to structure.
Unstructured Feedback	Informal feedback through verbal conversations, emails or suggestion boxes.	Difficult to analyze, Underreported issues, Not documented systematically
Pilot Projects using Digital Tools	Hospitals experimenting with new feedback-collection technologies, such as mobile apps and real-time patient surveys.	Not widely implemented. Requires staff training. Cost and infrastructure barrier.
Occasional Studies and Research Projects	One-time research initiatives conducted by hospital staff, students or external organizations to assess patient experience.	Lack of continuity. Not integrated into daily operations. Results take time.

A study carried out in three large hospitals in Brazil is described in [9]. Nine semi-structured interviews were conducted and hospital documents, such as feedback forms, action plans and reports, were also analyzed. NVivo 11 software was used to organize and analyze the information. It was found that hospitals use structured quality-improvement (QI) tools to analyze patient feedback and make meaningful changes. Some of such tools are:

- Plan-Do-Check-Action: Identify a problem based on patient feedback, implement a small change, measure the impact and if successful, apply the change hospital-wide.
- Ishikawa (Fishbone) Diagram: A visual tool to identify root causes of a problem by categorizing potential reasons.
- Pareto Analysis (80/20 Rule): It follows the 80/20 rule, meaning, 80% of patient complaints come from 20% of the problems, fixing that 20% can solve most issues.

The authors of [10] focused on creating simple and short questionnaires suitable for hospital patients with varying literacy levels. The patient experience monitor had two versions that were adult inpatient (14 items) and adult outpatient (15 items), both of them included key aspects, like emotional support, waiting time, privacy, clarity of information, communication and family involvement. From this study, it was found that even patients with low literacy found patient experience monitor easy to understand. The short format improved response rate.

While feedback collection is an important step in improving healthcare services, it becomes valuable when it is interpreted. Most patient responses are in unstructured formats, like free-text surveys, interviews or online reviews, as seen above and contain implicit information that is not immediately assessable. Manual review of such comments is resource-intensive and inconsistent. This is where sentiment classification becomes important. Sentiment classification helps reveal the underlying emotional tone of patient comments, whether they are satisfied, frustrated, in fear or express gratitude. By categorizing feedback into sentiment, such as positive, negative or neutral, healthcare providers can identify problem areas more efficiently. The techniques used for sentiment analysis are presented in the next section. Table 2 describes the patient-feedback datasets that have been collected and analyzed further to derive useful insights.

3. Sentiment Analysis Techniques

Sentiment is an opinion influenced by emotions. Automating the extraction of sentiments in unstructured data, such as reviews, comments or feedback, is an area of study under Natural-language Processing. Its objective is to automate extraction and interpretation of sentiments or data from text, providing insights into public sentiment, customer satisfaction and market dynamics.

Due to digitization of processes and the increase in the use of social media, the amount of reviews or feedback is enormous, making it impossible to process them manually. Therefore, there is a growing need for the use of AI-driven approaches to identify and extract the sentiment.

Table 2. Summary of patient-feedback datasets used in the reviewed studies.

Ref.	Data-collection Period	Dataset Description	Record Type	Open Source
[6] (2021)	January-February 2019	Longitudinal survey: preferences, experience, outcomes at admission/discharge	Open-ended questions answered by 254 patients	Available upon request
[11] (2020)	January 2008-October 2019	Synthesized findings from studies on patient feedback and review of interventions	20 studies having patient feedback (qualitative & quantitative)	Available upon request https: //shorturl.at/ z4cxg (supplementary data)
[12] (2024)	2018-2021	Norwegian national patient-experience surveys conducted by the Norwegian Institute of Public Health (NIPH)	2250 patient comments	No
[13] (2020)	January 2018-January 2019	Patient surveys data collected at Geisinger Holy Spirit Hospital covering various aspects of care and labeled by sentiment	2830 records of un-structured free-text comments	No
[14] (2020)	2016-2020	Three survey questions with binary responses related to respect received, clarity of explanation and attentive listening	3134 patient responses to survey Questions	No
[15] (2021)	-	Patient reviews for specific medications along with a 10-star rating	232 K free-text drug reviews	https://surl.li/ wjvtwk
[5] (2025)	-	Qualitative study exploring patient-feedback methods for e-Health in general practice	Interview transcripts of 13 patients, 8 GPs, 2 receptionists	No
[16] (2023)	-	Cancer-patient stories	Study 1-14, 391 random posts, study 2-30,037 posts	https://www. cancerconnection ca/s/ https://surl.li/uirjeq
[17] (2022)	January 2017-July 2017	Friends and family test (FFT) free-text, Patient feedback	69,285 responses	No
[10] (2020)	-	Questionnaires, interviews, pilot study	28 interviews, pilot study and surveys	https://pmc.ncbi.nlm.nih.gov/articles/PMC7725101/table/t0002/
[18] (2024)	-	Patient & family-member discussion posts on a medical forum	12,103 posts of patient narratives	https://patient.info/forums
[7] (2023)	2018	Negative feedback data from a digital platform of one hospital	Analysis of 134 reviews.	No
[19] (2022)	-	Five questions based on information provided, personal approach, collaboration among healthcare professionals organization of care and general feedback	534 responses of open-ended questionnaire	No
[20] (2021)	2019-2023	Classifying the complaint records using ML and NLP	1465 records having different complaints describing communication	No
[21] (2025)	January 2014-December 2014	Analyzed sentiment in patient comments using natural-language processing	1117 comments and ratings from 1 (worst) to 5 (best)	https://surl.li/ zcxygz

Due to digitization of processes and the increase in the use of social media, the amount of reviews or feedback is enormous, making it impossible to process them manually. Therefore, there is a growing need for the use of AI-driven approaches to identify and extract the sentiment. Recent advancements in artificial intelligence, machine learning, deep learning and generative AI, particularly large language models (LLMs), have greatly enhanced the precision and scalability of sentiment-analysis systems, establishing sentiment analysis as a crucial tool for examining extensive unstructured data. Sentiment analysis traditionally classifies text into positive, negative or neutral categories. However, advances in the field have led to the identification of nuanced sentiments, such as anger, joy, fear, toxicity, sadness and surprise.

Techniques for Extraction of Sentiment

In recent years, multiple strategies have emerged to improve the precision and scalability of sentiment classification. Conventional methods, such as the lexical-based approach, use sentiment dictionaries to assign polarity scores to individual words. Meanwhile, machine-learning methods rely on labeled datasets to train models that can identify sentiment patterns. In recent years, large language models (LLMs) have revolutionized the domain by comprehending complex linguistic nuances and context on an unprecedented scale. This transition from rule-based methods to data-driven and neural approaches highlights the evolving landscape of sentiment analysis, offering a range of strategies to address the various challenges in text analysis.

Before applying any sentiment-analysis technique, pre-processing of the text needs to be carried out. Some of the text pre-processing techniques are listed below:

1) Data cleaning - removing/handling emojis, URLs, HTML Tags, stop words, punctuation marks, spell checking, normalization, number removal, and converting into lowercase are some of the common data-cleaning techniques
2) Tokenization breaks down text into smaller units called tokens. The tokens can be a single character, word, phrase, sentence, paragraph, …etc.
3) Stemming is a process to find the root of a word by removing suffixes.
4) Lemmatization is a process that considers the context and part of speech to reduce words to their base forms, called lemmas.

Further, the techniques for classification of text into various sentiments are classified as below:

1) Lexicon-based Approach

The lexicon-based approach to sentiment analysis relies on dictionaries of words that are pre-assigned sentiment values, typically categorized as positive, negative or neutral. This method estimates the overall sentiment by summing the sentiment scores of individual words within a text. Its simplicity and transparency make it a popular choice, especially for domains where interpretability is critical or when the labeled data for training machine-learning models is scarce. Tools, such as SentiWordNet [22], VADER [23] and AFINN [24], are widely used in research and industry.

2) Machine Learning-based Approaches

Machine learning (ML)-based approaches have transformed sentiment analysis by moving beyond simple keyword matching to more sophisticated algorithms that can automatically learn patterns from data. These models do not require pre-defined lexicons and are capable of handling larger datasets and more complex language patterns. The key strength of machine learning approaches lies in their ability to generalize from data and to adapt across different domains, making them highly effective for sentiment analysis in areas like social media, product reviews and customer feedback [25]. Supervised machine learning is a prevalent approach in sentiment analysis, where models are trained on labeled datasets to classify text as positive, negative or neutral. This process generally involves data pre-processing, feature extraction and model training.

Feature Extraction

Feature extraction is crucial in converting text data into numerical vectors that the machine-learning model can process. Common methods for feature extraction include Bag-of-Words [26], TF-IDF [27], Word Embeddings [28]-[29]. Bag-of-Words is a simple and easy method which represents text by counting word frequency. Context and semantic meaning are lost in this process. TF-IDF weighs terms by their importance across documents and highlights rare, but important, words. Though computationally expensive, the technique is widely used in many text-mining applications. Word Embeddings (Word2Vec, GloVe) map words to continuous vector space, capturing semantic meaning, context and word relationships.

Model Training

Model training involves feeding the features into a machine-learning algorithm, which learns to predict the sentiment label based on the training data. Some of the most commonly used algorithms for sentiment classification include:

- **Linear Regression**: A simple model for prediction of continuous outcome based on a linear combination of input features [30].
- **Decision Tree**: A tree-based model that chooses the feature as a node of the tree based on metrics, like Gini-index and Entropy [31].
- **Naive Bayes**: Simple and effective for high-dimensional data [32].
- **Support Vector Machines (SVMs)**: this technique finds optimal hyper-planes for classification, performing well in high-dimensional spaces [33].
- **Logistic Regression**: A linear model commonly used for binary classification, such as predicting whether a review is positive or negative [34].
- **K Nearest Neighbor**: A lazy learner technique that does not learn a model and matches the unseen tuple at the time of prediction. Classification of the sample is based on the majority label among its k nearest neighbors. [35].
- **Random Forest**: Ensemble method that combines multiple decision trees. Prediction is based on the majority voting of the output of all models [36].

3) Deep Learning-based Approaches

Building upon the foundation laid by traditional machine-learning approaches, deep learning has emerged as a transformative force in sentiment analysis. While traditional models rely heavily on feature engineering and handcrafted rules, deep-learning models automatically learn representations from data, capturing complex linguistic patterns and contextual information. This sub-section highlights the contributions of CNNs, RNNs, LSTMs and GRUs, illustrating the transformative impact of deep learning in extracting sentiment from textual data. Convolutional Neural Networks [57] are a fast and high-performance technique that applies convolutional filters to extract n-gram features from text. Recurrent Neural Networks (RNNs) represent a slow, moderately performing technique that processes sequential data by maintaining hidden states, especially suitable for time-series data. Long Short- Term Memory (LSTM) deals with memory cells for long-term dependencies, suitable for long text, emotion recognition, speech processing. Gated Recurrent Units (GRUs) constitute a technique that reduces the complexities of LSTM by combining gates, making it suitable for text classification and machine translation.

4) Generative AI-based Approaches

In recent years, the advent of Generative AI (GenAI) and Large Language Models (LLMs) has significantly transformed the landscape of sentiment analysis. Unlike traditional machine learning and deep-learning approaches that require extensive labeled data and task-specific architectures, LLMs leverage large-scale pre-training on diverse datasets, enabling them to generalize across multiple tasks, including sentiment classification, with minimal fine-tuning. Large Language Models, such as OpenAI’s GPT series, Google’s BERT and Meta’s LLaMA, have set new benchmarks in natural-language understanding (NLU) and generation [37]. Their transformer-based architecture allows them to handle long-range dependencies, outperforming traditional recurrent neural networks (RNNs) and convolutional neural networks (CNNs) in various NLP tasks [38].

Transformer Architecture, the Backbone of LLMs: The transformative power of LLMs lies in the underlying transformer architecture, introduced by [38]. This architecture is based on the self-attention mechanism, which enables models to weigh the significance of different words in a sentence, regardless of their position. Unlike RNNs, which process sequences step by step, transformers process entire sequences simultaneously, drastically improving efficiency and scalability. This parallelization allows transformers to model long-range dependencies more effectively, which is critical for capturing complex sentiment patterns in lengthy reviews or documents.

The self-attention mechanism facilitates context-aware sentiment analysis by dynamically adjusting attention to relevant words. For example, in a sentence like "The movie was surprisingly good despite its slow start," the transformer architecture can attribute higher attention weights to "surprisingly good," correctly identifying the overall positive sentiment.

Zero-shot, Few-shot and Fine-tuning Approaches: LLMs have the capability of classifying sentiments based on the prompts given. Various types of prompts, such as zero-shot and few-shot can be used for learning. For example, models such as GPT-3 can classify sentiments even without direct training by utilizing prompt engineering techniques. By presenting the model with instances of positive, negative and neutral sentiments, researchers can steer the model toward producing precise predictions [39]. This versatility minimizes the necessity for labeled datasets and greatly speeds up the implementation in practical scenarios. Further, fine-tuning BERT on social-media datasets having informal and noisy data improves the sentiment-classification accuracy [40] and RoBERTa, a variant of BERT, optimizes the pertaining techniques and works on larger datasets [41].

4. Sentiment Analysis on Patient Feedback

This section addresses RQ2: How is sentiment analysis applied to patient feedback and what AI techniques (ML, DL, LLMs) are commonly used. The reviewed literature has been organized by approach type — ML, DL and LLMs. The feature-extraction and classification techniques employed in the reviewed studies are presented in Tables 3 and 4. Table 3 outlines the ML and DL approaches used for feedback analysis, while Table 4 summarizes the techniques applied in LLMs, respectively. The tables also give the performance achieved by different techniques. The following observations can be made from Table 2:

1) Approximately 43% of the datasets used in the reviewed studies were unstructured, while about 29% were structured and 29% were based on survey responses.
2) Majority of the studies categorized the sentiments as positive, negative and neutral. Maehlum et al. [12] used four sentiment categories - positive, negative, neutral and mixed, where mixed indicates sentences containing both positive and negative polarity. Similarly, Cho et al. [49] also defined positive aspects as care and kind and negative aspects as pain and rude.
3) Data cleaning was also observed to be an important part of all studies to improve model performance. Moreover, text cleaning and pre-processing techniques, such as tokenization, lemmatization, stop-word removal, stemming and lowercasing have been utilized in majority of the studies.

The bar chart in Figure 2 represents the different feature extraction techniques that have been used in the reviewed studies along with the study count. It can be observed that TF-IDF is the most widely used feature extraction technique in analyzing patient feedback data.

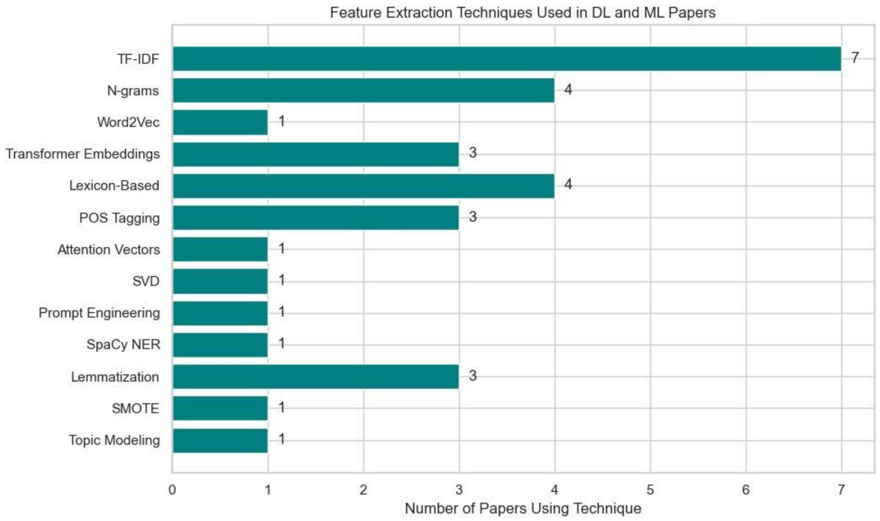


Figure 2. Feature-extraction techniques used in the analysis.

While Tables 3 and 4 summarize a wide range of studies applying various NLP techniques to patient

Table 3. Summary of NLP, ML and DL techniques used in patient-experience analysis.

Ref.	Feature-extraction Techniq.	Classification Techniques	Performance Metrics
[4]	TF-IDF	Supervised (Support Vector Machine (SVM), Naive Bayes (NB)), Unsupervised (Linear Discriminant Analysis (LDA), Factorial LDA)	Precision up to 88%; SVM accuracy 72%)
[13]	N-gram, Bigrams, Part-of-Speech (POS) Tagging, Word Frequency, Word Clouds	Artificial Neural Networks (ANN - Keras-Sequential model with dense and dropout layers)	Precision-0.83, Recall-0.82, F1-0.82, Support-103 sample
[20]	Word level TF-IDF, N-gram level TF-IDF(n=2)	SVM, Multifactor Logistic Regression (LR), Multinomial NB	Accuracy (up to 0.91), F1-Score, Precision, Recall, AUC (up to 0.94)
[19]	TF-IDF, N-gram	Finetuned Multilingual Bert, NMF for topic modeling	F1-Score (Positive: 0.97, Negative: 0.63), Machine-Human Topic Match: 90%, Topic Representativeness: 80.9
[21]	LIWC-22, Meaning Extraction Method (MEM), Principal Component Analysis (PCA)	Multivariable Linear Regression	Not given
[17]	Bag of words, tri-gram analysis.	Decision Tree (DT), Random Forest (RF), SVM, K-Nearest Neighbour (KNN), NB and Gradient Boosted Trees (GB)	SVM F1-score 94%
[42]	TF-IDF, Bag of Words, Name entity recognition, Word embedding	Transformer models (RoBERTa) and CNNs	RoBERTa F1-Score: Neurology (1.0), Combined datasets (0.995). CNN: 0.760.
[43]	Name Entity Recognition, TF-IDF, BERT	RF, GB models	85–90%
[16]	TF-IDF, Topic modeling	Topic classification, LDA.	87%
[14]	BERT, Bag of Words	RF, LR, DT and Social Network Analysis	RF: 87.6% (courtesy), 81.9% (clarity, listening).
[44]	Tokenization, lemmatization, Domain-specific lexicons	SVM, NB, DT	F1-score: 60%
[45]	TF-IDF, POS Tagging, BERT	Machine learning models for sentiment categorization	78.2–87%
[46]	Bag of words, TF-IDF	Sentistrength (for sentiment analysis), LDA	89.3% (general), 92.6% (healthcare), 90.8% (life
[47]	Word count, TF-IDF, Boolean features	NB, Multinomial NB, SVM, LR, RF	81% (cleanliness), 84% (dignity), 89% (recommendation).
[48]	N-grams, SNOMED CT, BERT	Rule-based NLP, SVM	AUC: 0.997; Sensitivity: 88%; Specificity: 96%.
[21]	Topic modeling	Topic modeling to identify themes (e.g., communication, logistics).	78.5%–87% across different aspects of care
[49]	TF-IDF, Sentiment lexicons, bag of words	LR, t-test/ANOVA	78.5%–87% across different aspects of care
[13]	TF-IDF from lemmatized, synonym-standardized text	Sequential Deep Neural Network (Keras); 3 dense layers with dropout	Accuracy peaked at epoch 35; ReLU + Softmax
[15]	TF-IDF, Bi-grams, Lexicon-based (Bing) SMOTE	Artificial Neural Network (ANN), SVM, Logistic Regression	SVM: Acc. = 0.720, AUC = 0.725
[50]	TF-IDF vectorization, 1–4 grams, Harvard emotional dictionary	N-gram Deep Learning model; also compared with RF, NB, Linear Regression	N-Gram model: Acc. = 89.4%
[51]	UMLS mapping, Symptom dictionaries, Term frequency, Lexicon usage, Clustering, Patient-authored symptom terms	Rule-based NLP, Machine Learning (SVM, RNN, Logistic Regression), Text Mining	F1-scores up to 90%, Precision/Recall/AUC (e.g., AUC = 0.899); task-dependent metrics like Jaccard Index for symptom clusters
[52]	Concept extraction, Topic modeling (LDA), Word embeddings; NLP pipelines using MetaMap, cTAKES,	Hybrid of SVM, CRFs, Deep Neural Networks; MetaMap, cTAKES	Accuracy: up to 92.68%; F1-scores: 0.54–0.83; AUC: up to 0.94; Task-specific benchmarks like SemEval

feedback, a few studies are discussed in greater detail here. These were chosen, because they use new or advanced methods, apply powerful AI models, like LLMs, work well on large-scale real-world data or combine human insight with AI tools. These examples will help us better understand the latest trends to use sentiment analysis in healthcare.

Table 4. Studies utilizing large language models (LLMs) for patient-experience analysis.

Ref.	Architecture	Embedding / Features	Performance Metrics
[12]	ChatNorT5 (T5-based, 808M), NorMistral (Mistral 7B-based)	Transformer embeddings; instruction-tuned LLMs	F1: ChatNorT5 = 42.4% (4-class), 89.3% (2-class); NorMistral = 39.9% (4-class), 89.1% (2-class)
[53]	Llama2-70B, Mistral-7B, GPT- 3.5; Chatbot + Dialogue Management System	LLM embeddings, Prompt Engineering, User Profile memory, SVD, Reddit/Chatbot transcripts	Llama2 > GPT-3.5 in 40–44% of summarization tasks; GPT-4 used as evaluator; promising pilot results for chatbot system
[18]	DeBERTa, BERT, Bi-LSTM, LSTM, ChatGPT-3.5 (few- shot)	Word embeddings, Transformer-based ABSA (DeBERTa)	ChatGPT-3.5: F1 =90%; ABSA-BERT: F1 = 73.2%; BiLSTM: Acc.= 85%; Manual eval.: Cohen's Kappa = 0.87

4.1 Studies Employing ML/DL for Analyzing Sentiment in Patient Feedback

Several studies applied traditional ML methods to classify patient feedback into positive, negative and neutral sentiment categories. Feature engineering techniques, like TF-IDF, n-grams, POS tagging, have been applied followed by supervised classification algorithms, such as SVM, Naive Bayes or Logistic Regression.

The authors of [20] collected 1817 Chinese complaint cases from two hospitals from 2015 to 2019 and divided them into four categories. First, the Chinese text was translated to English using ChatGPT-3.5 and tokenization was carried out using jieba (Chinese NLP library). The features were then extracted, followed by balancing the dataset using Synthetic Minority Over-sampling Technique (SMOTE). ML techniques were then employed for classification purposes, out of which SVM gave the best accuracy value. Another study, [17], worked on patient feedback collected through the Friends and Family Test (FFT) system in the UK's National Health Service (NHS). Nearly 10% of the responses (6,900 comments) were manually labeled by an annotation team to create a training dataset for model training and themes and sentiments were derived for each comment. The study used 10 core themes adapted from the NHS Patient Experience Framework. Six ML models were then trained using the annotated dataset to automatically classify the remaining 90% of the responses, with SVM achieving the best performance. In 2021, the authors of [15] demonstrated sentiment analysis, topic modeling and text classification on the publicly available drug-review dataset. Relying on the Bing sentiment lexicon where each word is tagged as either positive or negative, sentiment analysis was performed on reviews for four specific drugs (two of which had higher positive sentiments). Further, they grouped the text data by topic (topic modeling) and manually labeled each topic by looking at the most frequent words associated with it. They also assigned good and bad labels to the reviews based on star ratings, handled data imbalance through SMOTE and utilized ML models to classify the reviews.

In 2023, the authors of [16] combined design thinking with ML to make the process of understanding and analyzing patient experience in a more accurate, detailed and useful manner. In the first study, the authors used supervised ML to analyze 14,391 cancer forum posts. They also applied association rule mining to uncover relationships between topics, which helped in refining an initial journey map. In the second study, they used unsupervised learning to analyze 30,037 online patient stories, to identify hidden themes and map them to different stages of care. This was followed by designers looking at the most common topics found and labeling them to show what patients need and how they feel at different points in their care. This mix of computer analysis and human insight helped create detailed maps of the patient journey.

A few studies also worked on developing recommendation systems and automated analysis tools. The authors of [50] analyzed patient-written drug reviews obtained from Kaggle, to recommend the most suitable medicine for a health condition. After pre-processing the dataset with TF-IDF and N-Gram models, the reviews were classified as positive or negative using ML models. The sentiment analysis was carried out by using 1-gram to 4-gram models, with the 4-gram model achieving best results.

They further ranked the drugs by average sentiment score and built a drug-recommendation system based on it. However, the original dataset did not have a dedicated sentiment column and how the sentiments were computed for model training was not mentioned by the authors in the study. Further, the authors of [19] developed a new tool called AI-PREM, which combined an open-ended patient-experience questionnaire, an NLP pipeline to automatically analyze responses and a visual interface for easily understanding the results. Patients' responses were pre-processed and sentiment analysis was conducted using a fine-tuned multi-lingual BERT model to classify the feedback. For topic modeling, the authors used Non-negative Matrix Factorization (NMF) to group similar responses based on themes, with separate models created for each question and sentiment. An interactive three-layer dashboard was developed to visualize and interpret the results.

Researchers have also integrated Social Network Analysis (SNA) and DL techniques along with ML to enhance the analysis of patient feedback. In [13], the authors analyzed unstructured patient feedback using NLP and DL. First, free-text comments were pre-processed followed by exploratory data analysis using word clouds, frequency distributions and part-of-speech tagging to identify common themes and key concerns. The authors utilized a neural network model with a sequential architecture with dense and dropout layers to classify sentiments as positive, negative or neutral. This model was especially used to separate and label comments that had both positive and negative parts, by looking at each sentence one by one. This helped get a more detailed understanding of the feedback. Another study, [14], combined ML and Social Network Analysis (SNA) to develop a system that can both predict negative patient experiences and identify key doctors who have a direct impact on those experiences. The authors classified the responses into two classes - best response and all other responses. They utilized a variety of ML classifiers to predict negative patient experiences. Further, they utilized SNA (degree, betweenness and closeness centralities) to identify influential doctors who can help improve the overall patient experience.

4.2 Studies Employing LLMs for Analyzing Sentiment in Patient Feedback

A piece of research [12] in 2024 focused on Norwegian-language feedback from patients and developed a sentiment-labeled dataset from free-text patient-survey comments. The authors used two LLM architectures with zero and few-shot learning (to guide the model with no or minimal training examples) and achieved good classification results for binary labels - positive and negative. They used 48 custom prompts based on English datasets, translated into Norwegian. However, the models failed in the case of 4-class classification achieving less than 50% accuracy values. The study highlighted the importance of manual annotation to achieve good results. Another research, [18], collected patient posts from a health forum and identified aspects that patients talk about and checked whether people spoke positively, negatively or in a neutral way using DeBERTa neural network and ChatGPT-3.5. It was found that ChatGPT performed the best in understanding detailed feedback with few-shot learning (where a few examples are provided to the model in the prompt).

5. CHALLENGES

This section addresses RQ3 by discussing the key challenges related to the collection and analysis of patient feedback. Collecting and analyzing patient feedback is essential for improving healthcare quality. However, it comes with several practical and systemic challenges that must be addressed for these systems to be effective. First, the terms "patient satisfaction" and "patient experience" create confusion, since they are used interchangeably [54]. While satisfaction is subjective and based on expectations of an individual, experience is more objective and measures what actually happened during care. Hence, satisfaction may not accurately capture the quality of care. For example, two patients undergo the same surgery with identical medical outcomes. Patient A expected a painful recovery, but found it manageable leading to high satisfaction. Patient B expected a quick, painless recovery, but experienced discomfort leading to low satisfaction.

There can be many reasons for patients not giving feedback - low literacy in health, socio-economic inequalities, fear of being treated unfairly because of giving negative feedback and lack of trust in healthcare systems. In low-income and middle-income countries, many patients are unaware that feedback mechanisms even exist [55]. Moreover, there is an absence of clear guidelines and health workers also take feedback mechanisms as a threat rather than a scope to improve. They are reluctant to receive patient feedback fearing that negative feedback may harm their professional repute. Some institutions do not even integrate patient feedback into strategic planning effectively, since negative feedback over-shadows positive comments. Bias and reliability issues also arise while feedback is being collected, since it is influenced by the emotions and health conditions of the patients. Further, patients, being both a care recipient and a feedback provider, feel conflicted [56]. Also, healthcare professionals, being both experts and learners, are hesitant to invite feedback. Hence, there is an imbalance of power where patients may hesitate to provide negative feedback and professionals may feel vulnerable when receiving criticism. There is a lack of structured methods for engaging in feedback dialogues. Patients prefer verbal feedback for positive experiences, but written feedback when dissatisfied. Even after the feedback is collected, there are hardly any mechanisms for following it up and even if actions are taken, patients are hardly informed about them. Hence, participation is decreased over time.

Analyzing the collected feedback comments to get useful insights for decision-making can be expensive and time-consuming if carried out manually. Utilizing ML and DL techniques to process and analyze such unstructured data also requires careful intervention. These models should be carefully selected and validated, especially in healthcare contexts, where misclassification can have serious consequences. Further, LLMs like LLaMA and GPT are also very expensive to train and require significant resources.

6. Conclusion and Future Directions

This study has provided a thorough review of current methods for collecting and analyzing patient feed- back in healthcare. It examined both traditional tools, such as open-ended questionnaires and interviews and emerging digital platforms that support scalable and timely feedback collection. A particular emphasis was placed on sentiment analysis techniques, showcasing the application of machine learning (ML), deep learning (DL) and large language models (LLMs) to interpret unstructured patient responses. The review synthesized findings from recent studies, detailing the datasets used, feature-extraction strategies, classification approaches and performance outcomes. Furthermore, challenges and limitations associated with data collection, processing and analysis were discussed. By aligning sentiment analysis techniques with real-world feedback systems, this review supports the development of automated and patient-centered solutions that can enhance service quality and enable continuous healthcare improvement.

In future work, feedback systems should be designed to function across multiple platforms, such as mobile apps, websites, SMS, in-person interviews and voice input, to increase participation from diverse patient populations. Also, family members should be allowed to submit feedback on behalf of elderly or critically ill patients, to expand the scope of feedback collection. The process of feedback collection and analysis should be automated using NLP and AI tools to reduce manual efforts and analyze large amounts of data. Moreover, there is a lack of publicly available patient feedback datasets. Future work should focus on curating and sharing large-scale, representative datasets to improve the generalizability and robustness of sentiment-analysis models, across different demographics, languages and care settings. Lastly, feedback gathered must be fed directly into quality-improvement programs, performance evaluations and strategic planning.

References

[1] J. A. Wolf et al., "Reexamining “Defining Patient Experience”: The Human Experience in Healthcare," Patient Experience Journal, vol. 8, no. 1, pp. 16– 29, 2021.

[2] R. Kalaja, "Determinants of Patient Satisfaction with Health Care: A Literature Review," European Journal of Natural Sciences and Medicine, vol. 6, pp. 43–54, May 2023.

[3] B. Brown et al., "Clinical Performance Feedback Intervention Theory (CP-FIT): A New Theory for Designing, Implementing and Evaluating Feedback in Health Care Based on a Systematic Review and Meta-synthesis of Qualitative Research," Implementation Science, vol. 14, no. 1, pp. 1–20, 2019.

[4] M. Khanbhai et al., "Applying Natural Language Processing and Machine Learning Techniques to Patient Experience Feedback: A Systematic Review," BMJ Health Care Informatics, vol. 28, p. e100262, March 2021.

[5] M. Nasori, M. Mak-van der Vossen, M. Holtrop and J. Bont, "Exploring Effective Patient Feedback Methods for e-Health in General Practice," BMC Primary Care, vol. 26, p. 40, 2025.

[6] R. Gualandi et al., "What Does the Patient Have to Say? Valuing the Patient Experience to Improve the Patient Journey," BMC Health Services Research, vol. 21, pp. 1–12, 2021.

[7] S. M. Bez, I. Georgescu and M. S. Farazi, "TripAdvisor of Healthcare: Opportunities for Value Creation through Patient Feedback Platforms," Technovation, vol. 121, p. 102625, 2023.

[8] J. Kaipio et al., "Improving Hospital Services Based on Patient Experience Data: Current Feedback Practices and Future Opportunities," Studies in Health Technology and Informatics, vol. 247, pp. 266–270, 2018.

[9] S. Berger, A. M. Saut and F. T. Berssaneti, "Using Patient Feedback to Drive Quality Improvement in Hospitals: A Qualitative Study," BMJ Open, vol. 10, no. 10, p. e037641, 2020.

[10] C. M. Bastemeijer et al., "Patient Experience Monitor (PEM): The Development of New Short-form Picker Experience Questionnaires for Hospital Patients with a Wide Range of Literacy Levels," Patient Related Outcome Measures, vol. 11, pp. 221–230, 2020.

[11] E. Wong, F. Mavondo and J. Fisher, "Patient Feedback to Improve Quality of Patient-centred Care in Public Hospitals: A Systematic Review of the Evidence," BMC Health Services Research, vol. 20, no. 1, p. 530, 2020.

[12] P. Mæhlum et al., "It‘s Difficult to be Neutral–human and LLM-based Sentiment Annotation of Patient Comments," Proc. of the 1st Workshop on Patient-oriented Language Processing (CL4Health), pp. 8–19, Torino, Italia, May 2024.

[13] K. Nawab, G. Ramsey and R. Schreiber, "Natural Language Processing to Extract Meaningful Information from Patient Experience Feedback," Applied Clinical Informatics, vol. 11, pp. 242–252, March 2020.

[14] V. Bari et al., "An Approach to Predicting Patient Experience through Machine Learning and Social Network Analysis," J. of the American Medical Informatics Association, vol. 27, pp. 1834–1843, 2020.

[15] C. J. Harrison and C. J. Sidey-Gibbons, "Machine Learning in Medicine: A Practical Introduction to Natural Language Processing," BMC Medical Research Methodology, vol. 21, no. 1, p. 158, 2021.

[16] J. Jung, K.-H. Kim, T. Peters, D. Snelders and M. Kleinsmann, "Advancing Design Approaches through Data-driven Techniques: Patient Community Journey Mapping Using Online Stories and Machine Learning," Int. Journal of Design, vol. 17, no. 2, 2023.

[17] M. Khanbhai et al., "Using Natural Language Processing to Understand, Facilitate and Maintain Continuity in Patient Experience across Transitions of Care," Int. Journal of Medical Informatics, vol. 157, p. 104642, 2022.

[18] O. S. Alkhnbashi, R. Mohammad and M. Hammoudeh, "Aspect-based Sentiment Analysis of Patient Feedback Using Large Language Models," Big Data and Cognitive Computing, vol. 8, no. 12, 2024.

[19] M. M. van Buchem et al., "Analyzing Patient Experiences Using Natural Language Processing: Development and Validation of the Artificial Intelligence Patient Reported Experience Measure (AI-PREM)," BMC Medical Informatics and Decision Making, vol. 22, no. 1, p. 183, 2022.

[20] X. Li, Q. Shu, C. Kong, J. Wang, G. Li, X. Fang, X. Lou and G. Yu, "An Intelligent System for Classifying Patient Complaints Using Machine Learning and Natural Language Processing: Development and Validation Study," Journal of Medical Internet Research, vol. 27, p. e55721, 2025.

[21] A. Azarpey et al., "Natural Language Processing of Sentiments Identified in Patient Comments Associated with Less than Top-rated Care," J. of Patient Experi., vol. 12, p. 23743735251323677, 2025.

[22] A. Esuli and F. Sebastiani, "SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining," Proc. of the 5th Int. Conf. on Language Resources and Evaluation (LREC), pp. 417–422, Genoa, Italy, 2006.

[23] C. J. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," Proc. of the 8th Int. AAAI Conf. on Weblogs and Social Media, vol. 8, no. 1, Association for the Advancement of Artificial Intelligence, 2014.

[24] F. Nielsen, "Afinn: A New Word List for Sentiment Analysis," [Online], Available at: https://github.com/fnielsen/afinn, 2011.

[25] M. Wankhade, A. C. S. Rao and C. Kulkarni, "A Survey on Sentiment Analysis Methods, Applications and Challenges," Artificial Intelligence Review, vol. 55, no. 8, pp. 5731–5780, 2022.

[26] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," Proc. of the European Conf. on Machine Learning (ECML), pp. 137–142, Springer, 1998.

[27] G. Salton and C. Buckley, "Term-weighting Approaches in Automatic Text Retrieval," Information Processing & Management, vol. 24, no. 5, pp. 513–523, 1988.

[28] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv preprint, arXiv: 1301.3781, 2013.

[29] J. Pennington et al., "GloVe: Global Vectors for Word Representation," Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543, ACL, 2014.

[30] G. A. Seber and A. J. Lee, Linear Regression Analysis, John Wiley & Sons, 2nd Edn., 2012.

[31] J. R. Quinlan, Induction of Decision Trees, vol. 1, Springer, 1986.

[32] I. Rish, "An Empirical Study of the Naive Bayes Classifier," IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, vol. 3, no. 22, pp. 41–46, 2001.

[33] C. Cortes and V. Vapnik, "Support-vector Networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.

[34] D. W. Hosmer, S. Lemeshow and R. X. Sturdivant, Applied Logistic Regression, John Wiley & Sons, 3rd Edn., 2013.

[35] T. Cover and P. Hart, "Nearest Neighbor Pattern Classification," IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21–27, 1967.

[36] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[37] T. B. Brown et al., "Language Models Are Few-shot Learners," Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901, 2020.

[38] A. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems, vol. 30, Curran Associates, Inc., 2017.

[39] Y. Wu and G. Hu, "Exploring Prompt Engineering with GPT Language Models for Document-level Machine Translation: Insights and Findings," Proc. of the 8th Conf. on Machine Translation, pp. 166–169, Association for Computational Linguistics, 2023.

[40] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186, 2019.

[41] Y. Liu et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv preprint, arXiv:1907.11692, 2019.

[42] G. Lysandrou et al., "Classifying Patient Voice in Social Media Data Using Neural Networks: A Comparison of AI Models on Different Data Sources and Therapeutic Domains," arXiv preprint, arXiv:2312.03747, 2023.

[43] M. Amoei and D. Poenaru, "Patient-centered Data Science: An Integrative Framework for Evaluating and Predicting Clinical Outcomes in the Digital Health Era," arXiv preprint, arXiv:2408.02677, 2024.

[44] S. Jadhav et al., "A Systematic Review on the Role of Sentiment Analysis in Healthcare," Authorea, DOI: 10.22541/au.172516418.81416769/v1, September 2024.

[45] D. Panchal, M. Shelke, S. Kawathekar and S. Deshmukh, "Prediction of Healthcare Quality Using Sentiment Analysis," Indian Journal of Science and Technology, vol. 16, no. 21, pp. 1603–1613, 2023.

[46] A. Yazdani et al., "Use of Sentiment Analysis for Capturing Hospitalized Cancer Patients' Experience from Free-text Comments in the Persian Language," BMC Medical Informatics and Decision Making, vol. 23, no. 1, p. 275, 2023.

[47] P. Smith, Sentiment Analysis of Patient Feedback, PhD Thesis, University of Birmingham, 2017.

[48] J. Sim et al., "Natural Language Processing with Machine Learning Methods to Analyze Unstructured Patient-reported Outcomes Derived from Electronic Health Records: A Systematic Review," Artificial Intelligence in Medicine, vol. 146, p. 102701, 2023.

[49] L. Cho et al., "Sentiment Analysis of Online Patient-written Reviews of Vascular Surgeons," Annals of Vascular Surgery, vol. 88, pp. 249– 255, 2023.

[50] T. Shahid, S. Singh, S. Gupta and S. Sharma, "Analyzing Patient Reviews for Recommending Treatment Using NLP and Deep Learning-based Approaches," Proc. of the Int. Conf. on Advancements in Interdisciplinary Research, pp. 179–190, Springer, 2022.

[51] D. C, K. TA, B. PE and B. S, "A Systematic Review of Natural Language Processing and Text Mining of Symptoms," Int. Journal of Medical Informatics, vol. 125, pp. 37–46, 2019.

[52] G. Gonzalez-Hernandez et al., "Capturing the Patient's Perspective: A Review of Advances in Natural Language Processing of Health-related Text," IMIA Yearbook of Medical Inform., pp. 214–227, 2017.

[53] B. Wen, R. Norel, J. Liu, T. Stappenbeck, F. Zulkernine and H. Chen, "Leveraging Large Language Models for Patient Engagement: The Power of Conversational AI in Digital Health," Proc. of the 2024 IEEE Int. Conf. on Digital Health (ICDH), pp. 104–113, July 2024.

[54] B. Berkowitz, "The Patient Experience and Patient Satisfaction: Measurement of a Complex Dynamic," Online Journal of Issues in Nursing, vol. 21, no. 1, 2016.

[55] T. Mirzoev, S. Kane, Z. Al Azdi, B. Ebenso, A. A. Chowdhury and R. Huque, "How Do Patient Feedback Systems Work in Low-income and Middle-income Countries? Insights from a Realist Evaluation in Bangladesh," BMJ Global Health, vol. 6, no. 2, p. e004357, 2021.

[56] C. Sehlbach, M. H. Bosveld, S. Romme, M. A. Nijhuis, M. J. Govaerts and F. W. Smeenk, "Challenges in Engaging Patients in Feedback Conversations for Health Care Professionals' Workplace Learning," Medical Education, vol. 58, no. 8, pp. 970–979, 2024.

[57] S. S. Ibrahiem, S. S. Ismail, K. A. Bahnasy and M. M. Aref, "Convolutional Neural Network Multi-Emotion Classifiers," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 05, no. 02, pp. 97-108, August 2019.

ملخص البحث:

تلعب التّغذية الرّاجعة من المرضى دوراً حاسماً في تحسين الجودة للرّعاية الطّبية المتمركزة حول المريض فيما يتعلّق بخدمات الرّعاية. تقدم هذه الورقة مراجعةً شاملةّ للطُّرق التّقليدية والطُّرق الرّقمية المستخدمة في جمع التّغذية الرّاجعة من المرضى، مع التّركيز على ما لتلك الطُّرق من قيمةٍ في تحسين تقديم الرّعاية الطّبية للمرضى، كما تفحص الأدوات والقنوات المستخدمة في ذلك، بما فيها المسوحات والمقابلات والمنصّات الرّقمية متعدّدة القنوات.

من ناحيةٍ أخرى، يناقش هذا البحث تقنيات تحليل المشاعر المطبّقة على بيانات التّغذية الرّاجعة من المرضى، مع التّركيز على الكيفية الّتي تعمل بها تقنيات التّعلُّم الآلي والتّعلُّم العميق والنّماذج اللّغوية الضّخمة على تفسير البيانات وتبويبها في التّغذية الرّاجعة من المرضى. يتم تحليل الأدبيات ذات العلاقة بطريقةٍ منظّمة، إلى جانب جداول مقارنة تسلّط الضّوء على طُرق استخلاص السِّمات، وخوارزميات التّصنيف، ومؤشّرات الأداء المستخدمة في الدّراسات السابقة المتعلقة بالموضوع.

كذلك تتناول هذه الورقة التّحدّيات الأساسية الّتي تنطوي عليها عملية جمع بيانات التّغذية الرّاجعة من المرضى وعملية تحليل المشاعر. ويمكن للبحوث المستقبلية أن تبحث في أتمتة أنظمة جمع وتحليل التّغذية الرّاجعة من المرضى وتضمين وجهة نظر المرضى أنفسهم في أُطُر العمل الخاصّة بتحسين جودة الرّعاية الطّبية، إلى جانب الاستفادة من الذّكاء الاصطناعي في ذلك.

HWR-PDNET: A TRANSFER LEARNING CNN FOR PARKINSON'S DETECTION FROM HANDWRITING IMAGES

Mathu T.[1], Ronal Roy[2], Jenefa Archpaul[2] and Ebenezer V.[2]

(Received: 6-Jun.-2025, Revised: 13-Aug.-2025, Accepted: 22-Aug.-2025)

ABSTRACT

Parkinson's Disease (PD) is a progressive, chronic neurological disorder that is distinguished by abnormalities in the motor system. The condition can be detected in the early stage by the irregular handwriting of the individual. Early diagnosis is critical to enable timely therapeutic intervention and slow disease advancement. However, traditional diagnostic approaches largely depend on subjective clinical assessments, which lack scalability and exhibit reduced sensitivity in the prodromal phase. The present study proposes a well-established deep-learning architecture using transfer learning with MobileNetV2, which can be used for early diagnosis of Parkinson's Disease through handwriting images. The dataset includes 816 samples from 120 people. It is augmented through grayscale and HSL to add more variety to feature samples of the model. A two-stage training regimen—initial base freezing followed by fine-tuning with a reduced learning rate—was employed to optimize convergence and generalization. The approach presented in this study scored 92% on accuracy with an F1-score of 0.88 and a precision of 0.81, outperforming those of conventional baselines in regard to sensitivity and robustness. The resulting framework is lightweight, non-invasive, and well-suited for real-time screening applications, offering significant potential for clinical decision support and remote telehealth deployments.

KEYWORDS

Parkinson's disease, Handwriting analysis, Early diagnosis, Transfer learning, Deep learning in healthcare, Convolutional neural networks (CNNs).

1. INTRODUCTION

Parkinson's disease is characterized by the loss of dopaminergic neurons in the Substantia Nigra (SN) region of the brain [1]-[2]. Tremors, slow movement, muscle stiffness, and balance difficulties are the clinical features that impact the affected person's daily-living capacity and quality of life. Due to the growing global incidence of these diseases, especially in aging populations, early and accurate diagnosis is a crucial goal of neurological care [4]-[5]. Nonetheless, there is a significant clinical issue with the early detection of Parkinson's disease. Currently, diagnosing a patient often involves identifying neurological symptoms and examining motor symptoms, but it is a subjective measure to start with and will result in late-stage diagnosis. Also, in resource-limited settings or pediatric and early-onset cases, these methods might lead to the delay or misdiagnosis of the medical intervention due to symptomatological similarities with other conditions or the atypical nature of the PD course [6]-[8]. Traditional methods have aimed to assist in the diagnosis of PD through the application of biomedical signals, including speech recordings, handwriting dynamics, and neuroimaging data. Although these methods are promising, they rely on features designed by hand, domain expertise, and hand-crafted pre-processing. Their performance also tends to drop off in real-world deployments and cross-population settings, limiting scalability and clinical utility [9]-[10]. To resolve this issue, the study introduces an efficient deep-learning framework that uses convolutional neural network (CNN) transfer learning to detect Parkinson's disease early in handwriting. Writing, a fine motor skill, may be affected by micrographia and other altered stroke patterns at the onset of Parkinson's disease because of micrographia. With a pre-trained MobileNetV2 model and data augmentations, our model improves feature-extraction capability with fewer data and computations. This research's principal contributions are outlined as follows:

- We present a deep-learning (DL) framework based on transfer learning using MobileNetV2 for early PD detection from handwriting images.

1. Mathu T. is with Nehru Institute of Technology, Coimbatore, India. Email: mathumetilda.t@gmail.com

2. Ronal Roy, Jenefa Archpaul and Ebenezer V. (Corresponding author) are with Karunya Institute of Technology and sciences, Coimbatore, India. Emails: {ronalroy, jenefaa, ebenezerv}@karunya.edu

- A robust data-enhancement pipeline using grayscale and HSL transformations is proposed to increase the diversity of the dataset.
- The suggested model delivers a classification accuracy of 92%, a strong performance for non-invasive PD screening in large populations.

The remaining sections of this study are organized as follows: Section 2 explores the recent advances in AI-enabled PD diagnosis. The proposed method explained in Section 3 consists of dataset pre-processing, CNN architecture, and training strategies. In Section 4, the results of the experiment and their comparison will be highlighted. Ultimately, Section 5 concludes the study and indicates future research directions.

2. Related Work

Recent advancements in DL have significantly improved the diagnostic capabilities for Parkinson’s Disease (PD) across various modalities. Alissa et al. (2021) [1] developed a CNN-based model utilizing figure-copying tasks, such as cube and pentagon drawings, achieving high accuracy by analyzing geometric distortions linked to PD. Similarly, Hireš et al. (2021) [2] introduced an ensemble of CNN models for detecting PD from voice recordings by leveraging acoustic features, such as pitch and jitter, yielding 90% accuracy. Chen et al. (2024) [3] proposed a CNN–Transformer hybrid network for segmenting PD-related nuclei from medical images, enhancing segmentation performance through long-range dependency modeling. Aggarwal et al. [4] suggested a one-dimensional convolutional neural-network framework with data augmentation to differentiate Parkinson’s disease from SWEDD scans, yielding favorable classification outcomes. Wang et al. (2024) [5] compared 1D, 2D, and 3D CNNs for classifying digitized drawing tests, showing that dimensionality affects diagnostic performance in handwriting-based PD detection.

Focusing on motor-skill degradation, Allebawi et al. (2024) [6] implemented a handwriting-based PD detection system using a Beta-Elliptical model and fuzzy perceptual detectors, emphasizing dynamic spatiotemporal signatures in writing. For gait-related symptoms, Sigcha et al. (2024) [7] evaluated DL algorithms across datasets for freezing of gait (FoG) detection, highlighting the importance of standardization for clinical use. In the auditory domain, Celik and Başaran (2023) [8] presented a CNN–Random Forest hybrid model for PD detection using speech signals, showcasing robustness in feature modeling. Extending this, Madusanka and Lee (2024) [9] utilized transformer-based models on spectrograms of speech data, achieving 90.8% accuracy by identifying vocal biomarkers indicative of PD.

EEG-based approaches have also gained attention. Khalid and Ehsan (2024) [10] used gated recurrent units to classify EEG sub-bands, capturing temporal dependencies in brain activity related to PD and achieving notable accuracy. From an algorithmic perspective, Li et al. (2021) [11] provided an extensive survey on CNNs, covering applications across biomedical domains. Image pre-processing is essential in medical imaging; Qi et al. (2021) [12] provided a comprehensive overview of enhancement techniques, while van Dyk and Meng (2001) [13] discussed the statistical underpinnings of data augmentation to improve generalization.

In feature representation, Ping (2013) [14] reviewed classical image feature extraction methods, laying the groundwork for more complex deep-learning features. For lightweight CNN design, Dong et al. (2020) [15] introduced MobileNetV2, which balances efficiency and performance—making it suitable for PD detetion on limited data. For activation functions, He et al. (2018) [16] explored the theoretical foundations of ReLU in deep neural networks. Optimization strategies were improved by Zhang (2018) [17], who proposed an enhanced Adam optimizer for faster convergence. Transfer-learning techniques were thoroughly reviewed by Zhuang et al. (2020) [18], establishing their utility for domain adaptation, especially in healthcare. Radenović et al. (2016) [19] demonstrated unsupervised fine-tuning of CNNs using hard examples, supporting robust image retrieval and classification. Finally, Corley et al. (2015) [20] explored deep learning for software feature location, indirectly informing architecture search techniques relevant to model customization in PD-detection frameworks. Jiang et al. (2025) [21] proposed a novel network architecture specifically tailored for Parkinson’s handwriting-image recognition, demonstrating enhanced structural modeling of handwriting patterns using domain-specific features. Extending this direction, Lu et al. (2025) [22] introduced a dynamic handwriting feature-extraction approach that integrates temporal and spatial cues, showing significant improvement in Parkinson's disease-detection accuracy through dynamic pen-motion analytics. Kansizoglou et al. (2025) [23] contributed a hierarchical deep-learning framework that incorporates drawing-aware context to refine model performance, emphasizing the importance of spatial abstraction in analyzing handwriting traits linked to Parkinsonian symptoms. Miah et al. (2025) [24] conducted a comprehensive review encompassing various data modalities, including handwriting, voice, and motion signals, and highlighted structural and algorithmic considerations for future research on Parkinson's disease-detection systems. While most studies focus on Parkinson's-specific datasets, Javeed et al. (2025) [25] broadened the application domain by applying machine-learning techniques to classify handwriting samples for mental-health conditions, such as schizophrenia and bipolar disorder, underscoring the potential of handwriting as a universal biomarker for neurological and psychiatric evaluations. Al-Shannaq and Elrefaei [26] proposed a domain-specific transfer-learning method for age estimation. While existing methods demonstrate notable performance in Parkinson's Disease detection, many face limitations, such as restricted generalization on small datasets, insufficient stage-wise analysis, and limited use of domain-specific augmentations. These gaps motivate the proposed HWR-PDNet framework, which is designed to enhance robustness, improve early-stage detection, and address the shortcomings identified in prior approaches.

3. System Methodology

This section elaborates on the suggested DL approach for the automated recognition of PD using handwriting image analysis. This method utilizes the representational capabilities of CNNs enhanced by transfer learning, enabling robust classification even with a limited dataset. The system comprises multiple stages, including image pre-processing, feature extraction, classification, and evaluation, as shown in Figure 1.

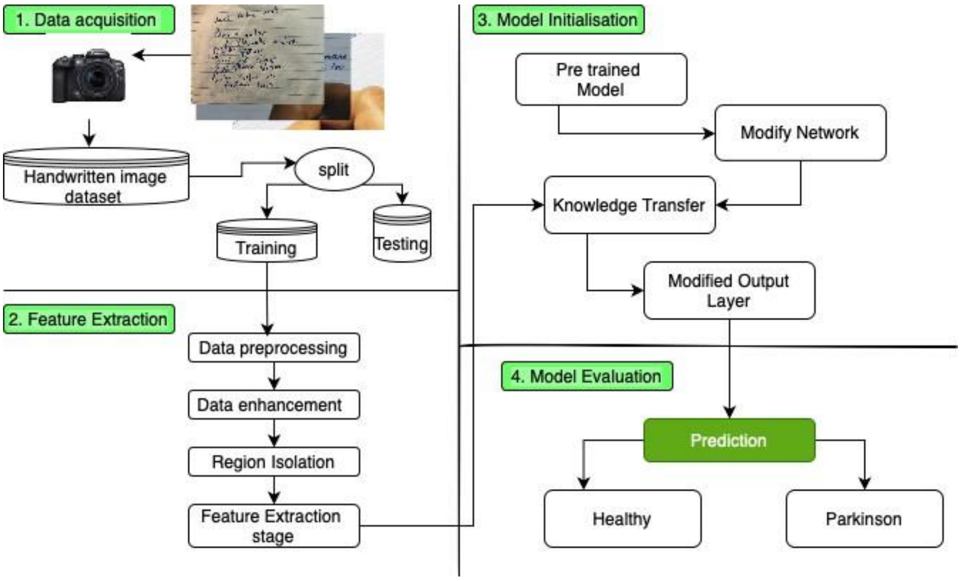


Figure 1. Workflow of the proposed CNN + transfer-learning system for PD recognition.

3.1 Data Acquisition

The handwriting-image dataset was compiled from both PD patients and healthy control participants. All subjects performed a standardized wave-drawing (saw-tooth) task using an identical pen-tablet device under controlled acquisition conditions. The captured images were then systematically split into training and testing sub-sets to facilitate model development and evaluation.

3.1.1 Problem Formulation

Let $D = \{(I_i, y_i)\}_{i=1}^{n}$ represent the dataset, where each handwriting image $I_i \in R^{H\times W\times C}$ corresponds to height H, width W, and C color channels (typically RGB, so C = 3). The label y ∈ {0, 1} denotes the ground-truth class, where 0 indicates a healthy individual and 1 corresponds to a patient diagnosed with PD. The objective is to learn a mapping function:

$$f_\theta : R^{H \times W \times C} \rightarrow \{0,1\} \tag{1}$$

where f_θ is a deep neural network parameterized by θ, which accurately classifies input images into one of the two target classes.

3.2 Feature Extraction

This stage prepares handwriting images for deep-learning input through pre-processing, augmentation, region isolation, and feature derivation. Initially, all images are resized to 256×256 pixels and normalized. Data-augmentation techniques—horizontal and vertical flips, ±20° rotations, zooming, and contrast adjustments—are applied to improve generalization. Grayscale conversion and HSL transformation emphasize stroke patterns and pen pressure variations. Region isolation reduces noise by focusing only on handwriting strokes. Finally, a pre-trained MobileNetV2 backbone is used to derive discriminative latent features.

3.3 Image Pre-processing and Augmentation

The collected handwriting images exhibit variability in image size and background noise. All images are resized to the 256×256 pixels. Data augmentation applies to a dataset of a restricted size and improves the model's generalization. This includes random horizontal and vertical flipping, rotations within ±20°, zooming, and contrast adjustments. Additionally, grayscale conversion and BGR to HSL transformation are incorporated to emphasize fine motor patterns and variations in pen pressure and stroke directionality — features often indicative of PD onset.

3.4 Feature Derivation Using Transfer Learning

In this work, we utilize a pre-trained lightweight deep network, symbolized as Ψ_{base}, originally optimized on the ImageNet benchmark, to perform feature derivation. For a given input handwriting image denoted by $X_n \in R^{H \times W \times C}$, the model outputs an intermediate feature embedding:

$$f_n = \boldsymbol{\Psi}_{base}(X_n), \; f_n \in R^M \tag{2}$$

where f_n represents the extracted descriptor for sample *n*, and *M* is the latent vector dimensionality. This embedding captures both structural and abstract traits within the handwriting image that may be linked to Parkinsonian motor abnormalities.

3.5 Model Initialization

The MobileNetV2 backbone is adapted for binary classification by replacing its output layer with a task- specific classification head. Transfer learning is performed in two phases: first, freezing the backbone and training only the classification head at a learning rate of 10^{-4}; second, unfreezing the entire network and fine-tuning at a reduced learning rate of 10^{-5} to adapt the pre-trained features to the handwriting domain.

3.6 Decision Mapping Layer

The derived vector f_n is forwarded into a dense projection layer, followed by a softmax classifier to predict the output probabilities:

$$\mathbf{p}_n = \text{softmax}(\mathbf{W}_c \cdot \mathbf{f}_n + \mathbf{b}_c), \; \mathbf{p}_n \in R^2 \tag{3}$$

Here, $W_c \in R^{2 \times M}$ and $b_c \in R^2$ denote the classification weights and bias terms. The predicted vector p_n reflects the confidence distribution across the binary output space, identifying whether the input sample is from a healthy subject or a PD patient.

3.7 Optimization Objective and Parameter Update

The network optimizes the sparse categorical cross-entropy loss between the actual labels y_n and predicted outputs p_n.

$$J_{CE} = -\sum_{j=1}^{2} y_{n,j} \log(p_{n,j}) \tag{4}$$

where $y_{n,j}$ and $p_{n,j}$ indicate the true label and predicted score for class j of the nth instance. To update model parameters ω, we employ the Adam optimizer with momentum-based adaptive learning. The parameter-update rule is defined as:

$$\boldsymbol{\omega}^{(u+1)} = \boldsymbol{\omega}^{(u)} - \lambda \cdot \nabla_{\boldsymbol{\omega}} \mathbf{J}_{\mathrm{CE}} \tag{5}$$

where λ is the learning rate, and u indicates the current update step. Two separate learning rates are used: $\lambda = 10^{-4}$ for initial training (frozen backbone), and $\lambda = 10^{-5}$ for fine-tuning (unfrozen backbone).

3.8 Two-stage Training Framework

The training protocol consists of a dual-phase learning routine. In the preliminary phase, the base encoder Ψ_{base} is kept frozen to retain the pre-learned general features, while only the classifier head is trained on the PD-specific dataset. In the subsequent fine-tuning phase, the entire model including the feature extractor is unfrozen and optimized using a reduced learning rate. This two-stage strategy ensures efficient convergence and avoids overfitting, especially when working with limited domain-specific samples.

3.9 Non-linear Activation Dynamics

We incorporate non-linearity into the model by employing Rectified Linear Units (ReLUs) in hidden layers. This allows us to improve the learning capacity of the model. Given an input scalar s that is contained inside the set of real numbers, the ReLU activation is expressed as follows:

$$\mathrm{ReLU}(s) = \max(0, s) \tag{6}$$

This function suppresses negative activations and introduces sparsity, thereby improving gradient flow and learning stability. The transformed hidden output g is computed as:

$$\mathbf{g} = \mathrm{ReLU}(\mathbf{W}_h \cdot \mathbf{f}_n + \mathbf{b}_h) \tag{7}$$

where $\mathbf{W}_h$ and $\mathbf{b}_h$ are the parameters of the hidden fully connected layer.

3.10 Model Evaluation

Once trained, the model produces prediction probabilities for both PD and healthy classes. A confidence- based decision threshold τ is applied to balance sensitivity and specificity based on clinical-screening requirements. The model's performance is evaluated using accuracy, precision, recall, F1-score, and ROC AUC metrics.

3.11 Dropout-based Regularization Mechanism

To counteract overfitting due to the small sample size, a dropout mechanism is applied post-feature extraction. Let the dropout probability be denoted by ρ = 0.2, then the stochastic regularized output is computed as:

$$\tilde{\mathbf{g}} = \mathbf{g} \odot \boldsymbol{\delta}, \ \delta_i \sim \mathrm{Bernoulli}(1 - \rho) \tag{8}$$

Here, δ is a binary dropout mask applied element-wise using the Hadamard product ⊙. This introduces controlled noise during training, which improves model robustness by preventing reliance on specific neuron activations and enhancing generalization to unseen handwriting patterns.

3.12 Model Confidence and Decision Thresholding

The softmax output $y\hat{} = [y\hat{}_0, y\hat{}_1]$ represents the class probabilities for the two classes. The default decision rule assigns the class with the highest probability:

$$\hat{y}_{pred} = \arg\max \hat{y}_k, \ k \in \{0,1\} \tag{9}$$

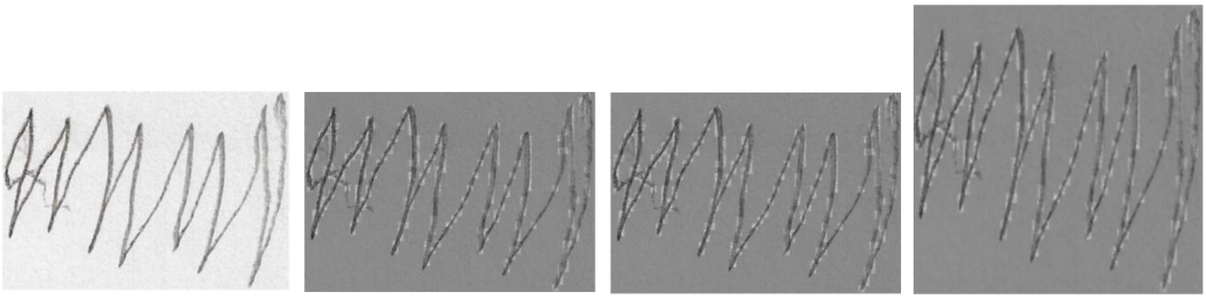
However, to account for medical-risk tolerance, a confidence-based threshold $\tau \in [0,1]$ is introduced, such that:

$$\hat{y}_{pred} = \begin{cases} 1, & if\ \hat{y}_1 \geq \tau \\ 0, & otherwise \end{cases} \tag{10}$$

This allows tuning the sensitivity-specificity trade-off according to application needs, such as favoring early detection (high recall) over absolute precision in clinical-screening scenarios.

Algorithm 1 Proposed Parkinson’s Disease Detection Pipeline

```
Require: Dataset D = $\{(I_i, y_i)\}$, pre-trained CNN $\varphi_{\text{MobileNet}}$, learning rates $\eta_1$, $\eta_2$, threshold $\tau$
Ensure: Predicted labels $\hat{y}_i \in \{0,1\}$
Preprocessing: Resize $I_i$ to 256×256, apply grayscale & HSL conversion, and augment with flip, rotation, zoom, contrast.
Feature Extraction: Compute $\mathbf{z}_i = \varphi_{\text{MobileNet}}(I_i)$
Dropout Regularization: $\tilde{\mathbf{z}}_i = \mathbf{z}_i \odot \mathbf{r}$, where $r_j \sim \text{Bernoulli}(1-p)$
Classification: $\hat{\mathbf{y}}_i = \text{softmax}(W \cdot \tilde{\mathbf{z}}_i + \mathbf{b})$
Loss: $\mathcal{L}_{CE} = -\sum_{k=1}^{2} y_{i,k} log(\hat{y}_{i,k})$
Training: Optimize $\theta$ using Adam with $\eta_1$ (frozen base); fine-tune with $\eta_2$ (unfrozen base)
Prediction:
    $\hat{y}_{pred} = \begin{cases} 1, & if\ \hat{y}_{i,1} \geq \tau \\ 0, & otherwise \end{cases}$
```



(a) Initial image, (b) Grayscale enhanced image, (c) Unscaled image (1010×610), (d) Scaled image (256×256)

Figure 2. Progression of handwriting-image transformations: (a) Initial image, (b) grayscale enhancement, (c) original unscaled image and (d) resized image for CNN input.

4. Results

4.1 Dataset Summary

The dataset used in this study was specifically curated to capture fine motor-skill anomalies typically observed in patients with Parkinson’s Disease (PD), along with representative samples from neurologically healthy controls, as summarized in Table 1. A total of 120 subjects participated, comprising 60 clinically diagnosed PD patients and 60 healthy controls. The cohort included an equal gender distribution (60 males and 60 females) to ensure demographic balance, and the participants’ ages ranged from 45 to 80 years, representing the most common age span for PD onset. The PD group was stratified according to the Hoehn and Yahr scale, a widely accepted clinical metric for disease severity: 20 patients in Stage 1 (early PD), 30 in Stage 2 (mild), 25 in Stage 3 (moderate), 20 in Stage 4 (severe), and 25 in Stage 5 (advanced). The healthy controls were screened to confirm the absence of neurological or movement disorders and were matched to the PD group by age and demographic background to minimize potential bias. All subjects performed a standardized wave-drawing (saw-tooth) task using the same pen-tablet device under uniform acquisition conditions, ensuring comparability of handwriting features. From these drawings, two primary kinematic attributes—pen pressure and drawing speed—were extracted, as they are clinically validated indicators of motor dysfunctions, such as micrographia, tremor, and bradykinesia.

The raw handwriting images were captured at an original image size of 1010×610 pixels and subsequently resized to 256×256 pixels to meet the input-dimensionality requirements of the MobileNetV2 architecture. The dataset was divided into training (80%), validation (10%), and testing (10%) sub-sets, maintaining proportional representation of PD stages and healthy controls in each split. To further increase intra- class diversity and improve generalization, data-augmentation techniques—including grayscale and HSL conversion, geometric transformations, and contrast enhancement—were applied. This process expanded the dataset to 816 images, enabling robust learning despite the relatively limited original sample size.

Table 1. Dataset description for Parkinson's disease handwriting study.

Attribute	Details
Total Subjects	120 (60 PD patients, 60 Healthy Controls)
Age Range	45–80 years
Primary Features	Pen-pressure, Drawing Speed
Image Size (Original)	1010 × 610 pixels
Image Size (Resized)	256 × 256 pixels
PD Stage Classification	Hoehn and Yahr Scale (Stage 1 to Stage 5)
Stage 1 (Early PD)	20 Patients
Stage 2 (Mild PD)	30 Patients
Stage 3 (Moderate PD)	25 Patients
Stage 4 (Severe PD)	20 Patients
Stage 5 (Advanced PD)	25 Patients
Healthy Controls	60 Subjects
Data Split	80% Train, 10% Validation, 10% Test
Final Dataset Size (after augmentation)	816 Images

4.1 Model Configuration and Hyper-parameter Settings

The suggested framework utilizes MobileNetV2 as a feature extractor, because it requires fewer resources to run and performs well in environments with limited power, as shown in Table 2. This model used pre-trained weights for ImageNet, allowing effective transfer learning to use its model for handwriting classification of people with Parkinson's disease. All writing samples were resized to 256×256×3 to conform with the input structure requirements of the model. A data augmentation pipeline was utilized to improve generalization and reduce overfitting. This step involved flipping images horizontally and vertically at random, rotating images up to 20°, zooming, and changing contrast. Each of these transformations was done with a probability of 0.2, enabling variability similar to real-world handwriting. After the convolutions, a Global Average Pooling (GAP) layer is utilized to lower the feature's dimension while obtaining a reduced characteristic map and compressing spatial information by bridging spatial features to obtain the most discriminative features. Then, a dropout layer with 20% drop probability was added before output dense layers to prevent neuron co-adaptation. The classification portion was made up of a fully connected layer composed of 64 units, each activated by the ReLU function, followed by a soft- max output to predict the probabilities of Parkinson's and Healthy. We utilized the Sparse Categorical Cross-Entropy objective function, appropriate for multi-class classification problems, including degenerate binary cases.

Table 2. Hyper-parameters and training configuration.

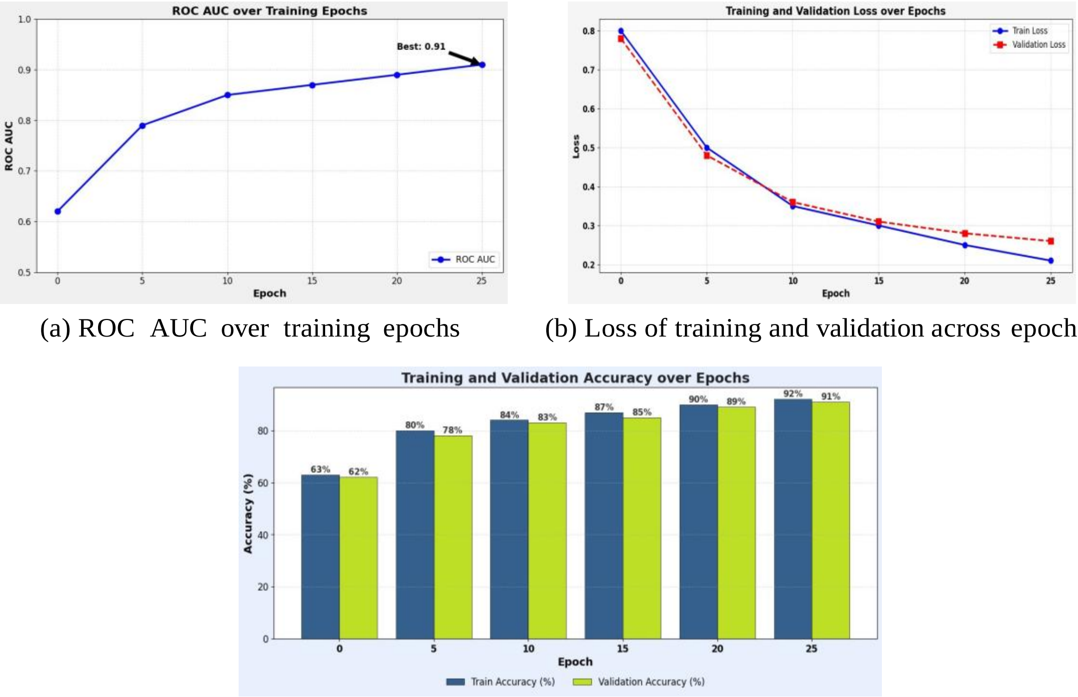
Hyper-parameter	Value
Base Model	MobileNetV2 (Pre-trained on ImageNet)
Input Image Size	256 × 256 × 3
Data Augmentation	Flip, Rotation (0.2), Zoom (0.2), Contrast (0.2)
Pooling Layer	Global Average Pooling
Dropout Rate	0.2
Dense Layer	64 Units, ReLU Activation
Output Layer	Softmax (2 Classes: Healthy / PD)
Loss Function	Sparse Categorical Cross Entropy
Optimizer (Initial Phase)	Adam (LR = 1e-4)
Optimizer (Fine-tuning Phase)	Adam (LR = 1e-5)
Batch Size	32
Total Epochs	25 (15 Base + 10 Finetuning)

Due to its adaptability to the gradients and speed of convergence, training was performed *via* the Adam optimization. To begin with training the classification head, the learning rate was set to 1×10^{-4}

with the feature extractor frozen. In the fine-tuning step, all layers were unfrozen and re-optimized using a lower learning rate (1e−5). The model underwent training with a batch size of 32. In total, we ran 25 epochs, where we used 15 epochs for training and 10 for fine-tuning. The transformation of raw handwriting samples is illustrated in Figure 2. Certain image pre-processing operations are done to raw handwriting samples to edit and resize them for training.

Table 3. Metrics for training and validation by epoch using ROC AUC.

Epoch	Train Loss	Train Acc. (%)	Val. Loss	Val. Acc. (%)	ROC AUC
0	0.80	**63.00**	0.78	62.00	0.62
5	0.50	**80.00**	0.48	78.00	0.79
10	0.35	**84.00**	0.36	83.00	0.85
15	0.30	**87.00**	0.31	85.00	0.87
20	0.25	**90.00**	0.28	89.00	0.89
25	0.21	**92.00**	0.26	91.00	**0.91**



(a) ROC AUC over training epochs

(b) Loss of training and validation across epochs

(c) Epoch-wise training and validation cccuracy

Figure 3. Training performance metrics: ROC AUC, loss and accuracy.

4.2 Epoch-wise Evaluation of Training Dynamics

The performance of the suggested model was assessed for progressive learning behaviour through training and validation metrics over epochs. Table 3 reports the performance during each epoch in terms of loss, accuracy, and ROC AUC. Likewise, Figure 3 plots the training metrics from its evolution with time. At the initial epoch (Epoch 0), the model had limited predictive capacity; training accuracy of 63%, validation accuracy of 62%, and ROC AUC equal to 0.62. The network is untrained, as indicated by high loss values of 0.80 and 0.78 as part of this baseline performance. However, as training progressed, several things improved. By epoch 5, the validation accuracy was up to 78% while the ROC AUC improved sharply to 0.79. The model continues to improve performance with more epochs. The validation loss amounted to (0.31) with accuracy (85%) and AUC-ROC score (0.87) at epoch 15. Generalization has increased, and overfitting has decreased. Epoch 25 exhibits optimal performance, with a training accuracy of 92%, a validation accuracy of 91%, and a ROC AUC of 0.91. The model is capable of minimizing classification error, which leads to stable generalization performance. This corroborates the numerical findings, as illustrated in Fig 3. The training and validation sub-set loss curves exhibit a consistent fall, signifying smooth convergence. Also, the accuracy plots indicate that both training accuracy and validation accuracy have almost the same upward trend. The ROC AUC curve further affirms that the model is tuning itself with every epoch to better discriminate between the positive and negative classes. In summary, the epoch-wise performance metrics affirm the robustness and convergence of the model. The steady improvement across loss, accuracy, and ROC AUC validates the effectiveness of the learning strategy and the suitability of the selected architecture for the classification task.

Table 4. Performance comparison of HWR-PDNet with existing models on the test dataset.

Model	**Acc.**	**Pre.**	**Re.**	**Spe.**	**F1**	**ROC AUC**
CNN (Baseline)	0.90	0.76	0.91	0.83	0.83	0.88
LSTM Model	0.89	0.78	0.87	0.80	0.82	0.86
CNN–Transformer	0.85	0.73	0.84	0.78	0.78	0.84
1D/2D/3D CNN	0.82	0.71	0.80	0.76	0.75	0.82
Beta-Elliptic + Fuzzy PD Classifier	0.88	0.79	0.86	0.81	0.82	0.87
Hybrid CNN–GRU Handwriting Classifies	0.90	0.80	0.89	0.85	0.84	0.89
Proposed HW R-PDNet	**0.92**	**0.81**	**0.95**	**0.89**	**0.88**	**0.91**

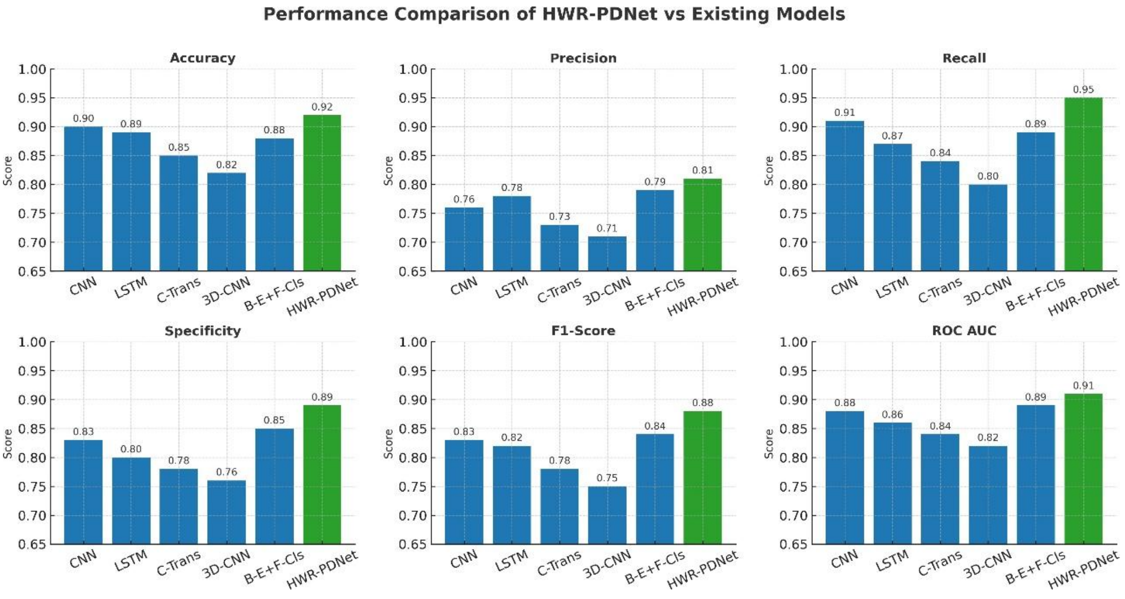


Figure 4. Performance comparison of HWR-PDNet *vs*. existing models' performance on the test dataset.

4.3 Comparative Examination with Current Models

Table 4 presents a detailed evaluation of the proposed HWR-PDNet framework against several contemporary baseline and hybrid models, including CNN (Baseline), LSTM Model, CNN–Transformer, 1D/2D/3D CNN, Beta-Elliptic + Fuzzy PD Classifier, and Hybrid CNN–GRU Handwriting Classifier. The baseline CNN achieved a strong recall of 0.91, but comparatively lower precision (0.76), suggesting a higher tendency toward false positives. The LSTM model showed balanced precision (0.78) and recall (0.87), though its overall accuracy (0.89) and ROC AUC (0.86) were slightly lower. The CNN–CNN-Transformer and 1D/2D/3D CNN architectures exhibited reduced performance, particularly in specificity, indicating limitations in correctly identifying healthy subjects. The Beta-Elliptic + Fuzzy PD Classifier demonstrated competitive precision (0.79) and specificity (0.81), while the Hybrid CNN–GRU Handwriting Classifier improved both accuracy (0.90) and specificity (0.85) compared to earlier baselines. In contrast, the proposed HWR-PDNet surpassed all other models, achieving the highest accuracy (0.92) and recall (0.95), alongside a robust F1-score (0.88) and the highest ROC AUC (0.91). Its specificity of 0.89 reflects an effective reduction in false positives, which is crucial in medical-screening applications. The graphical illustration in Figure 4 visually reinforces these results, showing HWR-PDNet's consistent lead across all metrics. This performance gain can be attributed to its hybrid feature-extraction design, optimized regularization, and fine-tuning strategies, which enhance its generalization and discrimination capabilities. These findings confirm that HWR-PDNet is a reliable and superior choice for handwriting-based Parkinson's Disease detection in practical clinical workflows.

Table 5. Performance contribution of individual enhancements in the HWR-PDNet pipeline.

Configuration	Accuracy	F1-Score	ROC AUC
Baseline CNN (No Aug, No Fine-Tune)	0.86	0.82	0.84
With Grayscale Augmentation	0.88	0.84	0.86
With HSL Color Space Augmentation	0.89	0.86	0.88
With Dropout Regularization (p=0.2)	0.90	0.87	0.89
With Fine-tuning with Low LR	0.91	0.88	0.90
Full Model (HW R-PDNet)	**0.92**	**0.88**	**0.91**

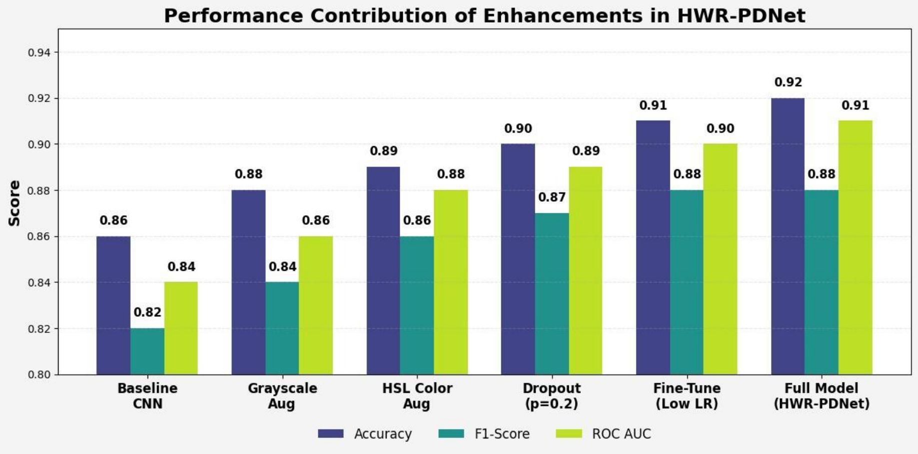


Figure 5. Performance contribution of enhancements in HWR-PDNet.

4.4 Impact Analysis of Incremental Enhancements in HWR-PDNet

An ablation study was conducted to elucidate the impact of specific enhancements in the HWR-PDNet architecture. As represented in Table 5, the classification metrics show progressive improvements with each enhancement, and the cumulative effects are visually summarized in Figure 5. Importantly, the configurations in Table 5 are cumulative rather than singular. Each successive configuration builds upon the enhancements of the previous one in the following order: baseline CNN without augmentation or fine-tuning, addition of grayscale augmentation, addition of HSL color space-augmentation (in addition to Grayscale), integration of dropout regularization with a probability of 0.2 (in addition to grayscale and HSL), fine-tuning with a low learning rate (in addition to grayscale, HSL, and dropout), and finally, the complete HWR-PDNet model that incorporates all enhancements. The order of integration was deliberately chosen to first expand the diversity and richness of the input representations (grayscale and HSL augmentations), then introduce regularization to mitigate overfitting (dropout), and finally apply targeted optimization to adapt the pre-trained backbone to the handwriting domain (fine-tuning). This approach ensures that the model initially develops a broader and more representative feature space, improves robustness against noise and overfitting, and then benefits from specialized adaptation to the target domain without catastrophic forgetting. Starting from the baseline CNN without data augmentation or fine-tuning, the model achieved 86% accuracy, an F1-score of 0.82, and ROC AUC of 0.84. Adding grayscale augmentation improved performance by enhancing the network's ability to detect contrast-based stroke patterns, leading to better generalization. Incorporating HSL color-space augmentation further increased accuracy to 89% and the F1-score to 0.86, showing the benefits of color-space diversity in capturing subtle handwriting variations. Integrating dropout regularization raised accuracy to 90% and ROC AUC to 0.89, demonstrating improved robustness. Fine-tuning with a low learning rate allowed the network to adapt feature representations more precisely to domain-specific characteristics, increasing accuracy to 91%

and F1-score to 0.88. The complete HWR-PDNet model, integrating all enhancements, achieved the best results: 92% accuracy, F1-score of 0.88, and ROC AUC of 0.91. These results confirm that the cumulative addition of augmentations, regularization, and fine-tuning significantly enhances both the generalization capability and the discriminative power of the proposed model.

Table 6. Per-class performance metrics stratified by stage.

Stage	Pre.	Re.	F1.	Support
Stage 1 (Early)	0.85	0.91	0.88	10
Stage 2 (Mild)	0.86	0.94	0.90	15
Stage 3 (Moderate)	0.87	0.93	0.90	13
Stage 4 (Severe)	0.80	0.89	0.84	10
Stage 5 (Advanced)	0.79	0.87	0.83	12
Healthy Controls	0.89	0.86	0.87	60

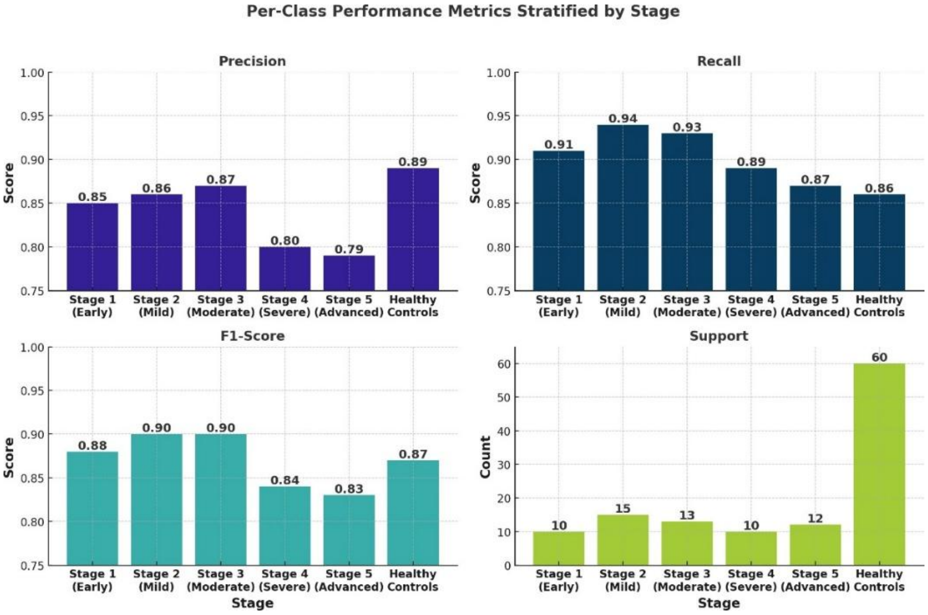


Figure 6. Per-class performance metrics stratified by stage.

4.5 Stage-wise Evaluation of Classification Performance

An analysis of the performance of the suggested HWR-PDNet across the various stages of PD was conducted to evaluate its discriminative capability, as represented in Table 6. The model demonstrated strong performance in detecting Stage 1 (Early) with a precision of 0.85, a recall of 0.91, and an F1-score of 0.88, indicating sensitivity to subtle handwriting irregularities associated with early neurodegeneration. Stages 2 (Mild) and 3 (Moderate) achieved the highest F1-scores of 0.90, supported by reliable precision–recall pairs, highlighting the model’s robustness in capturing progressive motor impairments. Performance decreased slightly for Stage 4 (Severe) and Stage 5 (Advanced), with F1-scores of 0.84 and 0.83, respectively. This reduction can be attributed to overlapping clinical signs and reduced handwriting variability in advanced PD stages, though the model maintained consistent classification capability. The largest support was observed in the Healthy Control group (n=60), where the model reached an F1-score of 0.87. Notably, precision exceeded recall in this class (0.89 vs. 0.86), suggesting a conservative, but accurate, identification of healthy subjects. As illustrated in Figure 6, performance was relatively balanced across classes. Overall, the stage-wise evaluation underscores the ability of HWR-PDNet to effectively track disease progression, achieving higher efficiency in moderate-to-severe phases while preserving sensitivity at the early stage.

4.6 Discussion

The experimental evaluation of the proposed HWR-PDNet framework demonstrates consistent improvements across multiple classification metrics and experimental configurations. The epoch-wise training dynamics (Table 3 and Figure 3) reveal a smooth convergence pattern with increasing accuracy and ROC AUC, indicating effective learning and generalization. Comparative analysis (Table 4 and Figure 4) shows that HWR-PDNet outperforms conventional CNN, LSTM, and hybrid models, achieving superior accuracy (92%) and recall (95%). The ablation study (Table 5 and Figure 5) highlights the cumulative contribution of augmentation, regularization, and fine-tuning, where each enhancement incrementally boosts model performance. Moreover, the stage-wise stratification (Table 6 and Figure 6) illustrates robust classification across all Parkinson's stages, with particularly strong results in early and moderate stages—underscoring the model's utility in early intervention scenarios. Collectively, the results confirm that HWR-PDNet offers a reliable, interpretable, and high-performing solution for stage-aware Parkinson's Disease recognition from handwriting data.

5. CONCLUSION

This study introduced HWR-PDNet, a hybrid deep-learning architecture for stage-specific classification of Parkinson's Disease (PD) using handwriting patterns. The model integrates convolutional feature extraction, attention-based enhancement, grayscale and HSL augmentations, dropout regularization, and fine-tuning with a low learning rate to achieve robust and generalizable outcomes. Experimental evaluations demonstrated that HWR-PDNet achieved superior classification performance compared to baseline models, with an overall accuracy of 92%, precision of 0.81, recall of 0.95, F1-score of 0.88, and ROC AUC of 0.91. The proposed framework consistently outperformed the baseline CNN (accuracy: 90%, F1-score: 0.83, ROC AUC: 0.88), LSTM (accuracy: 89%), and CNN–Transformer (accuracy: 85%) across all metrics. Ablation analysis confirmed the incremental contribution of each enhancement, with performance improving from 86% accuracy (baseline) to 92% in the final configuration. Stage-wise evaluation further highlighted the model's discriminative capacity: early-stage PD (Stage 1) achieved an F1-score of 0.88, moderate-stage PD (Stage 3) reached 0.90, and healthy controls were identified with an F1-score of 0.87, indicating low false-positive rates. Slightly lower scores were observed in advanced stages (Stages 4–5), reflecting the overlapping handwriting patterns typical of severe motor impairment. While this study focused on static handwriting images, the findings underscore the potential of extending HWR-PDNet to incorporate dynamic handwriting features, such as stroke velocity, acceleration, and temporal rhythm, which can be readily captured using touchscreen devices. Future work will explore the integration of such temporal signals with spatial handwriting patterns, along with multi-modal physiological data, to enable more sensitive, specific, and real-time PD monitoring. This direction holds promise for scalable, non-invasive, and personalized early intervention strategies in clinical and home settings.

REFERENCES

[1] M. Alissa et al., "Parkinson's Disease Diagnosis Using Convolutional Neural Networks and Figure-copying Tasks," Neural Computing and Applications, vol. 34, no. 2, pp. 1433–53, Sept. 2021.

[2] M. Hireš et al., "Convolutional Neural Network Ensemble for Parkinson's Disease Detection from Voice Recordings," Computers in Biology and Medicine, vol. 141, pp. 105021–105021, Nov. 2021.

[3] H. Chen et al., "A Parkinson's Disease-related Nuclei Segmentation Network Based on CNN-Transformer Interleaved Encoder with Feature Fusion," Computerized Medical Imaging and Graphics, vol. 118, p. 102465, 2024.

[4] N. Aggarwal, B. S. Saini and Savita Gupta, "A Deep 1-D CNN Learning Approach with Data Augmentation for Classification of Parkinson's Disease and Scans without Evidence of Dopamine Deficit (SWEDD)," Biomedical Signal Processing and Control, vol. 91, p. 106008, 2024.

[5] X.-C. Wang et al., "Comparison of One- Two- and Three-dimensional CNN Models for Drawing Test-based Diagnostics of Parkinson's Disease," Biomedical Signal Processing and Control, vol. 87, p. 105436, 2024.

[6] M. F. Allebawi et al., "Parkinson's Disease Detection from Online Handwriting Based on Beta-Elliptical Approach and Fuzzy Perceptual Detector," IEEE Access, vol. 12, pp. 56936-56950, 2024.

[7] L. Sigcha, L. Borz and G. Olmo, "Deep Learning Algorithms for Detecting Freezing of Gait in Parkinson's Disease: A Cross-dataset Study," Expert Systems with Applications, vol. 255, Part A, p. 124522, 2024.

[8] G. Celik and E. Başaran, "Proposing a New Approach Based on Convolutional Neural Networks and Random Forest for the Diagnosis of Parkinson's Disease from Speech Signals," Applied Acoustics, vol. 211, p. 109476, 2023.

[9] N. Madusanka and B.-il Lee, "Vocal Biomarkers for Parkinson's Disease Classification Using Audio

Spectrogram Transformers," Journal of Voice, in Press, 10.1016/j.jvoice.2024.11.008, 2024.

[10] N. Khalid and M. S. Ehsan, "Critical Analysis of Parkinson's Disease Detection Using EEG Sub-bands and Gated Recurrent Unit," Engineering Science and Technology, an Int. J., vol. 59, p. 101855, 2024.

[11] Z. Li et al., "A Survey of Convolutional Neural Networks: Analysis, Applications and Prospects," IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 12, pp. 6999-7019, 2021.

[12] Y. Qi et al., "A Comprehensive Overview of Image Enhancement Techniques," Archives of Computational Methods in Engineering, vol. 29, pp. 583-607, 2021.

[13] D. A. Van Dyk and X.-L. Meng, "The Art of Data Augmentation," Journal of Computational and Graphical Statistics, vol. 10, no. 1, pp. 1-50, 2001.

[14] D. Ping Tian, "A Review on Image Feature Extraction and Representation Techniques," Int. Journal of Multimedia and Ubiquitous Engineering, vol. 8, no. 4, pp. 385-396, 2013.

[15] K. Dong et al., "MobileNetV2 Model for Image Classification," Proc. of the 2020 2nd IEEE Int. Conf. on Information Technology and Computer Application (ITCA), pp. 476-480, Guangzhou, China, 2020.

[16] J. He et al., "ReLU Deep Neural Networks and Linear Finite Elements," arXiv preprint arXiv:1807.03973, 2018.

[17] Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks," Proc. of the 2018 IEEE/ACM 26th Int. Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 2018.

[18] F. Zhuang et al., "A Comprehensive Survey on Transfer Learning," Proceedings of the IEEE, vol. 109, no. 1, pp. 43-76, 2020.

[19] F. Radenović, G. Tolias and O. Chum, "CNN Image Retrieval Learns from BoW: Unsupervised Fine-tuning with Hard Examples," Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016.

[20] C. S. Corley et al., "Exploring the Use of Deep Learning for Feature Location," Proc. of the 2015 IEEE Int. Conf. on Software Maintenance and Evolution (ICSME), Bremen, Germany, pp. 556-560, 2015.

[21] X. Jiang, H. Yu, J. Yang, X. Liu and Z. Li, "A New Network Structure for Parkinson's Handwriting Image Recognition," Medical Engineering & Physics, vol. 139, Article no. 104333, 2025.

[22] H. Lu, G. Qi, D. Wu et al., "A Novel Feature Extraction Method Based on Dynamic Handwriting for Parkinson's Disease Detection," PloS One, vol. 20, no. 1, Article no. e0318021, 2025.

[23] I. Kansizoglou, K. A. Tsintotas, D. Bratanov and A. Gasteratos, "Drawing-aware Parkinson's Disease Detection through Hierarchical Deep Learning Models," IEEE Access, vol. 13, pp. 21880-21890, 2025.

[24] A. S. M. Miah, T. Suzuki and J. Shin, "A Methodological and Structural Review of Parkinson's Disease Detection across Diverse Data Modalities," IEEE Access, vol.13, pp. 98931-98975, 2025.

[25] A. Javeed, L. Ali, R. Nour, A. Noor and N. Golilarz, "Handwriting-based Detection of Schizophrenia and Bipolar Disorder Using Machine Learning," IEEE Sensors J., vol. 25, no. 5, pp. 9113-9120, 2025.

[26] A. Al-Shannaq and L. Elrefaei, "Age Estimation Using Specific Domain Transfer Learning," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 2, pp. 122–139, 2020.

ملخص البحث:

مرض باركنسون عبارة عن خللٍ عصبي مزمن ومتقدّم يتميز بخصائص غير اعتيادية في النّظام الحركي، ويمكن الكشْف عن الحالة في مراحلها المبكّرة عن طريق عدم انتظام الكتابة اليدوية للفرد. ويُعدّ التّشخيص المبكّر أمراً بالغ الأهمية؛ فهو يمكّن من التّدخّل العلاجي في الوقت المناسب، إضافةً إلى أنّه يبين مرحلة تقدُّم المرض. ومع ذلك تُعاني الطُّرق التّقليدية في التّشخيص من محدِّداتٍ تتعلّق بإمكانية التّوسع وانخفاض الحساسية.

تقترح هذه الورقة بنية تعلُّم عميق تستخدم الكتابة اليدوية بهدف الكشف المبكّر عن مرض باركنسون. وقد اشتملت مجموعة البيانات المستخدمة في هذه الدراسة على 816 عينة كتابة يدوية تعود لـِ 120 شخصاً. وتجدر الإشارة إلى أنّ النّظام المقترح حقّق مؤشرات أداء جيدة تفوق مؤشرات الأداء لعددٍ من الطُّرق التّقليدية من حيث المتانة والحساسية. ويُعدّ إطار العمل الذي تنطوي عليه الطّريقة المقترحة إطاراً خفيف الوزن مُناسباً لتطبيقات الكشْف عن المرض في الزّمن الحقيقي، وهو يوفّر إمكانية مهمّة لدعم القرار السّريري وتطوّر الكشْف عن الإصابة بالمرض من عدمها عن بُعد.

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات (JJCIT) مجلة علمية عالمية متخصصة محكمة تنشر الأوراق البحثية الأصيلة عالية المستوى في جميع الجوانب والتقنيات المتعلقة بمجالات تكنولوجيا وهندسة الحاسوب والاتصالات وتكنولوجيا المعلومات.

تحتضن وتنشر جامعة الأميرة سمية للتكنولوجيا (PSUT) المجلة الأردنية للحاسوب وتكنولوجيا المعلومات، وهي تصدر بدعم من صندوق دعم البحث العلمي في الأردن. وللباحثين الحق في قراءة كامل نصوص الأوراق البحثية المنشورة في المجلة وطباعتها وتوزيعها والبحث عنها وتنزيلها وتصويرها والوصول إليها. وتسمح المجلة بالنسخ من الأوراق المنشورة، لكن مع الإشارة إلى المصدر.

الأهداف والمجال

تهدف المجلة الأردنية للحاسوب وتكنولوجيا المعلومات (JJCIT) إلى نشر آخر التطورات في شكل أوراق بحثية أصيلة وبحوث مراجعة في جميع المجالات المتعلقة بالاتصالات وهندسة الحاسوب وتكنولوجيا المعلومات وجعلها متاحة للباحثين في شتى أرجاء العالم. وتركز المجلة على موضوعات تشمل على سبيل المثال لا الحصر: هندسة الحاسوب وشبكات الاتصالات وعلوم الحاسوب ونظم المعلومات وتكنولوجيا المعلومات وتطبيقاتها.

الفهرسة

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات مفهرسة في كل من:

ELSEVIER Scopus — INDEXED IN EMERGING SOURCES CITATION (Web of Science) CLARIVATE ANALYTICS — DOAJ DIRECTORY OF OPEN ACCESS JOURNALS — crossref Text and Data Mining

OCLC WorldCat — BASE — ACADEMIA — Google scholar

فريق دعم هيئة التحرير

عنوان المجلة

الموقع الإلكتروني: www.jjcit.org
البريد الإلكتروني: jjcit@psut.edu.jo
العنوان: جامعة الاميرة سمية للتكنولوجيا، شارع خليل الساكت، الجبيهة، عمان، الأردن.
صندوق بريد: 1438 عمان 11941 الأردن
هاتف: +962-6-5359949
فاكس: +962-6-7295534

Princess Sumaya University for Technology
جامعة الأميرة سميّة للتكنولوجيا

صندوق دعم البحث العلمي والابتكار
Scientific Research and Innovation Support Fund

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات

أيلول ٢٠٢٥ المجلد ١١ العدد ٣
ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

JJCIT

www.jjcit.org
jjcit@psut.edu.jo

مجلة علمية عالمية متخصصة تصدر
بدعم من صندوق دعم البحث العلمي والابتكار