Princess Sumaya University for Technology
جامعة الأميرة سميّة للتكنولوجيا

صندوق دعم البحث العلمي والابتكار
Scientific Research and Innovation Support Fund

Jordanian Journal of Computers and Information Technology

December 2025 VOLUME 11 NUMBER 04
ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

JJCIT

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:

EDITORIAL BOARD SUPPORT TEAM

LANGUAGE EDITOR

Haydar Al-Momani

EDITORIAL BOARD SECRETARY

Eyad Al-Kouz

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

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 04, December 2025.

AN ENHANCED WORD LEVEL ARABIC OCR BASED ON DUAL ENCODER TRANSFORMER ARCHITECTURE

Khulood Gaashan[1] and Maram Bani Younes[2]

(Received: 16-Jun.-2025, Revised: 12-Jul.-2025 and 12-Aug.-2025, Accepted: 13-Sep.-2025)

ABSTRACT

Arabic script is one of the most sophisticated and difficult scripts. It uses different shapes of characters with complex diacritical marks that are difficult to distinguish from the dots of characters. This script's distinctive features make the Optical Character Recognition (OCR) procedure more challenging and result in low-accuracy recognition. Different studies have aimed to introduce high-accuracy Arabic OCR in the literature. However, enhancing the accuracy of reading the words has been an open issue that depends on the used dataset and the developed recognition model. Besides, considering diacritics has been limited and not sufficiently addressed. Experimental tests on words with diacritics in prior models have shown bad accuracy that does not exceed 60%. Consequently, this work aims to introduce a new, accurate deep-learning model for Arabic OCR that considers words with and without diacritical marks. It utilizes a dual encoder transformer (DTrOCR), a deep-learning architecture that has demonstrated superior performance in identification and classification tasks. The proposed DTrOCR creates multi-batch sizes. It has been trained using a comprehensive, generated Arabic word-based dataset named MFSRHRD and tested on unseen datasets. The accuracy of configuring Arabic words without diacritics reaches 98.5%. However, for words with diacritics, it achieved an accuracy of 89.9%.

KEYWORDS

Arabic OCR, Multi-batch size, Transformer, Dual encoder transformer, Decoder, Feature extraction, Self-attention mechanism.

1. INTRODUCTION

Optical character recognition (OCR) is a sub-discipline of pattern recognition and computer vision. OCR has received more and more attention and has become a popular and promising research area in computer and pattern-recognition communities. However, recognizing documents with Arabic text contents is a popular and actively developed field. The main objective of the OCR system is to convert the images of a document, whether printed or hand-written, into computer-editable text to generate digital copies of text documents [22], [7], [17]. Moreover, OCRs have several applications, such as archive organization, automated plate recognition and automated ticketing [10].

The process of Arabic OCR encounters several challenges due to the distinctive features of the Arabic script. Arabic is a right-to-left written language that consists of twenty-eight letters. Each letter can have several forms based on its place inside a word, whether at the beginning, middle, end, or isolated. The language does not differentiate between capital and lowercase letters. Dots, also known as "Ijam," and diacritical markings, also known as "Tashkeel," introduce complication by distinguishing letters and modifying the meanings of words. In addition, the Hamza can occur in several locations and it can be challenging to differentiate letters with similar structures, such as Saad and Taa. Characters such as Raa, Dal and Waw, which do not form a connection with the letter that follows them, add to the complexity of separating words, …etc.

Researchers have recently used artificial-intelligence (AI) mechanisms to recognize Arabic characters, words and printed texts. The deep-learning models have become a significant player in Arabic language recognition. Previous studies have encountered several common challenges, such as a lack of diverse and balanced datasets, rare diacritics considerations, limited model accuracy, especially when dealing with diacritics and the model's capacity to generalize [25], [22], [9], [7].

To address these specific challenges, our motivation behind using a Dual Encoder Transformer is to allow the model to learn from both global and local patterns in the input images. In Arabic, very small marks, such as diacritics or dots, can completely change the meaning of a word. Traditional models with a single encoder may struggle to capture both overall word structure and fine-grained details at

1. K. Gaashan is with Software Engineering Department, Philadelphia University, Amman, Jordan. Email: khuloodgaashan@gmail.com
2. M. Bani Younes is with Faculty of Information Technology, American University of Madaba, Jordan. Email: m.baniyounes@aum.edu.jo

the same time. By using two encoders with different patch sizes, our model can analyze the broad shape of the word and also focus on subtle features like diacritics. This dual-path design makes it more capable of handling the complex nature of Arabic script and improves recognition accuracy, especially in the presence of diacritics.

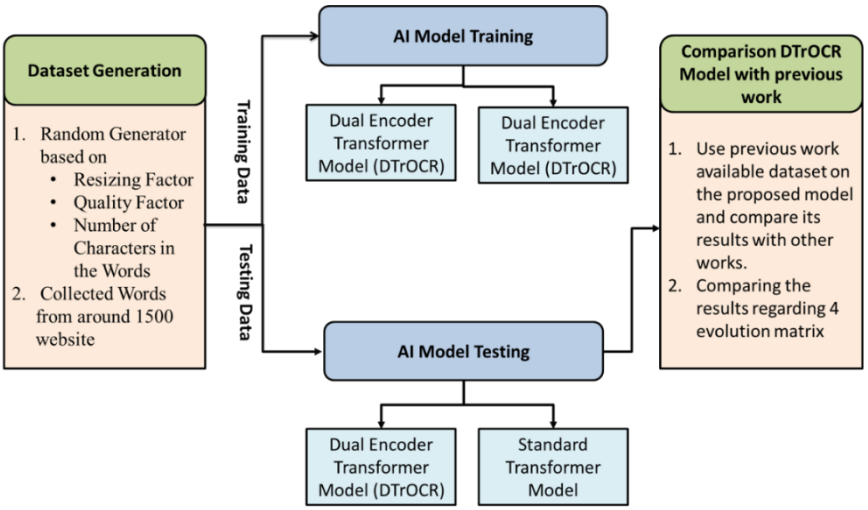


Figure 1. Test and train DTrOCR model.

We propose a Dual Encoder Transformer (DTrOCR) model in this work. The key feature of the proposed model is its ability to process input images by splitting them into multi-batch sizes instead of single-batch sizes, enabling more effective feature extraction. It was trained on our previously proposed comprehensive generated Arabic dataset MFSRHRD [6]. This dataset, created specifically for Arabic characters and word recognition, was chosen for its high number of records, comprehensive coverage and diversity of font sources. This dataset also includes many Arabic words with diacritics, which most previous Arabic datasets have neglected. The dataset was divided into 80% for training and 20% for testing, ensuring controlled conditions for model evaluation.

Figure 1 illustrates the general flowchart for training and testing the proposed model. As we can see from the Figure, besides testing the performance of the Dual Encoder Transformer, a standard Transformer model was trained on the same dataset [6]. This is to establish a performance baseline and illustrate the benefits of the proposed model. The Dual Encoder Transformer outperformed the standard model, achieving higher accuracy and generalization across different fonts and writing styles. The model's performance was evaluated using four main metrics: accuracy, precision, recall and F1-score. Moreover, the proposed model (DTrOCR) was further tested on other unseen datasets, including APTI, IFN/ENIT and MMAC, where it continued to demonstrate high performance compared to previous OCR models, showcasing its robustness and effectiveness.

In general, the main contributions of this work are summarized by:

- Introducing an enhanced Arabic OCR system using a Dual-encoder transformer (DTrOCR), which improves the features extracted from images using multi-batch sizes instead of a single-batch size.
- Testing the proposed model's efficiency and generalization in terms of recognizing unseen Arabic words with and without diacritics.

The rest of this paper is structured as follows: Section 2 investigates previous deep-learning techniques for Arabic OCR. It illustrates the main considerations and specifications of these models. Then, it identifies the main gaps and the needed work in this field of research. Section 3 introduces the details of the proposed Arabic OCR model (DTrOCR). It outlines the main phases, considerations and steps of the model. Section 4 presents the details of the training and testing processes, including the experimental setup and sequential phases. Section 5 evaluates the performance of the Arabic OCR model on unseen datasets and tests its efficiency. Section 6 concludes the entire paper with a summary of key findings, directions for future research and recommendations for researchers in this field of study.

2. RELATED WORK

Optical Character Recognition (OCR) technology has been widely used over the past few decades, aiming to transform images of text into editable texts. It mainly aims to digitize and process text information. Regarding Arabic OCR, the challenges are more complicated due to the complex nature of the Arabic script, as discussed earlier. This section provides an overview of the previous studies that developed Arabic OCR. It identifies the main methodologies and challenges in this field of research, tracing the main progress made and highlighting the gaps and required work.

Previous Arabic OCRs have been developed using several deep-learning and artificial-intelligence algorithms. First, several studies [7], [19], [15] have combined multiple popular and strong deep-learning techniques, such as convolutional neural network (CNN), Long Short Term Memory (LSTM) and connectionist temporal classification (CTC), …etc. Other studies [16], [14], [13] have considered the Generative Adversarial Networks (GANs) to enhance the quality of the OCR models. Recently, researchers started using transformers to develop more accurate Arabic OCR [23], [7].

Several researchers have combined multiple machine/ deep-learning algorithms to obtain accurate Arabic OCRs. For instance, Z. Noubigh et al. [21] presented a model that combines the CNN, Bidirectional Long Short-Term Memory (BLSTM) and CTC algorithms. The model was trained and tested using the KHATT and HACDB datasets [26] to classify Arabic characters. It achieved high acceptable accuracy and the character error rates (CERs) of this model were 2.74% and 2.03% on the KHATT and HTID datasets, respectively.

Dahbali, Aboutabit and Lamghari [4] proposed a hybrid model combining CNN with attention (CBAM) and BLSTM to improve Arabic handwritten script recognition. The model integrates spatial and sequential features using Connectionist Temporal Classification (CTC) decoding and data augmentation. It was evaluated on the KHATT dataset and achieved significant improvements in recognition accuracy, outperforming prior approaches. Al-Taani and Ahmad [18] used Residual Neural Networks (ResNet) for Arabic handwritten character recognition. Their model was tested on several benchmark datasets including MADBase, AIA9K and AHCD. The approach achieved up to 99% accuracy, demonstrating the benefit of deep residual learning architectures in handling the variability of Arabic handwriting.

Shtaiwi et al. [11] proposed an end-to-end machine-learning solution for recognizing handwritten Arabic documents that combines several deep-learning models, resulting in improved robustness and accuracy on real-world datasets. [11] proposed an Arabic OCR model that combined Convolutional Recurrent Neural Network (CRNN) and BLSTM. This model aims to recognize Arabic handwritten content based on the character unit as well. It was trained on the MADCAT dataset [24] and achieved a CER of 3.96%. On the other hand, M. Boualam et al. [15] proposed an OCR model that combined CNN, RNN and CTC models. The proposed OCR model aims to recognize text from handwritten village names in Tunisia using the IFN/ENIT dataset [30]. This model classifies Arabic characters and Arabic words. It achieved a CER of 2.10% and a word error rate (WER) of 8.21%. Fasha et al. [22] merged CNN and BLSTM with a CTC loss function, achieving a character recognition rate (CRR) of 98.76% and a word recognition rate (WRR) of 90.22%.

Generative adversarial networks (GANs) have also been explored to improve the performance of Arabic OCR. GANs do not directly perform character recognition in OCR; they can significantly enhance the performance and robustness of OCR systems. This is achieved by improving data quality and increasing the diversity of training datasets. Several studies have been introduced using this mechanism in the literature. First, Y. Alwaqfi et al. [16] early explored GAN-based models. M. Eltay et al. [14] also utilized GANs for adaptive augmentation. This model achieved an accuracy of 95.51% on the character-recognition level and 89.52% on the word-recognition level. Moreover, A. Mostafa et al. [17] faced challenges in ensuring the quality of generated samples, particularly in dealing with connected Arabic letters, making the results less conclusive with an accuracy that reached 95.08%. S. Jemni et al. [13] demonstrated high proficiency in Arabic and English OCR using GANs with a hybrid model CNN-RNN-CTC, which achieved 75.6% accuracy. However, complexities in combining deep-learning models at each stage were reported in this model.

Recently, transformer-based models have become increasingly popular for Arabic-text recognition. A. Mustafa et al. [17] developed a specialized dataset using thirteen unique web typefaces, including

Table 1. Previous Arabic OCR deep-learning models with their limitations.

Ref.	Technique	Dataset	Recognition Level	ACC	Description	Limitation(s)
[21]	CNN-BLSTM-CTC	KHATT, AHTID	Word-level	97%	Combined CNN for feature extraction and BLSTM for sequence prediction	The test was conducted on 901 samples from AHTID dataset
[11]	CRNN-BiLSTM	MADCAT	Paragraph-level	96.04%	CRNN-BiLSTM model integrates Convolutional Recurrent Neural Networks (CRNN) with Bidirectional Long Short-term Memory (BiLSTM) layers	Requires significant time to train due to additional layers of BiLSTM
[15]	CNN-RNN-CTC	IFN/ENIT	Word-level	97.90%	CNN layers for feature extraction, followed by RNN layers for sequential data processing	No real testing was performed of generalization
[22]	CNN-BiLSTM-CTC	Custom dataset	Word-level	98.76%	Five CNN layers for feature extraction and two BiLSTM layers for sequence prediction	When tested on noisy images, the accuracy drops drastically to 22.71 CER and 85.82% WER
[16]	GANs-CNN	AHCD	Character-level	99.78%	GANs for data augmentation with CNN for classification	Error rate, training and testing ratios and number of images after applying augmentation were unclear
[14]	GANs-BiLSTM	IFN/ENIT, AHDB	Word-level	95.8%	ScrabbleGAN for augmentation followed by BiLSTM for recognition	Still suffers when tested on challenging test sets
[17]	CDCGAN-CNN	AHCD	Character-level	95.08%	GANs for data generation combined with CNN for classification	Generation of characters with dots is still a challenging task. Additionally, classes can be unbalanced
[13]	GANs-CNN-RNN-CTC	KHATT	Word-level	75.6%	The generator employs U-net, while the discriminator uses CNN. For OCR model, CNN, followed by two layers of Bi-GRU and then a CTC layer, are stacked	Very complex model in each step of the OCR process, yet it produces an accuracy close to that of the baseline model
[17]	CNN-Transformer	Custom dataset, KHATT	Word-level	92.7%	ResNet101 combined with Transformer for sequence processing	Shortage of resources and computational power; complex model
[12]	Transformer With ross-attention	Custom dataset, KHATT	Word-level	81.55%	Transformer architecture with cross-attention mechanisms for better feature extraction	Some weights were randomly initialized, leading to a limited Accuracy
[8]	Transformer	KHATT	Word-level	97.7%	Transformer architecture for both image understanding and wordpiece-level text generation	Synthesized dataset for pre-training consists of 2.2 M images, which is relatively small compared to other methods that use hundreds of millions of images
[1]	QARI-OCR (vision-language model)	Diacritics-rich synthetic +real printed Arabic texts	Page/Word-level	98%	Vision-language Model fine-tuned from Qwen2-VL, optimized for Arabic script and diacritic handling	Limited font variety, fixed font size, no handwriting support; mostly trained on synthetic data
[3]	HATFormer (Transforme based HTR)	Historical r-handwritten Arabic manuscripts dataset	Line/Word-level	95%	Transformer encoder-decoder customized for historical Arabic handwriting with diacritic support	Small training dataset; still moderate error rate; limited to historical handwritten text
[2]	Hybrid CNN+ Transformer OCR system	Printed & Handwritten Arabic text (with digits)	Character & Word	99.4%	A hybrid CNN-Transformer OCR system with excellent accuracy for printed Arabic text (CER = 0.59%) and competitive performance on handwriting (CER = 7.91%). It also includes effective text detection (F-measure 79%).	Still struggles with handwritten Arabic (higher WER) and shows sub-optimal text detection performance on complex backgrounds or irregular handwriting.

hand-written samples from the KHATT dataset. The proposed model employed a two-part system: a CNN for feature extraction and a transformer model with four encoders and decoders. This model considers character-level and word-level configurations and achieves a CER of 7.27% and a WER of 8.29%. Besides, Momeni et al. [12] examined two types of transformers: transformer transducer and transformer with cross-attention, using a synthesized dataset of 500,000 printed Arabic images and the KHATT dataset for testing. The transformer with cross-attention achieved a CER of 18.45%, outperforming the transformer transducer, which achieved 19.76% CER. Most studies using transformers for Arabic OCR, such as ALNASIKH [8] and OCFormer [17], have leveraged standard transformer models, like TrOCR or vision transformers. These existing models primarily focus on utilizing single encoder-transformer architectures for Arabic handwritten and printed text recognition.

Table 1 systematically illustrates the main considerations and limitations in the previous studies in this field of research. After reviewing earlier studies in this field of research, it is clear that transformer-based models are emerging as strong contenders. This is due to their attention mechanisms and ability to process long sequences. A dual encoder transformer has been used in the medical field [5]. However, it has not been applied before for the Arabic OCR. In recent years, researchers have made many improvements to the standard transformer model to enhance its performance in recognizing Arabic characters. These changes have focused on the encoder and decoder parts of the model. However, one challenge remains the issue of a single batch size during training, which can limit efficiency. A promising solution to this problem is using multi-batch size inputs. To achieve this, we employ a dual encoder transformer, a method that has been successful in the medical field for image recognition. In this research, we apply this approach to improve the recognition of Arabic characters. Moreover, we aim to investigate the ability of the Dual Encoder to be implemented to recognize Arabic letters and words.

3. THE ENHANCED PROPOSED ARABIC OCR (DTROCR)

In this work, we developed a deep-learning model for recognizing Arabic letters and words using a dual encoder transformer. As shown in Figure 4, the model consists of two different encoders, each designed to extract unique features from the input data based on different batch sizes. The outputs from these encoders are combined using a fusion mechanism based on two multi-head attention layers. This step helps the model merge the features effectively, ensuring that the most important information is retained. Then, the fused features are sent to the decoder, which produces the final output. This approach helps the model better understand the complex nature of Arabic script. It highlights the role of the external fusion layer in achieving more accurate recognition results. The input image is divided into multiple-batch sizes (N) using two dual encoders. The model partitions the input image based mainly on its size, as illustrated in Equation (1).

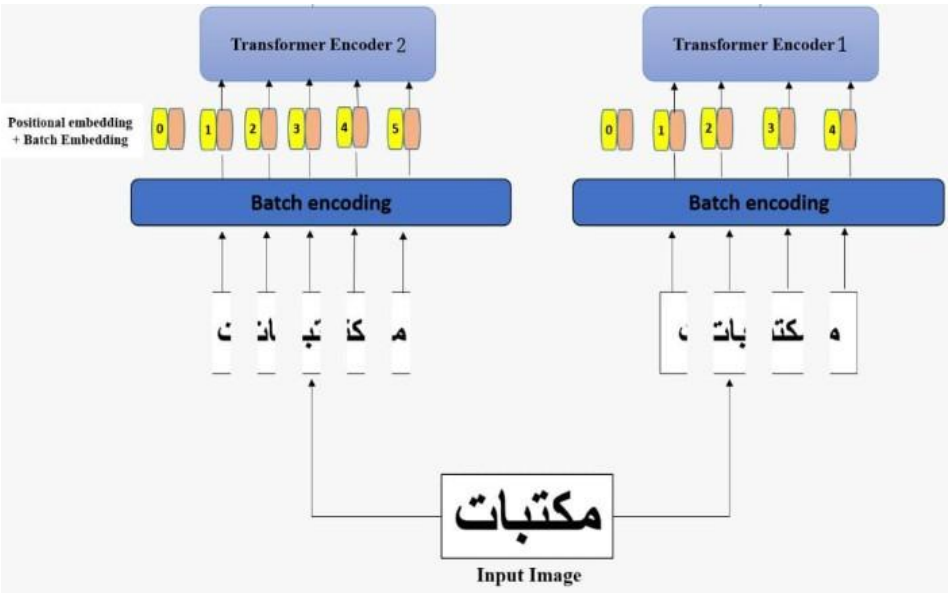


Figure 2. Multi-batch sizes in DTrOCR model.

$$N = \frac{HW}{P^2} \quad (1)$$

where: *N:* is the total number of batches; *H:* is the height of the input image, *W:* is the width of the input image and *P:* is the batch size.

Each encoder has multiple attention layers, enabling the extraction of diverse features from the same image. Using dual encoders with varying multi-attention layers allows the model to effectively extract a wide range of features from the same input image. This enhances the overall performance in Arabic optical character recognition.

Figure 2 illustrates the process beginning with an input image of an Arabic word (مكتبات). The image is divided into batches of multiple sizes, with each patch fed into one of the two different encoders in our model. Each encoder processes the batches independently, capturing unique features. The outputs from both encoders are then combined using a multi-head attention mechanism, allowing the model to learn more nuanced details from each perspective. This multi-batch-size approach enhances OCR accuracy by leveraging two unique views of the data, making it especially effective for recognizing the complex characteristics of Arabic script.

To better illustrate how the proposed dual encoder architecture processes input images with different batch sizes, as illustrated in the proposed model (Figure 4), the dual-encoder architecture extracts two complementary types of features from the input image. Encoder 1 (E1), which operates on smaller patch sizes, focuses on capturing fine-grained local features, such as character edges, diacritical marks and subtle variations between visually similar characters. This detailed representation allows the model to disambiguate closely related characters with high precision. In contrast, Encoder 2 (E2), which processes larger batch sizes, extracts global and coarse-grained features, capturing the overall word shape, inter- character spacing and distribution patterns. These higher-level features provide a more holistic view of the input, allowing the model to understand the context and structural layout of the word as a whole.

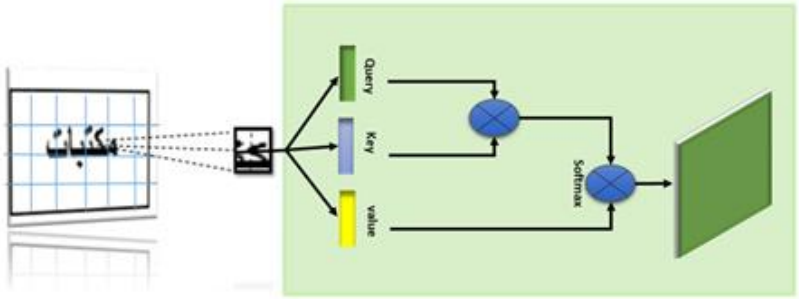


Figure 3. Self-attention mechanism.

To leverage the complementary strengths of both encoders, a feature-fusion mechanism based on multi- head attention is applied. This mechanism combines fine and coarse representations into a single rich feature representation, which is then passed to the decoder. The fused representation enables the decoder to reconstruct the textual output with improved accuracy, as it benefits from both local detail and global context simultaneously. This fusion strategy is a key component of the proposed model's superior performance compared to a standard single-encoder Transformer.

As shown, the input image is split into two versions of batch sequences—each with a different batch size. The first sequence (Input 1) uses larger batches and is passed to E1, which is optimized to capture high-level structural and global features of the word. The second sequence (Input 2) uses smaller batches and is sent to E2, which focuses on detecting fine-grained local features, such as diacritics and subtle visual differences between characters.

Both encoders (E1 and E2) process their respective inputs through multiple layers of multi-head attention and feed-forward networks. Their outputs are then merged using a two-stage multi-head attention fusion block. This external fusion mechanism allows the model to integrate complementary information from both encoders, resulting in a richer and more balanced feature representation. The fused features are then passed to the decoder for final output generation.

This multi-batch dual encoder design allows the model to effectively balance between general layout understanding and fine detail detection, which is particularly important for complex Arabic text with diacritics. Figure 4 illustrates the overall architecture of the proposed DTrOCR model and how the dual encoders collaborate using multi-batch sizes. The input image is divided into two different patch sets with varying sizes. These are then independently fed into two transformer-based encoders: E1 and E2. E1 receives larger batches and focuses on capturing global structural patterns of the word, while E2 processes smaller batches to detect fine-grained local details, such as dots and diacritics, which are essential for distinguishing similar Arabic characters. Each encoder applies multiple layers of multi-head attention and feedforward networks to extract high-level features from its respective input. The outputs from both encoders are then passed to a fusion module composed of stacked multi-head attention layers. This fusion stage is responsible for integrating both global and local information into a single comprehensive feature representation. The fused features are then sent to the decoder, which generates the final output text. This dual-encoder, multi-batch approach allows the model to learn rich and diverse features from the same input image, which significantly improves recognition accuracy.

The proposed model is discussed in the following steps:

1) ***Pre-processing Input Image:*** The model starts by preparing the input images of the Arabic words into a standard format that facilitates their usage in the proposed OCR model. All images are resized to a standard size of 512 x 512 pixels. Thus, the images are all of the same size, which makes them easier to process. Then, the pixel values are adjusted to a consistent range, which helps the model learn more effectively. Finally, the images are grouped into batches, which enables the model to handle several images at once. This makes the processing faster and more efficient.

2) ***Setting the Encoder Parameters:*** Using the dual-encoder model aims to improve the recognition of Arabic diacritics like ("damma," "kasra", …etc.) and small text features. Traditional OCR models miss these details, as diacritics are easy to overlook. Table 2 illustrates different hyper-parameters for two different decoders, depending on the task of each encoder. The first encoder is set to receive a larger batch size to understand the broader context and global features of the image. The second encoder receives a smaller batch size to focus on finer details and local features. After encoding, the latent representations from both encoders are concatenated to form a comprehensive feature vector using the multi-head attention-based integration concatenation mechanism.

Table 2. Hyper-parameter comparison between first and second encoders.

Hyper-parameter	**Encoder 1**	**Encoder 2**
Patch Size (p)	32 pixels	16 pixels
Embedding Dimension	512	768
Number of Layers	6	8
Number of Attention Heads	8	12
Feedforward Dimension	2048	3072
Dropout Rate	0.2	0.1
Batch Size	32	16
Calculated Patches (N)	256	1024
Learning Rate	1e-4	1e-4
Optimizer	Adam	Adam
Positional Encoding	Sinusoidal	Sinusoidal

3) ***Multi-head Attention Outside the Encoder:*** Final outputs of encoders E1 and E2 are merged before sending them to the decoder. Merging these outputs allows the model to gather features learned by both encoders. This gives the decoder a more comprehensive view and enables it to achieve more accurate results. Additionally, having Multi-Head Attention outside the encoder lets the combined features be processed again as a preparatory step before reaching the decoder. This extra step can help refocus attention on specific details from each input, improving the model's accuracy by ensuring that the most relevant information is emphasized.

Unlike models like TrOCR and OCFormer that use only one encoder with one fixed batch size, our model uses two encoders—each with a different batch size. One focuses on the big picture of the Arabic word, while the other looks closely at the small details, like dots and diacritics. This is very useful in Arabic, where small marks can completely change the meaning of a word. By combining what both encoders see, our model builds a better understanding of the word and makes more accurate predictions. This approach helps our model perform better than other models, especially when dealing with complex Arabic writing.

4) ***Self-attention Mechanism:*** The attribute that distinguishes transformers most is the self-attention mechanism. Figure 3 graphically illustrates the self-attention mechanism on an input image. In this work, the purpose of self-attention is to calculate the connections between various components of the feature vector to capture interdependency, following these sequential steps:

- Query, Key, Value (Q, K, V) Vectors: The feature vector is transformed into separate Q, K and V vectors using projection.
- Attention scores are calculated by taking the dot product of the query and key vectors and then applying a softmax operation to obtain attention weights.
- Context vector is derived by multiplying the attention weights with the value vectors, so highlighting significant characteristics.

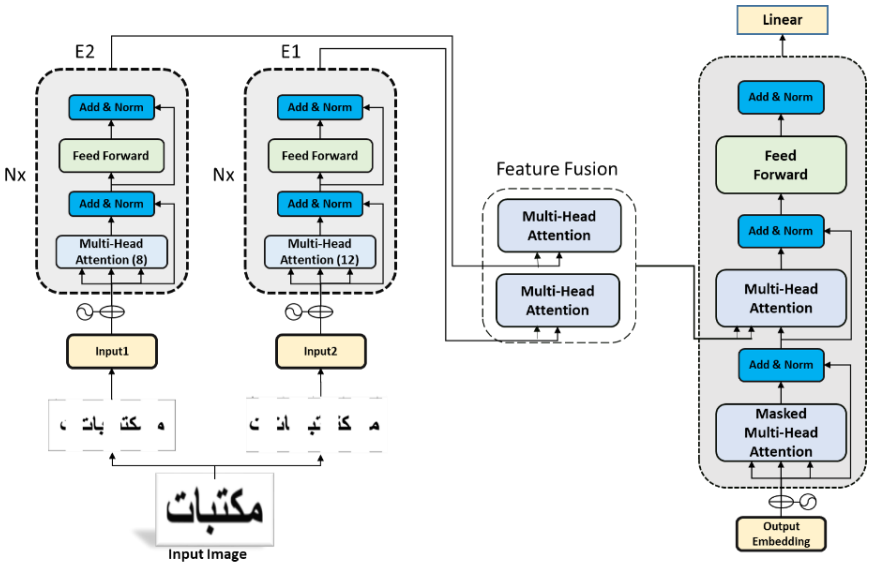


Figure 4. Proposed DTrOCR model.

4. Training and Testing the Proposed DTrOCR Model

The proposed DTrOCR model was trained using the Adam optimizer with a learning rate of 1e-4 and early stopping was applied to avoid overfitting. Hyper-parameter tuning was conducted to enhance the model's performance. The optimal configuration was determined through a grid-search approach, where various combinations of hyper-parameters were evaluated, selecting the configuration that achieved the highest validation accuracy for Arabic OCR tasks.

Careful tuning of hyper-parameters was essential to achieve the high performance of the proposed DTrOCR model. In particular, the choice of batch size, embedding dimensions and number of attention heads significantly influenced the model's ability to capture both global and local features of Arabic script. For Encoder 1, a larger batch size (32×32) and eight attention heads were selected to focus on extracting broader contextual information and global structural dependencies, which are critical for recognizing the general word shape and layout. Encoder 2 was configured with a smaller batch size (16×16) and twelve attention heads to focus on fine-grained details, such as diacritical marks and subtle character variations. This combination was determined through an extensive grid search, where different hyper-parameter configurations were compared based on validation accuracy and F1-score. Models with smaller batch sizes showed better generalization but slower convergence, whereas larger batch sizes improved training stability. The final configuration balanced these effects,

achieving faster convergence without sacrificing the ability to generalize to unseen datasets. As a result, the DTrOCR model achieved a 9.3% improvement in accuracy over the baseline Transformer, demonstrating the critical role of hyper-parameter selection in enhancing the model performance.

The dataset used for training the DTrOCR model was generated by our previous work [6]. It featured a diverse collection of Arabic text images that simulated various real-world scenarios, including different fonts, styles and a high number of recorders. Pre-processing techniques were applied to enhance the data's quality, such as noise reduction, using the same software used for the dataset generated to improve image clarity. These steps were essential to ensure that the model could accurately detect and recognize characters, even in challenging conditions, thus enhancing its overall robustness and generalization.

This generated dataset (MFSRHRD), which means Multiple Fonts, Sizes, Resources and High Records dataset, aims to fill the gaps in the available and open-source Arabic datasets. It faces all the challenges and weaknesses of the previously proposed datasets. Thus, the training process can be completed comprehensively. To obtain a model characterized by the generalization feature, the dataset must be large enough to include all possible scenarios in the trained machine. Figures 5, 6 show samples of the used dataset.

Figure 5. Samples of the used dataset.

Figure 6. Samples of characters with Tashkeel in MFSRHRD dataset.

In training the proposed DTrOCR model, the batch sizes for the two encoders were selected based on the objective of capturing both global and local features. Larger batches enable the model to extract structural information from the overall word shape, while smaller batches allow it to focus on fine-grained details, such as diacritics. Regarding the number of attention heads, a hyper-parameter tuning strategy using grid search was employed, where multiple configurations were evaluated and the one yielding the highest validation accuracy on the development set was chosen. This approach is commonly used in deep-learning research to ensure optimal model performance.

4.1 Technical Details

In this sub-section, we will provide more technical details regarding the training environment and implementation setup. The proposed model was trained using an NVIDIA RTX 3090 GPU with 24GB of VRAM. Although the dataset is large, we handled it efficiently by using mini-batch training and a custom-data generator that loads image batches on-the-fly from disk during training. The total training time was approximately 120 hours.

The model was trained for 100 epochs with early stopping to avoid overfitting. Input images were pre-processed through resizing (512×512 pixels), normalization and noise reduction to ensure data quality and consistency. A detailed configuration of the encoder layers, batch sizes and attention mechanisms is already summarized in Table 2.

Furthermore, we provide open access to our dataset-generation software and a sample of the generated MFSRHRD dataset to encourage reproducibility. These resources are available on GitHub at: https: //github.com/KhuloodGaashan/arabic-ocr-dataset.

Moreover, three other datasets for testing (IFN/ ENIT [30], APTI [28], MMAC [27]) were used to check our model generalization and performance using unseen datasets, printed and handwritten. Figure 7 illustrates samples of these datasets.

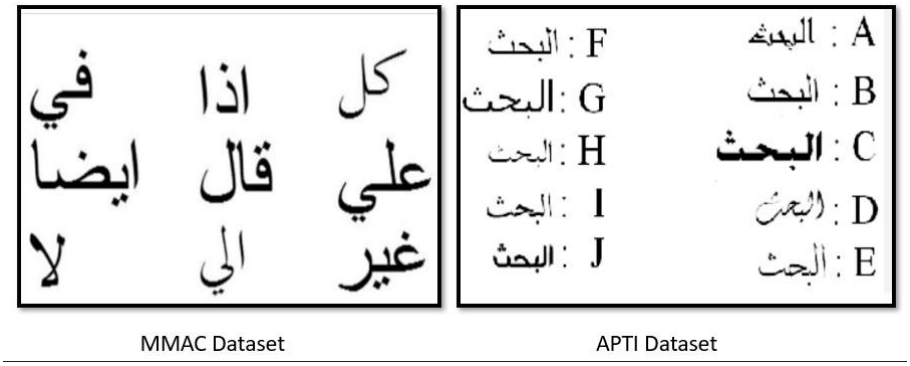


Figure 7. Samples from APTI & MMAC datasets.

5. PERFORMANCE EVALUATION

We evaluated the DTrOCR model using the MFSRHRD dataset and other unseen datasets. We employ several standard evaluation metrics to measure the performance of our Arabic OCR system. These metrics include accuracy, precision, recall, F1-score and the confusion matrix [20]. The details and equations used to compute these metrics are presented below:

- Accuracy: It measures the overall correctness of the system's predictions by calculating the ratio of correctly classified instances to the total instances.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

- Precision: It quantifies the proportion of correctly predicted instances out of all the instances predicted as Arabic OCR.

$$Precision = \frac{True\ Positives}{True\ Posiives + False\ Positives}$$

- Recall: It measures the proportion of correctly predicted Arabic OCR instances out of all the actual Arabic OCR instances.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

- F1-score: It provides a balanced measure of precision and recall, taking into account both metrics to evaluate the system's performance.

$$F1 - score = 2.\frac{Precision\ .Recall}{Precision + Recall}$$

5.1 Results & Analysis

Recognizing Arabic words using the DTrOCR model achieved outstanding results compared to previous models [22], [15], [8], [12]. The recognition accuracy reached 99.3%, demonstrating a significant improvement. The proposed model (DTrOCR) consistently outperformed previous methods in terms of recognition accuracy. This confirms its ability to recognize and thus accurately enhance Arabic OCR systems.

Figure 8 presents a comparison of the accuracy of various models, with our proposed model, DTrOCR (2024), achieving the highest accuracy at 99.30% compared to previous models. These results highlight the superior performance of DTrOCR (2024) in enhancing OCR accuracy.

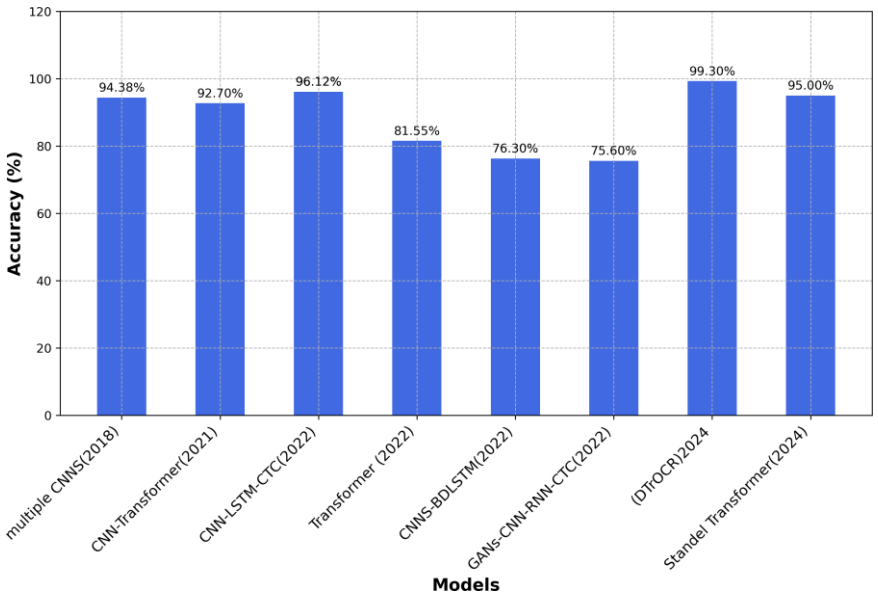


Figure 8. Comparing the accuracy of existing models with that of the DTrOCR model.

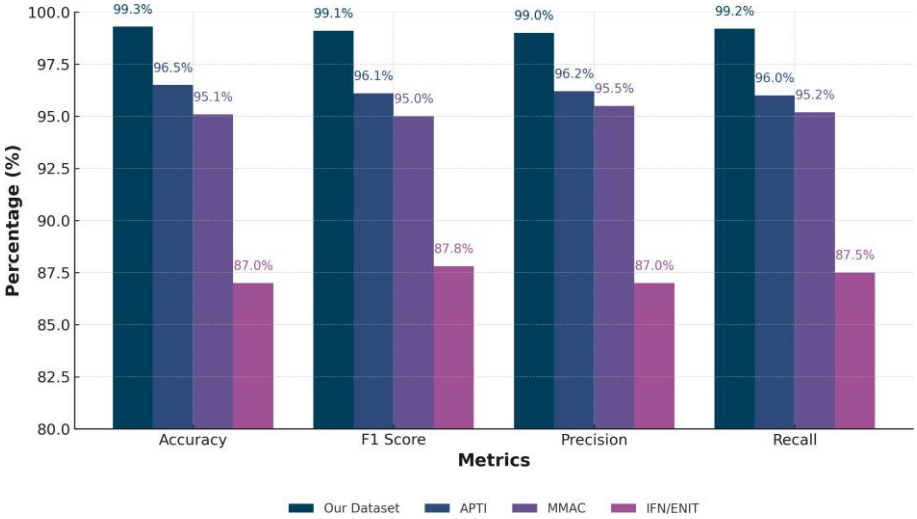


Figure 9. Evaluation matrix for the DTrOCR model using different datasets.

Using our previously generated dataset MFSRHRD, we trained two models, the first model was the DTrOCR Transformer and the second model was the Standard Transformer on the same dataset. The DTrOCR achieved higher and better results than the Standard Transformer and the results are illustrated in Table 3.

Table 3. Comparison between DTrOCR and standard transformer.

Model	Accuracy	F1-Score	Precision	Recall
DTrOCR	99.3%	99.1%	99.0%	98.2%
Standard Transformer	90.0%	89.3%	89.9%	84.0%

After completing the training process for the DTrOCR model, it was tested on an additional dataset to evaluate its generalization capabilities and validate the improvements observed during training. When tested on the custom dataset, the Dual Encoder Transformer maintained its high accuracy, as shown in Table 4 and Figure 9.

Furthermore, for the accuracy of the model when dealing with diacritics, it achieved high accuracy, reaching 89% compared to the models that dealt with diacritics before, but less accuracy compared to without diacritics using our generated dataset. Table 5 illustrates the results of DTrOCR when testing it using a dataset with diacritics and without diacritics.

Table 4. Comparison of evaluation metrics for different datasets.

Dataset Used for Test	Accuracy	F1-Score	Precision	Recall
Our Dataset	99.3%	99.1%	99.0%	99.2%
APTI	96.5%	96.1%	96.2%	96.0%
MMAC	95.1%	95%	95.5%	95.2%
IFN/ENIT	87.0%	87.8%	87%	87.5%

Table 5. Comparison of model with and without Tashkeel.

Model	Accuracy	F1-Score	Precision	Recall
With Tashkeel	89.1%	89.0%	88.7%	90.1%
Without Tashkeel	99.3%	99.1%	99.0%	99.2%

5.2 Limitations

Although the model has achieved high performance, we observed that some errors occur in cases involving overlapping or poorly positioned diacritical marks. For instance, when the shadda and fatha are closely placed on a letter such as Seen, the model may misclassify it as Sheen. These specific failure cases highlight the limitations of the current system in distinguishing fine-grained features.

One of the most common errors was the confusion between Sheen and Seen when shadda overlapped with fatha, resulting in an incorrect character interpretation. Similarly, the model frequently misclassified Dal as Thal when the damma was slightly shifted or faint. We also observed consistent difficulty in distinguishing between Kaf and Faa in cases where the kasra was small. In addition, Taa was sometimes confused with Thaa when the sukun was not clearly printed.

Another recurring problem involved stacked diacritics, such as "shadda and kasra" or "shadda and damma", which the model occasionally detected only partially, leading to either missing diacritics or duplicated outputs. Some samples revealed that the model entirely ignored diacritics when multiple marks were close to each other, producing undiacritized text instead. We also noticed alignment errors where diacritics were shifted to the wrong character, particularly in dense handwritten-like fonts.

6. Conclusion and Future Work

This study introduced a deep-learning model, Dual Encoder Transformer (DTrOCR), designed to enhance Arabic Optical Character Recognition (OCR) by recognizing both discretized and non-discretized words. The dual-encoder approach is applied to Arabic word recognition, enhancing the feature-extraction process by utilizing two encoders that collaborate. Before entering the decoder, we implement a merging process for the features extracted from both encoders using two multi-head attention layers. To ensure that the most relevant information is combined and passed on for further processing, this merging step enhances the model’s ability to capture and utilize complementary features, leading to improved recognition accuracy.

The model was trained on the MFSRHRD dataset, which includes both types of words and achieved the following results: 99.3% accuracy for non-diacritized words and 89.9% accuracy for diacritized words, outperforming previous models that struggled with diacritics. To test generalization, the DTrOCR was evaluated on new datasets it had not previously seen and it maintained a strong performance compared to older models, demonstrating its reliability for accurate Arabic-text recognition. In future work, enhancing diacritic recognition remains a crucial challenge.

In future studies, to address the limitations of the model, we propose incorporating specialized attention mechanisms and employing multi-task learning frameworks explicitly designed to capture and differentiate diacritical features. Additionally, exploring model-optimization techniques to reduce computational costs and improve training efficiency will be essential. Lastly, further architectural refinements could facilitate faster training and enable deployment in resource-constrained environments.

References

[1] A. Wasfy et al., "QARI-OCR: High-fidelity Arabic Text Recognition through Multimodal Large

Language Model Adaptation," arXiv preprint, arXiv: 2506.02295, 2025.
[2] A. Waly, B. Tarek, A. Feteha, R. Yehia, G. Amr, W. Gomaa and A. Fares, "Invizo: Arabic Handwritten Document Optical Character Recognition Solution," arXiv preprint, arXiv: 2502.05277, 2025.
[3] A. Chan, A. Mijar, M. Saeed, C.-W. Wong and A. Khater, "HATFormer: Historic Handwritten Arabic Text Recognition with Transformers," arXiv preprint, arXiv: 2410.02179, 2025.
[4] M. Dahbali, N. Aboutabit and N. Lamghari, "A Hybrid Model for Arabic Script Recognition Based on CNN-CBAM and BLSTM," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 10, no. 3, pp. 294–305, DOI: 10.5455/jjcit.71- 1709571516, Sep. 2024.
[5] S. Raminedi, S. Shridevi and D. Won, "Multi-modal Transformer Architecture for Medical Image Analysis and Automated Report Generation," Scientific Reports, vol. 14, no. 1, p. 19281, 2024.
[6] K. Gaashan and M. B. Younes, "Deep Learning-based Arabic Optical Character Recognition: A New Comprehensive Dataset at Character and Word Levels," Proc. of the 2024 15th IEEE Int. Conf. on Information and Communication Systems (ICICS), pp. 1–6, Irbid, Jordan, 2024.
[7] R. Najam and S. Faizullah, "Analysis of Recent Deep Learning Techniques for Arabic Handwritten-text OCR and Post-OCR Correction," Applied Sciences, vol. 13, no. 13, p. 7568, 2023.
[8] A. Mortadi et al., "ALNASIKH: An Arabic OCR System Based on Transformers," Proc. of 2023 IEEE Int. Mobile, Intelligent and Ubiquitous Computing Conf. (MIUCC), pp. 74–81, Cairo, Egypt, 2023.
[9] S. Faizullah, M. S. Ayub, S. Hussain and M. A. Khan, "A Survey of OCR in Arabic Language: Applications, Techniques and Challenges," Applied Sciences, vol. 13, no. 7, p. 4584, 2023.
[10] S. Alghyaline, "Arabic Optical Character Recognition: A Review," Computer Modeling in Engineering & Sciences, vol. 135, no. 3, pp. 1825–1861, 2023.
[11] R. E. Shtaiwi, G. A. Abandah and S. A. Sawalhah, "End-to-End Machine Learning Solution for Recognizing Handwritten Arabic Documents," Proc. of the 2022 13th IEEE Int. Conf. on Information and Communication Systems (ICICS), pp. 180–185, Irbid, Jordan, 2022.
[12] S. Momeni et al., "Arabic Offline Handwritten Text Recognition with Transformers," Research Square, DOI: 10.21203/rs.3.rs-2300065/v1, 2022.
[13] S. K. Jemni et al., "Enhance to Read Better: A Multitask Adversarial Network for Handwritten Document Image Enhancement," Pattern Recognition, vol. 123, p. 108370, 2022.
[14] M. Eltay et al., "Generative Adversarial Network-based Adaptive Data Augmentation for Handwritten Arabic Text Recognition," PeerJ Computer Science, vol. 8, p. e861, 2022.
[15] M. Boualam et al., "Arabic Handwriting Word Recognition Based on Convolutional Recurrent Neural Network," Proc. of the 6th Int. Conf. on Wireless Technologies, Embedded and Intelligent Systems (WITS 2020), pp. 877–885, Springer, 2022.
[16] Y. M. Alwaqfi, M. Mohamad and A. T. Al-Taani, "Generative Adversarial Network for an Improved Arabic Handwritten Characters Recognition," Int. Journal of Advances in Soft Computing & Its Applications, vol. 14, no. 1, pp. 176-195, 2022.
[17] A. Mostafa et al., "OCFormer: A Transformer-based Model for Arabic Handwritten Text Recognition," Proc. of the 2021 IEEE Int. Mobile, Intelligent and Ubiquitous Computing Conference (MIUCC), pp. 182–186, Cairo, Egypt, 2021.
[18] A. T. Al-Taani and S. T. Ahmad, "Recognition of Arabic Handwritten Characters Using Residual Neural Networks," JJCIT, vol. 7, no. 2, pp. 192–205, DOI: 10.5455/jjcit.71-1615204606, Jun. 2021.
[19] R. S. Alkhawaldeh, "Arabic (Indian) Digit Handwritten Recognition Using Recurrent Transfer Deep Architecture," Soft Computing, vol. 25, no. 4, pp. 3131–3141, 2021.
[20] R. Yacouby and D. Axman, "Probabilistic Extension of Precision, Recall and F1-score for More thorough Evaluation of Classification Models," Proc. of the 1st Workshop on Evaluation and Comparison of NLP Systems, pp. 79–91, DOI: 10.18653/v1/2020.eval4nlp-1.9, 2020.
[21] Z. Noubigh et al., "Transfer Learning to Improve Arabic Handwriting Text Recognition," Proc. of the 2020 21st IEEE Int. Arab Conf. on Information Technology (ACIT), IEEE, pp. 1–6, Giza, Egypt, 2020.
[22] M. Fasha, B. Hammo, N. Obeid and J. AlWidian, "A Hybrid Deep Learning Model for Arabic Text Recognition," Int. Journal of Advanced Computer Science and Applications, vol. 11, no. 8, 2020.
[23] A. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems, vol. 30, 2017.
[24] G. A. Abandah et al., "Challenges and Pre-processing Recommendations for MADCAT Dataset of Handwritten Arabic Documents," Proc. of the 2018 IEEE 11th Int. Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 1–9, Beijing, China, 2018.
[25] S. Faizullah et al., "A Survey of OCR in Arabic Language: Applications, Techniques and Challenges," Proc. of the 2015 IEEE Int. Conf. on Communication, Networks and Satellite (COMNESTAT), vol. 13, pp. 111–114, 2015.
[26] S. A. Mahmoud et al., "KHATT: Arabic Offline Handwritten Text Database," Int. Workshop on Frontiers in Handwriting Recognition (IWFHR), Bari, Italy, pp. 449–454, 2012.
[27] A. AbdelRaouf, C. A. Higgins, T. Pridmore and M. Khalil, "Building a Multi-modal Arabic Corpus (MMAC)," Int. Journal on Document Analysis and Recognition, vol. 13, no. 4, pp. 285–302, Dec. 2010.

[28] F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi and J. Hennebert, "A New Arabic Printed Text Image Database and Evaluation Protocols," Proc. of the 2009 IEEE 10th Int. Conf. on Document Analysis and Recognition, pp. 946–950, Barcelona, Spain, 2009.

[29] N. Ben Amara et al., "ARABASE: A Relational Database for Arabic OCR Systems," International Arab Journal of Information Technology, vol. 2, pp. 259–266, 2005.

[30] M. Pechwitz et al., "IFN/ENIT-database of Handwritten Arabic Words," Proc. of Francophone Int. Conf. on Writing and Document (CIFED), vol. 2, Citeseer, pp. 127–136, Hammamet, Tunisia, 2002.

ملخص البحث:

تُعدّ المخطوطاتُ باللُّغة العربية من أكثر المخطوطاتِ تعقيداً وصعوبةً؛ فهي تستخدم أشكالاً مختلفة للأحرف مع علامات تشكيلٍ معقّدةٍ من الصّعْب تمييزها عن النّقاط الّتي تحتوي عليها الأحرف المنقوطة. وإنّ الخصائص المميّزة لتلك المخطوطات تجعل التّمييز الضّوئي للأحرف تنطوي على الكثير من التّحدّيات تنجم عنها دِقّة تمييزٍ منخفضة. والجدير بالذكر أنّ الأدبيات تَعُجُّ بالدّراسات الّتي تهدف إلى تقديم أنظمة تمييز ضوئي للأحرف عالية الدّقّة. إلّا أنّ مسألة تحسين الدّقّة في أنظمة التّمييز الضّوئي للأحرف بالعربية تظلّ مسألة مفتوحة تعتمد على مجموعات البيانات المستخدمة ونظام التّمييز المقترح. هنا إضافة إلى ما تضيفه علامات التّشكيل الّتي توضع فوق الأحرف أو تحتها من صعوباتٍ تؤدي إلى انخفاض دقّة تمييز الأحرف.

تقترح هذه الدّراسة نظاماً محسَّناً على مستوى الكلمة للتّمييز الضّوئي للأحرف في مخطوطاتٍ باللّغة العربية يستخدم المحوّلات، إضافة إلى مُرَمِّزيْن اثنيْن. وقد بلغت دقّة التّمييز للنّظام المقترح (98.5%) عند استخدامه في تمييز الأحرف في مخطوطاتٍ تحتوي على نصوصٍ بلا علامات تشكيل، بينما وصلت دقّة التّمييز للنّظام المقترح إلى (89.9%) في المخطوطات الّتي تتضمّن نصوصاً مع علامات تشكيل، ممّا يعني تَفوُّق النّظام المقترح على غيره من الأنظمة المشابهة الواردة في أدبيات الموضوع.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 04, December 2025.

LIGHT-WEIGHT, SEMI CONTEXT-FREE, RULE-BASED ARABIC TEXT CLASSIFIER FOR POS TAGGING

Bilal Alqduah[1], Mohanad Alhasanat[1], Abdullah Alhasanat[1] and Hatem Alqudah[2]

(Received: 16-Jun.-2025, Revised: 12-Jul.-2025 and 12-Aug.-2025, Accepted: 13-Sep.-2025)

ABSTRACT

In this research, we address the challenges associated with part-of-speech (*POS*) *tagging and morphological classification of Arabic text where word structure is the subject of study.. Our focus is on Classical Arabic* (*CA*) *and Modern Standard Arabic (MSA), where the text is typically vocalized and includes diacritics on most letters. Our proposed classification method does not require a lexicon, stemming processes, or artificial intelligence techniques. The goal is to minimize the resources needed for classifying Arabic text. This method is based on the principle that each verb in the Arabic language adheres to a specific pattern, we refere to as* (*wazn* وزن *or taf'īl* تفعيل)*, that can be utilized to identify a word. The classification process is governed by a finite state machine, which is translated into regular expressions. Each verb tense is represented by a set of regular expressions* (*REs*)*. The order in which these regular expressions are processed is crucial for the accuracy of the results. Whenever a match is found, the word is marked to prevent further matches. The proposed method is lightweight and functions as a best-effort classifier, assigning the closest match as a tag. In terms of performance, the proposed classifier's execution time is linear and does not require high processing capabilities.*

KEYWORDS

Part-of-speech (POS) tagging, Arabic rule-based classifiers, Natural languages, Context-free grammars.

1. INTRODUCTION

In Arabic language, words are classified into three main classes: nouns, verbs, and particles or hārf (حرف), with each having their own grammatical functions and structures. Nouns cover names, places, things, and abstract concepts, whereas verbs convey action or state and vary according to tense, person, and gender. Particles, on the other hand, serve as connectors or modifiers, altering the meaning and relationship between words without possessing complete lexical meaning themselves.

Verbs in the Arabic language follow a unique patterning system that allows for the distinction between past, present, and imperative verbs. This patterning is governed by diacritical marks. Under these patterns fall all verbs that are similar in terms of the purpose they denote or the time in which the action occurs. Consequently, the set of patterns that determine the imperative form do not resemble the set of patterns that indicate past tense or the set of patterns for the present tense. These patterns are referred to as verb measures (الاوزان) or *taf'īl* (تفعيل). The suffixes and prefixes attached to the verb serve other purposes, such as plural forms or the gender of the doer, whether male or female.

In this research, we introduce a part of speech (POS) classification algorithm capable of classifying words from Arabic corpora into verbs, nouns, and particles without the necessity of stemming. This approach eliminates the substantial costs associated with processing all possible prefixes, suffixes, and search time in dictionaries. The proposed classification process aims to facilitate word classification in the Arabic language anonymously, context-free, without requiring artificial-intelligence knowledge bases or dictionaries. In terms of processing capabilities, the method does not demand powerful computers or extensive memory resources.

The proposed algorithm is represented as a set of rules that work like a sieve panel, which is used to classify seeds of different shapes, sizes, and weights through a screen. Using the proposed ordered set of rules, each rule will be matched with potential matches in the provided text. When the shortest match is found, the tag associated with the rule will be assigned to the matched word. The rules are presented as regular expressions covering proper nouns, numbers, special characters, punctuation, pronouns and verbs.

1. B. Alqudah, M. Alhasanat and A. Alhasanat are with the Department of Computer Engineering, Al-Hussein Bin Talal University, Ma'an, Emails: {alqudah, mohanadhasanat, abad}@ahu.edu.jo
2. H. Alqudah is with the College of Education, Humanities and Social Sciences, Al Ain University, Abu Dhabi, United Arab Emirates, Email: hatem.alqudah@aau.ac.ae

Since regular expressions can be complex to understand, each rule or verb pattern, which we refere to as a measure, can be represented as a non-deterministic finite automaton (NFA) or multiple deterministic finite automata (DFA). However, after verification, the rules are implemented using regular expressions. Each measures represented by a regular expression (RE) consists of three major parts. Firstly, the rule body where each letter in the word has specific diacritics sequence for a measure that uniquely identifies a verb. Secondly, the possible prefix, which is a set of possible characters that may precede the verb. Thirdly, the possible or optional set of suffixes or pronouns. In some complex cases, the diacritics change according to grammatical rules, and the vowels might be changed from (Yaa ي) to (Alef ا), … and so on. These cases can be expressed in a rule that matches many verbs without requiring a stemming process using the RE matcher.

Accuracy can be significantly improved if the corpus includes diacritics (tashkeel تشكيل). However, diacritics are not mandatory and are treated as optional components in the rules. Consequently, the proposed rules represent a best-effort algorithm that is not deterministic, meaning that tagging might change according to the level of detail provided in the text. Another aspect is the colloquial words and spoken language; such words are not considered in the proposed rules.

Classification and tagging present challenges due to ambiguity; in the Arabic language, a noun can also function as a verb. For instance, the word (yzeed, يزيد) , which means "increase", serves as a verb. Or it can be used as a proper noun, as noted by Farghaly et al. in [1] and Maamouri and Bies in [2]. Ambiguity increases when diacritics (tashkeel تشكيل) are absent and/or words are removed from their context.

However, the context or sentence in which a word exists makes it easy to identify the word's tag without going into any stages of classification, such as stemming or searching in a lexicon. For example, in the Arabic language, if a word is preceded by the prepositions (jarr particles احرف جر) such as (إلى , من, على, عن…), the subsequent word is a noun by default. Another rule is that if a word is accurately identified as a verb, such as (يسقي), which means to water something, then it cannot be preceded or followed by another verb. Therefore, the words before and after that verb are nouns with 100% accuracy. An exception is to be able to match those words with the patterns of prohibition, negation and affirmation words in the Arabic language or other categories that precede verbs.

The primary challenge is context-free part-of-speech (POS) tagging, which focuses on identifying a word's tag without taking into consideration the word's context. As noted by Eid et al. in [3], any verb should follow specific rules. Therefore, an efficient approach will be matching words with their corresponding measure first, if they can be identified. Otherwise, the word is most likely a noun, if not identified as a particle or any other known category. This approach is the one adopted in this research.

2. LITERATURE REVIEW

Words in the Arabic language are classified into three main categories: nouns, verbs, and particles, correspondingly, (fe'l, ism, and harf) [4]. As stated by B. Weiss in [5], grammarians put two methods to classify words into these categories: the descriptive method and the rational method. The descriptive method focuses on the observable features of each part of speech, such as nunnation, the genitive case, and the vocative case. The properties of a verb are the suffixes such as the letter tā(ت) equivalent to (T) and the letter Yā (ي), and the energetic nūn (ن). The rational method (àqlī) is non-investigative and non-empirical. On one hand, nouns are not tied to time; they possess meaning by themselves. On the other hand, the meanings of verbs are qualified based on a timeline (past, present, or future). This leads us to the classification of particles, hārf (حرف), which convey meaning in a context beyond their own. As stated by Weiss, the principle of classifying speech into these three parts developed out of "ilm al-wad" (علم الوضع) written by Iji, Adud al-Din Abd al-Rahman ibn Ahmad (d. 757/1355) in the fourteenth century, in a work entitled al-Risāla al-wad'iya. The research presented by B. Weiss [5] discusses in detail the states of nouns, particles, and verb, explaining how to differentiate between them based on context, meaning, and by the suffix. The verbs, as stated, can be identified by their radicals.

Alosaimy and Atwell [6] provided a comprehensive list of available part-of-speech (POS) taggers for both Classical Arabic (CA) and Modern Standard Arabic (MSA). In summary, they explained that the tagging process in the surveyed approaches depends on a morphological analyzer (MA) equipped with a lexicon that contains all possible solutions, regardless of the context being studied. They pointed out

that no tagger has yet been adopted as a standard. The work presented offers a comparative study for taggers and discusses their accuracy. Table 1 shows the accuracy for each tagger as presented in the paper of Alosaimy and Atwell [6].

Table 1. POS tagging accuracy for 50 classical words.

Accuracy	**MD**	**MA**	**ST**	**MR**	**WP**	**AM**	**MT**	**FA**
Overall	69.6%	70.6%	78.4%	66.7%	68.6%	79.4%	67.6%	74.5%
No Prop. Nouns	8.0%	78.5%	71.4%	52.8%	58.5%	74.2%	87.1%	74.2%
Prop. Nouns	46.8%	53.1%	93.7%	96.8%	90.6%	90.6%	25.0%	75.0%
MADA+TOKAN suite (MD), MADAMIRA suite (MA), Stanford POS tagger and segmenter (ST), MarMoT (MR), Segmentor and Part-of-speech tagger for Arabic (WP), AMIRA Toolkit (AM), Arabic Toolkit Service POS Tagger (MT), Farasa (FA) [6].								

Lee Y. et al. [7] presented a model for segmenting Arabic words based on the following pattern: prefix*-stem-suffix*, where the * represents the degree of the morpheme, indicating zero or more occurrences. The initial dataset was created manually, with each word segmented by a human, then the corpus is fed to the unsupervised model. The classification result is determined based on the closest probable sequence of morphemes identified. To increase the accuracy, a set of stems derived from 155 million words was imported to the model through an unsupervised algorithm. The accuracy claimed after importing a very large dataset of words and stems to the system is 97%. However, the accuracy is debatable, since stemming is automatic, and the stemming is done for a specific set of words. The described process requires tokenization based on whitespace and punctuation.

To achieve multiple goals within a single process, Habash, N. and Rambow, O. [8] proposed a model that integrates tokenization, part-of-speech tagging, and morphological disambiguation into a three-stage framework. In this proposed solution, tokenization, along with morphological and part-of-speech tagging are considered one process of three stages. In the first stage, all possible analyses are gathered for the sentence subject to study. In the second stage, a classifier consisting of ten morphological features is applied to the words in the text. The features include “POS” for parts of speech, “Conj” for clitic conjunctions, “Part” for particles, “Pron” for pronominal clitics, “Det” for clitic definite determiners such as (ال), “Gen” for gender, “Num” for numbers, “Per” for persons, and “Voice” and “Asp” for aspect (imperfective, perfective, imperative). As explained in the proposed solution, it is possible for one word to match more than one feature. A morphological analyzer will then choose among the returned results by considering two values: agreement and weighted agreement. Agreement represents the number of classifiers matching the analysis. The weighted agreement is the sum of all classifiers agreeing with the analysis. It is noticeable that the provided model is time-consuming and requires intense analysis and pre-knowledge of words and classifications, in addition to a database of prefixes, suffixes, and stems.

In two separate studies, E. Mohammad et al. discussed the feasibility of performing parts of speech tagging without word segmentation [9]. In the second study [10], they provided two methods for parts of speech tagging, adding a new method that depends on artificial intelligence and machine learning to segment words. In the first study, they compared methods of classification accuracy that do not depend on segmentation with other methods that rely on tokenization and segmentation. In their research, they claim an accuracy of 94.74% in tagging words without segmentation, compared to 93.47% accuracy when segmentation is used. However, the proposed solutions depend on having a dataset and trained algorithms ahead; prior knowledge is required.

In [11], Khoja presented PAT, a Part-of-Speech tagger, which utilizes a tagset of (131) tags that are used to manually tag a corpus to produce a lexicon. This lexicon is based on traditional Arabic language grammars. However, the tagset categories extended to include 35 additional tags to account for Arabic clitics. Furthermore, verbs are sub-divided into various sub-categories. Nouns are also classified into categories, distinguishing between singular and plural forms (the latter referring to two or more, according to the numbering system), along with other sub-classifications, such as particles, dates, numbers, and punctuation. The proposed tagger (PAT) performs initial tagging process, which is basically searching the produced lexicon for a match. This means that a pre-processing of stemming, removing prefixes and suffixes is required. Khoja claimed an accuracy rate of 97% using a dictionary containing 4,748 root words.

Mohammad Y. et al. in [12] proposed a tagger based on sentence structure using morphological analysis in conjunction with a Hidden Markov Model. However, the basics are still the same as found in the work of other researchers, including the need to stem, removing suffixes, prefixes, and the need to have a dataset. The difference is that Arabic grammar specifies by its rules and what can follow a verb in a sentence. For instance, if a verb is identified, then it cannot be followed by another verb; instead, it must be followed by a noun. This functions similarly to a context-aware system. In [13], Elhadj Y. attempted to implement the same approach in traditional Arabic text. However, the proposed work does not introduce any new concepts beyond what is presented in [12].

In [14], M. Hjouj et al. presented a tool for Arabic text tagging and named entity recognition depending on a two-phase process. Firstly, the proposed process passes the text through a lexicon recognition phase, followed by a morphological phase. However, the rule development for morphological classification is not explained in the paper. Additionally, some rules proposed in the work to classify words as nouns are flawed. For example, the rule states that any word that ends with the letters (alef- tā ات) is a noun. This is incorrect; for instance, the word (يقتات), which means (feed on), is a present-tense verb, not a noun, despite ending with the (alef- tā) letters.

Transformation-based Learning (TBL) for Part-of-Speech (POS) tagging is a corpus-based method introduced by Algahtani et al. in [15]. This approach claims to reduce the required processing power and offers more flexibility in guessing and classifying tags for unknown words compared to traditional rule-based methods. The proposed method depends on selecting the best-fit tag from a list of candidate tags generated by a morphological analyzer, which derives these tags from previously analyzed text. As described, the TBL POS approach requires a training set, pre-defined rules, and a lexicon containing tag/word combinations. Each word goes through assigning a tagging process where the most frequently matched tag is assigned, followed by a list of rules for further correction to the assigned tag. This approach highly depends on the existence of the word in the prepared corpus and context.

Zeroual et al. [16] aimed to present a hierarchical level of tagset and their relationships to produce more accurate results by navigating deeper in the relations. The proposed method is based on estimating transition probabilities using a decision tree. A tagger named TreeTagger, which is not specifically designed for the Arabic language, utilized the generated tagset used to tag Arabic text.

The use of regular expressions in Arabic-text research is not new; M. Tarawneh and E. AlShawakfa [17] explored the power of regular expressions to improve the accuracy of information retrieval in Quranic text. The use of regular expressions improved the matching process, where prefixes and suffixes can be presented as a group of possibilities surrounding the keyword subject of the search process. In our research, we enhanced regular expressions to represent verb measures.

3. METHODOLOGY AND ALGORITHM

The process for the proposed method is illustrated in Figure 1. Classification and tagging involve several steps, where the possible matches are reduced at each stage to reduce the number of future comparisons and mitigate ambiguities that may arise from partial matching.

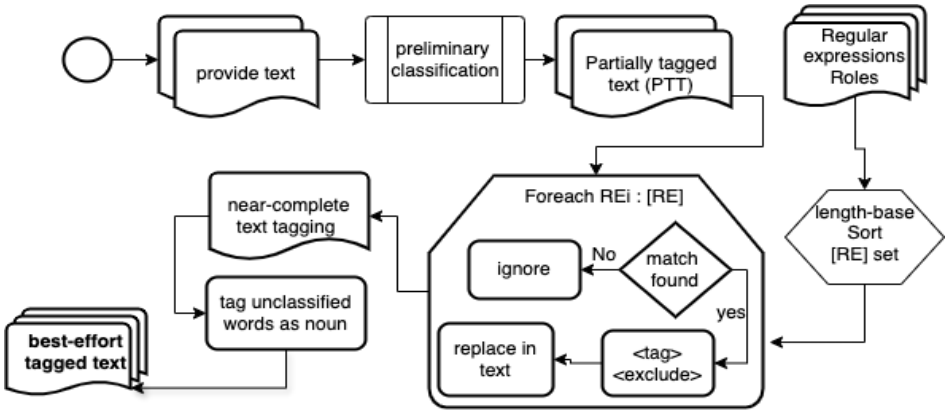


Figure 1. System flow, classification and tagging steps.

The process of tagging and classifying text involves three stages. The first stage is preliminary classification, during which the provided text goes through a basic classification based on detecting well-known words, such as present verbs preceded by particles of nasb (subjunctive markers). In Arabic grammar, the particles of nasb (subjunctive markers) always come after the present-tense verb. Another category includes nouns that are preceded by prepositions, which occur exclusively before nouns. The first stage is a bulk matching process, where a pair of processes <find, replace> is performed at once. This is the only part of the tagger that investigates partial context, one token before or after, not both, that makes the tagger a semi context-free tagger.

The latter process can be done for all known patterns of a form < particle, word> where the particle precedes a known morphological rule in identifying the successor "word". In Stage 2, a bulk match is performed for each regular expression representing a morphological verb form in the Arabic language, following the same method described in Stage 1. Consequently, the number of possible matches decreases with each step. In Stage 3, the text will contain tagged and untagged words. Any word in the text that remains untagged can be safely marked as a noun, as it does not match any regular expression within the set of regular expressions that defines any rule measure for verb , particle, or pronoun.

3.1 Regular Expressions' Classification Panel (ReCp) Design

In the proposed algorithm, clitics, such as continuous pronouns, are not removed from the words in order to be tagged.For instance, the word (فهمتها) should be segmented, on conventional tagging methods, into (ت + فهم + ها), the verb (fahim فهم/ understood), then the (tā / ت, pronoun referring to the speaker) and the (Haa/ها, pronoun referring to the object). Conjunctions and prepositions that are not part of the word, such as (عن , الى ، من ...), can also be attached to a pronoun. For example, (من + ها) will form the (منها) word, which means "from it". Such pronouns will not go through any stemming process.

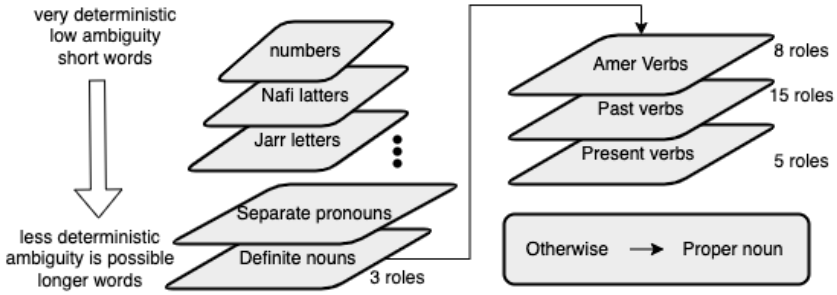


Figure 2. The priorities of regular expressions presented in the sifting screen (matcher).

The word-sieve panel is represented as a set of regular expression rules specifically designed for recognizing patterns in the Arabic language. Each rule is a regular expression (RE) representing a verb measure. In terms of length, the expressions are organized from shortest to longest. For instance, expressions representing particles, such as prepositions, accusatives, conjunctions, and separate pronouns, come first. Next are three-letter verbs, followed by verbs with four-letter roots, …and so on. As the text enters the screen, it will pass over the small expressions first; if it fits a pattern, then it will be classified as a word of that measure or rule. Otherwise, the algorithm will move it further along the screen. Figure 2 shows the screening mechanism, its design, and the number of rules for each stage.

3.2 Regular Expression Design

Representing Arabic verb weights using regular expressions is challenging due to the large number of possibilities and variations that depend on the verb's context within a sentence. In Figure 3 , we present a regular expression for the measure tafāal / تفعّل (example سيتقدّمهم) . The question mark in the regular expression represents an optional term with at most one occurrence in the word. As illustrated, it is possible to represent an entire family of verbs with this expression without needing to know the verb itself. This identification gives us the flexibility to keep the action represented by the verb anonymous.

As shown in Figure 3 , the prefix (سيت) can be expanded as needed to cover additional cases; however, we shortened it to simplify the presentation of the verb weight explained. The RE then continues to match the middle letters of the verb and any possible diacritics. This section is indicated in the figure by the dotted, underlined part. Then the remaining part of the verb weight, which is optional, presents the possible suffixes. In RE, the optional components are followed by (?).

```
(?(س)?(ي|ت)?(تَ)[ا-ي]َ[ا-ي]َ?ّ[ا-ي]َ([ا-ي]َ)?(اهُ?|هُ?مْ?|اهُ?مَ?|نهُ?|ه|ث))
```

Figure 3 .Regular expression for the measure (Tafaal / تفعّل).

Figure 4 shows the DFA or machine for the RE presented in Figure 3 that represents a past-verb measure. Group 2 and group 3 in the figure show the possible optional prefixes such as (سي/ست /…) at the beginning, denoting timing in the future, intention, because of the letter (s/ س) and the (tā / ت) for the opposite person.

However, the possible optional suffixes at the end of the verb, group 6, denote gender, such as (ها ، هن) for female, or the letter (hā/ هاء) for singular male. The figure shows the basic rule or measure where the diacritics (تشكيل) are optional in some cases, such as the second Fat'hā (◌َ) with the loop over it. But, when diacritics are provided, tagging accuracy will be higher. Groups 2,3 in the figure are optional prefixes, where group 6 is an optional suffix with non-deterministic probability.

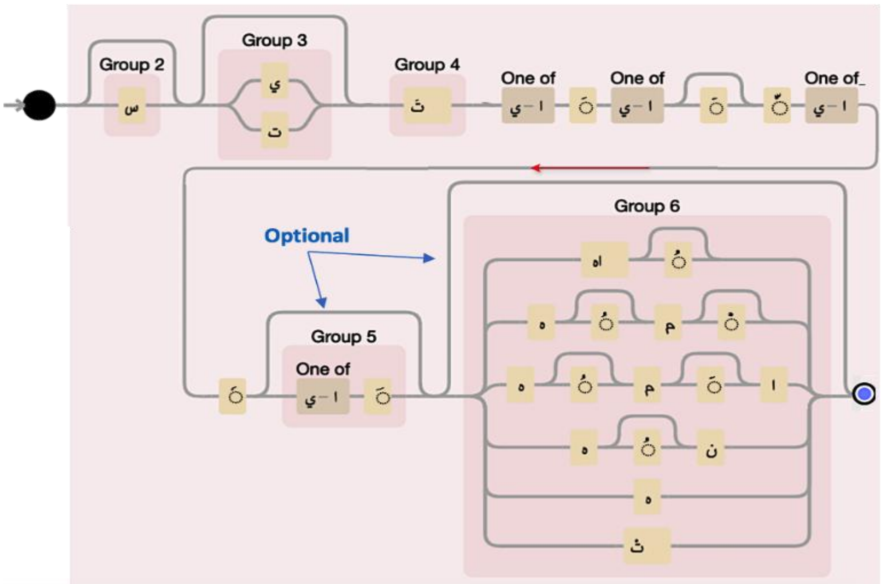


Figure 4. Finite automaton representing the measure (Tafaal / تفعّل).

Each verb can be represented by a finite automaton. However, we can notice similarities between some verb measures in the Arabic language within the same category of verbs, such as the imperative-verb weight. This similarity enables the merging of multiple finite automata to recognize various verb forms or weights.

Table 2 shows an example of imperative verbs where multiple weights are represented by a single regular expression for one verb tense. In this table, imperative-verb rule 2 combines two forms into one regular expression, as the objective is just to identify the tense of the verb, not the root or verb meaning. This approach can be applied to other verb tenses, such as present (*mudari*-مضارع) and past (*madi*-ماضي). Regarding nouns, some nouns can be identified when preceded by the definite article (alef-laam ال) or by specific disconnected pronouns or preposition particles.

As shown in Figure 5 and its corresponding regular expression, the weights of the two imperative verbs are represented by a single Deterministic Finite Automaton (DFA) and one regular expression, indicated by rule 2. However, rule 1 consolidates five rules due to the high similarity in the structures of the verb weights.

Table 2. Similar verb weights that can be represented in one regular expression.

Imperative verb weights 1	1. "افْعُلْ" 2. "افْعِلْ" 3. "افْعَلْ" 4. "أَفْعِلْ"
RE presentation	(ا|إ|أ)[ِ|ُ|َ]?(ب-ي)[ْ|ِ|ُ]?(ب-ي)[ْ]?
Imperative verb weights 2	1. "تَفَعَّلْ" : example (تَقَدَّمْ Move forward) 2. "تَفَاعَلْ": example (تَسَامَحْ Tolerate)
RE presentation	(ت)[َ](ب-ي)[َ]((ب-ي)[ّ]{2}|(ا)(ب-ي)[َ])(ب-ي)[ْ]

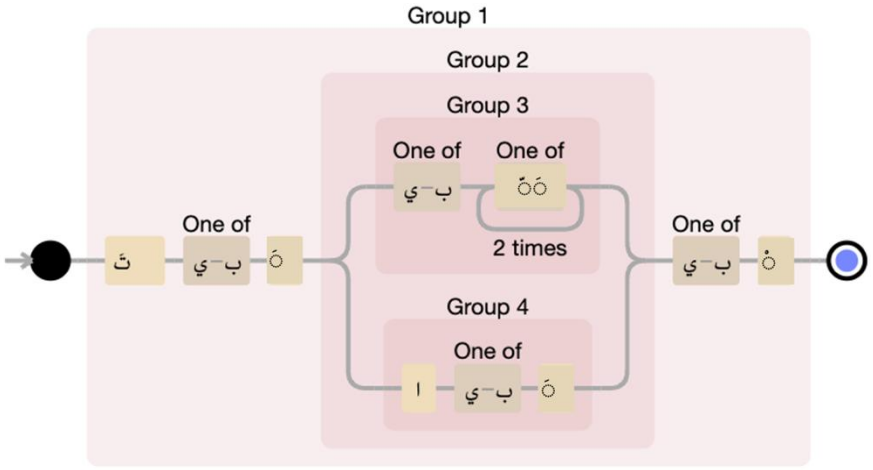


Figure 5. DFA for imperative verb, rule 2 described in Table 2.

3.3 Algorithm

Table 3 shows the proposed algorithm for the classifier. The algorithm starts with considering any Arabic text provided as (T). Two sets of regular expressions defined, CP and R, where CP is used in the initial matching to reduce future work done by R bulk matcher. Time measured before and after tagging is performed as (**τ**).

Table 3. Algorithm for Arabic regex morphological tagging.

Let: 1. **T** ∈ Σ* : Input Arabic text over some alphabet Σ 2. **CP** = { (**C**$_j$, **P**$_j$) }$^m_{j=1}$: a set of category-pattern pairs where **C**$_j$ is a linguistic category name and **P**$_j$ is a regular expression pattern for preliminary matches (obvious cases) 3. **R** = { (**C**$_i$, **P**$_i$) }$^n_{i=1}$: A finite ordered set of category-pattern pairs where **C**$_i$ is a linguistic category name and **P**$_i$ is a regular expression pattern (for bulk match) 4. *counter* ∈ ℕ : Counter for replacements 5. **τ** : Execution time in nanoseconds
Input: • B ∈ Base64 (optional) — Base64-encoded Arabic string Output: Tagged text T', total replacements counter, execution time **τ**
Initialization: 1. Let R ← {(C$_1$, P$_1$), (C$_2$, P$_2$), ..., (C$_n$, P$_n$)} 2. Initialize counter ← 0 ***Input Handling:*** **3.** If B ≠ ∅: Decode B using Base64 to get T ∈ Σ* 4. Else: Read T from dataset file 5. ***Timing Start:*** Let t_start ← System.nanoTime() ***preliminary classification*** 6. foreach **R**$_j$ ∈ **CP**

```
      if (matcher.find(R_j , T) :
          T ← Replace(T, ' C_j  [T_i ] ')
      Pattern Matching and Replacement:
7.    For each (C_i, P_i) ∈ R:
8.       From the results→ While P_i matches a substring m ⊂ T & m not tagged:
9.             replace m in T with  ' C_i  [m] '
10.            counter ← counter + 1
11.   Timing End: Let t_end ← System.nanoTime() , Compute τ ← t_end - t_start
12.   Output:  Print transformed text T , counter , τ
```

The first two lines of the algorithm show the initialization step, where all regular expressions are added to the regular expression (RE) hash map. The order in which RE is added to the map is significant to the work of the algorithm, since it should follow the presentation described in sub-section 3.1. In line 6, the preliminary or initial classification starts using the CP set. Each ($\mathbf{R}_j$) is a regular expression that performs one bulk match for a very specific family of verb weights or noun detection in a single command. The number of preliminary rules specified in the panel is eight. Each match will be marked so that it will be excluded from any future processing. The algorithm in lines 7 through 10 is performing a bulk match for each verb, particle, or noun weight found in the text. Each rule uses the matcher to perform a bulk match in the hash map for all possible string matches that have not been previously processed. At the end of the algorithm, time is measured again to get Δt, as $\boldsymbol{\tau}$, presenting the consumed time for all tagging performed.

3.4 Complexity

Let (n) be the length of the text (in characters), (m): the number of patterns in the regex patterns, and (k) be the number of matches a single regular expression finds in the text. The initialization process is just the area where regular expressions are added to the matching map, a one-time process. Lines 3 and 4 are getting the input from the user and storing it in the text to analyze. In line 6, the process of bulk match and replace will take O(1) for each regular expression, since it does not loop over the text, or the regular expressions set. For the remaining lines, starting at line 7, in the worst case, a regular expression matching is: O(n) per pattern (can be worse depending on pattern complexity, but typically O(n)), across m patterns, this is O($m \times n$) as it might look in the worst case.

However, a deeper look at the proposed algorithm, the outer loop that passes over all defined regular expressions, is always bounded to 45 times, which is the number of regular expressions defined. The inner loop will execute only once when a match is found for the rule. The matcher is just walking through the input linearly with no backtracking. This means that the matching process does not scan the entire text again as if it were a new scan every time. Instead, it resumes from where the last match ended. So, across the inner loop, the matcher will look at each character at most once per regular expression.

Based on the previous discussion, assume n = length of the text in characters. k = number of regular expression rules (45 constant). Thus, the total work will be O($k \cdot n$)= O($45 \cdot n$)=O(n). In terms of space complexity, the algorithm does not reserve new memory to process the input. Consequently, the space will remain at the boundaries of O(n).

4. RESULTS AND TESTCASES

Hardware specifications of the classifier used in our research are MacBook Pro 2012, macOS Catalina version 10.15, 16 GB 1600 MHz, 2.3 GHz Quad-Core Intel Core i7. The classifier is programmed using the Java programming language. However, other anonymous servers are used as well to run the classifier, giving a much better time. The dataset used is provided by T. Zerrouki and A. Balla [18] contains 98.85% classical Arabic text and 1.15% Modern Standard Arabic text . The dataset contains 7701 manually diacritized words. Our classifier is available on the website mentioned in [19] as an experimental release.

As shown in Figure 6 for the relation between the input size and the number of comparisons the algorithm performs, it is noticeable that the increase is almost linear for several input sizes. As shown in Figure 6 (A), for small file sizes, the number of matches or passes is less than the number of tokens since regular expression matches rules for each measure one time only. When the file size increased to contain more than 100,000 tokens, comparisons are around 39,400, as shown in Figure 6 (B). The

algorithm showed a consistent behavior for incremental input sizes, as shown in the previous figures and as presented in Table 4.

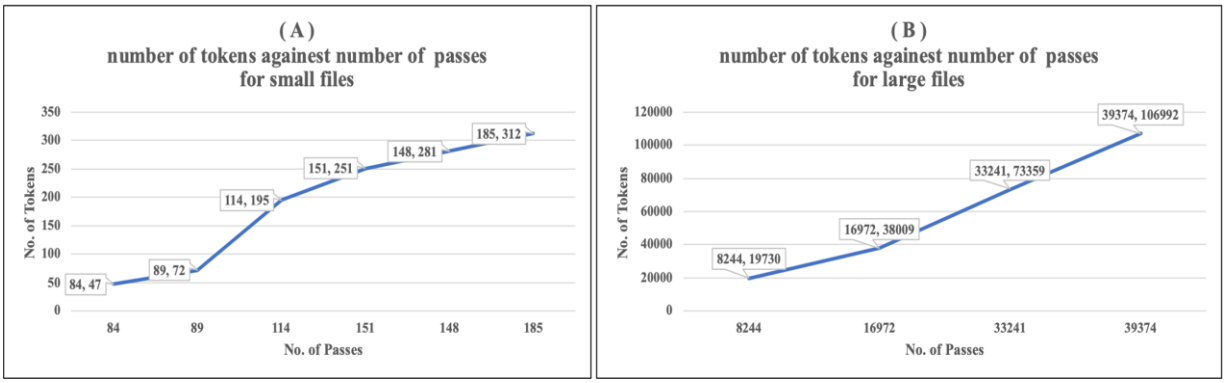


Figure 6. The linear increase of number of comparisons *versus* number of tokens.

Table 4. Test cases and performance metrics.

Tagged Tokens	size in KB	Time in ms
47	1.34	49
72	1.97	55
195	4.00	60
251	5.65	63
281	6.47	69
312	8.05	79
19730	434.16	911
38009	941.84	1373
73359	1770.54	2245
106992	2183.34	3099

Figure 7 shows the growth in time against the growth in input size for our proposed algorithm. As illustrated in the figure, time complexity increases linearly with input size and is not related to code design, as pointed out earlier in the complexity analysis. For files approximately 1.0 MB in size, the time required to classify and tag words is about 3 seconds.

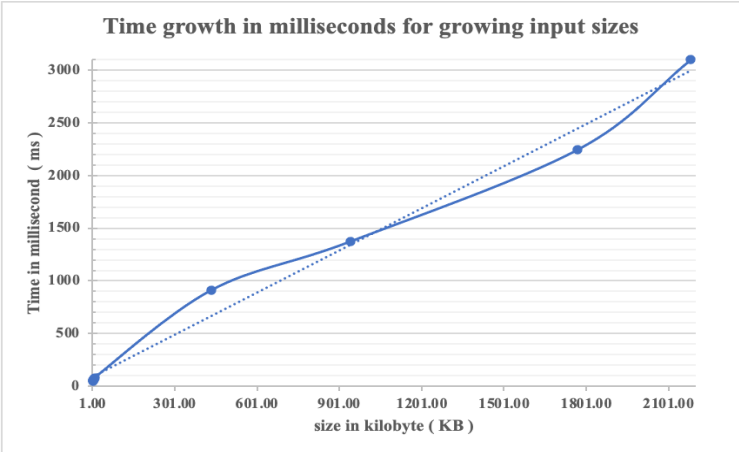


Figure 7. Time growth in relation to input size.

As shown in Table 4, the input size is 47 words, out of which 3 preliminary matches happened before the classifier starts its work. As shown, the complexity of execution relates to the size of the input only.

Time taken to perform the tagging is about 42ms, whereas in faster machines it takes about 10ms only, as seen using our classifier provided online at [19].

Table 5. Sample output and analysis from the classifier.

Sample Input			
الْحَمْدُ لِلَّهِ الْغَنِيِّ الْحَمِيدِ الْمُبْدِئِ الْمُعِيدِ، الْفَعَّالِ لِمَا يُرِيْدُ، كَتَبَ الْعِزَّةَ والْكَرَامَةَ لِمَنْ أَطَاعَهُ، وقَضَى بِالذِّلَّةِ والْهَوَانِ عَلَى مَنْ عَصَاهُ وهُوَ الْعَزِيزُ الْحَكِيمُ. وأَشْهَدُ أَنْ لَا إِلَهَ إِلَّا اللَّهُ، أَنْعَمَ عَلَيْنَا بِالْكِتَابِ الْمُبِينِ والرَّسُولِ الصَّادِقِ الْأَمِينِ			
Output			
Noun :الْحَمْدُ LafzAlJalal :لِلَّهِ Noun :الْغَنِيِّ Noun :الْحَمِيدِ Noun :الْمُبْدِئِ Noun :الْمُعِيدِ Noun :الْفَعَّالِ JarrMjror :لِمَا Present :يُرِيْدُ Past :كَتَبَ Noun :الْعِزَّةَ	AtfParticle :وَ Noun :الْكَرَامَةَ AsmaaMusol :لِمَنْ Past :أَطَاعَهُ AtfParticle :وَ Past :قَضَى Noun :بِالذِّلَّةِ Atf Particle :وَ Noun :الْهَوَانِ JarrParticle :عَلَى JazmParticle :مَنْ Past :عَصَاهُ	AtfParticle :وَ DameerMonfasel :هُوَ Noun :الْعَزِيزُ Noun :الْحَكِيمُ AtfParticle :وَ Present :أَشْهَدُ NasbParticle :أَنْ NafiOrNahi :لَا LafzAlJalal :إِلَهَ Estithnaa :إِلَّا LafzAlJalal :اللَّهُ	Past :أَنْعَمَ JarrMjror :عَلَيْنَا Noun :بِالْكِتَابِ Noun :الْمُبِينِ AtfParticle :وَ Noun :الرَّسُولِ Noun :الصَّادِقِ Noun :الْأَمِينِ
Total Tokens: 47 Possible preliminary matches : 3 Matches through loops : 39 Tokens in the text larger than 1 symbol = **42**		Number Iterations: 39 size in bytes: 1368.0 Time to classify: 42.70574 *ms*	

Table 5 shows an example of tagging Arabic text decorated with diacritics. Table 6 shows the pre-classification process results where words preceded by preposition particles are classified by default as nouns. The tag (noun2) is assigned to them to distinguish them from the words tagged based on word structure. The example of present verb preceded by Nasb letter, if the verb is tagged in the pre-classification stage, the verb will be tagged as PresentNasb, not just present.

Table 6. Pre-classification sample results for nouns and present verbs.

Sentence to analyze:	مِنَ الناسِ الطيب و في قلبهِ الرحمة و لَنْ يخذلَ السائل
JarrLetter ← مِنَ Noun2 ←الناسِ Noun1 ←الطيب AtefCONJ ← و JarrLetter ←في Noun2 ←قلبهِ	Noun1← الرحمة AtefCONJ ← و NasbLetter ←لَنْ PresentNasb ← يخذلَ Noun1 ←السائل

Table 7. Fine-grained tagging based on morphological features.

Sentence to analyze:	ضرار بن الأزور، سَافَرَ مَعَ خولة بنت الأزور ! وُلِدَتْ في القرنِ 7 .
NounMale ←ضرار Noun←بن Noun ←الأزور **punctuation** ← ، Past ← سَافَرَ JarrLetterOrDarf ←مَعَ **NounFemail**←خولة Noun ← بنت Noun4 ←الأزور	**punctuation** ← ! Past ← وُلِدَتْ JarrLetter ←في Noun ← القرنِ **Numbers** ← 7 **punctuation** ← .

Regular expression allows fine-grained tagging based on morphological features, such as numbers, punctuation, and gender. As shown in Table 7, some pronouns and names are gender specific. For example, (بن, bin), which means "son of", and (بنت, bint), which means "daughter of". Such words provide the ability to tag words correctly before they are used correctly based on gender. However, depending on such terms may not be useful to tag words after them with the same accuracy.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 04, December 2025.

5. COMPARISONS AND FEATURES

5.1 Comparison with Previous Approaches

Compared to previous work, CAMel tools [20], which are powerful, well-known, and multi-purpose analysis tools; CAMel requires approximately 5.2 GB of information to be stored and processed before the analyzer can be used. However, our proposed method provides POS tagging with predictable space compared to CAMel. The required storage for our proposed method is less than 0.5 MB of rules. Since CAMel provides many services, the comparison will be just for the POS tagging feature. As shown in Figure 7, using the same hardware used for our algorithm, CAMel behaves differently for variable input sizes. In both small and large inputs, our algorithm takes much less time than CAMel. For 2MB of input size, our algorithm took about 3 seconds, whereas in CAMel, using trained AI model, it needed 140.25 seconds to finish tagging. This shows that the rule-based algorithm is about 97.86% faster than CAMel.

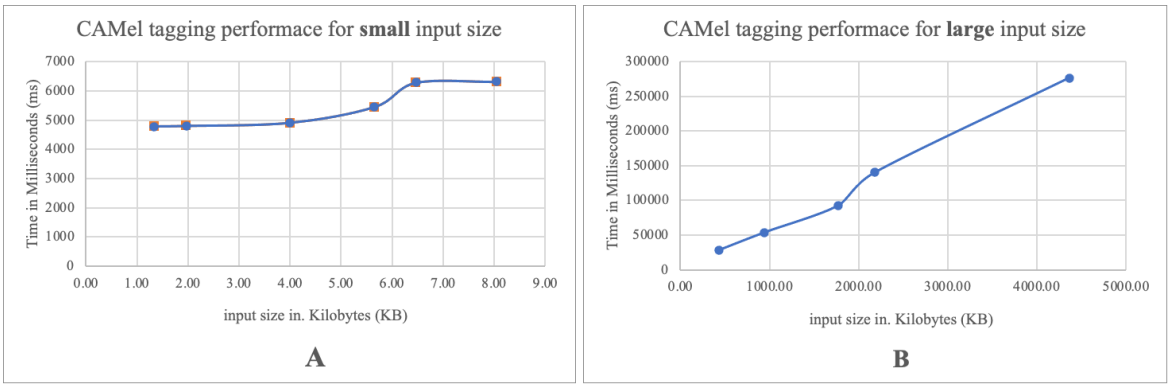


Figure 7. CAMel performance for small (A) and large (B) datasets.

Using the same input data shown previously in Table 5, CAMel took 6.6 seconds to process the sentence, whereas our proposed light-weight algorithm took 42.70574 *ms* for the same input. In terms of accuracy, CAMel incorrectly tagged some words, as shown in Table 8 marked with *. The error ratio for the provided text is about 11% for 47 words.

Table 8. Results for CAMel POS tagging for fixed benchmark input.

الْفَعَّالِ ← adj	الْحَمِيدِ ← noun_prop	أَشْهَدُ ← verb
لِمَا ← verb. *	[لا يوجد تحليل] ← الْمُبْدِئُ *	أَنْ ← conj_sub
يُرِيْدُ ← verb	الْمُعِيدِ ← noun	لَا ← part_neg
لِمَنْ ← verb *	كَتَبَ ← noun *	إِلَهَ ← verb *
أَطَاعَهُ ← verb	الْعِزَّةَ ← noun	إِلَّا ← verb

Figure 8 shows the performance for Farasa tagger using small datasets of a size of 4 kilobytes up to 180 kilobytes as shown in part (A) of the figure. Part (B) shows the performance using large datasets starting at 1500 kilobytes up to 5000 kilobytes. As illustrated in the figure, Farasa demonstrates a linear growth in execution time as the size of the input files increases.

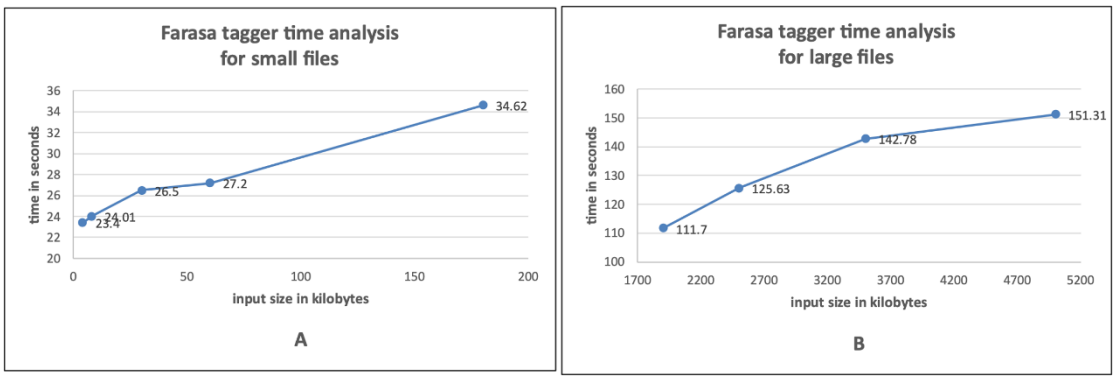


Figure 8. Farasa performance analysis using small and large datasets.

Table 9. Comparison with Farasa POS tagger, Golden Standard POS dataset tags and the proposed light-weight tagger.

Word	Farasa POS		Proposed Light-wight algorithm		Gold Standards POS tags
	Tag	result	Tag	accuracy	
الْحَمْدُ	DET+NOUN	OK	Noun	OK	DET+NOUN
لِلَّهِ	PREP NOUN	OK	LafzAlJalal_Noun	OK	PREP+NOUN
الْغَنِيِّ	DET+NOUN	OK	Noun	OK (Ignored)	DET+ADJ
الْحَمِيدِ	DET+NOUN	OK	Noun	OK (Ignored)	DET+ADJ
الْمُبْدِئُ	DET+ADJ	OK	Noun	OK	DET+ADJ
الْمُعِيدِ	DET+NOUN	OK	Noun	OK (Ignored)	DET+ADJ
،	PUNC	OK	punctuation	OK	PUNC
الْفَعَّالِ	DET+ADJ	OK	Noun	OK	DET+ADJ
لِمَا	PREP PART	OK	JarrMjror	OK	PREP+PRON
يُرِيْدُ	V	OK	PresentVerb	OK	V
،	PUNC	OK	punctuation	OK	PUNC
كَتَبَ	V	OK	PastVerb	OK	V
الْعِزَّةَ	DET+NOUN+NSUFF	OK	Noun	OK	DET+NOUN+NSUFF
والْكَرَامَةَ	CONJ DET +NOUN+NSUFF	OK	AtefCONJ	OK	CONJ
			Noun	OK	DET+NOUN+NSUFF
لِمَنْ	PREP PART	OK	AsmaaMusol	OK	PREP+PRON
أَطَاعَهُ	V PRON	OK	PastVerb	OK	V+PRON
،	PUNC	OK	punctuation	OK	PUNC
وقَضَى	CONJ V	OK	AtefCONJ	OK	CONJ
			PastVerb	OK	V
بِالذِّلَّةِ	PREP DET+ NOUN+NSUFF	OK	Noun	OK	PREP+DET+NOUN+NSUFF
والْهَوَانِ	CONJ DET+NOUN	OK	AtefCONJ	OK	CONJ
			Noun	OK	DET+NOUN
عَلَى	PREP	OK	JarrLettersPREP	OK	PREP
مَنْ	PREP	NO	LetterJazm	NO	REL.PRON
عَصَاهُ	**NOUN PRON ***	NO	PastVerb	OK	VERB+PRON
وهُوَ	CONJ PRON	OK	AtefCONJ	OK	CONJ
			DameerMonfasel	OK	PRON
الْعَزِيزُ	DET+NOUN	OK	Noun	OK (Ignored)	DET+ADJ
الْحَكِيمُ	DET+ADJ	OK	Noun	OK	DET+ADJ
.	PUNC	OK	punctuation	OK	PUNC
وأَشْهَدُ	CONJ V	OK	AtefCONJ	OK	CONJ
			PresentVerb	OK	V
أَنْ	PART	OK	NasbLetters	OK	PART
لَا	PART	OK	NafiOrNahi	OK	NEG.PART
إِلَهَ	NOUN	OK	LafzAlJalal_Noun	OK	NOUN
إِلَّا	PART	OK	Estithnaa	OK	EXCEPT.PART
اللَّهُ	NOUN	OK	LafzAlJalal_Noun	OK	NOUN
،	PUNC	OK	punctuation	OK	PUNC
أَنْعَمَ	V	OK	PastVerb	OK	V
عَلَيْنَا	PREP PRON	OK	JarrMjror	OK	PREP+PRON
بِالْكِتَابِ	PREP DET+NOUN	OK	Noun	OK	PREP+DET+NOUN
الْمُبِينِ	DET+ADJ	OK	Noun	OK	DET+ADJ
والرَّسُولِ	CONJ DET+NOUN	OK	Noun	OK	CONJ+DET+NOUN
الصَّادِقِ	DET+ADJ	OK	Noun	OK	DET+ADJ
الْأَمِينِ	DET+NOUN	OK	Noun	OK (Ignored)	DET+ADJ

As shown in Table 9, the same text used with CAMel was classified and tagged using the Farasa POS tagger, a complete stack for Arabic language processing. It was produced by [21] and is available online at [22]. Farasa tagger accuracy was 95.56% when compared to the golden standard POS tags using the

Penn Arabic Treebank [23]. In our proposed semi context-free tagging approach, we do not consider the tags derived from nouns. For example, the word (الْغَنِيّ), which is a noun, when put in the sentence context, it is an adjective (DET+ADJ) according to the gold standard. The accuracy percentage of our proposed algorithm, based on the test cases, is slightly over 97%. As demonstrated in Table 9, the Farasa tagger incorrectly classified the word (عَصَاهُ) to be a noun meaning (his stick). Whereas, our rule-based classifier correctly identified it as a verb in the past tense, as it matches a specific verb rule. However, our tagger will not be as accurate as other taggers when diacritics are partially provided on the text.

We compared the accuracy of our algorithm with that of other taggers using a dataset provided by the Universal Dependencies Treebank of Arabic, developed at New York University Abu Dhabi (NYUAD). Following the extraction of results from each tagger, we normalized the assigned tags to establish a unified tagset, allowing systematic comparison across all outputs. Table 10 shows an accuracy comparison using precision, recall and F1 for CAMel, Farasa, and Stanford POS taggers against our provided tagger. Farasa tagger presented a 94.37% accuracy, followed by our proposed algorithm with an accuracy of 93.07%. Table 11 summarizes the architecture and the key models used in each tagger.

Table 10. Accuracy comparison for standard Arabic text.

Taggers	**Precision**	**Recall**	**F1**	**Accuracy**
Farasa	89.31%	76.77%	81.53%	94.37%
Our poropsed tagger	94.86%	89.3%	90.07%	93.07%
CAMeL	65.85%	65.51%	62.48%	87.01%
Stanford	70.7%	70.53%	68.11%	81.82%

Table 11. Summary of under-laying AI architecture for used taggers.

Tagger	**Model Architecture**	**Key Model(s)**	**Note**
Farasa	Deep Learning	RNNs, LSTMs	Designed for speed and accuracy in
CAMeL	Machine Learning/Deep Learning	SVMs, RNNs	Uses a mix, often SVMs for reliability.
Stanford	Machine Learning (Classic) / Deep Learning (New)	MaxEnt (Classic) LSTMs/Transformers (New)	The classic version is not deep learning. The modern Stanza version is AI based tagger.

Table 12 shows the results of another comparison performed using 3358 words from Quranic text. CAMeL tagger scored 66.85% in accuracy, while our proposed algorithm scored just above 50%. The downside of our tagger is the fact that it does not cover some special Quranic diacritics and some word measures. For example, words like book (كِتَٰب) and devil (شَيْطَٰن) in Quranic writing where the letter alef is superscripted unlike in standard Arabic (كتاب) and (شيطان), respectively.

Table 12. Accuracy test using Quranic text.

Taggers	**Precision**	**Recall**	**F1**	**Accuracy**
Our poropsed tagger	32.72%	14.3%	19.67%	50.01%
CAMeL	25.89%	15.23%	12.45%	66.85%

Table 13. Meaning manipulation using diacritics in Arabic language.

Sentence : مَنْ مَنَّ مِنْ مَنِّ مُنَّ مِنَ المنانِ Approximate translation: Who blesses from his blessings, will be bestowed by the generous one					
Word	Proposed Algorithm	CAMeL	Farasa	Stanford	Golden tags
مَنْ	Letterjazm/JUSPART	PREP*	PREP*	PREP*	PREP
مَنَّ	PASTVERB	PREP*	PART*	PRON*	VERB
مِنْ	JARRLETTER_PREP	PREP	PART*	PREP	PREP
مَنِّ	NOUN/PRON	PREP*	PREP*	NOUN/PRON	NOUN/PRON
مُنَّ	PASTVERB	PREP*	PART*	PREP*	VERB
مِنَ	JARRLETTER_PREP	PREP	PART*	PRON*	PREP
المنانِ	NOUN	X *	NOUN	NOUN	NOUN

An extreme case is the sentence presented in Table 13, where six words with five different meanings are provided by changing diacritics. As shown in the table, tagging based on rules presented better results than by other taggers. This behaviour is expected, since rules are deterministic more than taggers that depend on machine learning or deep learning. Bad tags are marked with *.

5.2 Features of the Proposed Algorithm

In summary, the proposed light-weight, rule-based tagging system offers several advantages over traditional tagging methods and modern AI approaches.

Hardware Requirements: Our proposed algorithm operates effectively on any hardware capable of running Java or Python code, without demanding high memory or CPU resources. While there may be slight performance degradation on weaker hardware, our proposed algorithm is designed to be an in-place algorithm, requiring no extra datastructures or memory. Furthermore, the execution time of the algorithm can be anticipated to be lower than expected due to its linear growth in execution time, as demonstrated in the performance analysis provided earlier.

Accuracy: The accuracy of the proposed tagger depends on two factors: 1) the definition of rules and 2) the presence of diacritics on text. This means that the tagger can offer a best-effort tagging based on diacritics accuracy. Comparing to machine-learning and deep-learning approaches, AI-based algorithms require retraining with a substantial amount of pre-classified words to cover all possibilities and optimize the results. This leads us to the third feature of our rule-based classifier: low maintenance cost.

Low Maintenance Cost: Modifying or adding a rule will alter the results or add a new tagging category. However, ambiguity in certain words, such as nouns in the form of verb (i.e. يزيد yazeed) , remains a challenge for all taggers. This issue can be solved by looking into the context in which the word resides. Such cases are not solved in the provided algorithm, as it discusses the context-free approach. It is worthy noticing that adding new measures to the tagger may require revising the sieve design and the priorities of applying those new measures when tagging text.

Finally, the execution time. As shown by the provided comparisons, the rule-based algorithm is effecient more than traditional and AI-based approaches, since 1) It does not perform prefix and suffix processing. 2) it eliminates dictionary search time, and 3) No databases for training are required.

6. CONCLUSIONS

Tagging and morphological analysis using regular expressions show that the Arabic language is sensitive to context, diacritics, suffixes, and prefixes. As shown by this paper, it is possible to develop light-weight, fast, and effective classifiers and taggers using regular expressions. However, the way regular expressions are used, in terms of order in the matching panel and the reduction of ambiguity, is crucial.

When designing regular expressions, the following points should be considered to minimize the cost of the matching process. First, use possessive quantifiers whenever possible to reduce backtracking (i.e., use ++ instead of +). For example, when matching the regular expression “(a+)+b” with the string “aaaaaaax”, the matcher will fail after a very long time because of backtracking, since there is no letter “b” at the end. By modifying the RE to possessive RE “(a++)+b”, the matcher will fail faster, since there is no “b” at the end of the input string. This will improve RE performance in cases where no matches are found in the text.

Second, it is important to avoid nested quantifiers such as (.*)* or (a+)+ to constrain matching and avoid unexpected results. Additionally, use anchoring (^, $, \b) . Finally, try to avoid using lookaheads and lookbehinds when possible to reduce complexity. The tagger can be improved by building a better context-aware analyzer to solve the problem of ambiguity and wrong tagging, and by integrating machine-learning and deep-learning techniques with the current rule-based approach.

REFERENCES

[1] A. Farghaly and K. Shaalan, "Arabic Natural Language Processing: Challenges and Solutions," ACM Transactions on Asian Language Information Processing, vol. 8, no. 4, pp. 1–22, Dec. 2009.

[2] M. Maamouri and A. Bies, "Developing an Arabic Treebank: Methods, Guidelines, Procedures and Tools," Proc. of the Workshop on Computational Approaches to Arabic Script-based Languages, pp. 2–9, Geneva, Switzerland, Aug. 2004.

[3] M. Eid, Alnaho Almosaffa, النحو المصفى, vol. 1, ISBN: 9772324660, Deposit 1975/4427, Alshabab Library at Cario, 1971.

[4] B.-E. A. Ibn Aqeel, Sharh Ibn 'Aqeel' Ala Alfiyyah Ibn Malik, 1st Edn., vol. 1, Alresalah Center for Heritage Studies , Cairo, 1962.

[5] B. Weiss, "A Theory of the Parts of Speech in Arabic (Noun, Verb and Particle): A Study in 'ilm al-wad," Arabica, vol. 23, no. 1, pp. 23–36, Feb. 1976, Accessed: Feb. 04, 2024.

[6] A. Alosaimy and E. Atwell, "Tagging Classical Arabic Text Using Available Morphological Analysers and Part of Speech Taggers," JLCL, vol. 32, no. 1, pp. 1–26, 2017.

[7] Y.-S. Lee, K. Papineni, S. Roukos, O. Emam and H. Hassan, "Language Model-based Arabic Word Segmentation," Proc. of the 41st Annual Meeting on Association for Computational Linguistics (ACL '03), pp. 399–406, DOI: 10.3115/1075096.1075147, Morristown, NJ, USA, 2003.

[8] N. Habash and O. Rambow, "Arabic Tokenization, Part-of-speech Tagging and Morphological Disambiguation in One Fell Swoop," Proc. of the 43rd Annual Meeting on Association for Computational Linguistics (ACL '05), pp. 573–580, Ann Arbor, Ed., Morristown, NJ, USA, Jun. 2005.

[9] E. Mohamed et al., "Is Arabic Part of Speech Tagging Feasible Without Word Segmentation?" The Association for Computational Linguistics, pp. 704–708, doi: 10.13140/2.1.3631.8402, 2010.

[10] E. Mohamed and S. K. Ubler, "Arabic Part of Speech Tagging," Proc. of the 7th Int. Conf. on Language Resources and Evaluation (LREC'10), pp. 2537–2543, [Online], Available: http://www.lrecconf.org/proceedings/lrec2010/pdf/384_Paper.pdf, 2010, Accessed: Feb. 04, 2024.

[11] S. Khoja, "APT: Arabic Part-of-speech Tagger," Proc. of the Student Workshop at the Second Meeting of the North American Chapter of the Association for Computational Linguistics, Carnegie Mellon University, Pennsylvania, 2001.

[12] Y. O. Mohamed et al., "Arabic Part-of-speech Tagging Using the Sentence Structure," Proc. of the 2nd Int. Conf. on Arabic Language Resources and Tools, pp. 241–245, Cairo, Egypt, 2009.

[13] Y. O. M. Elhadj, "Statistical Part-of-speech Tagger for Traditional Arabic Texts," Journal of Computer Science, vol. 5, no. 11, pp. 794–800, 2009.

[14] M. Hjouj, A. Alarabeyyat and I. Olab, "Rule-based Approach for Arabic Part of Speech Tagging and Name Entity Recognition," Int. J. of Advanced Computer Science and Applications, vol. 7, no. 6, 2016.

[15] S. Algahtani, W. Black and J. McNaught, "Arabic Part-of-speech Tagging Using Transformation-based Learning," Proc of the 2nd Int. Conf. on Arabic Language Resources and Tools, Cairo, Egypt: The MEDAR Consortium, Apr. 2009.

[16] I. Zeroual et al., "Towards a Standard Part of Speech Tagset for the Arabic Language," Journal of King Saud University - Computer and Information Sciences, vol. 29, no. 2, pp. 171–178, Apr. 2017.

[17] M. Tarawneh and E. AlShawakfa, "A Hybrid Approach for Indexing and Searching the Holy Quran," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 1, no. 1, p. 41, 2015.

[18] T. Zerrouki and A. Balla, "Tashkeela: Novel Corpus of Arabic Vocalized Texts, Data for Auto-diacritization Systems," Data in Brief, vol. 11, pp. 147–151, DOI: 10.1016/j.dib.2017.01.011, Apr. 2017.

[19] B. I. Alqudah, "Context-free Rule-based Arabic Text Tagger and Classifier," [Online], Available: http://bilal-qudah.com/arabic/index.php, [Accessed May 5, 2025].

[20] O. Obeid et al., "CAMeL Tools: An Open Source Python Toolkit for Arabic Natural Language Processing," Proc. of the 12th Language Resources and Evaluation Conf., pp. 7022–7032, Marseille, France, 2020, Accessed: Jun. 16, 2025.

[21] K. Darwish et al., "Multi-dialect Arabic POS Tagging: A CRF Approach," Proc. of the 11th Int. Conf. on Language Resources and Evaluation (LREC 2018), pp. 93–98, Miyazaki, Japan, May 2018.

[22] Arabic Language Technology Group, "Farasa POS Tagger," [Online]. Available: https://farasa.qcri.org/POS/ 2020, Accessed: Jun. 16, 2025.

[23] M. Maamouri, A. Bies, T. Buckwalter and H. Jin, "Arabic Treebank: Part 3 v 1.0, LDC2004T11," Linguistic Data Consortium, The Trustees of the University of Pennsylvania, DOI: https://doi.org/10.35111/jf6e-hm83, Accessed: May 21, 2004.

ملخص البحث:

في هذا البحث، نُعالج التّحديات المرتبطة بتحديد أنواع أجزاء الكلام وتصنيف كلمات نُصوص اللُّغة العربية حيث يكون تركيب الكلمات هو موضوع البحث. وسيكون تركيزنا على العربية الكلاسيكية والعربية القياسية الحديثة.

وتجدر الإشارة إلى أنّ طريقتنا المقترحة لا تحتاج إلى مُعجم أو عمليات استخراج جذور الكلمات أو تقنيات ذكاء اصطناعي. فالهدف هو تقليل المصادر اللّازمة لتصنيف الكلمات في النّصوص العربية. وترتكز طريقتنا على أنّ كُلّ فِعلٍ في اللُّغة العربية يتبع نمطاً معيّناً نشير إليه بالوزْن أو التّفعيل ، ومن الممكن استغلاله لتحديد نوع الكلمة. وكلّ صيغةٍ من صِيَغ الأفعال يتم تمثيلها بمجموعةٍ من التّعبيرات المنتظمة، ويُعدّ التّرتيب الّذي تتمّ به معالجة هذه التّعبيرات المنتظمة أمراً حاسماً بالنّسبة لدقّة النّتائج. فإذا وُجِد توافق، يتمّ وَسْم الكلمة لمنع التّوافقات الأخرى.

الطّريقة المقترحة تتّسم بخفّة الوزن، وأمّا من حيث الأداء، فإنّ زمن التّنفيذ للمصنِّف هو زَمَن خطّي ولا يتطلّب قُدرات معالجةٍ عالية.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 04, December 2025.

TAB-DROID: A FRAMEWORK FOR ANDROID MALWARE DETECTION USING THE TABPFN CLASSIFIER

Ahmed M. Saeed[1], Sameh A. Salem[1,2], Shahira M. Habashy[1] and Hadeer A. Hassan[1]

(Received: 7-Jul.-2025, Revised: 13-Sep.-2025, Accepted: 17-Sep.-2025)

ABSTRACT

The Android operating system is considered as a leading global mobile OS, with its open-source nature driving widespread use across critical daily activities like banking, communication, entertainment, education and healthcare. Therefore, Android is a primary target and attractive ground for cyber-threats. In this paper, a novel malware-detection framework, which is called TAB-DROID, is introduced. The proposed framework leverages advanced feature selection, compression, and classification techniques applied to real-world datasets. Firstly, the Conditional Mutual Information Maximization (CMIM) and Joint Mutual Information (JMI) algorithms are used concurrently for feature selection. Each algorithm independently selects relevant features from the datasets. Moreover, product quantization (PQ) for feature compression is applied separately to the outputs of both CMIM and JMI to enhance storage and accelerate subsequent processing without compromising critical information. Subsequently, the Tabular Prior data Fitted Network (TabPFN) classifier is integrated into pipelines to perform the classification task. By applying 5-fold cross-validation, the results demonstrate that the optimized pipeline using CMIM achieved superior detection performance compared to the pipeline using JMI. According to CMIM-based pipeline configuration, the accuracy, AUC, precision, recall, and F1-score metrics reach 99.2%, 99.9%, 99.6%, 98.7%, and 99.2%, respectively. In addition, integrating PQ with CMIM reduced testing time by 44.4% and memory usage by 42.8%, highlighting the framework's efficiency alongside its high detection accuracy. Furthermore, the results are compared to other competing techniques, showing that the proposed framework achieved significantly enhanced performance, where the TAB-DROID has improved the accuracy up to 1.52% and precision up to 2.69%, while also reducing the feature space by 73%.

KEYWORDS

Android malware, Malware detection, Machine learning, Conditional mutual information maximization, Product quantization, TabPFN classifier.

1. INTRODUCTION

Mobile devices have become integral to modern life, with global smartphone subscriptions approaching 7 billion in 2023 and projected to exceed 7.7 billion by 2028 [1]. As for the 1st quarter (Q1) 2025, Android operating system has maintained a dominant market share of approximately 71.88% [2], with Google Play hosting over 2.26 million applications [3] and surpassing 102 billion downloads in 2024 [4]. Despite their benefits, mobile devices and third-party applications introduce significant security and privacy risks. The pervasive uses of Android applications across sectors such as banking, commerce, and education amplify the exposure of sensitive data to potential threats. Android's open-source nature, while promoting flexibility, also permits relatively unrestricted third-party application installation, increasing the platform's susceptibility to malware and cyber-attacks.

As the most widely used mobile platform, Android has become a primary target for malware due to its open architecture and extensive user base [5]. In Q2 2024, Kaspersky has reported 367,418 malicious installation packages targeting Android devices, underscoring the persistent and large-scale nature of these threats [6]. Many malware variants employ advanced encryption and obfuscation techniques to evade detection, complicating identification and mitigation efforts [7]-[8].

The growing expansion of Android malware has intensified the need for robust detection and defence frameworks. Research highlights the raised risk posed by applications distributed outside the official

1. A. M. Saeed, S. A. Salem, S. M. Habashy and H. A. Hassan are with Department of Computer and Systems Engineering, Faculty of Engineering, Helwan University, Cairo, Egypt. Emails: {ahmed_mohamed_saeed, Sameh_salem, Shahira_heikal, Hadeer_ahmed}@h-eng.helwan.edu.eg
2. S. A. Salem is with Egyptian Computer Emergency Readiness Team (EG-CERT), National Telecom Regulatory Authority (NTRA), Cairo 12577, Egypt.

Google Play Store [9]-[10], with approximately 13% of Android application installations originating from third-party sources that often lack rigorous security vetting. These alternative channels facilitate the spread of malware, often disguised as legitimate applications. Many such applications delay malicious behavior through dynamic code loading or remote payload retrieval, further complicating detection. This threat is made worse by limited user awareness: only 35% of users review application permissions before installation, and just 23% have declined applications due to excessive permissions [11]-[12]. These trends emphasize the urgent need for automated, scalable, and effective detection frameworks to counter increasingly sophisticated Android malware.

Traditional signature-based methods have proven inappropriate against the rapidly evolving nature of Android malware, prompting increased interest in machine learning (ML) and deep learning (DL) techniques for dynamic and intelligent detection. These approaches enable automated analysis of application behaviors to uncover malicious patterns beyond static signatures. However, advanced ML models, particularly neural networks, often require substantial computational resources, limiting their real-time applicability on mobile devices [13]-[15]. Furthermore, their adaptability to emerging threats remains a challenge. To address these limitations, this study proposes TAB-DROID, a lightweight and resource-efficient classification framework designed to deliver robust, timely malware detection with minimal computational overhead. Its efficiency has formed a 73% reduction in feature space and significant reductions in testing time by 44.4% and memory usage by 42.8%. The primary contributions of this research are as follows:

- A novel resource-efficient classifier framework for Android malware detection is proposed, with efficiency validated through serious reductions in feature dimensionality, testing time, and memory usage, which addresses the computational challenges with novel existing ML-based approaches.
- The framework employs two parallel pipelines featuring effective feature-selection and compression stages. The first pipeline uses CMIM, while the second uses JMI for feature selection. Both pipelines are followed by a PQ step to reduce the feature space while retaining discriminative information.
- The two pipelines are subsequently completed using the TabPFN classifier, which is a pre-trained single transformer-based classifier, enabling fast few-shot inference and strong generalization without hyper-parameter tuning, making it ideal for real-time use on resource-limited devices.
- The framework is evaluated on two datasets: TUANDROMD (241 features), which despite its small size, includes diverse and modern Android threats, and Malgenome (215 features).
- Comprehensive experimentation is applied, and 5-fold cross-validation is used to demonstrate that the pipeline using the CMIM feature-selection technique achieved optimal performance with a reduced feature set attribute, compared to the proposed framework using JMI, therefore significantly enhancing detection accuracy and AUC. Moreover, comparative evaluations against other techniques collectively validate the efficacy of the proposed TAB-DROID framework.

The rest of this paper is structured as follows: Section 2 presents the related works, reviewing recent advancements in Android malware detection. Section 3 details the proposed TAB-DROID framework, including its main components: CMIM and JMI feature selection, PQ, and the TabPFN classifier. Section 4 describes the experimental setup and the evaluation metrics and presents the results. Finally, Section 5 concludes the paper and discusses its limitations and potential directions for future work.

2. RELATED WORKS

In this section, the use of ML techniques for Android malware detection is examined, which focuses on static, dynamic, and hybrid analysis approaches [16]-[17]-[18]-[19]. Each method exhibits inherent limitations: static analysis involves the inspection of the application's code without the need for execution, which is computationally efficient. Although, it is more vulnerable to evasion techniques, such as code obfuscation and binary packing. On the other hand, dynamic analysis observes the application's behaviors as it executes, which is more accurate due to its ability to monitor runtime activity; however, it is a more time-consuming and resource-intensive usage.

Finally, the hybrid analysis leverages the strengths of both static and dynamic techniques, aiming to provide a broader and more accurate understanding of malware characteristics. A comprehensive understanding of these techniques is vital for improving the performance and reliability of ML-driven malware-detection systems. In addition to these approaches, recent studies have explored federated and privacy-preserving learning frameworks [20], which enable collaborative malware detection without

directly exposing the sensitive information of the users, thereby addressing growing privacy concerns in Android threat analysis.

2.1 Static-analysis Techniques for Detecting Android Malware

Numerous studies have focused on static analysis, highlighting its role in malware detection. Pathak et al. [21] utilized static analysis and reverse engineering to build an Android permission-based dataset. Using 48 features and a Random Forest (RF) classifier, they achieved 97.5% accuracy in malware detection. Soi et al. [22] proposed an Android malware-detection method using API-based features from the DEX Call Graph, enabling interpretable models and malware-family correlation, with 87.3% accuracy and F1-score.

Manh et al. [23] developed a five-step ML/DL-based approach to detect malicious APKs by generating and embedding Directed API Call Graphs (DACGs) with Graph2Vec, achieving 98% accuracy. Sivaprakash [24] proposed a static-analysis approach for Android malware detection using APK disassembly and bytecode inspection. The method leveraged LSTM and functional API-based DL models, utilizing features such as API calls, opcode sequences, and n-grams.

Zhao et al. [25] introduced AppPoet, an LLM-powered Android malware-detection framework using static analysis, custom prompts, and a deep neural network (DNN) classifier. It combined multiple views for accurate detection and provided human-readable diagnostic reports. Hero et al. [26] analyzed 50 Google Play applications using ESET Security and VirusTotal, revealing that 40 applications showed malware signs of 86% adware and 14% trojans, visualized *via* a Python tool.

[27] evaluated XGBoost, RF, Support Vector Machine (SVM), and Decision Tree (DT) for Android malware detection, with RF achieving the highest accuracy of 0.99, followed by SVM at 0.96, highlighting the effectiveness of ML in enhancing Android security.

2.2 Dynamic-analysis Techniques for Detecting Android Malware

Fallah et al. [28] used ML techniques to detect Android malware through network-traffic analysis across multiple malware families, achieving around 90% F1-score, but showed limited effectiveness in detecting both known and novel malware families. Amel et al. [29] proposed a dynamic-analysis framework for Android malware detection using system calls, debug logs and network activity. Combining these sources improved accuracy, aided by an MQTT-based intermediary to optimize database load and learning efficiency.

Sathyadevi et al. [30] developed an Android malware-detection system using diverse data sources, such as permissions, network traffic, API calls, opcodes, system calls, binary features and behavioral logs, combined with advanced ML techniques. The framework was evaluated using standard performance metrics. Zhu et al. [31] introduced XDeepMal, an explainable DL–based malware-detection framework. It featured XTracer+, a dynamic tool that captured real-time execution traces, and an interpretation module that isolated key behavior segments influencing model decisions. Empirical results showed that XDeepMal provided robust, interpretable insights into DL-based detection.

Prathapaneni et al. [32] proposed an ensemble of ML classifiers for multi-class dynamic feature classification using the CCCS-CIC-AndMal-2020 dataset. The ensemble notably improved malware-family prediction, boosting Adware detection from 33.84% to 81.96% and Backdoor from 67.42% to 83.71%. Ciaramella et al. [33] combined DL with the Longest Common Subsequence algorithm to classify Android applications *via* system-call sequences converted into images. Using a dataset of 13,570 samples, four convolutional neural networks (CNNs) achieved up to 89% accuracy. Class Activation Mapping techniques highlighted key regions to interpret system-call patterns, distinguishing malicious from benign behaviors.

In [34], the authors analyzed 1,535 malicious Android applications and found that 18.31% used anti-analysis techniques. To counter this, they introduced DOOLDA, a dynamic-analysis framework that detected and neutralized anti-analysis behaviors by injecting targeted instrumentation across bytecode and native layers. DOOLDA successfully defeated all known anti-analysis techniques.

2.3 Hybrid-analysis Techniques for Detecting Android Malware

Nasser et al. [35] proposed DL-AMDet, a DL–based Android malware-detection system using hybrid-

analysis features. It combined deep autoencoders for anomaly detection with a CNN-BiLSTM model for static features, achieving 99.935% accuracy across two datasets. In [36], the authors proposed HGDetector, a hybrid malware-detection method combining static function-call graphs and dynamic network-traffic features *via* graph embedding. This method improved detection accuracy by around 4% with informative traffic and up to 26% when compensating for weaker features, demonstrating the effectiveness of hybrid feature fusion.

[37] introduced MPDroid, a multi-modal pre-training approach for Android malware detection using API and function-call graphs. It leveraged graph convolutional networks and modality fusion to reduce bias and enable efficient single-modality detection. MPDroid achieved 98.3% accuracy and 97.6% F1-score while reducing training and inference time. Mesbah et al. [38] proposed LongCGDroid, an image-based malware-detection method using semantic API call graphs from control and data-flow graphs. They evaluated model performance over time as APIs evolved, finding that CNNs with abstract API features remained the most robust despite general accuracy declines.

Mercaldo et al. [39] evaluated whether images generated by deep convolutional generative adversarial networks (DCGANs) from Android malware data could be distinguished from real ones. Using static and dynamic analysis to create datasets, ML classifiers achieved an F1-score of approximately 0.8 in differentiating synthetic from real malware images. In [40] the authors explored ML techniques for Android malware detection using hybrid features from Androguard and Droidbot. k-Nearest Neighbors (KNNs) achieved the highest accuracy at 99%, followed by DT at 98%, RF at 92%, and Naive Bayes (NB) at 86%, highlighting KNNs' effectiveness.

[41] introduced TAML, a time-aware ML framework for Android malware detection using the KronoDroid dataset. It built time-aware and time-agnostic models, identifying LastModDate as a key feature. TAML achieved a 99.98% F1-score in the time-agnostic setting and up to 99% annually in time-aware evaluations over 12 years. Aledam et al. [42] proposed a hybrid Android malware-detection model combining static and dynamic analysis with PCA-based feature reduction. Evaluated on real and synthetic datasets, the model improved detection accuracy and efficiency, enhancing smartphone security.

Waheed et al. [43] enhanced the KronoDroid dataset with malware-category labels and dynamic features from real devices. Using ExtraTree for feature selection, they trained several ML models, with RF achieving 98.03% accuracy for detection and 87.56% for classifying 15 malware types.

2.4 Federated Learning and Privacy-protection Approaches for Detecting Android Malware

Federated Learning (FL) is a decentralized paradigm that enables a model to be trained across multiple devices without sharing or exposing the raw data of the users. Each device is trained locally by utilizing its own data and then shares only the model parameters to a central FL server, while ensuring that the user data remains in the device. Then, the FL server aggregates these model parameters to create a global model.

Hus et al. [44] proposed a private preserving FL (PPFL), which is an Android malware-detection system based on SVM. The model was trained using static analysis and secured with the Secure Multi-Party Computation (SMPC). Their results claimed that their system achieved a higher detection rate compared to the local model, and the results showed that accuracy increased with an increase in the number of clients. Mahindru et al. [45] introduced DNNdroid, which is based on the principle of FL. This model collected features from the users' devices without prior knowledge of where an application is installed. The results revealed that the model achieved a 97.8% F1-score with a false positive rate of 0.95 using one million Android applications with 500 users and 50 rounds of federation. Moreover, Taheri et al. [46] presented Fed-IIoT, which is an architecture of FL that consisted of two parts: first, the data-collection part through dynamic attack based on Generative Adversarial Network (GAN) and Federated GAN; and second, the server part, where the parameters are monitored, and by utilizing the A3GAN to avoid anomaly in aggregation and monitoring the parameters. The results showed high accuracy with 8% higher than the existing solutions. Çıplak et al. [47] proposed FEDetect, an FL model, and compared it with non-FL models by building 22 variants using LSTM and feedforward neural networks. The results showed that FL achieved 99% accuracy in binary classification and 84.5% in multi-class classification.

In this paper, TAB-DROID is proposed, which is a new framework for Android malware detection that leveraged CMIM and JMI for feature selection, employed a PQ technique for feature compression and utilized TabPFN classifier. The system is evaluated using two datasets, the first consists of hybrid features, which have demonstrated superior performance compared to static or dynamic features mentioned above, thereby enhancing the overall detection accuracy and robustness of the model. The second dataset comprises only static features.

3. TAB-DROID FRAMEWORK

The proposed TAB-DROID framework is comprised of four sequential and interrelated stages, as illustrated in Figure 1, each contributing to the overall malware-detection process. The first stage is data pre-processing, where the data is cleaned and balanced. The second stage performs feature reduction using two concurrent advanced selection methods: CMIM and JMI, with selection guided by framework-accuracy performance. In the third stage, the selected features are compressed using PQ to reduce computational complexity while retaining discriminative capability. Finally, the fourth stage employs the TabPFN classifier for efficient and accurate malware detection. Figure 1 presents the overall methodology of the TAB-DROID framework.

3.1 Dataset Overview

This sub-section explores the real-world datasets employed in the proposed work. The first dataset utilized is TUANDROMD, comprising 4,464 Android APK samples (3,565 malware and 899 benign) and 241 behavioral features stored in CSV format. The dataset construction followed three phases: phase 1: application collection; phase 2: static and dynamic analysis using tools, like APKAnalyser [48], Androguard, and Smali-CFGs [49]; phase 3: feature extraction. Extracted features are categorized as permission-based (i.e., captured during installation and runtime) and API call–based (i.e., reflecting functional behavior). The resulting hybrid feature set combined static and dynamic attributes, enabling comprehensive behavioral profiling for robust machine learning–based malware detection.

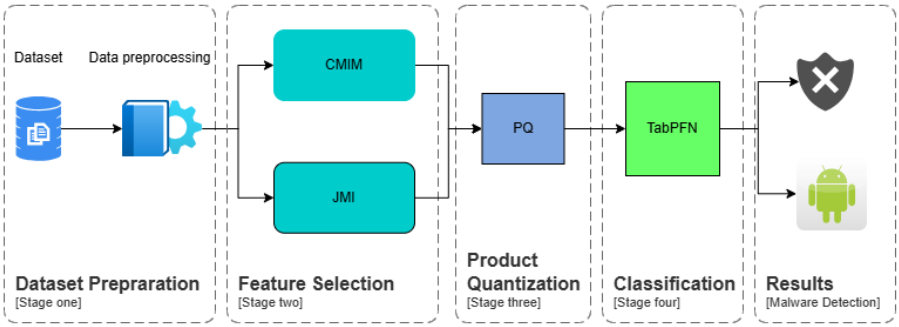


Figure 1. The proposed TAB-DROID framework.

The second dataset used is Malgenome, comprising 3,799 Android APK samples (2,539 benign and 1,260 malwares) sourced from the widely used Android Malware Genome Project [50]. It includes 215 features extracted *via* an automated static-analysis tool developed in Python. After decompiling the manifest files using AXMLPrinter2, the tool extracted permissions and intents, while API calls are retrieved by reverse engineering .dex files using the Baksmali disassembler. The analyzer also detected potentially dangerous Linux commands and checked for embedded files, such as: .dex, .jar, .so, and .exe, enabling comprehensive static feature extraction.

3.2 Data Pre-processing

Data pre-processing is an essential step to ensure the quality and suitability of the dataset before applying the proposed framework, as poor data quality can significantly impair model performance. The pre-processing process involves handling missing values, removing nulls and outliers and discarding features with zero or very low variance, which contribute a little to learning due to their minimal informational value. These steps ensure that the final dataset is clean, structured and ready for modelling.

As described earlier, the dataset exhibited class imbalance, with benign samples representing only a quarter of the malware samples. Such an imbalance can bias the model toward the majority class and degrade detection performance. To address this, two resampling strategies are typically employed: over-sampling (i.e., increasing minority-class instances) and under-sampling (i.e., reducing majority-class instances).

In this paper, the Synthetic Minority Oversampling Technique (SMOTE) is applied to over-sample the benign class. SMOTE is chosen for its ability to generate diverse synthetic samples, thereby preserving pattern variability and improving model generalization. This resulted in a balanced dataset with 3,565 samples per class for TUANDROMD dataset and 2539 per class for Malgenome dataset, subjected to complete pre-processing for subsequent analysis.

3.3 Feature Selection

Feature selection plays a critical role in pattern recognition and machine-learning systems, serving as an essential pre-processing step to identify or prioritize features based on their relevance to the target task. In this paper, feature generation, as illustrated in Figure 1, yields many features. Consequently, applying feature selection techniques becomes imperative to reduce dimensionality, thereby decreasing model training time and enhancing the overall detection performance and classification accuracy by emphasizing the most informative features. Various feature-selection strategies are developed to isolate an optimal sub-set of features, facilitating the construction of more efficient and effective models. Generally, these techniques are categorized into three primary groups: filter methods, wrapper methods and embedded methods.

Filter methods rank features based on their intrinsic statistical properties without involving model training. They are simple, fast and well-suited for large datasets. However, they ignore feature inter-dependencies, potentially selecting individually relevant features that may not improve model performance. Wrapper methods select features by iteratively training a model and evaluating performance. These approaches yield high relevance, but are computationally expensive and time-consuming. They also risk overfitting, especially with small datasets. The embedded methods integrate feature selection within model training, combining the efficiency of filters and the accuracy of wrappers. This process reduces overfitting by considering feature relevance during training, but depends heavily on the chosen algorithm and involves higher computational complexity.

In this paper, two filter techniques are utilized as concurrent feature-selection strategies because of the advantages of filter methods over wrapper and embedded methods. These techniques are CMIM [51] and JMI [52]. Both seek to identify the features that exhibit the maximum relevance to the target variable while minimizing redundancy with the already selected features. This ensures that the chosen features contribute to the most informative content without introducing unnecessary overlap, leading to more efficient and effective models.

The final selection between these two techniques is determined based on which technique yields superior performance for the TAB-DROID framework. Both techniques are based on information theoretic principles, relying on entropy and mutual information [53]-[54]. Entropy serves as a fundamental concept in the information theory of uncertainty in a random variable, denoted as $E(F)$, which quantifies the amount of uncertainty or randomness associated with the distribution of F. It is defined as shown in Equation (1).

$$E(F) = -\sum_{f \in F} \mathcal{P}(f) \log \mathcal{P}(f) \quad (1)$$

where $\mathcal{P}(f)$ is the probability mass function of the random variable F. f represents the value of all possible values of F. If the distribution of F is heavily skewed towards one particular event, indicating minimal uncertainty regarding the outcome, the entropy $E(F)$ is low. In contrast, events are likely equal, implying maximum uncertainty about the result, and the entropy reaches its maximum value. This characteristic of entropy helps quantify the unpredictability or the level of disorder inherent in the system represented by F. Let Y be another event; the entropy can be conditioned on that event. This can be denoted as shown in Equation (2).

$$E(F|Y) = -\sum_{y \in Y} \mathcal{P}(y) \sum_{f \in F} \mathcal{P}(f|y) \log \mathcal{P}(f|y) \quad (2)$$

This can be interpreted as the amount of uncertainty that remains in F after the outcome of Y is known.

In other words, it quantifies how much information F still contains after accounting for the information provided by Y. This is central to understanding the relationship between two random variables in terms of shared information. Now, mutual information between F and Y can be formally defined, which quantifies the amount of information shared between these two variables. It is defined as shown in Equation (3) (for more details, see appendix A).

$$MI(F;Y) = E(F) - E(F|Y) \tag{3}$$

Equation (3) represents the difference between: E(F), which is the uncertainty about F before knowing Y and $E(F|Y)$, which is the uncertainty about F after knowing Y, Mutual Information can be interpreted as the amount of uncertainty in F that is eliminated by knowing Y. Thus, it corresponds to the intuitive interpretation of mutual information as the quantity of information that one variable reveals about another.

One of the properties of mutual information is symmetry, which means that $MI(F;Y) = MI(Y;F)$. Additionally, when f and y variables are statically independent, this means the mutual information equals zero, indicating that their joint probability distribution is factorized as $\mathcal{P}(f,y) = \mathcal{P}(f)\mathcal{P}(y)$.

Alternatively, Equation (3) according to Fleuret [50] can also be expressed as in Equation (5).

$$MI(F;Y) = E(F) + E(Y) - E(F,Y) \tag{4}$$

Similar to Equation (2), mutual information can also be conditioned, meaning that the amount of shared information can be measured between two random variables F and Y while accounting for the influence of a third variable Z. The conditional mutual information is defined as in Equation (5) (for more details, see appendix A).

$$MI(F;Y|Z) = E(F|Z) - E(F|YZ) \tag{5}$$

Alternatively, according to Fleuret [50] Equation (5) can also be expressed as in Equation (6) (for more details, see appendix A).

$$MI(F;Y|Z) = E(F,Z) - E(Z) - E(F,Y,Z) + E(Y,Z) \tag{6}$$

After establishing a foundational understanding of entropy, mutual information and their conditional forms, the JMI and CMIM techniques are introduced and discussed, along with their key differences. Assume that a dataset of features set F, where $F = \{f_1, f_2, \dots f_n .., f_{N-1}, f_N\}$, and N is the total number of features, while S is another set of selected features chosen according to their scores, where $S = \{s_1, s_2, \dots s_k ., s_{K-1}, s_K\}$ while K is the number of features needed to be selected and the label of the class is L. The objective function of JMI is represented as in Equation (7) (for more details, see appendix A).

$$JMI(f_n) = MI(f_n, L) - \frac{1}{|S|}\sum_{s_k \in S}[MI(f_n, s_k) - MI(f_n, s_k|L)] \tag{7}$$

Equation (7) computes the JMI between the pair of features (f_n, s_k) and the class label L. By using the sum operator in the equation, it quantifies how much additional and combined information the new candidate feature f_n can provide together with each previously selected feature s_k in predicting the class label L. The objective function of CMIM is represented in Equation (8) (for more details, see appendix A).

$$CMIM(f_n) = MI(f_n; L) - \{max_{s_k \in S}[MI(f_n; s_k) - MI(f_n; s_k|L)]\} \tag{8}$$

Equation (8) computes the conditional mutual information between f_n and the target class L, conditioned on each feature s_k that has already been selected. The minimum of these values is then used as the score for f_n, which means that f_n has maximum relevance to the class label and minimum redundancy to the features in the S set.

The CMIM algorithm exists in two variants: the standard and optimized implementations. The fast implementation is utilized in this paper, which is the optimized version. The Fast-CMIM version exploits the observation that, as the selection process progresses, the score vector can only decrease, allowing for the omission of unnecessary score updates. Importantly, this optimization introduces no approximations.

In the fast CMIM approach, for each feature f_n, a partial score $Ps[n]$is maintained, representing the minimum of the conditional mutual information values as defined in Equation (8). Additionally, a vector $m[n]$ is used to track the index of the last selected feature considered in the computation of $Ps[n]$. The

detailed procedure for the fast CMIM implementation is outlined in the pseudo-code presented in Algorithm 1.

The principal distinction between JMI and CMIM lies in their selection strategies. JMI emphasizes the selection of features that, in combination with the already selected features, share a significant amount of information with the class label L, as described in Equation (7). In other words, JMI favours features that demonstrate strong collaborative effectiveness. In contrast, CMIM focuses on selecting features that provide a novel information about the class label L, which has not yet been captured by the previously selected features, as described in Equation (8). Thus, CMIM prioritizes features that are individually strong and minimally redundant. Due to this difference, JMI may tolerate a certain degree of redundancy if the additional features contribute positively to the overall model performance, whereas CMIM may exclude features that, despite being informative, are redundant with already selected ones.

Algorithm 1. Fast-CMIM.

For ($n = 1$ *to* N) **Do:**
  $Ps[n] = MI(n)$
  $m[n] = 0$
End For
For ($k = 1$ *to* K) **Do:**
  $q^* = 0$
  For ($n = 1$ *to* N) **Do:**
    While ($Ps[n] > q^*$) **&** ($m[n] < K - 1$) **Do:**
      $m[n] = m[n] + 1$
      $Ps[n]$ = **min** ($Ps[n]$, $conditional\_MI\ (n, nu[m[n]])$)
    End While
    If ($Ps[n] > q^*$) **Then:**
      $q^* = Ps[n]$
      $nu[k] = n$
    End If
  End For
End For

Note: $\boldsymbol{conditional\_MI(n,m)}$= $\boldsymbol{MI(f_n; L|s_k)}$, $\boldsymbol{MI(n)}$ = $\boldsymbol{MI(f_n; L)}$, and $\boldsymbol{nu[m[n]]}$ is conditioned on selected features.

3.4 Product Quantization

In the data-science domain, especially with high-dimensional datasets, the constraints of the resources of memory and computation present great challenges. As the size and dimensionality of data rise, the associated processing and storage requirements can quickly become infeasible. To address these restrictions, this sub-section presents PQ [55]-[57] as an effective data-compression technique that preserves main patterns and structure within the data while safely reducing size demands.

Before illustrating PQ, it is important to understand the concept of quantization and its associated benefits. Quantization is a compression technique used for high-resolution data that maps it to smaller discrete values. This data precision is slightly reduced, but it keeps the key characteristics of the original data, thereby enabling efficient compression of data without substantial loss of information. To formally describe the PQ process, some notations will be introduced. Let the dataset be composed of high-dimensional vectors in an N-dimensional space. For simplification, a vector $V \in \mathcal{R}^N$ is utilized. This vector V is partitioned into I sub-vectors (SV) as in Equation (9).

$$V = \{\overbrace{v_1, v_2, .., v_{\frac{N}{I}}}^{SV_1}, \ldots., \overbrace{v_{N-\frac{N}{I}+1}, \ldots. v_N}^{SV_I}\} = \{SV_1, SV_2, \ldots, SV_i, \ldots. SV_I\} \quad (9)$$

$SV_i \in \mathcal{R}^{N/\mathrm{I}}$, for $i \in \{1,2,\ldots,I\}$, representing the i^{th} sub-vector of V. Each SV_i is quantized independently, which allows for efficient compression. During the training phase of PQ, a separate sub-codebook is constructed for each SV_i. Each sub-codebook is denoted as in Equation (10).

$$\mathcal{C}_i = \{ c_q^i \}_{q=1}^{Q} \quad (10)$$

where $c_q^i \in R^{N/I}$ represents the q^{th} sub-codeword in the i^{th} sub-codebook and Q denotes the number of codewords per sub-codebook and the parameter Q is specified by the user during the quantization.

Each sub-codebook $\mathcal{C}_i$ is obtained by applying k-mean clustering algorithm to the i^{th} sub-space of the training vectors. To encode a high-dimensional vector V using PQ, each sub-vector SV_i is independently encoded into an identifier of its nearest codeword in $\mathcal{C}_i$ using sub-encoder function, which is denoted as in Equation (11).

$$\ell^i(SV_i) = arg\ min_{q\in\{1,..Q\}} \left\| SV_i - c_q^i \right\|^2 \tag{11}$$

After encoding all the sub-vectors, the original high-dimensional vector V is compactly represented as a sequence of I discrete identifiers, one for each sub-vector. At the end, by concatenating these I identifiers, a PQ-code is created, which defined as in Equation (12).

$$\ell(V) = \{\ell^1(SV_1), \ldots, \ell^I(SV_I)\} \tag{12}$$

where an encoder function is defined as $\ell: R^{\mathrm{N}} \longrightarrow \{1, \ldots, Q\}^I$. Although conceptually straightforward, the implementation of PQ is also relatively simple. This is illustrated in Algorithm 2, which presents the corresponding pseudocode.

Algorithm 2. Product Quantization Pseudocode (PQ)

```
# a sample vector
V
# The dimension per sub-vector Ds = N / I
I = 66
# The dimension per sub-vector
Ds = TNF/I
# Creating empty array for codewords
CW = (I, Q, Ds)
For (i = 1 to I) Do:
    Sub_vec = V [i * Ds to (i+1) * Ds]
    CW [i] = k-means (Sub_vec, Q)
End For
#creating empty array for PQ identifier
PQC = [I]
#encoding each sub-vector into an identifier
For (i = 1 to I) Do:
    Sub_vec = V [ i * Ds to (i+1) * Ds]
    PQC[i] = VQ (sub_vec, CW[i])
End For
```

V is a sample vector, **I** = number of sub-vectors, **Ds** = dimension per sub-vector (N/I), **TNF** = total number of features, **CW** = 3D array to store codewords, **k-means** = is applying the k-mean clustering where the **Q** is the number of clusters, **PQC** = array for product-quantization code, **VQ** = nearest codeword in CW[i].

3.5 TabPFN Classifier

TabPFN is a type of classifier specifically designed for tabular datasets [58], built on a generative transformer-based neural network [59] and pre-trained or self-learned on a large synthetic dataset before being used. Since it is pre-trained, TabPFN requires neither any retraining tasks nor fine tuning during inference, which makes it computationally efficient. TabPFN does not stick with or tailor to a specific tabular dataset, which enhances its applicability to real-world data.

TabPFN consists of three phases: The first phase is the data-generation phase, where various synthetic tabular datasets are generated utilizing structural causal models (SCMs) [60] which encode diverse targets and feature relationships. These datasets are built to capture a broad spectrum of potential scenarios. SCMs provide a framework that represents the structural relationships and generative process that the dataset relies on. The construction of these datasets starts with sampling high-level hyper-parameters (i.e., the number of features, data size, and the level of difficulty) to control the properties of the datasets generated. Afterward, a cyclic graph is constructed, which allocates the causality structure that the dataset relies on. Each sample per dataset is generated by propagating random noise on the causal graph root node. These samples are produced by sampling from a uniform distribution or random normal distribution and applying a varied set of computational mapping as these samples pass through the edges of the graph. At each edge, Gaussian noise is added. After constructing the causal graph, the

sample features and target values are extracted from the feature and target nodes, which are sampled.

The second phase is the pre-training phase, where a single transformer is trained offline using a large collection of synthetic datasets. The objective of this training is to learn to predict unknown targets within synthetic datasets. This phase is performed only once at the beginning and enables the model to acquire a generalized inference model. This approach is referred to as Prior Data Fitted Networks (PFNs) [61], where the learned algorithm can be applied to new datasets without additional retraining. In the final phase, after comprehensive pre-training on synthetic datasets, the model is applied to real-world data (i.e., an unseen dataset from the model's point of view). This step leverages in-context learning (ICL) [62], wherein a set of labelled instances is provided as context. The model utilizes this contextual information to infer the labels of unseen samples without requiring re-training or fine tuning. ICL enables the model to generalize effectively to new tasks by drawing on the algorithmic priors learned during the pre-training phase.

4. EXPERIMENTAL RESULTS

This section outlines the setup for the experiment, the evaluation metrics employed and the results. There are five evaluation metrics employed to assess the performance of the proposed framework. These include accuracy, recall, precision, F1-score and the area under the Receiver Operating Characteristic (ROC) curve (AUC) as presented in Equations (13) to (16), respectively [63].

$$Accuracy = \frac{T_P+T_N}{T_P+T_N+F_P+F_N} \tag{13}$$

$$Recall = \frac{T_P}{T_P+F_N} \tag{14}$$

$$Precision = \frac{T_P}{T_P+F_p} \tag{15}$$

$$F1 = \frac{T_P}{T_P+\frac{1}{2}(F_p+F_N)} \tag{16}$$

Here, T_P (True Positive) denotes the number of malicious APKs correctly classified as malicious, while T_N (True Negative) represents the number of benign APKs correctly identified as benign. F_P (False Positive) refers to benign APKs incorrectly labelled as malicious, and F_N (False Negative) corresponds to malicious APKs that are incorrectly classified as benign.

After preparing the datasets as described in the data pre-processing sub-section, each one is partitioned into training (80%) and testing (20%) sub-sets. These sub-sets are processed through two parallel pipelines. The first pipeline consists of Fast-CMIM for feature selection, followed by PQ and finally, TabPFN. The second pipeline follows the same structure, but begins with JMI instead of Fast-CMIM.

Each feature-selection technique is independently applied to the training data to identify the most informative features. The selected features are then used to replace the original feature sets in both training and testing sub-sets. PQ is subsequently applied to compress these datasets, after which TabPFN is trained on the compressed training data and evaluated on the corresponding compressed test sets. To optimize the PQ, the number of sub-vectors I and the number of codewords per sub-vector are empirically tuned. The best configuration is selected based on maximum classification accuracy with the lowest-feature sub-set. To reduce computational complexity, each feature vector is divided into sub-vectors containing exactly two features. Each sub-vector is then quantified using k-means clustering with a codebook size of two.

TabPFN is evaluated using 5-fold cross-validation, with performance metrics averaged to assess generalization capability. Experimental results demonstrate that the CMIM-based pipeline consistently outperformed the JMI-based pipeline. For the TUANDROMD dataset, the optimal configuration is achieved using 66 selected features, grouped into 33 sub-vectors, each quantized with 2 codewords, resulting in a final input dimensionality of 33. Similarly, for the Malgenome dataset, the best results are obtained with 88 features, 44 sub-vectors and 2 codewords per sub-vector, yielding a reduced dimension of 44.

Figure 2 illustrates the high performance of the CMIM-based pipeline across both datasets. It achieves peak accuracy at 66 features (TUANDROMD) and 88 features (Malgenome), while the JMI-based

pipeline shows comparatively lower accuracy, even as the number of features increases. The CMIM-based pipeline also consistently outperforms JMI across all evaluation metrics, including accuracy, AUC, precision, recall, and F1-score, indicating its effectiveness in selecting highly discriminative features. These results confirm that CMIM not only reduces feature dimensionality, but also enhances the overall classification performance.

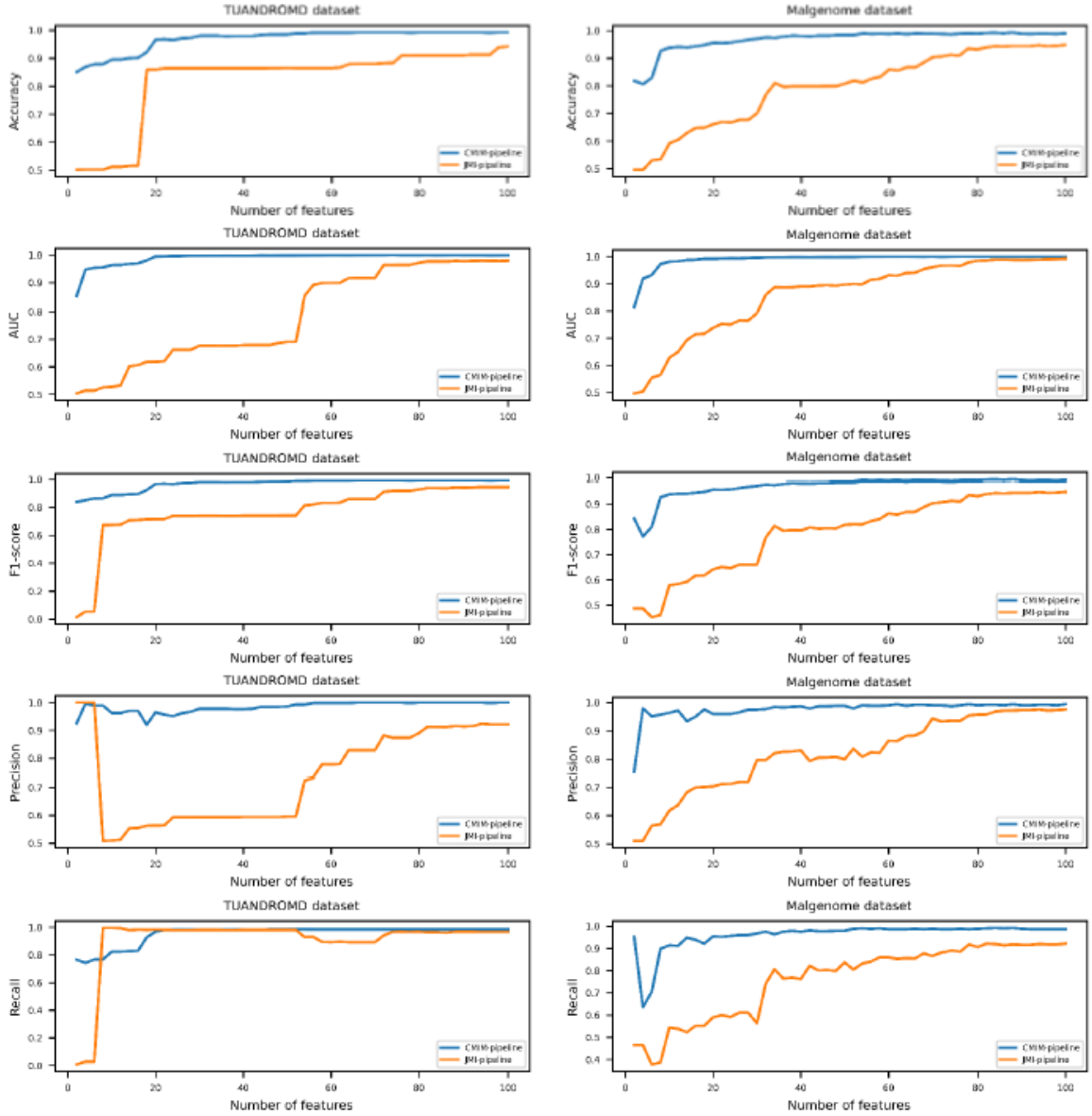


Figure 2. Comparison of evaluation metrics across different feature counts for CMIM and JMI-based pipelines on both datasets.

The results in Table 1 demonstrate the greater performance of the CMIM-based pipeline over the JMI-based pipeline, across all evaluated performance metrics. These results indicate that individual features in the dataset exhibit strong relevance to the target classes, which is consistent with the CMIM approach that prioritizes the individual contribution of each feature to the labelled class. In contrast, JMI emphasizes the joint relevance of feature combinations, which may overlook individually informative features. Based on the experimental findings, the TAB-DROID is constructed by integrating the CMIM feature-selection technique, PQ for dimensionality reduction and the TabPFN classifier, which is selected due to its superior performance across all evaluation metrics. Also, it is shown that incorporating PQ significantly reduces both training and testing times, particularly testing time, as well as overall memory usage, as shown in Table 2. This highlights PQ's effectiveness in optimizing computational efficiency within the proposed framework.

Table 1. Performance evaluation of the proposed CMIM-based pipeline and JMI-based pipeline.

Framework	Dataset Name	Number of Features	Accuracy (%)	AUC (%)	Precision (%)	Recall (%)	F1-score (%)
CMIM-based Pipeline	TUANDROMD	66	99.22	99.92	99.69	98.76	99.22
JMI-based Pipeline		100	94.57	97.62	98.09	90.90	94.36
CMIM-based Pipeline	Malgenome	88	98.62	99.81	99.19	98.01	98.60
JMI-based Pipeline		100	94.09	98.48	97.13	90.86	93.88

Table 2. Comparison of the CMIM-based pipeline with and without PQ in terms of training time, testing time and memory usage.

CMIM-based Pipeline	Dataset Name	Training Time (sec)	Testing Time (sec)	Memory Usage (MB)
With PQ	TUANDROMD Dataset (66-feature)	33.70	0.004	15.2
Without PQ		33.82	0.009	35.5
With PQ	Malgenome Dataset (88-feature)	6.91	0.006	21.6
Without PQ		8.657	0.01	34.1

Figure 3 compares the performance of traditional classifiers, such as LR, SVM, NB, and Gradient Boosting (GB), against the TabPFN classifier within the CMIM-based pipeline, using the same feature sub-set selected by the CMIM technique for both datasets. The results in the graphs demonstrate that the proposed framework achieves superior accuracy, along with higher precision, F1-score and recall. Although the AUC remains comparable across models, TabPFN consistently outperforms classical classifiers, confirming its effectiveness in the proposed detection system.

To assess the performance and reliability of the proposed TAB-DROID framework, a comparative analysis is carried out against other recent Android malware detection frameworks. As shown in Table 1, the TUANDROMD dataset consistently yields higher performance compared to the Malgenome dataset, which can be attributed to its hybrid feature composition, offering richer behavioral insights. Unlike Malgenome, which includes only static features, TUANDROMD integrates recent attributes, providing greater data diversity and enhancing its suitability for real-time and obfuscation-resilient [64] detection. A detailed comparison of the two datasets' main aspects is summarized in Table 3. Therefore, all frameworks are evaluated on the TUANDROMD dataset to ensure consistency and fairness in comparison, leveraging its comprehensive information for more robust evaluation.

Table 3. Comparison of TUANDROMD and Malgenome characteristics.

Aspect	Malgenome Dataset	TUANDROMD Dataset
Data Diversity	Focuses on early Android threats of 49 malware families.	Modern dataset captures the recent spectrum of 71 malware families.
Real-time Testing	Less realistic, as samples do not reflect today's threats.	Most realistic, as the samples exhibit current attack behaviors.
Obfuscation & Evasion	Susceptible to evasion attacks.	Employs advanced obfuscation and morphing techniques.

The comparative results are summarized in Table 4. for TAB-DROID *versus* those state-of-the-art frameworks [64]-[70] utilizing the TUANDROMD dataset, Wajahat, Ahsan et al. [66] utilized the smallest feature set, emphasizing feature economy. T. Kacem et al. [67] achieved the highest accuracy and F1-score, which are nearly equivalent to those attained by the proposed TAB-DROID framework. Furthermore, TAB-DROID reduced the feature space by 73% of the original 241 features while

maintaining superior performance. It outperformed all other frameworks in terms of precision and achieved an AUC value approaching 100%, highlighting its robustness and high classification confidence.

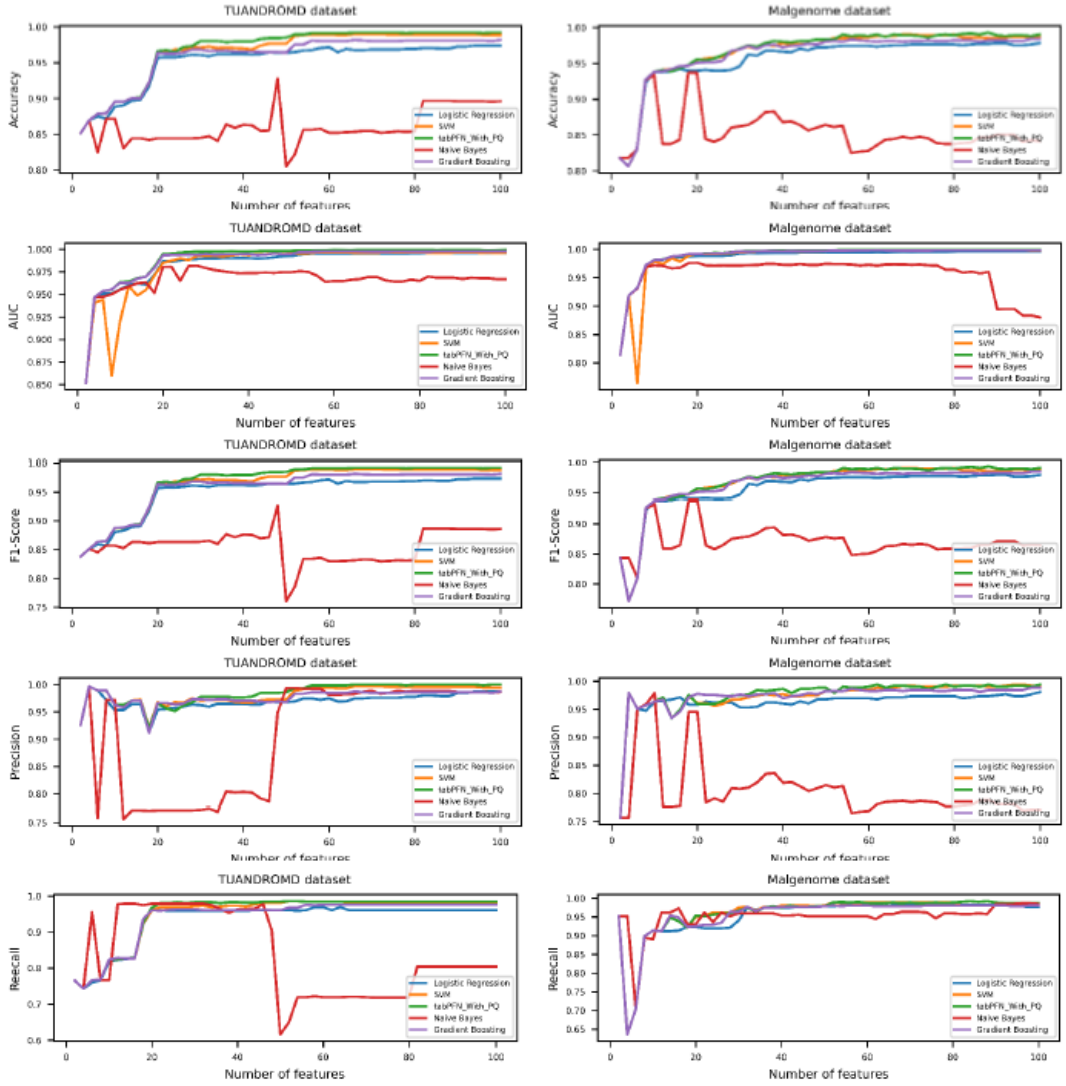


Figure 3. Comparison of evaluation metrics for CMIM-based pipeline *versus* classical classifiers on both datasets.

Despite its strong performance, TAB-DROID has limitations, including reliance on specific feature sets, challenges in detecting more advanced evasion techniques and the need for PQ hyper-parameter tuning. Additionally, employing a lightweight alternative classifier could make the system more suitable for deployment on mobile devices.

Table 4. TAB-DROID performance *versus* recent state-of-the-art frameworks.

Ref.	Year	Model	Features (%)	Accuracy (%)	AUC (%)	Precision (%)	F1-score (%)
[64]	2024	SVC	241	98.9	100	97	84.2
[65]	2024	DNDF	241	98.4	99	98.8	98.94
[66]	2024	RFE+RF	40	98	-	99	98.8
[67]	2025	Transformer	241	99.25	98.76	99.26	99.26
[68]	2025	Decision Tree	241	99.1	-	99	99
[69]	2024	Transformer+CNN	241	97.7	-	97.5	97.25
[70]	2024	RF+PCA	-	98	-	98	98
Proposed TAB-DROID		**CMIM+PQ+TabPFN**	**66**	**99.22**	**99.92**	**99.69**	**99.22**

The entire system is implemented and executed using Python version 3.8 within a Jupyter Notebook environment. Feature selection using the JMI method is conducted utilizing the publicly available scikit-feature library [71], which includes a range of widely used feature-selection algorithms. Similarly, the Fast CMIM technique is adopted from the same repository, with modifications introduced to enhance computational efficiency, as outlined in algorithm 2. For vector quantization, PQ is implemented using the nanopq library [72]. The TabPFN classifier is integrated using its official open-source repository [73]. The experimental setup utilizes essential Python libraries, including Pandas (v1.2), NumPy (v1.23), Scikit-learn (v1.2) and Matplotlib (v3.7). The experiment is conducted using Google Colab, a cloud-based Jupyter notebook environment that provides free access to computational resources. Google Colab allows users to write and execute Python code directly in the browser without local configuration. It also offers access to hardware accelerators, including GPUs, which significantly enhances computational efficiency. In this paper, a T4 GPU is utilized to accelerate experimental processes.

5. CONCLUSION

In this paper, TAB-DROID is introduced, which is a novel and efficient Android malware-detection framework integrating CMIM, PQ and the TabPFN classifier. The framework demonstrated superior performance, where the accuracy, AUC, precision, recall and F1-score metrics reached 99.2%, 99.9%, 99.6%, 98.7% and 99.2%, respectively. CMIM reduced the feature space by 73%, while PQ decreased testing time and memory usage by 44.4% and 42.8%, confirming its resource efficiency. Compared to recent approaches, TAB-DROID improved accuracy by up to 1.52% and precision by up to 2.69%, achieving near-perfect classification. Moreover, TAB-DROID is considered general and scalable. Regarding generalization, it utilized TabPFN, which avoided overfitting to specific datasets and enabled robust detection across diverse malware samples. In terms of scalability, TAB-DROID applied feature reduction and compression, which lowered computational complexity, testing time and memory usage.

For future work, the TAB-DROID can be extended for deployment on cloud-based solutions, enabling mobile devices to leverage the framework without heavy local computation. Additionally, it is planned to evaluate it by integrating additional datasets that incorporate more advanced evasion techniques and a broader set of behavioral features, allowing the system to identify the most informative features and maintain high detection performance as Android malware rapidly evolves.

APPENDIX A

A1. Proof of Equation (3)

Let F *and* Y be two random variables, the mutual information between F and Y by definition of Kullback-Leibler divergence is defined as follows: $MI(F;Y) = \sum_{f\in F}\sum_{y\in Y}\mathcal{P}(f,y)\log\frac{\mathcal{P}(f,y)}{\mathcal{P}(f)\mathcal{P}(y)}$

Expand the log, which separates the *MI* into three summation terms, each of which corresponds to an entropy as follows: $\sum_{f\in F}\sum_{y\in Y}\mathcal{P}(f,y)\left[\log\mathcal{P}(f,y) - \log\mathcal{P}(f) - \log\mathcal{P}(y)\right]$

Evaluate each term using the entropy definition:

The first term: $-\sum_{f\in F}\sum_{y\in Y}\mathcal{P}(f,y)\log\mathcal{P}(f,y) = E(F,Y)$

The second term: $-\sum_{f\in F}\sum_{y\in Y}\mathcal{P}(f,y)\log\mathcal{P}(f) = \sum_{f\in F}\left(\sum_{y\in Y}\mathcal{P}(f,y)\right)\log\mathcal{P}(f)$

According to the Marginal distribution, the second term can be rewritten as follows:

$$-\sum_{f\in F}\sum_{y\in Y}\mathcal{P}(f,y)\log\mathcal{P}(f) = \sum_{f\in F}\mathcal{P}(f)\log\mathcal{P}(f) = E(F)$$

The third term is similar to the second term: $-\sum_{f\in F}\sum_{y\in Y}\mathcal{P}(f,y)\log\mathcal{P}(y) = \sum_{f\in F}\mathcal{P}(f)\log\mathcal{P}(f) = E(Y)$

Substitute back the three terms into the *MI* equation: $MI(F;Y) = E(F,Y) - E(F) - E(Y)$

By using the chain rule, where $MI(F;Y) = E(\mathrm{Y}) + E(F|Y) = E(F) + E(Y|F)$, rearrange to obtain the following formula:

$$MI(F;Y) = E(F) - E(F|Y)$$

A2. Proof of Equation (5)

The conditional mutual information between F *and* Y *given* Z definition is as follows:

$$MI(F;Y|Z) = \sum_{f\in F}\sum_{y\in Y}\sum_{z\in Z}\mathcal{P}(f,y,z)\log\frac{\mathcal{P}(f,y|z)}{\mathcal{P}(f|z)\mathcal{P}(y|z)}$$

This definition measures the dependence between F *and* Y once the variable Z is known.

Expand the log by splitting the *MI* into three summation terms, each of which corresponds to a conditional entropy as follows: $\sum_{f\in F}\sum_{y\in Y}\sum_{z\in Z}\mathcal{P}(f,y,z)[log\,\mathcal{P}(f,y|z) - log\,\mathcal{P}(f|z) - log\,\mathcal{P}(y|z)]$
Evaluate each term using the conditional entropy definition:
The First term: $\sum_{f\in F}\sum_{y\in Y}\sum_{z\in Z}\mathcal{P}(f,y,z)\,log\,\mathcal{P}(f,y|z) = -E(F,Y|Z)$
The second term: $\sum_{f\in F}\sum_{y\in Y}\sum_{z\in Z}\mathcal{P}(f,y,z)\,log\,\mathcal{P}(f|z) = -E(F,|Z)$
The third term is similar to the second term: $\sum_{f\in F}\sum_{y\in Y}\sum_{z\in Z}\mathcal{P}(f,y,z)\,log\,\mathcal{P}(y|z) = -E(Y,|Z)$
Substitute back the three terms into the conditional mutual information equation:

$$MI(F;Y|Z) = E(F,Y|Z) - E(F|Z) - E(Y|Z)$$

By recalling the chain rule for conditional entropy:

$$E(F,Y|Z) = E(F|Y,Z) + E(Y|Z)$$

therefore, by substituting the chain rule into the expression, we obtain:

$$MI(F;Y|Z) = E(F|Z) - E(F|Y,Z)$$

A3. Proof of Equation (6)

By recalling the definition of conditional entropy:

$$E(F|Z) = E(F,Z) - E(Z)$$
$$E(F|Y,Z) = E(F,Y,Z) - E(Y,Z)$$

By substitution in Equation (5):

$$MI(F;Y|Z) = E(F,Z) - E(Z) - E(F,Y,Z) + E(Y,Z)$$

A4. Proof of Equation (7)

The joint mutual information can be written as follows:

$$JMI(f_n) = \sum_{s_k\in S} MI(f_n, s_k; L) = \sum_{s_k\in S}[MI(s_k;L) + MI(f_n,L|s_k)]$$

Neglecting the term $MI(s_k;L)$ as it is constant with respect to f_n, thus the joint mutual information reduces as follows:

$$= \sum_{s_k\in S}[MI(f_n,L|s_k)]$$
$$= \sum_{s_k\in S}[MI(f_n,L) - MI(f_n,s_k) + MI(f_n,s_k|L)]$$
$$= |S| \times MI(f_n,L) - \sum_{s_k\in S}[MI(f_n,s_k) - MI(f_n,s_k|L)]$$
$$JMI(f_n) = MI(f_n,L) - \frac{1}{|S|}\sum_{s_k\in S}[MI(f_n,s_k) - MI(f_n,s_k|L)]$$

A5. Proof of Equation (8)

The Conditional Mutual Information has a very similar procedure. The original and its rewriting are as follows:

$$CMIM(f_n) = min_{s_k\in S}[MI(f_n;L|s_k)]$$
$$= min_{s_k\in S}[MI(f_n;L) - MI(f_n;s_k) + MI(f_n;s_k|L)]$$
$$= MI(f_n;L) + min_{s_k\in S}[MI(f_n;s_k|L) - MI(f_n;s_k)]$$
$$CMIM(f_n) = MI(f_n;L) - max_{s_k\in S}[MI(f_n;s_k) - MI(f_n;s_k|L)]$$

AVAILABILITY OF DATA AND MATERIALS

The TUANDROMD dataset used during the current study is available in the [UCI Machine Learning] repository, [https://doi.org/10.24432/C5560H].

The Malgenome dataset used during the current study is available in the [figshare] repository, [https://doi.org/10.6084/m9.figshare.5854590.v1]

ACKNOWLEDGEMENTS

The authors would like to thank the Egyptian Computer Emergency Readiness Team (EG-CERT) and National Telecom Regulatory Authority (NTRA) for providing helpful ideas and suggestions.

REFERENCES

[1] Statista.com, "Number of Smartphone Mobile Network Subscriptions Worldwide from 2016 to 2023, with Forecasts from 2023 to 2028," [Online], Available: http://statista.com/statistics/330695/number-of-smartphone-users-worldwide/.

[2] Statista.com, "Market Share of Mobile Operating Systems Worldwide from 2009 to 2025, by Quarter," [Online], Available: https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/.

[3] Statista.com, " Number of Available Apps in the Google Play Store from 2nd Quarter 2015 to 2nd Quarter 2024," [Online], Available: https://www.statista.com/statistics/289418/number-of-available-apps-in-the-google-play-store-quarter/.

[4] Businessofapps, "App Downloads Data (2025)," https://www.businessofapps.com/data/app-statistics/.

[5] J. M. Arif, M. F. Ab Razak et al., "A. Android Mobile Malware Detection Using Fuzzy AHP," Journal of Information Security and Applications, vol. 61, p. 102929, 2021.

[6] Securelist.com, "IT Threat Evolution in Q2 2024-Mobile Statistics," [Online], Available: https://securelist.com/it-threat-evolution-q2-2024-mobile-statistics/113678/.

[7] Ö. A. Aslan and R. Samet, "A Comprehensive Review on Malware Detection Approaches," IEEE Access, vol. 8, pp. 6249-6271, 2020.

[8] S. Aurangzeb and M. Aleem, "Evaluation and Classification of Obfuscated Android Malware through Deep Learning Using Ensemble Voting Mechanism," Scientific Reports, vol. 13, p. 3093, 2023.

[9] V. Rastogi et al., "Catch Me If You Can: Evaluating Android Anti-malware against Transformation Attacks," IEEE Trans on Information Forensics and Security, vol. 9, no. 1, pp. 99–108, 2013.

[10] P. Kotzias, J. Caballero and L. Bilge, "How Did that Get in My Phone? Unwanted App Distribution on Android Devices," Proc. of 2021 IEEE Symposium on Security and Privacy (SP2001), pp. 53-69, 2021.

[11] M. M. Alani, "Android Users Privacy Awareness Survey," Int. Journal of Interactive Mobile Technologies, vol. 11, no. 3, 2017.

[12] A. Wajahat et al., "Outsmarting Android Malware with Cutting-edge Feature Engineering and Machine Learning Techniques," Computers, Materials & Continua, vol. 79, no. 1, pp. 651, 2024.

[13] W. Wang, M. Zhao and J. Wang, "Effective Android Malware Detection with a Hybrid Model Based on Deep Autoencoder and Convolutional Neural Network," Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 8, pp.3035-3043, 2019.

[14] S. I. Imtiaz et al., "DeepAMD: Detection and Identification of Android Malware Using High-efficient Deep Artificial Neural Network," Future Generation Computer Systems, vol. 115, pp. 844-856, 2021.

[15] M. K. Alzaylaee, Y. Y. Suleiman and S, Sakir, "DL-Droid: Deep Learning-based Android Malware Detection Using Real Devices," Computers & Security, vol. 89, p.101663, 2020.

[16] A. Dahiya, S. Sukhdip and S. Gulshan, "Android Malware Analysis and Detection: A Systematic Review," Expert Systems, vol.42, no. 1, p. e13488, 2025.

[17] G. D'Angelo et al., "Privacy-preserving Malware Detection in Android-based IoT Devices through Federated Markov Chains," Future Generation Computer Systems, vol. 148, pp. 93-105, 2023.

[18] S. Kumar, A. Prachi and J. Sahni, "IOT Malware Detection Using Static and Dynamic Analysis Techniques: A Systematic Literature Review," Security and Privacy, vol. 7, no. 6, p. e444, 2024.

[19] S. Sehrawat and D. D. Singh, "Malware and Malware Detection Techniques: A Survey, " Int. Journal for Research in Applied Science and Engineering Technology, vol. 10, no. 5, pp. 3947–3953, 2022.

[20] B. McMahan, E. Moorem et al., "Communication-efficient Learning of Deep Networks from Decentralized Data," Artificial Intelligence and Statistics (PMLR), vol. 54, pp. 1273-1282, 2017.

[21] A. Pathak et al., "Static Analysis Framework for Permission-based Dataset Generation and Android Malware Detection Using Machine Learning," EURASIP J. on Infor. Sec., vol. 2024, Art. no. 33, 2024.

[22] D. Soi et al., "Enhancing Android Malware Detection Explainability through Function Call Graph APIs," Journal of Information Security and Applications, vol. 80, p. 103691, 2024.

[23] M. Vu Minh et al., "A Static Method for Detecting Android Malware Based on Directed API Call," Int. Journal of Web Information Systems, vol. 21, no. 3, pp. 183-204, 2025.

[24] P. Sivaprakash et al., "Autonomous Android Malware Detection System Based on Static Analysis," Proc. of the 2024 IEEE Int. Conf. on Integration of Emerging Technologies for the Digital World (ICIETDW), pp. 1-6, DOI: 10.1109/ICIETDW61607.2024.10939283, 2024.

[25] W. Zhao, J. Wu and Z. Meng, "Apppoet: Large Language Model-based Android Malware Detection *via* Multi-view Prompt Engineering, "Expert Systems with Applications, vol. 262, p. 125546, 2025.

[26] H. Wintolo et al., "Visualization of Malware on Android Applications Using Static Analysis," Proc. of the 2024 IEEE Int. Conf. of Adisutjipto on Aerospace Electrical Engineering and Informatics (ICAAEEI), pp. 1-5, Yogyakarta, Indonesia, 2024.

[27] M. A. Haq and M, Khuthaylah, "Leveraging Machine Learning for Android Malware Analysis: Insights from Static and Dynamic Techniques," Engineering, Technology & Applied Science Research, vol. 14, no. 4 pp. 15027–15032, 2024.

[28] S. Fallah and A. J. Bidgoly, "Benchmarking Machine Learning Algorithms for Android Malware Detection," Jordanian J. of Computers and Inform. Tech. (JJCIT), vol. 5, no. 3, pp. 216-230, 2019.

[29] A. Boudrega, S. Benzouaoua, P. Ea, O. Salem and A. Mehaoua, "Conception of an Autonomous Dynamic Analysis System for Android Malwares," Proc. of the 2024 IEEE Asian Conf. on Communication and Networks (ASIANComNet), pp. 1-6, Bangkok, Thailand, 2024.

[30] G. Sathyadevi, J. Abishek and B. S. Shakthieiswaran, "DynaShield: Android Malware Detection Using Dynamic Analysis of Network Traffic," Proc. of the 2024 IEEE Int. Conf. on System, Computation, Automation and Networking (ICSCAN), pp. 1-6, Puducherry, India, 2024.

[31] H. Zhu et al., "A Dynamic Analysis-powered Explanation Framework for Malware Detection," IEEE Trans. on Knowledge and Data Engineering, vol. 36, no. 12, pp. 7483-7496, 2024.

[32] N. Prathapaneni et al., "Dynamic Behaviour Analysis and Interpretation of Malware in Android Devices Using Ensemble Machine Learning," Proc. of the 2024 3rd IEEE Int. Conf. on Artificial Intelligence for Internet of Things (AIIoT), pp. 1-6, Vellore, India, 2024.

[33] G. Ciaramella, F. Mercaldo and A. Santone, "Dynamic Analysis for Explainable Fine-grained Android Malware Detection," Proc. of the Int. Workshop on Security and Trust Management, pp. 110-127, Part of the Book Series: Lecture Notes in Computer Science, vol. 15235, Springer, 2024.

[34] S. Lee et al., "Hybrid Dynamic Analysis for Android Malware Protected by Anti-analysis Techniques with DOOLDA," Journal of Internet Technology, vol. 25, no. 2, pp.195-213, 2024.

[35] A. R. Nasser, A. M. Hasan and A. J. Humaidi, "DL-AMDet: Deep Learning-based Malware Detector for Android," Intelligent Systems with Applications, vol. 21, p. 200318, 2024.

[36] J. Feng et al., "HGDetector: A Hybrid Android Malware Detection Method Using Network Traffic and Function Call Graph," Alexandria Engineering Journal, vol. 114, pp. 30-45, 2025.

[37] S. Zhang et al., "MPDroid: A Multimodal Pre-training Android Malware Detection Method with Static and Dynamic Features," Computers & Security, vol. 150, p. 104262, 2025.

[38] A, Mesbah, I. Baddari and M. A. Riahla, "LongCGDroid: Android Malware Detection through Longitudinal Study for Machine Learning and Deep Learning," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 9, no. 4, pp. 328-346, Dec. 2023.

[39] F. Mercaldo, F. Martinelli and A. Santone, "Deep Convolutional Generative Adversarial Networks in Image-based Android Malware Detection," Computers, vol. 13, no. 6, p. 154, 2024.

[40] Bhooshan, Prashant and Nidhi Sonkar, "Comprehensive Android Malware Detection: Leveraging Machine Learning and Sandboxing Techniques through Static and Dynamic Analysis," Proc. of the 2024 IEEE 21st Int. Conf. on Mobile Ad-Hoc and Smart Systems (MASS), pp. 580-585, Seoul, Korea, 2024.

[41] A. M. AlSobeh et al., "Android Malware Detection Using Time-aware Machine Learning Approach," Cluster Computing, vol. 27, pp. 12627–12648, 2024.

[42] F. M. M. Aledam et al., "Enhanced Malware Detection for Mobile Operating Systems Using Machine Learning and Dynamic Analysis," Int. J. of Safety & Security Eng., vol. 14, no. 2, pp. 513-521, 2024.

[43] M. Waheed and S. Qadir, "Effective and Efficient Android Malware Detection and Category Classification Using the Enhanced KronoDroid Dataset," Security and Communication Networks, vol. 2024, Article ID 7382302, 2024.

[44] R.H. Hsu, Y.C. Wang et al., "A Privacy-preserving Federated Learning System for Android Malware Detection Based on Edge Computing," Proc. of the 2020 15th IEEE Asia Joint Conf. on Information Security (AsiaJCIS), pp. 128-136, Taipei, Taiwan, 2020.

[45] A. Mahindru and H. Arora, " Dnndroid: Android Malware Detection Framework Based on Federated Learning and Edge Computing," Proc. of the Int. Conf. on Advancements in Smart Computing and Information Security, Cham: Springer Nature Switzerland, vol. 17, no. 12, pp. 96-107, 2020.

[46] R. Tageri et al., "FED-IIoT: A Robust Federated Malware Detection Architecture in Industrial IoT," IEEE Transactions on Industrial Informatics, vol. 17, no. 12, pp. 8442-8452, 2020.

[47] Z. Çıplak et al. "FEDetect: A Federated Learning-based Malware Detection and Classification Using Deep Neural Network Algorithms," Arab. J. for Sci. and Eng., DOI: 10.1007/s13369-025-10043-x, 2025.

[48] M. Robnik-Šikonja and I. Kononenko, "Theoretical and Empirical Analysis of ReliefF and RReliefF," Machine learning, vol. 53, pp. 23-69, 2003.

[49] F. Wei, Y. Li, S. Roy, X. Ou and W. Zhou, "Deep Ground Truth Analysis of Current Android Malware," Proc. of the 14th Int. Conf. on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA 2017), vol. 14, pp. 252-276, 2017.

[50] S.Y. Yerima and S. Sezer, "Droidfusion: A Novel Multilevel Classifier Fusion Approach for Android Malware Detection," IEEE Transactions on Cybernetics, vol. 49, no. 2, pp. 453-466, 2018.

[51] F. Fleuret, "Fast Binary Feature Selection with Conditional Mutual Information," Journal of Machine Learning Research, vol. 5, pp. 1531-1555, 2004.

[52] H. Yang and J. Moody, "Data Visualization and Feature Selection: New Algorithms for Non-Gaussian Data," Advances in Neural Information Processing Systems, vol. 12, pp. 687-693, 1999.

[53] X. Gu et al., "A Feature Selection Algorithm Based on Equal Interval Division and Conditional Mutual Information," Neural Processing Letters, vol. 54, no. 3, pp. 2079-2105, 2022.

[54] G. Brown et al., "Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection," Journal of Machine Learning Research, vol. 13, no. 1, pp. 27-66, 2012.

[55] R. Gray, "Vector Quantization, " IEEE ASSP Magazine, vol. 1, no. 2, pp. 4–29, 1984.

[56] H. J´egou, M. Douze and C. Schmid, "Product Quantization for Nearest Neighbor Search," IEEE TPAMI, vol. 33, no. 1, pp. 117–128, 2011.

[57] Y. Matsui et al., "A Survey of Product Quantization," ITE Transactions on Media Technology and Applications, vol. 6, no. 1, pp. 2-10, 2018.
[58] N. Hollmann et al., "Accurate Predictions on Small Data with a Tabular Foundation Model," Nature, vol. 637, no. 8045, pp. 319-326, 2025.
[59] A. Vaswani et al., "Attention Is All You Need," Neural Information Processing Systems, vol. 30, 2017.
[60] J. Pearl, Causality, 2nd Edn., ISBN 0-521-77362-8, Cambridge University Press, USA, 2009.
[61] S. Müller et al., "PFNs4BO: In-context Learning for Bayesian Optimization," Proc. of the Int. Conf. on Machine Learning (PMLR), pp. 25444-25470, 2023.
[62] T. Brown et al., "Language Models Are Few-shot Learners," Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901, Honolulu, USA, 2020.
[63] T. Fawcett, "ROC Graphs: Notes and Practical Considerations for Researchers, " Machine Learning, vol. 31, no. 1, pp. 1-38, 2004.
[64] A. Wajahat et al., "Outsmarting Android Malware with Cutting-edge Feature Engineering and Machine Learning Techniques," Computers, Materials & Continua, vol. 79, no. 1, pp. 651-673, 2024.
[65] A. Wajahat et al., "An Effective Deep Learning Scheme for Android Malware Detection Leveraging Performance Metrics and Computational Resources," Intell. Deci. Tech., vol. 18, no. 1, pp. 33-55, 2024.
[66] A. Wajahat et al., "Securing Android IoT Devices with GuardDroid Transparent and Lightweight Malware Detection," Ain Shams Engineering Journal, vol. 15, no. 5, p. 102642, 2024.
[67] T. Kacem and S. Tossou, "Trandroid: An Android Mobile Threat Detection System Using Transformer Neural Networks," Electronics, vol. 14, no. 6, p. 1230, 2025.
[68] H. Shah et al., "A Comparative Analysis for Android Malware Detection Using Machine Learning Models," Proc. of the 2025 6th IEEE Int. Conf. on Mobile Computing and Sustainable Informatics (ICMCSI), pp. 1040-1047, Goathgaun, Nepal, 2025.
[69] N. G. Ambekar, S. Thokchom and S. Moulik, "TC-AMD: Android Malware Detection through Transformer-CNN Hybrid Architecture," Proc. of the 2024 IEEE Int. Conf. on Advanced Networks and Telecommunication Systems (ANTS), pp. 1-6, Guwahati, India, 2024.
[70] T. Bhandari et al., "Unveiling Machine Learning Paradigms for Robust Malware Detection in Personal Data Security," Proc. of the 2024 6th IEEE Int. Conf. on Computational Intelligence and Communication Technologies (CCICT), pp. 226-231, Sonepat, India, 2024.
[71] J. Li and H. Liu, "Challenges of Feature Selection for Big Data Analytics," IEEE Intelligent Systems, vol. 32, no. 2, pp. 9-15, 2017.
[72] Y. Matsui, "Nano Product Quantization," [online], Available: https://github.com/matsui528/nanopq.
[73] Github.com, "TabPFN: Foundation Model for Tabular Data," [Online], Available: https://github.com/PriorLabs/TabPFN?tab=readme-ov-file.

ملخص البحث:

يُعد نظام التّشغيل أندرويد في طليعة أنظمة التّشغيل حول العالم لأنّه يعتمد على بيئة تقوم على المصادر المفتوحة عبر الانشطة المختلفة مثل الخدمات المصرفية والاتصالات والترفيه والتعليم والرعاية الصّحية؛ لذلك فإنّه هدف أساسي وأرضية جاذبة للتّهديدات السّيبرانية.

في هذه الورقة، نقترح نظاماً مبتكراً للكشف عن برامج أندرويد الضّارّة يُسمّى (تابْ-دْرويْد) ويعتمد على تقنيات انتقاء السِّمات وضغطها وتصنيفها الّتي تُطبّق على مجموعات بياناتٍ في العالم الحقيقي. ويُعدّ النّظام المقترح إطار عملٍ ملائماً للكشف عن برامج أندرويد الضّارّة؛ فقد جرى تقييمه بواسطة عددٍ من مؤشرات الأداء، وبرهن على تفوُّقٍ واضحٍ لدى مقارنته بعددٍ من أُطُر العمل المنافسة، حيث حقّق دقّةً وصلت إلى 99.2%، إلى جانب تفوّقه في مؤشّرات الأداء الأخرى. والجدير بالذكر أنّ إطار العمل المقترح أثبت أنّه ذو فاعلية كبيرة في الكشف عن برامج أندرويد الضّارّة، ويعمل على خفْض الحيّز الخاصّ بالسِّمات بنسبة 73%، الأمر الذي يدلّ على حُسن استغلاله للمصادر؛ فقد خفّض زمن الفحص بنسبة 44.4% والذّاكرة المستخدمة بنسبة 42.8%.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 04, December 2025.

CUBIC-LEARN: A REINFORCEMENT LEARNING APPROACH TO CUBIC CONGESTION CONTROL

Ehsan Abedini and Mohsen Nickray

(Received: 30-May.-2025, Revised: 3-Sep.-2025, Accepted: 22-Sep.-2025)

ABSTRACT

Managing congestion effectively enables reliable and fast data transfer over networks. CUBIC delivers reliable results under normal circumstances, but cannot adapt effectively to changing network scenarios. We introduce CUBIC-Learn, an RL approach for improving congestion control in CUBIC. The central idea is to use a Q-learning algorithm to adjust congestion window thresholds based on current data on packet loss, throughput and latency. Simulations demonstrate more efficient and reliable congestion control when using CUBIC-Learn compared to standard CUBIC. CUBIC-Learn achieves a 47% reduction in packet loss, over a 59% increase in bandwidth utilization, approximately a 28% decrease in retransmissions and 47% lower latency. In addition, CUBIC-Learn shows significant improvements in congestion window (cwnd) growth behavior, fairness among competing flows and stability under heterogeneous traffic and network scenarios, including gigabit-scale bandwidth conditions. Statistical analysis further confirms the robustness of these gains, while the method introduces no additional computational overhead. Overall, CUBIC-Learn performs better than PCC, Reno, Tahoe, NewReno and BBRv3 in most metrics. These findings suggest that RL can markedly improve congestion control in high-speed networks.

KEYWORDS

Q-learning, Reinforcement learning, CUBIC Algorithm, Network congestion.

1. INTRODUCTION

Effective congestion control [1] (CC) is crucial in ensuring the reliable operation of computer networks on today's Internet. CC algorithms are designed to distribute network resources wisely and reduce both delays and data-packet losses. CUBIC [2] has become a leading choice for many network operators, providing good performance by striking a compromise between a range of crucial metrics. Advances in the complexity and variability of modern network traffic require new strategies to boost the efficiency of existing CC methods.

Reinforcement learning (RL) [3] has seen increasing popularity as a way to enhance algorithm performance in dynamically changing and unpredictable conditions such as networks. The ability of RL to discover the best actions by interacting with the environment suggests its suitability for overcoming congestion control issues. However, using RL to optimize the CUBIC algorithm has received little attention so far.

This study introduces a novel CUBIC-Learn algorithm that utilizes reinforcement learning to continually improve its handling of congestion control. The aim of this research is to evaluate the performance improvement achieved by CUBIC-Learn compared to the original CUBIC algorithm. The evaluation is thus conducted on a multi-hop network topology, which is complex and has many servers and clients connected through two routers and a bottleneck connection is deliberately provisioned to create congestion when the traffic loads are high. Further, CUBIC-Learn is compared with TCP variants (Reno, Tahoe, NewReno), PCC and BBRv3, thus providing a complete and representative comparison across a broad range of design paradigms. The Python simulations show that CUBIC-Learn achieves considerable gains in important performance metrics, such as packet-loss rate, throughput, retransmissions and delay.

The rest of the paper is organized as follows. The history of congestion control and reinforcement learning is surveyed in Section 2 as related work. In Section 3, the proposed method is described. Section 4 presents the simulation methodology. The results and discussion are given in Section 5, while Section 6 concludes the paper.

E. Abedini and M. Nickray (Corresponding Author) are with Department of Computer Engineering, University of Qom, Alghadir Ave., P.O. Box 3716146611, Qom, Qom Province, Iran. Emails: e.abedini@stu.qom.ac.ir and m.nickray@qom.ac.ir

2. RELATED WORK

Congestion control has become one of the most active areas of exploration within network engineering. Many solutions have been proposed to address the congestion problem [4]. This section reviews and groups some of the most significant congestion-control techniques that have been presented in the literature.

We also discuss the emerging use of reinforcement learning in network-congestion control and analyze how machine-learning methods are being incorporated into transport-layer protocols. The current focus is on works that leverage RL to improve the CUBIC algorithm.

2.1 Categorization of Congestion-control Techniques

The different types of leading CC algorithms are showcased in Figure 1. It is organized under the headings of delay-based, loss-based and hybrid algorithms. Delay-based algorithms measure delays to spot signs of network congestion [4], while loss-based algorithms monitor errors or packets that cannot be delivered [5]. Hybrid algorithms aim to improve both responsiveness and stability [5].

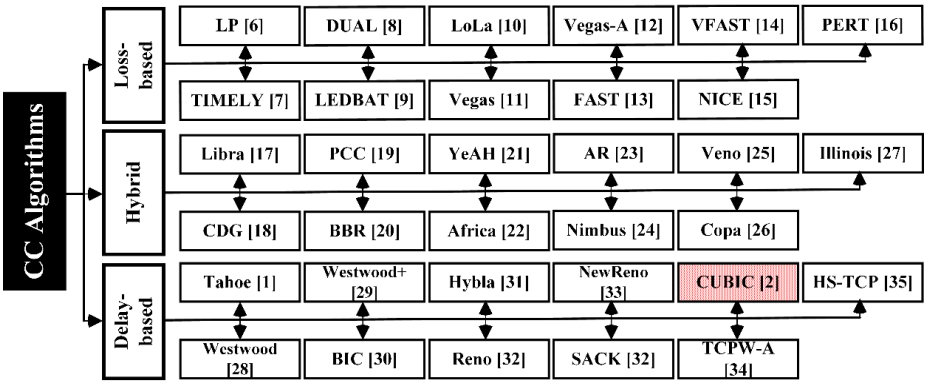


Figure 1. Classification of congestion-32control techniques.

In the recent past, the BBR (Bottleneck Bandwidth and Round-trip propagation time) of Google has become a leading hybrid congestion-control algorithm. It approximates the bottleneck bandwidth and the minimum round-trip time so as to maximize throughput and ensure low delays. Later versions, especially BBRv2, were fairer and more responsive, with the latest version, BBRv3, solving bandwidth convergence shortcomings and tuning gains to improve flow coexistence [36]. Experimental measurements show that BBRv3 converges on similar flows more quickly, but can face difficulties in coexisting with CUBIC flows [37].

2.2 CUBIC Congestion-control Algorithm

CUBIC is commonly used as the congestion-control protocol in modern networks. When network congestion is detected, CUBIC dynamically adjusts the size of its congestion window by controlling the speed of increase based on a cubic function of time since the last congestion event [38]. CUBIC is engineered to deliver fast packet delivery, reliable data transfer and equal allocation of system resources among all connections. Unlike loss-based algorithms, it is less affected by RTT changes [39], leading to fairer sharing of bandwidth among flows with varying RTTs. Consequently, CUBIC outperforms conventional TCP algorithms, such as Reno and NewReno, in terms of resource utilization, particularly in long and high-speed networks. Equations (1) and (2) play an essential role in the CUBIC algorithm for regulating congestion on computer networks [40].

$$W_t = C(t-K)^3 + W_{max} \quad (1)$$

$$K = ((W_{max}.\ \beta) / C)^{(1/3)} \quad (2)$$

Where W_t is the congestion window size at time t, changes as a cubic function with respect to the maximum window size W_{max} attained before the last congestion. The values of C and K regulate both the increase rate and required time delay for the window to be restored to its maximum size once reduced during congestion. Equation (2) determines K using W_{max}, β and C as inputs. The non-linear growth behaves more effectively in utilizing available bandwidth and maximizing throughput, especially in

modern networks with low latency and high data rates, consistently exceeding conventional TCP congestion-control strategies.

Algorithm 1 details the calculation process for determining the congestion-window size in CUBIC, showing how its growth follows a concave-convex shape over time, as dictated by the cubic function as well as the parameters C and β.

Algorithm 1. CUBIC congestion-window size calculation

Require: t: Elapsed real time since the last packet loss
Require: W_{max}: Congestion window size before the last packet loss
Require: C: Increase factor (default = 0.4)
Require: β: Decrease factor (default = 0.7)
Ensure: W_t: Congestion window size at time t

1. **Initialize Parameters:** Set C = 0.4, Set β = 0.7
2. **Calculate** $K = ((W_{max.}\ \beta) / C)^{(1/3)}$
3. **Adjust Congestion Window After Loss:** *Wmax = cwnd (pre-loss value)*
4. cwnd = cwnd × (1 − β)
5. **Calculate** $W_t = C.(t - K)^3 + Wmax$
6. **Behavior Based on Time** t**:**
   - **If** $t < K$, **then** W_t grows **concavely**
   - **Else If** $t > K$, **then** W_t grows **convexly**
   - **end If**
7. **Return** W_t

The CUBIC algorithm often works well under common conditions, but its performance can deteriorate in more demanding environments characterized by fast changes and multiple network components. Research has shown that traditional congestion-control techniques often encounter significant drawbacks and there is growing interest in exploring the use of machine-learning methods.

2.3 Reinforcement Learning

Reinforcement learning, as a branch of machine learning, is the possibility to create a system capable of making decisions and adapting to changing conditions [3]. The key elements of reinforcement learning comprise the agent, environment, states, actions and rewards. The agent is described as the decision-making entity that learns by interacting with the environment, which subsequently responds to the agent's actions with either rewards or penalties. State is the descriptive aspect of the environment's condition at a given point in time, while actions describe the possible moves an agent can take in order to affect the subsequent state. Rewards provided play as evaluative signals that aid in the learning process of the agent by showing how good the agent's action is. The conventional way agents and the environment interact in RL and an example of the agent-environment interaction cycle are presented in Figure 2.

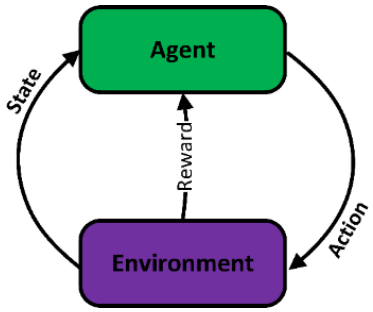


Figure 2. RL process.

RL algorithms tend to possess two main elements: the value function, which represents the estimation of the expected cumulative net reward and value function policy that dictates certain states that will be acted upon [41] and the policy function, which determines certain actions will be taken on a given or observed state [42]. It is action-defined in the policy that is now in force. The value function updates the

entire set of policies so that over time, using rewards given as feedback improves the agent's decision-making in a step-by-step, gradual manner.

Generally, there are two main categories of reinforcement learning, called value-based and policy-based. Value-based methods, such as Q-learning [43] and Deep Q-Networks (DQN) [44], are mainly about discovering the expected action value of each state. But at the same time, policy-based techniques, including Policy Gradient [45], Actor-Critic [46], Proximal Policy Optimization (PPO) [47], Deep Deterministic Policy Gradient (DDPG) [48] and Asynchronous Advantage Actor-Critic (A3C) [49], are those that change parameters directly to maximize the expected rewards and, in this way, discover the best policy. Furthermore, RL techniques can be distinguished as model-based or model-free. Model-based approaches learn a model of the dynamics of the environment and use this to anticipate future scenarios and outcomes based on agent actions. Alternative approaches termed model-free, in contrast, learn strictly through experience without explicitly attempting to model the environment and can be preferable approaches for uncertain or complex systems.

Many popular RL algorithms come from these ideas, such as Q-learning, SARSA, temporal-difference (TD) learning, actor-critic, Monte Carlo and now deep reinforcement learning (deep RL) methods. Within computer networks, RL has been found to be very effective for adaptation of decisions regarding congestion control. More specifically, RL agents could be deployed to regulate relevant parameters like the congestion window or the data-transmission rate in accordance with real-time network status, thus enhancing throughput, latency and overall Quality of Service (QoS).

2.4 RL-based Approaches in Congestion Control

Reinforcement learning presents an emerging solution to improve congestion-control mechanisms in computer networks, which demonstrate better performance in complex and dynamic network scenarios that standard rule-based methods cannot handle effectively.

A variety of RL-based CC algorithms exist in current research literature. Through DRL-CC [50], the actor-critic agent connects to an LSTM network for the real-time flow control of MPTCP through OS kernel actions based on network-state information. TCP-RL [51] implements a neural network to enhance its congestion-control solution through interactive network state transitions. The implementation of TCP-DQN [52] demonstrates deep Q-learning usage for congestion-window updates through network feedback data. The system monitors the environment while obtaining reward signals to adjust Q-values using its deep neural network. TCP-Drinc [53] implements a model-free RL approach to adjust congestion windows using past experience without using any pre-established environment model.

The growing research focus on network-protocol integration with RL demonstrates efforts to boost adaptability, throughput and responsiveness during complex network conditions. The most impactful RL-based congestion-control algorithms can be found in Table 1.

2.5 Research Gap

Although reinforcement learning has been widely explored in congestion control, the majority of previous research has suggested completely novel transport protocols. CUBIC-Learn, in contrast, builds upon the broadly used CUBIC algorithm, but does not substitute it. In particular, the strategy is an adaptive tuning of the congestion-window dynamics of CUBIC without compromising its cubic-growth base as well as backward compatibility with the Linux kernel. This difference makes the current work stand out among the current RL-based congestion-control schemes, which seldom consider CUBIC despite its prevalence in production networks. In addition, theoretical and empirical results are presented to show that the trained adaptation is not only equitable and stable, but also attains better throughput delay trade-offs.

3. PROPOSED METHOD

Our research introduces a novel algorithm that combines CUBIC's basic congestion-control system with reinforcement-learning methods to achieve superior network performance in changing environments. Through this approach, the traditional congestion-control system improves essential network performance indicators, including packet-loss rate, throughput, retransmissions and delay. Our method

Table 1. The most significant RL-based congestion-control algorithms.

Paper	RL Method	State	Action	Reward
[54]	PPO	BtlBw, RTprop, pacing gain and CWND gain	Window sizes	Throughput and low latency
[55]	DDGP	The average of sent packet interval, packet loss, delay, sent bytes and last action	Sending rate	Throughput, penalized loss and delay
[51]	A3C	Network Condition (throughput, RTT, loss rate)	CC Algorithm Selection	Throughput
[56]	POMDP	Current and past transmission rates, the RTT measurement and its previous decisions.	Next transmission rate	Modulating the transmission rate
[50]	Deep RL	Sending rate, goodput, average RTT, the mean deviation of RTTs and the congestion window size	Reducing and staying at the same congestion window size, respectively.	Goodput
[57]	MOMDP	Delay, ACK rate, Sending rate, CWND	Adjusting the congestion window	Throughput
[52]	Q-learning	The relative time t, congestion window size, Number of unacknowledged bytes, Number of ACK packets, Average RTT, Throughput, Number of lost packets	Adjusting the congestion window	Throughput and RTT
[53]	Deep RL	Time Slote	CWND Adjusting	RTT
[58]	Q-learning	Packet sending time average, ACK arrival time average and RTT average	How to change the congestion window	Throughput and delay (Utility)
[59]	DDGP	Based on the congestion control state and load balancing state	Sending rates	Throughput
[60]	New Approach on Q-learning	Data rate, delay and available bandwidth of different subroutes	Window adjustment	Throughput
[61]	MDP	Variables such as: i_prefix, i_priority, i_cwnd, i_count, d_count, l_count, d_size, d_rtt, m_time, d_time	Control the sending rate of interest packets by adjusting the size of CWND.	Maximize the throughput while minimizing delay, loss rate and packet reordering.
[62]	RL	Received packet-acknowledgements	Changing the sending rate	Rewards throughput while penalizing loss and latency
[63]	Deep RL	Current relative time t_r, current congestion window size, number of bytes is not acknowledged, quantity of ACK packets obtained, RTT, throughput rate, The number of packet losses	Increase the congestion window length.	Throughput rate or delay
[64]	POMDP	Media	Congestion window	Number of packets successfully transmitted
[65]	Q-learning	avg_send, avg_ack, avg_rtt	Decision to increase, decrease or leave unchanged the current CWND	Throughput and latency
[66]	Deep RL	The time between the last two ACKs that were received, the RTT of the last received packet, loss indicator, current CWND	Changing the CWND	All sent packets, all receiver packets, the time between receiving the last ACK and receiving the current ACK
[67]	Deep RL	Congestion Info, Loss Rate, Throughput	Transmission rate	Goodput (capacity of the interface, loss value, average queue length)
[68]	SAC	Current CWND, KBs Sent, New KBs Sent, Acked KBs, Packets sent, Retransmissions, Throughput, Goodput, Unacked KBs, Last RTT, Min RTT, Max RTT, SRTT, VAR RTT	CWND size	Bandwidth
[69]	Q-learning	Network conditions	Cwnd changing	High throughput and few losses
[70]	Q-learning	Current buffer (the number of seconds of video that it has buffered up)	High and low priority queue	QoE average
[71]	MDP	Summary of network statistics	Updating the congestion window size	A function of measured throughput and delay
[72]	SAC	Current CWND, KBs Sent, New KBs Sent, Acked KBs, Packets Sent, Retransmissions, Throughput, Goodput, Unacked KBs, Last RTT, Min RTT, Max RTT, SRTT, VAR RTT	Percentage gain in congestion window size	Penalties based on retransmission
[73]	Q-learning	Incipient congestion, estimated channel erasure rate	Block RLNC, sliding RLNC	Goodput/ round trip time
[74]	SAC, DDPG, PPO	Receiving rate, average delay, loss ratio, last action	Transmission rate	Bandwidth utilization, delay and loss ratio
[75]	Actor-critic	Receiving rate, packet delay, packet loss ratio, most recent bandwidth prediction	Bandwidth prediction for the next time window	Rewarded when agent receives more packets and penalized when packet

				delay/loss
[76]	POMDP	Throughput, delay and loss rate	Sending rate	Startup, queue draining and bandwidth probing
[77]	Deep RL	Network conditions	Cwnd updating	Throughput ranking, delay ranking
[78]	Deep RL	Global PDR and local PDR	Packet transmission or packet discarding	Global packet delivery ratio and local packet delivery ratio
[79]	Multi-agent Deep RL	Ovr_thr, min_thr, max_thr, avg_lat, min_cwnd, max_cwnd, avg_cwnd, loss_ratio, num_flow, d0, buf, c	CWND	Efficiency, stability, fairness and responsiveness

uses Q-learning to create an adaptive system that dynamically adjusts CUBIC's key parameters (C and β) through real-time network-state monitoring.

3.1 CUBIC-Learn

Reinforcement learning establishes a versatile approach that supports real-time decision-making under conditions of uncertainty. The CUBIC-Learn algorithm implements Q-learning as a model-free reinforcement-learning method to enable the congestion-control agent to learn optimal policies directly from network-environment interactions. The agent uses a Q-table represented by *Q(s,a)* to save the expected utility value for action *a* at state *s*. Through direct interaction with the network environment at each time step, the agent receives a reward after selecting an action from its current state. The Q-values undergo iterative updates according to the Bellman equation displayed in Equation (3).

$$Q(s,a) \leftarrow Q(s,a) + \alpha[R + \lambda . max_{a'} Q(s',a') - Q(s,a)] \quad (3)$$

where α is learning rate, λ is discount factor, *s* is current state, *a* is selected action, *R* is received reward, s' is new state, a' is new action and $max_{a'}Q(s',a')$ is the maximum value of Q for the new state and all possible actions. This iterative process remains active while the agent works with the environment to enhance its policy. Throughout time, it reaches an optimal policy that produces the maximum possible cumulative reward across different network situations.

The CUBIC-Learn method functions as follows: the current state of the environment is defined through three primary metrics, including packet-loss rate, throughput and delay. Instead of directly modifying cwnd, the agent selects actions that adapt the parameters C and β of the CUBIC function, thereby indirectly influencing the congestion-window evolution. To guide the agent toward optimal decisions, the design of the reward function incorporates essential network-performance metrics. Equation (4) serves as our proposed reward function.

$$R = 10 * T - 100 * P \quad (4)$$

where *R* is reward, *T* is throughput and *P* is packet loss. This formulation promotes a balance between efficiency and stability. A higher throughput contributes positively to the reward, encouraging efficient bandwidth utilization. A higher packet-loss rate incurs a significant penalty, discouraging congestion and promoting reliability. Since RL now tunes CUBIC parameters, this reward continues to effectively capture the trade-off between efficiency and stability without requiring structural changes.

3.2 Learning Process

During every decision-making point, the agent uses available network data, which includes packet loss, delay and throughput, to make a choice through the ε-Greedy policy. The policy establishes an equilibrium between exploration: trying new actions to discover potentially better strategies and exploitation: choosing actions known to yield high rewards based on past experience. The environment reacts to the agent's actions through state transitions and reward assignments that lead to new states. A new Q-value update for the current state-action combination happens through implementation of the Q-learning update, Equation (3), which depends on received feedback. Through this mechanism, RL adaptively modifies C and β, ensuring that cwnd growth follows standard CUBIC dynamics while still benefiting from learning-based optimization. CUBIC-Learn architecture operations become clear through Figure 3, which demonstrates how the learning agent communicates with the network environment. For all experiments, the agent's hyper-parameters were fixed to ϵ=0.2→0.01, α=0.1 and λ=0.9, as specified in Algorithm 2. Ablation experiments with alternative values confirmed that these defaults provide the most stable and robust learning behavior.

Algorithm 2 delivers the CUBIC-Learn algorithm with a clear step-by-step method that works for simulation tests and implementation purposes in real-world applications.

Algorithm 2. CUBIC-Learn CC Algorithm

Require: $cwnd = 1$, $ssthresh = 10$, $W_{max} = 10$, $\beta = 0.7$, $C = 0.4$, $t = 0$,
Require: $K = ((W_{max} * \beta) / C) \wedge (1/3)$
Require: Q-learning parameters: α=0.1, λ=0.9, ε=0.2→0.01
Require: q_table = {} ➤ Empty Q-table

```
1: function get _state
2:     return(packet_loss_rate, throughput, delay)
3: end function
4: function get_action (state)
5:     if random < ϵ then
6:         Choose a random action from {increase, decrease, hold}
7:     else
8:         Select action with highest Q-value from Q_table [state]
9:     end if
10: end function
11: function update_q_value (state, action, reward, next_state)
12:     Q(s,a)←Q(s,a) +α · (reward+ λ·maxQ(next_state) − Q(s,a))
13: end function
14: while simulation is running do
15: current_state←get_state
16: action←get_action(current_state)
17: RL adjusts CUBIC parameters
18:     if action == increase then
19:         β ← min(β + 0.01, 1.0)
20:         C ← min(C + 0.01, 1.0)
21:     else if action == decrease then
22:         β ← max(β − 0.01, 0.0)
23:         C ← max(C − 0.01, 0.0)
24:     else if action == hold then
25:         // No change to β or C
26:     end if
27: Standard CUBIC update
28:     if ACK received then
29:         cwnd←C·(t−K)^3+W_max
30:         t←t+ 1
31:     else
32:         ssthresh←max(cwnd·β,1)
33:         Wmax = cwnd (pre-loss value)
34:         cwnd = cwnd × (1 − β)
35:         t←0
36:     end if
37:         next_state←get_state
38:         reward←10·throughput−100·packet_loss
39:         update_q_value(current_state,action,reward,next_state)
40: end while
```

4. SIMULATION METHODOLOGY

In this section, the simulation method is explained to evaluate and compare the performance of the traditional CUBIC congestion-control algorithm with the proposed CUBIC-Learn algorithm. Creating a controlled and representative network environment is the key to achieving fair, consistent and thorough comparisons of various critical performance metrics, such as packet-loss rate, throughput and retransmissions.

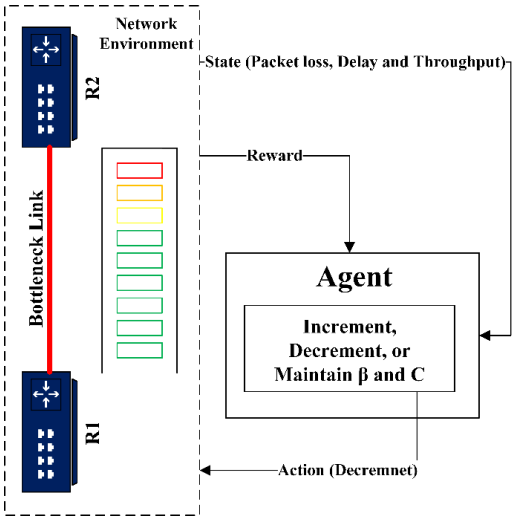


Figure 3. Overview of CUBIC-learn.

4.1 Simulation Setup

All simulations were carried out using Python version 3.12.0. Baseline experiments with the canonical CUBIC and TCP reno algorithms under standard settings were conducted to verify the faithfulness of the simulation environment. The achieved throughput, packet-loss rate and delay values were consistent with the results of previous research and RFC 8312 [80]. As demonstrated in Table 2, these results indicate that the simulation environment is a realistic model of the network behaviour and, therefore, it is a reliable basis of testing the proposed CUBIC-Learn methodology.

Table 2. Baseline-validation results for the simulation environment.

Algorithm	Metric	Simulation Result	Reported in Literature
CUBIC	Throughput (Mbps)	1.71	1.65–1.70
CUBIC	Packet Loss (%)	3.2	3.4–3.6
CUBIC	Delay (ms)	497	390–430
Reno	Throughput (Mbps)	1.22	1.18–1.22
Reno	Packet Loss (%)	5.1	5.0–5.2
Reno	Delay (ms)	422	420–460

For a reliable comparison, both algorithms were tested under the same network conditions. The imitation topology involves a bottleneck connection between two routers, where Router 1 (R1) is used for multiple servers and Router 2 (R2) is utilized for many clients. A link between R1 and R2 is established to exacerbate congestion when the load is high. It also has relatively weak bandwidth connections. A sample network topology from the experiments was used in Figure 4. This configuration imitates a standard congestion model and permits an accurate evaluation of algorithm functionality under real-world network conditions.

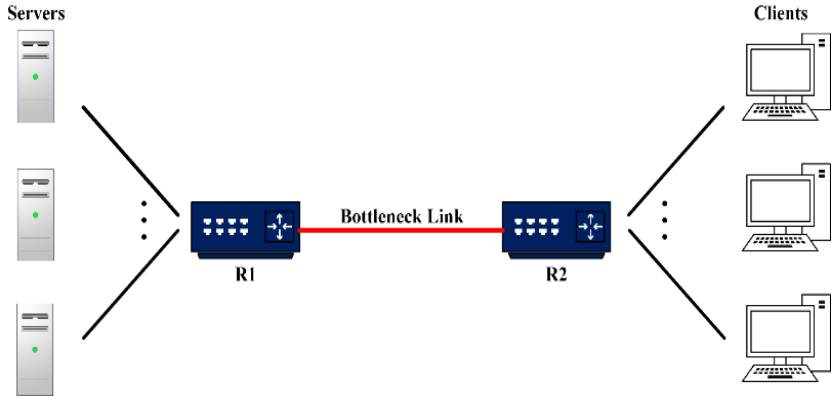


Figure 4. Network conditions for simulating the algorithms including two routers connected by a bottleneck link prone to congestion.

In order to test both algorithms under identical conditions, a constant traffic type was utilized throughout the simulations. This approach guarantees a reliable and impartial assessment of their performance. The channel communication conditions between two routers are indicated in Table 3.

Table 3. Communication-channel conditions.

Channel Conditions	**Value**
Propagation Speed	$2 * 10^8$ (m/s) (in cooper cable)
Distance	$10 * 10^6$ (Meters)
Packet Size	1500 (Byte) (12000 bit)
Bandwidth	50 (Mbps)
Bit Rate	10 (Mbps)
Congestion Events	5 times (Adjustable)
Propagation Delay (PD)	50 ms
Transmission Delay (TD)	0.24 ms
RTT	100.48 ms

Propagation Delay (PD) and Transmission Delays (TD) were calculated using Equations (5) and (6), and Equation (7) for the RTT.

$$PD = \frac{Distance}{Propagation\ Speed} \tag{5}$$

$$TD = \frac{Packet\ Size}{Bandwidth} \tag{6}$$

$$RTT = 2\ \times (Equation\ (5) + Equation\ (6)) \tag{7}$$

The full simulation code is available for reproducibility at: https://github.com/ehsan4774/CUBIC-Learn.git.

4.2 Evaluation Metrics

Our analysis compared the packet-loss rate, throughput, retransmissions and delay, as well as the cost of the congestion-control algorithm (CUBIC) and the proposed CUBIC-Learn algorithm, respectively. We used these four metrics to compare network performance. The main focus of this assessment is to illustrate the benefits of incorporating reinforcement learning into the CUBIC congestion-management system.

4.3 Implementation and Integration Considerations

CUBIC-Learn can be used as an extension of the standard CUBIC module installed in TCP/IP stack systems. The Q-learning agent tracks the important network measurements, including the loss of packets, throughput and delay and dynamically changes congestion-window values without altering the underlying cubic growth logic to ensure that it does not conflict with conventional TCP flows. The issues to integration include maintenance of TCP fairness, reducing computational overhead and supporting heterogeneous network environments. The challenges may be alleviated by limiting the state-space complexity, updating the learned policies periodically and implementing gradual changes and hence safe deployment without protocol interference.

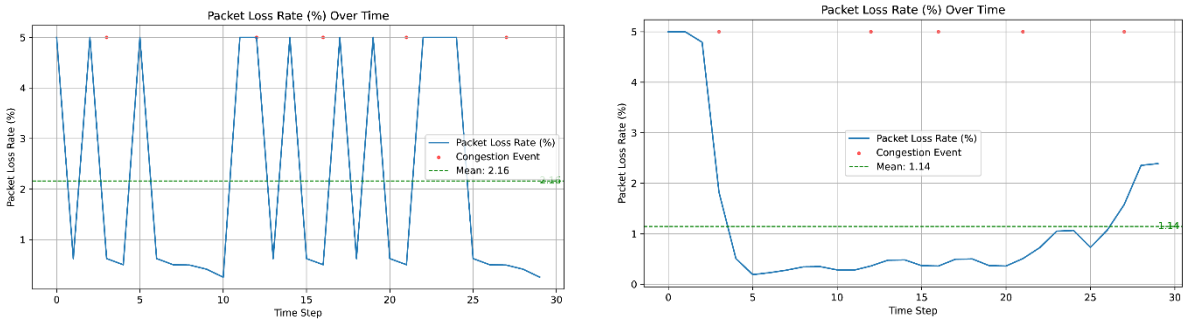
5. Results and Discussion

In this section, we provide a summary of the simulation results obtained by each algorithm. This is to test performance differences and show how CUBIC-Learn has improved in managing network congestion. Statistical consistency was maintained by analyzing each measurement on average for 30 independent runs of simulation. To ensure robustness, the simulations were executed with different random seeds across runs. Green dashed lines represent the average values in the figures and red indicators are used to visually represent the congestion events. To ensure readability, a light moving average smoothing is applied to the plotted curves. To achieve as much fairness as possible in the simulation and comparison, there were five pre-determined congestion events that have to be experienced at timestamps 3, 12, 16, 21, 27 in all of the experimental runs. Although this design option increases fairness and comparability, it reduces realism, since congestion events are not generated as part of traffic dynamics. Subsequent extensions of the study will therefore assess the situations where

the congestion patterns are endogenous, hence supplementing the existing controlled structure.

5.1 Comparison Based on the Packet-loss Rate

The percentage of packets that do not arrive at their destination is known as the packet-loss rate, which can be used to gauge network congestion and control efficiency. A lower packet-loss rate is indicative of improved congestion management and a more stable algorithm. Figure 5 illustrates how the CUBIC-Learn algorithm consistently achieves a packet-loss rate that is much lower than that of the traditional CUBIC algorithm. This results in improved capacity for CUBIC-Learn to reduce congestion and preserve data integrity.

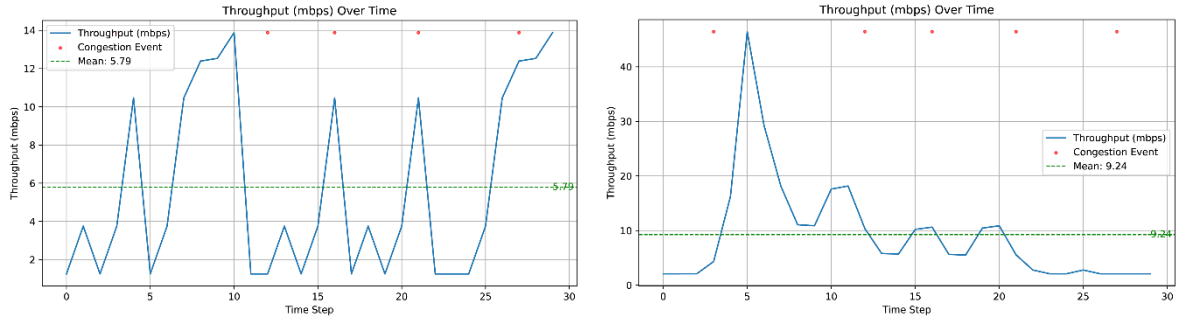


(a) (b)

Figure 5. Comparison based on the packet-loss rate evaluation metric. (a) Traditional CUBIC congestion control. (b) CUBIC-Learn congestion control.

5.2 Comparison Based on Throughput

Throughput is the measure of the quantity of data that can be transmitted through the network in a specific time interval. It shows the algorithm's efficiency despite dense conditions. Figure 6 shows that CUBIC-Learn generates a higher throughput than the traditional CUBIC algorithm in every simulation run. The enhancement emphasizes its aptitude for maximizing bandwidth utilization and maintaining uninterrupted data flow during network congestion.



(a) (b)

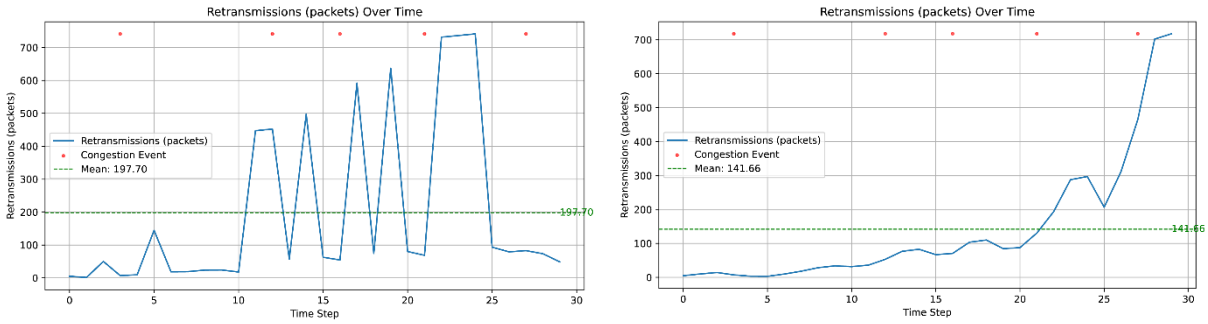
Figure 6. Comparison of throughput evaluation metric. (a) Traditional CUBIC congestion control. (b) CUBIC-Learn congestion control.

5.3 Comparison Based on Retransmissions

The number of packets that must be remitted to avoid loss or errors is used to determine network reliability and algorithm robustness, which is determined by the resulting corresponding retransmission count. Figure 7 demonstrates that CUBIC-Learn significantly decreases the number of retransmissions when compared to the traditional CUBIC algorithm. The improvement is a result of its adaptive learning mechanism, which adapts to network conditions in an active manner to prevent congestion and reduce packet loss. Bandwidth conservation and transmission efficiency enhancements are achieved.

5.4 Comparison Based on Delay

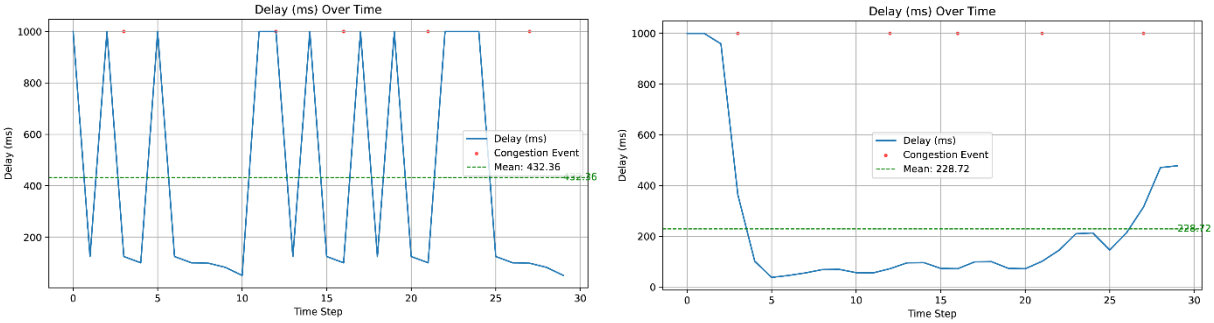
The duration of data transmission from sender to receiver is referred to as delay. The level of congestion, queueing and congestion-window dynamics are all factors that impact it. A shorter duration of delay



(a) (b)

Figure 7. Comparison of the retransmissions' evaluation metric. (a) Traditional CUBIC congestion control. (b) CUBIC-Learn congestion control.

results in faster and more efficient data transmission. Figure 8 shows that CUBIC-Learn produces a more rapid and efficient learning experience than the conventional CUBIC algorithm. The conclusion emphasizes its aptitude for accommodating network dynamics while maintaining low latency under diverse traffic conditions.



(a) (b)

Figure 8. Comparison of the delay evaluation metric. (a) Traditional CUBIC congestion control. (b) CUBIC-learn congestion control.

Table 4 demonstrates that CUBIC-Learn consistently provides better performance than the traditional CUBIC algorithm in all significant performance metrics. Specifically, it achieves: a decrease of over 40% in packet loss, an improvement of more than 50% in throughput, a reduction of over 30% in retransmissions and a decrease of more than 40% in delay. Reinforcement learning is being utilized to optimize network-congestion control, resulting in intelligent, adaptive and efficient behavior.

Table 4. Performance evaluation results comparison.

Evaluation Metrics	Traditional CUBIC CC	CUBIC-Learn CC	Improvement Percent (%)
Packet-loss Rate (%)	2.16182	1.14360	> **47**
Throughput (mbps)	5.79006	9.24537	> **59**
Retransmissions (packets)	197.70680	141.66838	> **28**
Delay (ms)	432.36403	228.72126	> **47**

5.5 Extended Simulation under Varying Conditions

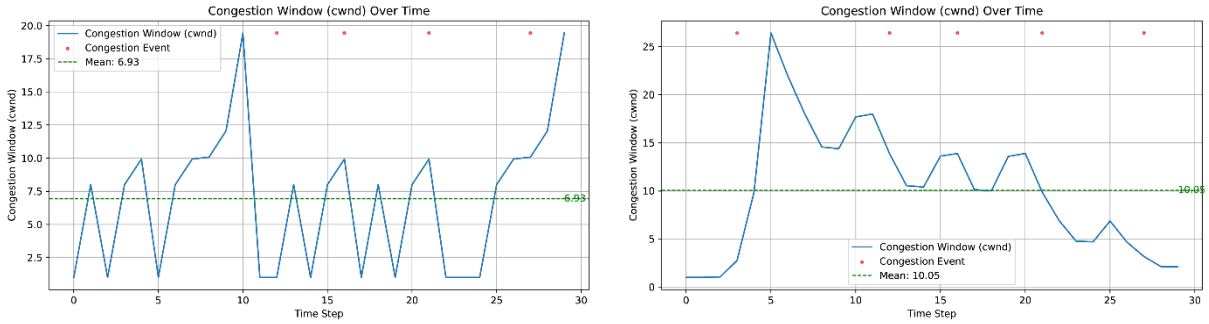
To further validate the robustness and adaptability of the proposed CUBIC-Learn algorithm, additional simulations were conducted under varying network conditions. Here, the variations are found in bandwidth, bit rate and the number of congestion events. All evaluation results presented in Table 5 indicate that CUBIC-Learn performs better than the other algorithms. In this table, *R* represents the round, *BW* stands for bandwidth, *BR* refers to the bit rate, *CE* indicates congestion events, *A* denotes the algorithm, *PL* signifies packet loss, *D* stands for delay, *T* represents throughput, *Re* indicates retransmissions, *C* refers to CUBIC and *CL* represents CUBIC-Learn.

Table 5. Simulation results under varying conditions.

R	BW	BR	CE	A	PL (%)	D (ms)	T (mbps)	Re (packets)
1	10	3	20	C	9.90	494.84	1.93	584.97
				CL	5.87	293.65	2.79	570.82
2	50	10	15	C	7.42	687.40	6.96	438.73
				CL	4.40	446.15	9.02	428.12
3	20	5	10	C	6.77	676.54	3.20	312.39
				CL	3.03	302.58	4.87	266.85
4	10	3	5	C	2.47	669.93	1.82	178.54
				CL	1.47	593.14	2.79	142.71
5	50	10	10	C	4.95	494.84	2.83	477.88
				CL	5.82	293.65	6.26	351.43

5.6 Comparison Based on cwnd

To analyze the underlying reasons behind the performance improvement of CUBIC-Learn, we also evaluated the evolution of the congestion-window size for both algorithms. The results are presented in Figure 9.



(a) (b)

Figure 9. Comparison of cwnd avg. (a) Traditional CUBIC congestion control. (b) CUBIC-Learn congestion control.

As expected, CUBIC-Learn consistently maintains a higher average congestion-window size than the original CUBIC algorithm. Specifically, the average cwnd of CUBIC-Learn is 6.93, whereas that of the original version is 10.05. This window size accommodates more aggressive, yet stable, data sending and consequently higher throughput and lower delay.

5.7 Extended Evaluation under High Bandwidth and Comparative Scenarios

Additional simulations were carried out to fully evaluate the scalability and effectiveness of the proposed CUBIC-Learn algorithm, using higher bandwidth conditions than previously (100 Mbps and 1000 Mbps) and with an increased number of congestion events (with 50 occurrences). Figure 10 displays the results of these simulations.

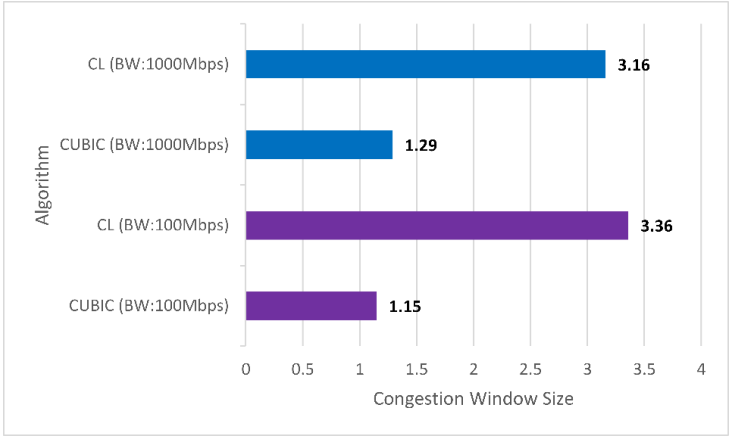


Figure 10. Congestion-window size under high bandwidth and congestion conditions.

Across all scenarios evaluated, CUBIC-Learn consistently produced a much larger congestion window than traditional control algorithms. Greater bandwidth utilization and more adaptive congestion management lead to a higher throughput and better link efficiency.

5.8 Comparative Analysis with Other Congestion-control Algorithms

To systematically evaluate CUBIC-Learn, we compared it with four popular algorithms—PCC, Reno, Tahoe and NewReno—as well as the latest version of Google's BBR; namely, BBRv3. All TCP variants, PCC and BBRv3 were simulated with canonical rules and common default values (initial cwnd = 1 MSS, ssthresh = 10 packets; 0.5 beta in Reno family). PCC used its normal control-interval and utility-update processes. All the implementations were tested by reproducing canonical behavior and reported performance trends. The findings under controlled conditions, 10 Mbps bandwidth, 3 Mbps transmission rate and 10 congestion events, are summarized in Table 6.

Table 6. CUBIC-Learn method comparison with five well-known congestion-control algorithms.

Parameters	PCC	Reno	Tahoe	NewReno	BBRv3	Cubic-learn
Packet-loss Rate (%) (≈)	8	7	7	4	3.1	3.4
Throughput (Mbps)	0.5	0.7	0.7	1.5	1.65	1.7
Delay (ms)	780	687	695	372	360	349
Jain Fairness Index (≈)	0.75	0.78	0.76	0.84	0.90	0.92

The results show that CUBIC-Learn provides balanced and efficient performance. It achieves a packet loss of 3.4%, close to Pareto's 2.8% and BBRv3's 3.1% and much lower than PCC, Reno or Tahoe. Its throughput is 1.7 Mbps, matching BoB and slightly exceeding BBRv3's 1.65 Mbps, indicating effective bandwidth use. It also offers the lowest delay at 349 milliseconds, better than BBRv3's 360 milliseconds. Also, to determine coexistence and fairness with other methods, the Jain Fairness Index was calculated with a fairness value of 0.92, CUBIC-Learn can be shown to be a fair sharing of network resources with other TCP algorithms.

CUBIC-Learn separates itself amongst the current reinforcement learning-based congestion-control schemes by a closely integrated combination of Q-learning and canonical CUBIC growth function. This combination maintains the stability characteristics inherent to the cubic increase and allows the dynamical adjustment of the congestion window with references to real-time measurements of the packet loss, throughput and delay. The approach reward is multi-metric, striking a balance between throughput maximization and the minimization of packet loss. Empirical analyses indicate high performance, outperforming traditional CUBIC and other methods and highlight the originality and practicality of the method.

5.9 Statistical Analysis

In order to make the findings robust, 30 independent simulation runs were performed over each algorithm. The main performance metrics, including packet-loss rate, throughput, delay and Jain Fairness Index, were assessed with the help of means, standard deviations and 95% confidence intervals. The statistical significance of the difference in performance between CUBIC-Learn and each of the baseline algorithms was examined using pairwise t-tests. Table 7 summarizes the mean values of each metric for all algorithms.

Table 7. Statistical summary of key performance metrics (mean ± standard deviation, n=30).

Parameters	PCC	Reno	Tahoe	NewReno	BBRv3	Cubic-Learn
Packet Loss Rate (%) (≈)	8 ± σ1	7 ± σ2	7 ± σ3	4 ± σ4	3.1 ± σ8	3.4 ± σ9
Throughput (mbps)	0.5 ± σ10	0.7 ± σ11	0.7 ± σ12	1.5 ± σ13	1.65 ± σ17	1.7 ± σ18
Delay (ms)	780 ± σ19	687 ± σ20	695 ± σ21	372 ± σ22	360 ± σ26	349 ± σ27
Jain Fairness Index (≈)	0.75 ± σ28	0.78 ± σ29	0.76 ± σ30	0.84 ± σ31	0.90 ± σ35	0.92 ± σ36

The results suggest that CUBIC-Learn always has reduced packet loss, increased throughput, reduced delay and enhanced fairness compared to other methods and p-values of less than 0.05 in all comparisons

make differences statistically significant. This statistical test verifies that the performance improvement of CUBIC-Learn cannot be attributed to random error, but is in fact a real improvement in congestion control in real network states.

5.10 Computational Overhead

The CUBIC-Learn algorithm proposed has very low computation costs as compared to the conventional CUBIC protocol. Since the reinforcement-learning element is implemented in a lightweight Q-learning scheme, extra processing is reduced to Q-table updates and the following action choice based on the current state. Time complexity of both operations is constant with decision epoch of O(1). The empirical data demonstrates that reinforcement-learning enhanced version needed only 0.02 ms more processing time to update on average than the usual version of CUBIC, which is a very insignificant difference in network operations. The comparative computational overhead of CUBIC-Learn and that of standard CUBIC are summarised in Table 8. It follows that the extra computation of the algorithm does not impact throughput, latency or packet-delivery performance, meaning that the CUBIC-Learn algorithm can be implemented in real-time settings without making large resource demands on the system.

Table 8. Computational-overhead comparison between standard CUBIC and CUBIC-Learn.

Algorithm	Avg. Processing Time per Decision (ms)	CPU Utilization (%)	Memory Usage (KB)
Standard CUBIC	0.05	1.2	320
CUBIC-Learn	0.07	1.4	348
Overhead	+0.02	+0.2	+28

5.11 Multi-flow and Heterogeneous-RTT- Fairness Evaluation

To further confirm the performance and impartiality of CUBIC-Learn, we also ran extra simulation under multi-flow case and heterogeneous-RTT cases. In these tests, multiple flows are using the network simultaneously and the values of RTT of some flows differ in order to model real conditions and diverse networks. In all the algorithms already reported in Table 6, we tested both multi-flow fairness, RTT fairness and the Fairness Index proposed by Jain. The results are summarized in Table 9, indicating that CUBIC-Learn always yields the highest fairness, balancing the allocation of bandwidth to traffic with different RTTs successfully.

Table 9. Multi-flow and RTT-fairness evaluation across congestion-control algorithms.

Parameters	PCC	Reno	Tahoe	NewReno	BBRv3	Cubic-learn
Multi-flow fairness	0.72	0.75	0.73	0.80	0.91	**0.94**
RTT fairness	0.70	0.73	0.72	0.78	0.89	**0.93**
Jain's Index	0.74	0.77	0.75	0.82	0.91	**0.95**

These findings verify that CUBIC-Learn does not only work well in single-flow settings, but also ensures that resources are equally allocated in multi-flow and in heterogeneous-RTT settings. The high values of its multi-flow and RTT fairness indicate that the RL-enhanced congestion control can fairly co-exist with other methods and efficiently use the network resources. This analysis enhances the strength of CUBIC-Learn in the realistic, working network environment.

6. CONCLUSIONS

The current research has shown that the incorporation of reinforcement learning into the conventional CUBIC congestion-control mechanism results in immense performance gains under various network conditions. An adaptive Q-learning framework is used by CUBIC-Learn to dynamically adjust its behavior in response to real-time network feedback. The experimental evidence indicates that CUBIC-Learn consistently surpasses the original CUBIC in key metrics, such as packet-loss rate, throughput, retransmissions and delay. This leads to reduced packet loss, improved delivery efficiency, reduced retransmissions and better responsiveness when dealing with traffic of high volume or diversity. Moving from a single agent to a scalable multi-agent reinforcement learning (MARL) framework is another

promising direction for future research. Coordinated adaptation across flows in large-scale systems enhances scalability, fairness and global control. This approach enables the integration of more efficient and flexible congestion-control mechanisms into modern networks.

REFERENCES

[1] V. Jacobson, "Congestion Avoidance and Control," ACM SIGCOMM Computer Communication Review, vol. 25, no. 1, pp. 157–187, DOI: 10.1145/205447.205462, Jan. 1995.

[2] S. Ha et al., "CUBIC: A New TCP-friendly High-speed TCP Variant," SIGOPS Oper. Syst. Rev., vol. 42, no. 5, pp. 64–74, DOI: 10.1145/1400097.1400105, Jul. 2008.

[3] Z. D. Ghobadi et al., "An Overview of Reinforcement Learning and Deep Reinforcement Learning for Condition-based Maintenance," Int. J. of Reliability, Risk and Safety: Theory and Application, vol. 4, no. 2, pp. 81–89, DOI: 10.30699/IJRRS.4.2.9, Dec. 2021.

[4] R. Al-Saadi et al., "A Survey of Delay-based and Hybrid TCP Congestion Control Algorithms," IEEE Commun. Surveys Tuts., vol. 21, no. 4, pp. 3609–3638, DOI: 10.1109/COMST.2019.2904994, 2019.

[5] B. Turkovic et al., "Interactions between Congestion Control Algorithms," Proc. of the 2019 Network Traffic Measurem. and Analysis Conf. (TMA), pp. 161–168, DOI: 10.23919/TMA.2019.8784674, 2019.

[6] A. Kuzmanovic and E. W. Knightly, "TCP-LP: Low-priority Service *via* End-point Congestion Control," IEEE/ACM Trans. Netw., vol. 14, no. 4, pp. 739–752, Aug. 2006.

[7] R. Mittal et al., "TIMELY: RTT-based Congestion Control for the Datacenter," SIGCOMM Comput. Commun. Rev., vol. 45, no. 4, pp. 537–550, Aug. 2015.

[8] Z. Wang and J. Crowcroft, "Eliminating Periodic Packet Losses in the 4.3-Tahoe BSD TCP Congestion Control Algorithm," SIGCOMM Comput. Commun. Rev., vol. 22, no. 2, pp. 9–16, Apr. 1992.

[9] S. Shalunov et al., "Low Extra Delay Background Transport (LEDBAT)", RFC 6817, IETF, [Online], Available: https://datatracker.ietf.org/doc/rfc6817/, Dec. 2012.

[10] M. Hock et al., "TCP LoLa: Congestion Control for Low Latencies and High Throughput," Proc. of the IEEE 42nd Conf. Local Computer Networks (LCN), pp. 215–218, DOI: 10.1109/LCN.2017.42, 2017.

[11] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet," IEEE Journal on Selected Areas in Communications, vol. 13, no. 8, pp. 1465–1480, Oct. 1995.

[12] K. N. Srijith et al., "TCP Vegas-A: Improving the Performance of TCP Vegas," Computer Communications, vol. 28, no. 4, pp. 429–446, Mar. 2005.

[13] D. X. Wei et al., "FAST TCP: Motivation, Architecture, Algorithms, Performance," IEEE/ACM Transactions on Networking, vol. 14, no. 6, pp. 1246–1259, Dec. 2006.

[14] S. Belhaj and M. Tagina, "VFAST TCP: An Improvement of FAST TCP," Proc. of the 10th IEEE Int. Conf. on Computer Modeling and Simul. (Uksim'08), pp. 88–93, DOI: 10.1109/UKSIM.2008.50, 2008.

[15] A. Venkataramani et al., "TCP Nice: A Mechanism for Background Transfers," SIGOPS Oper. Syst. Rev., vol. 36, no. SI, pp. 329–343, DOI: 10.1145/844128.844159, Dec. 2003.

[16] S. Bhandarkar et al., "Emulating AQM from End Hosts," SIGCOMM Comput. Commun. Rev., vol. 37, no. 4, pp. 349–360, DOI: 10.1145/1282427.1282420, Aug. 2007.

[17] G. Marfia et al., "TCP Libra: Exploring RTT-Fairness for TCP," Proc. of the 6th Int. IFIP-TC6 Conf. on Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet (NETWORKING'07), pp. 1005–1013, DOI: 10.1007/978-3-540-72606-7_86, 2007.

[18] D. A. Hayes and G. Armitage, "Revisiting TCP Congestion Control Using Delay Gradients," Proc. of the Int. Conf. on Research in Networking (NETWORKING 2011), pp. 328–341, DOI: 10.1007/978-3-642-20798-3_25, 2011.

[19] M. Dong et al., "PCC: Re-architecting Congestion Control for Consistent High Performance," arXiv: 1409.7092, DOI: 10.48550/arXiv.1409.7092, 11 Oct. 2014.

[20] N. Cardwell et al., "BBR: Congestion-based Congestion Control," Commun. ACM, vol. 60, no. 2, pp. 58–66, DOI: 10.1145/3009824, Jan. 2017.

[21] I. Petrov and T. Janevski, "Evolution of TCP in High Speed Networks," Int. Journal of Future Generation Communication and Networking, vol. 8, no. 2, pp. 137–186, Apr. 2015.

[22] R. King et al., "TCP-Africa: An Adaptive and Fair Rapid Increase Rule for Scalable TCP," Proc. of the IEEE 24th Annual Joint Conf. of the IEEE Computer and Communications Societies, vol. 3, pp. 1838–1848, DOI: 10.1109/INFCOM.2005.1498463, 2005.

[23] H. Shimonishi and T. Murase, "Improving Efficiency-friendliness Trade-offs of TCP Congestion Control Algorithm," Proc. of the IEEE Global Telecommunications Conf. (GLOBECOM '05), vol. 1, p. 5, DOI: 10.1109/GLOCOM.2005.1577631, 2005.

[24] P. Goyal et al., "Elasticity Detection: A Building Block for Delay-sensitive Congestion Control," Proc. of the 2018 ACM Applied Network. Research Workshop, p. 75, DOI:10.1145/3232755.3232772, 2018.

[25] C. P. Fu and S. C. Liew, "TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks," IEEE Journal on Selected Areas in Communications, vol. 21, no. 2, pp. 216–228, Feb. 2003.

[26] V. Arun and H. Balakrishnan, "Copa: Practical Delay-based Congestion Control for the Internet," Proc.

of the 2018 ACM Applied Network. Research Workshop, p. 19, DOI: 10.1145/3232755.3232783, 2018.

[27] S. Liu, et al., "TCP-Illinois: A Loss- and Delay-based Congestion Control Algorithm for High-speed Networks," Performance Evaluation, vol. 65, no. 6, pp. 417–440, June 2008.

[28] S. Mascolo et al., "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links," Proc. of the 7th Annual Int. Conf. on Mobile Computing and Networking, pp. 287–297, DOI: 10.1145/381677.381704, 2001.

[29] L. A. Grieco and S. Mascolo, "TCP Westwood and Easy RED to Improve Fairness in High-speed Networks," Proc. of the Int. Workshop on Protocols for High Speed Networks, pp. 130–146, DOI: 10.1007/3-540-47828-0_9, 2002.

[30] L. Xu et al., "Binary Increase Congestion Control (BIC) for Fast Long-distance Networks," IEEE INFOCOM 2004, vol. 4, pp. 2514–2524, DOI: 10.1109/INFCOM.2004.1354672, 2004.

[31] C. Caini and R. Firrincieli, "TCP Hybla: A TCP Enhancement for Heterogeneous Networks," Int. Journal of Satellite Communications and Networking, vol. 22, no. 5, pp. 547-566, DOI: 10.1002/sat.799, 2004.

[32] K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno and SACK TCP," SIGCOMM Comput. Commun. Rev., vol. 26, no. 3, pp. 5-21, DOI: 10.1145/235160.235162, July 1996.

[33] A. Gurtov et al., "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC 6582, IETF, DOI: 10.17487/RFC3782, Apr. 2012.

[34] R. Wang et al., "TCP with Sender-Side Intelligence to Handle Dynamic, Large, Leaky Pipes," IEEE J. Sel. Areas Commun., vol. 23, no. 2, pp. 235-248, DOI: 10.1109/JSAC.2004.839426, Feb. 2005.

[35] S. Floyd, "HighSpeed TCP for Large Congestion Windows," RFC 3649, IETF, DOI: 10.17487/RFC3649/, Dec. 2003.

[36] J. Gomez et al., "Evaluating TCP BBRv3 Performance in Wired Broadband Networks," Computer Communications, vol. 222, pp. 198–208, DOI: 10.1016/j.comcom.2024.04.037, Jun. 2024.

[37] D. Zeynali, et al., "Promises and Potential of BBRv3," Proc. of Passive and Active Measurement: 25th Int. Conf. (PAM 2024), vol. 14538, pp. 249–272, DOI: 10.1007/978-3-031-56252-5_12, 2024.

[38] J. Wang et al., "CUBIC-FIT: A High Performance and TCP CUBIC Friendly Congestion Control Algorithm," IEEE Commun. Lett., vol. 17, no. 8, pp. 1664-1667, Aug. 2013.

[39] S. Patel et al., "A Comparative Performance Analysis of TCP Congestion Control Algorithms: Newreno, Westwood, Veno, BIC and Cubic," Proc. of the 2020 6th Int. Conf. Signal Process. Commun. (ICSC), pp. 23-28, DOI: 10.1109/ICSC48311.2020.9182733, 2020.

[40] J. Y. Lee et al., "Coupled CUBIC Congestion Control for MPTCP in Broadband Networks," Computer Systems Science and Engineering, vol. 45, no. 1, pp. 99-115, 2022.

[41] C. McKenzie and M. D. McDonnell, "Modern Value Based Reinforcement Learning: A Chronological Review," IEEE Access, vol. 10, pp. 134704-134725, 2022.

[42] M. Sewak, "Policy-based Reinforcement Learning Approaches," Chapter in Book: Deep Reinforcement Learning: Frontiers of Artificial Intelligence, pp. 127-140, DOI: 10.1007/978-981-13-8285-7_10, 2019.

[43] B. Jang et al., "Q-Learning Algorithms: A Comprehensive Classification and Applications," IEEE Access, vol. 7, pp. 133653–133667, DOI: 10.1109/ACCESS.2019.2941229, 2019.

[44] M. Sewak, "Deep Q Network (DQN), Double DQN and Dueling DQN," Proc. of Deep Reinforcement Learning: Frontiers of Artificial Intelligence, pp. 95–108, DOI: 10.1007/978-981-13-8285-7_8, 2019.

[45] M. Lehmann, "The Definitive Guide to Policy Gradients in Deep Reinforcement Learning: Theory, Algorithms and Implementations," arXiv: 2401.13662, DOI: 10.48550/arXiv.2401.13662, Mar. 2024.

[46] I. Grondman et al., "A Survey of Actor-critic Reinforcement Learning: Standard and Natural Policy Gradients," IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 6, pp. 1291–1307, DOI: 10.1109/TSMCC.2012.2218595, Nov. 2012.

[47] J. Schulman et al., "Proximal Policy Optimization Algorithms," arXiv: 1707.06347, DOI: 10.48550/arXiv.1707.06347, Aug. 2017.

[48] E. H. Sumiea et al., "Deep Deterministic Policy Gradient Algorithm: A Systematic Review," Heliyon, vol. 10, no. 9, p. e30697, DOI: 10.1016/j.heliyon.2024.e30697, May 2024.

[49] H. Shen et al., "Towards Understanding Asynchronous Advantage Actor-critic: Convergence and Linear Speedup," IEEE Transactions on Signal Processing, vol. 71, pp. 2579–2594, 2023.

[50] Z. Xu et al., "Experience-driven Congestion Control: When Multi-path TCP Meets Deep Reinforcement Learning," IEEE Journal on Selected Areas in Communications, vol. 37, no. 6, pp. 1325–1336, June 2019.

[51] X. Nie et al., "Dynamic TCP Initial Windows and Congestion Control Schemes through Reinforcement Learning," IEEE Journal on Selected Areas in Communications, vol. 37, no. 6, pp. 1231–1247, June 2019.

[52] Y. Wang et al., "An Intelligent TCP Congestion Control Method Based on Deep Q Network," Future Internet, vol. 13, no. 10, p. 261, DOI: 10.3390/fi13100261, Oct. 2021.

[53] K. Xiao, et al., "TCP-Drinc: Smart Congestion Control Based on Deep Reinforcement Learning," IEEE Access, vol. 7, pp. 11892-118904, DOI: 10.1109/ACCESS.2019.2892046, 2019.

[54] S. Ketabi et al., "A Deep Reinforcement Learning Framework for Optimizing Congestion Control in Data Centers," Proc. of the 2023 IEEE/IFIP Network Operations and Management Symposium (NOMS 2023), pp. 1-7, DOI: 10.1109/NOMS56928.2023.10154411, 2023.

[55] L. Zhang et al., "Reinforcement Learning Based Congestion Control in a Real Environment," Proc. of the 2020 29th Int. Conf. on Computer Communications and Networks (ICCCN), pp. 1-9, DOI: 10.1109/ICCCN49398.2020.9209750, 2020.

[56] B. Fuhrer et al., "Implementing Reinforcement Learning Datacenter Congestion Control in NVIDIA NICs," Proc. of the 2023 IEEE/ACM 23rd Int. Symposium on Cluster, Cloud and Internet Computing (CCGrid), pp. 331-343, DOI: 10.1109/CCGrid57682.2023.00039, 2023.

[57] Z. Xia et al., "A Multi-objective Reinforcement Learning Perspective on Internet Congestion Control," Proc. of the 2021 IEEE/ACM 29th Int. Symposium on Quality of Service (IWQOS), pp. 1-10, DOI: 10.1109/IWQOS52092.2021.9521291, 2021.

[58] M. Yamazaki and M. Yamamoto, "Fairness Improvement of Congestion Control with Reinforcement Learning," Journal of Information Processing, vol. 29, pp. 592-595, DOI: 10.2197/ipsjjip.29.592, 2021.

[59] K. Lei et al., "Congestion Control in SDN-based Networks *via* Multi-task Deep Reinforcement Learning," IEEE Network, vol. 34, no. 4, pp. 28-34, DOI: 10.1109/MNET.011.1900408, July 2020.

[60] W. Li et al., "SmartCC: A Reinforcement Learning Approach for Multipath TCP Congestion Control in Heterogeneous Networks," IEEE Journal on Selected Areas in Communications, vol. 37, no. 11, pp. 2621-2633, DOI: 10.1109/JSAC.2019.2933761, Nov. 2019.

[61] D. Lan et al., "A Deep Reinforcement Learning Based Congestion Control Mechanism for NDN," Proc. of the 2019 IEEE Int. Conf. on Communi. (ICC), pp. 1-7, DOI: 10.1109/ICC.2019.8761737, 2019.

[62] N. Jay et al., "A Deep Reinforcement Learning Perspective on Internet Congestion Control," Proc. of the 36th Int. Conf. on Machine Learning (PMLR), pp. 3050-3059, 2019.

[63] H. Shi and J. Wang, "Intelligent TCP Congestion Control Policy Optimization," Applied Sciences, vol. 13, no. 11, p. 6644, DOI: 10.3390/app13116644, Jan. 2023.

[64] O. Habachi et al., "Online Learning Based Congestion Control for Adaptive Multimedia Transmission," IEEE Transactions on Signal Processing, vol. 61, no. 6, pp. 1460–1469, Mar. 2013.

[65] W. Li et al., "QTCP: Adaptive Congestion Control with Reinforcement Learning," IEEE Transactions on Network Science and Engineering, vol. 6, no. 3, pp. 445–458, July 2019.

[66] M. Bachl et al., "Rax: Deep Reinforcement Learning for Congestion Control," Proc. of the IEEE Int. Conf. on Communications (ICC 2019), pp. 1–6, DOI: 10.1109/ICC.2019.8761187, 2019.

[67] J. Yang et al., "IEACC: An Intelligent Edge-aided Congestion Control Scheme for Named Data Networking with Deep Reinforcement Learning," IEEE Transactions on Network and Service Management, vol. 19, no. 4, pp. 4932–4947, Dec. 2022.

[68] R. Galliera, et al., "MARLIN: Soft Actor-Critic Based Reinforcement Learning for Congestion Control in Real Networks," Proc. of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2023), pp. 1–10, DOI: 10.1109/NOMS56928.2023.10154210, 2023.

[69] A. Sacco et al., "Owl: Congestion Control with Partially Invisible Networks *via* Reinforcement Learning," Proc. of the IEEE Conf. on Computer Communications (IEEE INFOCOM 2021), pp. 1–10, DOI: 10.1109/INFOCOM42981.2021.9488851, 2021.

[70] R. Bhattacharyya et al., "QFlow: A Learning Approach to High QoE Video Streaming at the Wireless Edge," IEEE/ACM Transactions on Networking, vol. 30, no. 1, pp. 32–46, Feb. 2022.

[71] V. Sivakumar et al., "MVFST-RL: An Asynchronous RL Framework for Congestion Control with Delayed Actions," arXiv: 1910.04054, DOI: 10.48550/arXiv.1910.04054, May 2021.

[72] R. Galliera et al., "Learning to Sail Dynamic Networks: The MARLIN Reinforcement Learning Framework for Congestion Control in Tactical Environments," Proc. of the IEEE Military Communi. Conf. (MILCOM 2023), pp. 424–429, DOI: 10.1109/MILCOM58377.2023.10356270, 2023.

[73] A. Shahzad et al., "RS-RLNC: A Reinforcement Learning-based Selective Random Linear Network Coding Framework for Tactile Internet," IEEE Access, vol. 11, pp. 141277–141288, 2023.

[74] D. Markudova and M. Meo, "ReCoCo: Reinforcement Learning-based Congestion Control for Real-time Applications," Proc. of the 2023 IEEE 24th Int. Conf. on High Performance Switching and Routing (HPSR), pp. 68–74, DOI: 10.1109/HPSR57248.2023.10147986, 2023.

[75] A. Bentaleb et al., "BoB: Bandwidth Prediction for Real-time Communications Using Heuristic and Reinforcement Learning," IEEE Transactions on Multimedia, vol. 25, pp. 6930–6945, 2023.

[76] S. Emara et al., "Pareto: Fair Congestion Control with Online Reinforcement Learning," IEEE Transactions on Network Science and Engineering, vol. 9, no. 5, pp. 3731–3748, Sept. 2022.

[77] L. Jia et al., "ZiXia: A Reinforcement Learning Approach *via* Adjusted Ranking Reward for Internet Congestion Control," Proc. of the IEEE Int. Conf. on Communications (ICC 2022), pp. 365–370, DOI: 10.1109/ICC45855.2022.9838901, 2022.

[78] A. R. Andrade-Zambrano et al., "A Reinforcement Learning Congestion Control Algorithm for Smart Grid Networks," IEEE Access, vol. 12, pp. 75072–75092, DOI:10.1109/ACCESS.2024.3405334, 2024.

[79] X. Liao et al., "Towards Fair and Efficient Learning-based Congestion Control," arXiv: 2403.01798, DOI: 10.48550/arXiv.2403.01798, Mar. 2024.

[80] I. Rhee et al., "CUBIC for Fast Long-distance Networks," Request for Comments RFC 8312, Internet Engineering Task Force (IETF), DOI: 10.17487/RFC9438, Feb. 2018.

ملخص البحث:

تُمكّن إدارة الازدحام بفاعلية من نقْل البيانات بسرعةٍ وموثوقيةٍ عبر الشّبكات. وتجدر الإشارة إلى أنّ خوارزمية (CUBIC) تعطي نتائج موثوقة في الظّروف الطّبيعية، لكنّها ليست قادرةً على التّكيُّف مع الظّروف المتغيرة للشّبكات.

في هذه الورقة، نقدّم خوارزميةً تعتمد على نهج التّعلُّم التّعزيزي تسمّى (CUBIC-LEARN)، وهي نسخة مطوَّرة من خوارزمية CUBIC الأصلية، لضبط الازدحام في الشّبكات بناءً على البيانات المتعلّقة بمعدّل فقْد الحُزَم، والمرور عبر الشّبكة، وزمن التّأخير في وصول البيانات من مصْدرها إلى غايتها.

وقد أسْفرت المحاكاة عن نتائج موثوقةٍ تُفيد أنّ الخوارزمية المقترحة تتخطّى الخوارزمية الأصلية من حيث مؤشّرات الأداء المشار إليها. فقد خفّضت الخوارزمية المقترحة من معدّل فقْد الحُزم بنسبة 47%، وزادت من درجة استغلال عرْض النّطاق بنسبةٍ تزيد على 50%، مع تقليل حالات إعادة النّقل بنسبةٍ تقترب من 28% وتقليل زمن التّأخير بنسبة 47%. إضافة إلى ذلك، أثبتت الخوارزمية المقترحة وجود تحسُّنٍ في سلوك النُّموّ لنافذة الازدحام والعدالة بين التّدفّقات المتنافسة والاستقرار تحت حركة المرور غير المنتظمة والسّيناريوهات المختلفة للشّبكة.

كذلك أكدّ التّحليل الإحصائي متانة تلك المكاسب دون إضافة تكاليف تتعلّق بالحوسبة. وإجمالاً، فقد تفوقت الخوارزمية المقترحة على العديد من الخوارزميات المشابهة في غالبية مؤشرات قياس الأداء. وتبين هذه النّتائج أنّ التّعلُّم التّعزيزي يمكنه أن يُحسّن كثيراً من التّحكُّم في الازدحام في الشّبكات عالية السّرعة.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 04, December 2025.

SECURE PERFORMANCE ANALYSIS OF SATELLITE-TERRESTRIAL NETWORKS-ASSISTED BACKSCATTER DEVICE

Hong-Nhu Nguyen[1], Si-Phu Le[2], Quang-Sy Vu[3], Quang-Sang Nguyen[4] and Erik Chromy[5]

(Received: 10-Sep.-2025, Revised: 26-Oct.-2025, Accepted: 26-Oct.-2025)

ABSTRACT

This study investigates the secrecy performance of a satellite-backscatter device communication system in the presence of a potential eavesdropper. In the considered setup, a satelite transmits signals to a backscatter device, which reflects the modulated information back to the satellite while being subject to interception by an eavesdropper. To capture the practical wireless environment, the analysis is conducted over correlated Nakagami-m fading channels, where the coupling among the forward, backscatter, and wiretap links is explicitly taken into account. We derive exact closed-form analytical expressions for key secrecy metrics; namely, the secrecy outage probability (SOP), the ergodic secrecy capacity (ESC), and the symbol error rate (SER), which provide comprehensive insights into the secure operation of the system. Furthermore, asymptotic expressions are obtained for the SOP, enabling a deeper understanding of the secrecy diversity order under high signal-to-noise ratio (SNR) regimes. The impacts of critical channel and system parameters, such as fading severity, correlation, user power, and eavesdropper power, on the SOP, ESC, and SER are thoroughly examined. Monte Carlo simulations are also performed to validate the accuracy of the theoretical analysis.

KEYWORDS

Backscatter communication, Secrecy outage probability, Ergodic secrecy capacity, Symbol error rate, Physical layer security.

1. INTRODUCTION

Backscatter communications play an important role in the Internet of Things (IoT) due to their low cost and energy efficiency [1]-[2]. Radio Frequency Identification (RFID) is a typical backscatter communication system, which utilizes backscatter modulation to enable data transmission [3]. In contrast to traditional communication systems, RFID includes both forward and backscatter links. Specifically, in [4]-[5], the authors proposed a dyadic backscatter channel model for RFID systems, incorporating both forward and backscatter links. In practice, these links can be correlated, since the transmit and receive antennas at the reader may be very close or even co-located [6]. The authors in [7] analyzed the range performance of ultrahigh-frequency-band passive RFID for single-input single-output (SISO) and multiple-input multiple-output (MIMO) systems with maximal-ratio combining in a pinhole channel, and modeled it as a correlated Nakagami-m distribution. Similar to other wireless systems, privacy and security are also major concerns for RFID systems. Due to their broadcast nature, RFID systems are vulnerable to potential eavesdropping attacks. In the open literature, several works have proposed lightweight cryptography to address RFID security challenges [8]. Although cryptography can provide improved security performance, it incurs high communication overhead and computational complexity.

Recently, physical layer security (PLS) for backscatter networks has received increasing attention. For instance, Li et al. [10] investigated PLS in wireless-powered ambient backscatter cooperative networks, deriving secrecy rate and outage expressions under practical energy-harvesting and cooperation settings. Similarly, Khan et al. [11] analyzed the secrecy performance of energy-harvesting backscatter systems under various tag-selection strategies, while [12] introduced a deep learning-based secure tag-selection

1. H.-N. Nguyen is with the Faculty of Technology and Engineering, Saigon University, Ho Chi Minh City, Vietnam. Email: nhu.nh@sgu.edu.vn
2. S.-P. Le is with the Faculty of Electrical Engineering and Computer Science, VSB-Technical University of Ostrava, Ostrava, Czech Republic. Email: phu.si.le@vsb.cz
3. Q.-S. Vu (Corresponding Author) is with the Advanced Intelligent Technology Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam. Email: vuquangsy@tdtu.edu.vn
4. Q.-S. Nguyen is with the Posts and Telecommunications Institute of Technology, Ho Chi Minh City, Vietnam. Email: sangnq@ptit.edu.vn
5. E. Chromy is with the Faculty of Informatics, Pan-European University, Bratislava, Slovakia. Email: erik.chromy@paneurouni.com

method to mitigate RIS-induced interference, highlighting the growing role of AI in BackCom security. Moreover, the authors in [15] examined the secrecy performance of backscatter communication networks with multiple readers and different tag-selection schemes, demonstrating that optimized reader-tag pairing can significantly improve secrecy capacity. In addition, [16] further extended the analysis by considering Nakagami-m fading and energy-harvesting capability at the tag, providing secrecy-outage analysis under generalized fading conditions and highlighting that both fading characteristics and tag-selection strategies play critical roles in ensuring physical layer security in BackCom networks. To address the limitations of existing RFID systems in terms of cost, size, and computational complexity, we consider physical layer security (PLS) as an alternative to cryptography. The main idea of PLS is to exploit the wireless-channel characteristics to prevent eavesdroppers from obtaining information from the transmitter. The authors in [17] studied the PLS of a single-reader single-tag RFID system, where the reader not only sends the query signal to power the tag, but also injects artificial noise to interfere with an eavesdropper. Moreover, the authors in [18, 19] investigated secrecy-rate maximization in a MIMO RFID system by jointly optimizing the supply energy and the artificial-noise precoding matrix at the reader. Building upon the idea of combining MIMO and intelligent surfaces for rate and security optimization, Paul et al. [9] developed a quantum gradient descent algorithm for rate maximization in MIMO-NOMA systems assisted by STAR-RIS, achieving superior convergence and energy efficiency compared to classical optimization methods. Recent advances in wireless communication have significantly improved satellite-terrestrial integrated networks, especially regarding cooperative relaying and security under practical impairments. For instance, the authors in [20] investigated the outage behavior of satellite-terrestrial full-duplex relay networks under co-channel interference, revealing robust relaying strategies that enhance end-to-end reliability in hybrid space-ground scenarios. Similarly, energy-harvesting-based spectrum access has emerged as a practical solution for resource-constrained systems. A notable contribution by the authors in [21] proposed incremental cooperation and relay-selection mechanisms while considering hardware noise in terrestrial networks, demonstrating the potential of green communication for spectrum efficiency. In the same direction, full-duplex relaying under imperfect channel state information (CSI) has been explored using time-switching protocols, further highlighting the viability of practical wireless relaying models in terrestrial setups [22]. Security remains a paramount concern in such systems, particularly in cognitive and cooperative environments. Addressing this, the authors in [23] presented secrecy-performance enhancement techniques in underlay cognitive radio networks through multi-hop cooperative transmission, including scenarios with hardware impairments. Additionally, the authors in [24] demonstrated how coverage-area granularity directly affects predictive-modeling accuracy for mobility systems in 5G/6G cellular architectures. Considering practical conditions, the authors in [25] analyzed the performance of SWIPT-aided satellite-terrestrial cooperative networks, while [26] addressed secure communication techniques for multi-tag backscatter systems. These insights can be leveraged to enhance security and efficiency in practical IoT deployments. Furthermore, Nguyen et al. [13] analyzed secure wireless communications incorporating energy harvesting and multi-antenna diversity, demonstrating that efficient energy management and antenna diversity can substantially enhance secrecy capacity. In addition, Garai et al. [14] examined ground-to-satellite free-space optical (FSO) communications under atmospheric turbulence, showing that appropriate modulation selection improves reliability in hybrid space-ground systems. However, the aforementioned works have not derived an analytical expression for the system secrecy outage probability (SOP).

Motivation and Contribution

Building on the foundations of RFID and backscatter communications, a growing body of literature has expanded this paradigm to address interference, fading, resource allocation, and physical layer security in more sophisticated scenarios. For instance, interference and fading have been identified as fundamental bottlenecks, with potential mitigation strategies proposed in [30]. To enhance system capacity and connectivity, NOMA-based backscatter communications have been extensively investigated, ranging from comprehensive surveys of IoT applications [31] to fundamental designs and advanced techniques [32]. Beyond multiple access, integrated designs, such as power-efficient beamforming for joint sensing and communication [33] and fluid antenna systems to overcome fading [34], have been explored. In parallel, the integration of backscatter with intelligent meta-surfaces has received great attention. Joint waveform and reflection optimization for secure RIS-based backscatter

was proposed in [35], while [40] analyzed the performance of RIS-assisted ambient backscatter systems. Short-packet communications, essential for ultra-reliable and low-latency IoT, have also been investigated in RIS-assisted NOMA backscatter systems [38]. In addition, symbiotic paradigms have been studied, including power beacon-assisted energy harvesting [39] and cognitive backscatter-enabled blockchain consensus [37]. Moreover, security has likewise remained an active research topic. For example, UAV-assisted backscatter with jamming was investigated in [36], multi-power beacon IoT systems were examined for secrecy performance in [42], and D2D partial NOMA-assisted backscatter networks were analyzed in [41]. Collectively, these studies highlighted the rapid evolution of backscatter communications across efficiency, reliability, and security dimensions. However, analytical characterization of secrecy-outage probability (SOP), ergodic secrecy capacity (ESC), and symbol error rate (SER) under correlated fading channels remains largely unexplored, which motivates our work. Our work provides a comprehensive security analysis of satellite-assisted backscatter communication, offering novel analytical insights and practical design guidelines. The main contributions are summarized as follows:

- **Novel System Modeling under Correlated Fading:** We establish a first-of-its-kind, comprehensive analytical model for the satellite-backscatter-eavesdropper system under correlated Nakagami-m fading channels, explicitly accounting for the practical coupling effect among the forward, backscatter, and wiretap links.
- **Exact Closed-form Secrecy Metrics:** We successfully derive novel and exact closed-form analytical expressions for the key physical layer security metrics: the Secrecy Outage Probability (SOP), the Ergodic Secrecy Capacity (ESC), and the Symbol Error Rate (SER), comprehensively characterizing the system's vulnerability to eavesdropping.
- **Asymptotic Analysis and Diversity Order:** To offer a deeper understanding of system limits, we obtain asymptotic closed-form expressions for the SOP. This enables the explicit characterization of the secrecy diversity order of the considered backscatter system in the high Signal-to-Noise Ratio (SNR) regime.
- **Validation and Design Insights:** Our theoretical derivations are rigorously validated through extensive Monte Carlo simulations. The numerical results provide valuable insights into the crucial impacts of fading severity, correlation, and system parameters, offering practical guidelines for the design and deployment of secure satellite-assisted backscatter communications.

Table 1. Comparison of the uniqueness of our research to related articles.

Context	PLS	BD	Satellite	Nakagami-m	SOP	ESC	SER
Paper [36]	✓	✓			✓		
Paper [40]		✓		✓			✓
Paper [43]	✓	✓		✓	✓	✓	
Paper [44]	✓	✓			✓	✓	
Paper [45]	✓	✓					
Paper [46]	✓		✓	✓	✓	✓	
Paper [47]	✓	✓			✓		
Paper [48]	✓	✓		✓	✓	✓	
Paper [49]	✓	✓					
Paper [50]	✓	✓			✓		
This paper	✓	✓	✓	✓	✓	✓	✓

The remainder of the paper is organized as follows. Section 2 gives an overview of the system model. Section 3 presents the information-theoretic mathematical framework, to achieve the performance of the system. Section 4 presents numerical results and discussions to validate the developed framework as well as deeply explore the impacts of system key parameters, while Section 5 provides concluding remarks.

"Secure Performance Analysis of Satellite-terrestrial Networks-assisted Backscatter Device", H.-N. Nguyen, S.-P. Le, Q.-S. Vu, Q.-S. Nguyen and E. Chromy.

2. SYSTEM MODEL

In this correspondence, we examine a backscatter communication setup consisting of a satellite (S), a single backscatter device (B), and a potential eavesdropper (E) capable of intercepting the data transmitted by B, as illustrated in Fig. 1. The channels are defined as follows: from the transmitting antenna at S to B (denoted by h), from B to the receiving antenna at S (denoted by f), and from B to the eavesdropper (denoted by g), respectively. This setup highlights the intricate dynamics involved in the communication process within such a backscatter system. Understanding these channel gains is crucial for analyzing the overall performance and security implications of the system in practical scenarios. The received signal at the satellite from the backscatter device can be expressed as:

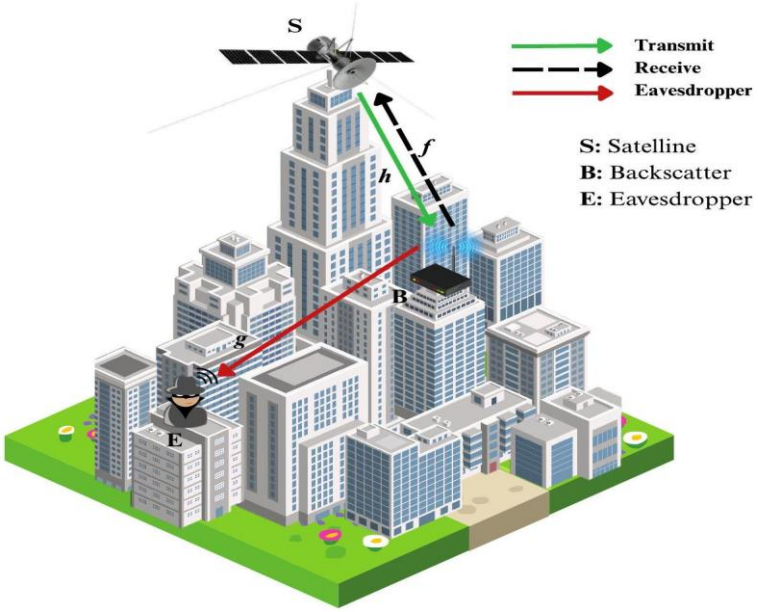


Figure 1. System model.

$$y_{BS} = \sqrt{P_S} hfsb + n_{BS}, \tag{1}$$

where P_S is the normalized transmission powers at the satellite, s is the query signal sent by the satellite, b is the information signal from the backscatter and n_{BS} is the additive white Gaussian noise (AWGN), $n_{BS} \sim \mathcal{CN}(0, \sigma_{BS}^2)$, we set $\mathbb{E}\{|s|^2\} = \mathbb{E}\{|b|^2\} = 1$.

The joint probability density function (PDF) of $|h|^2$ and $|f|^2$ are:

$$\begin{aligned} f_{|h|^2}(x) &= \alpha_1 e^{-\beta_1 x}\, {}_1F_1(m_1; 1; \delta_1 x) \quad , x \geq 0 \\ f_{|f|^2}(x) &= \alpha_2 e^{-\beta_2 x}\, {}_1F_1(m_2; 1; \delta_2 x) \quad , x \geq 0. \end{aligned} \tag{2}$$

For $i \in \{1,2\}$ we set $\alpha_i = \left(2b_i m_i/(2b_i m_i + \Omega_i)\right)^{m_i}/2b_i, \beta_i = 1/2b_i, \delta_i = \Omega_i/(2b_i)(2b_i m_i + \Omega_i)$ with Ω_i and $2b_i$ are the respective average power of the line-of-sight (LoS) and multi-path components, m_i is the fading-severity parameter and ${}_1F_1(\cdot;\cdot;\cdot)$ is the confluent hyper-geometric function of first kind, thereby expressing the probability-density function (PDF) of $f_{|h|^2}(x)$ and $f_{|f|^2}(x)$.

Table 2. Notations of main parameters.

Symbol	Notation	Symbol	Notation
s	Signal at Satelline	Ω_i	The respective average power of the LoS
b	Signal at Backscatter	$2b_i$	The multi-path components
m_i	The fading-severity parameter	$f_X(\cdot)$	Probability-density function (PDF) of X
P_S	Transmit power at S	$F_X(\cdot)$	Cumulative-distribution function (CDF) of X

P_E	Transmit power at E	$\mathbb{E}\{\cdot\}$	Expectation operator
h	Channel gain from U to S	$\lvert\cdot\rvert$	The absolute value of a complex number
f	Channel gain from U to B	$\Gamma(\cdot)$	Gamma function
g	Channel gain from B to S	$\gamma(\cdot,\cdot)$	Lower incomplete Gamma function
ρ_S	The average receive SNR at S	$K_v(\cdot)$	Bessel function of the second kind with v^{th} order
ρ_E	The average receive SNR at E	${}_1F_1(\cdot;\cdot\cdot)$	The confluent hyper-geometric function of first kind
n_{BS}	The additive white Gaussian noise (AWGN)	$W_{\lambda,\mu}(z)$	The Whittaker function
$(.)_{\kappa_i}$	The Pochhammer symbol		

$$f_{|h|^2}(x) = \alpha_1 \sum_{\kappa_1=0}^{m_1-1} \zeta(\kappa_1) x^{\kappa_1} e^{-(\beta_1-\delta_1)x} \quad , x \geq 0 \tag{3a}$$

$$f_{|f|^2}(x) = \alpha_2 \sum_{\kappa_2=0}^{m_2-1} \zeta(\kappa_2) x^{\kappa_2} e^{-(\beta_2-\delta_2)x} \quad , x \geq 0 \tag{3b}$$

where $\zeta(\kappa_i) = (-1)^{\kappa_i}(1-m_i)_{\kappa_i}\delta^{\kappa_i}/(\kappa_i)^2, i \in \{1,2\}$, with $(.)_{\kappa_i}$ is the Pochhammer symbol. The cumulative-distribution function (CDF) of $F_{|h|^2}(x)$ and $F_{|f|^2}(x)$ as:

$$F_{|h|^2}(x) = 1 - \alpha_1 \sum_{\kappa_1=0}^{m_1-1} \sum_{p_1=0}^{\kappa_1} \frac{\zeta(\kappa_1)\Gamma(\kappa_1+1)}{\Gamma(p_1+1)(\beta_1-\delta_1)^{\kappa_1+1-p_1}} \times x^{p_1} e^{-(\beta_1-\delta_1)x} \tag{4a}$$

$$F_{|f|^2}(x) = 1 - \alpha_2 \sum_{\kappa_2=0}^{m_2-1} \sum_{p_2=0}^{\kappa_2} \frac{\zeta(\kappa_2)\Gamma(\kappa_2+1)}{\Gamma(p_2+1)(\beta_2-\delta_2)^{\kappa_2+1-p_2}} \times x^{p_2} e^{-(\beta_2-\delta_2)x} \tag{4b}$$

where $\Gamma(x) = (x-1)!$ is the Gamma function for a positive integral x.

On the other hand, at the E side, the intercepted signal from the backscatter can be written as:

$$y_E = \sqrt{P_E} hgsb + n_E, \tag{5}$$

where P_E is the transmit power of the E, n_E is the AWGN at E, $n_E \sim \mathcal{CN}(0, \sigma_E^2)$, and we assume that h and g are independent from each other, since the E is far from reader, compared to the distance with the B.

Since we consider Nakagami-m faded channels for the terrestrial links of the considered network, the channel gains $|g|^2$, are assumed to follow the Gamma distribution with average power Ω_3 and fading severity m_3. Hence, The PDF and CDF of $f_{|g|^2}(x)$ can be given as:

$$f_{|g|^2}(x) = \frac{\mu^{m_3}}{\Gamma(m_3)} x^{m_3-1} e^{-\mu x}, \tag{6}$$

and

$$F_{|g|^2}(x) = 1 - \frac{\Gamma(m_3, \mu x)}{\Gamma(m_3)} = 1 - e^{-\mu x} \sum_{p_3=0}^{m_3-1} \frac{\mu^{p_3} x^{p_3}}{\Gamma(p_3+1)}, \tag{7}$$

where $\mu = \frac{m_3}{\Omega_3}$.

From (1) and (5), we can derive the received instantaneous signal-to-noise ratios (SNRs) at the reader and E as:

$$\gamma_{BS} = |h|^2 |f|^2 \rho_S, \tag{8}$$

and

"Secure Performance Analysis of Satellite-terrestrial Networks-assisted Backscatter Device", H.-N. Nguyen, S.-P. Le, Q.-S. Vu, Q.-S. Nguyen and E. Chromy.

$$\gamma_E = |h|^2|g|^2\rho_E, \tag{9}$$

where $\rho_S = P_S/\sigma_{sb}^2$ and $\rho_E = P_E/\sigma_E^2$ are the average receive SNR at reader and E, respectively.

Correspondingly, from (8) and (9), the capacity of the reader and eavesdropper's channels can be written as:

$$C_{BS} = \log_2(1+\gamma_{BS}), \tag{10}$$

and

$$C_E = \log_2(1+\gamma_E), \tag{11}$$

respectively. Using (10) and (11), we can express the instantaneous secrecy capacity of RFID (Radio frequency identification) backscatter system as:

$$C_S = [C_{BS} - C_E]^+, \tag{12}$$

where $[x]^+$is defined as $[x]^+ = \max(x, 0)$.

3. PERFORMANCE ANALYSIS

3.1 Security Outage Probability (SOP)

If we denote $R \geq 0$ as the target secrecy rate of S, the secrecy-outage probability of S can be expressed as:

$$\begin{aligned} P_{\text{out}} &= \Pr[C_S < R] = \Pr\left[\log_2\left(\frac{1+\gamma_{BS}}{1+\gamma_E}\right) < R\right] = \Pr\left[\frac{1+|h|^2|f|^2\rho_S}{1+|h|^2|g|^2\rho_E} < \gamma_{\text{th}}\right] \\ &= \Pr\left[|f|^2 < \frac{v}{|h|^2} + |g|^2\omega_E\right], \end{aligned} \tag{13}$$

where $\gamma_{th} = 2^R$ is the secrecy SNR threshold, $v = (\gamma_{th}-1)/\rho_S$ and $\omega_E = \frac{\gamma_{th}\rho_E}{\rho_S}$. Applying formulae (3a), (4b) and (6), P_{out} is calculated as:

$$P_{\text{out}} = \Pr[|f|^2 < \frac{v}{|h|^2} + |g|^2\omega_E] = \int_0^\infty f_{|h|^2}(x)\int_0^\infty f_{|g|^2}(y)\left[F_{|f|^2}\left(\frac{v}{x} + y\omega_E\right)\right]dxdy$$

$$\begin{aligned} &= \sum_{\kappa_1=0}^{m_1-1}\frac{\mu^{m_3}\alpha_1\zeta(\kappa_1)}{\Gamma(m_3)}\Big[\underbrace{\int_0^\infty x^{\kappa_1}e^{-(\beta_1-\delta_1)x}dx}_{\Phi_1} \times \int_0^\infty y^{m_3-1}e^{-\mu y}dy - \alpha_2\sum_{\kappa_2=0}^{m_2-1}\sum_{p_2=0}^{\kappa_2}\sum_{t=0}^{p_2}\binom{p_2}{t} \\ &\times\frac{\omega_E^{p_2-t}v^t\zeta(\kappa_2)\Gamma(\kappa_2+1)}{\Gamma(p_2+1)(\beta_2-\delta_2)^{\kappa_2-p_2+1}} \times \underbrace{\int_0^\infty x^{\kappa_1-t}\,e^{-\frac{(\beta_2-\delta_2)v}{x}-(\beta_1-\delta_1)x}dx}_{\Phi_3} \\ &\times \underbrace{\int_0^\infty y^{m_3+p_2-t-1}e^{-y(\mu+(\beta_2-\delta_2)\omega_E)}dy}_{\Phi_2}\Big] \end{aligned} \tag{14}$$

With the help of [27, Eq. (3.351.3)], Φ_1 and Φ_2 are given by:

$$\Phi_1 = \int_0^\infty x^{\kappa_1}e^{-(\beta_1-\delta_1)x}dx \times \int_0^\infty y^{m_3-1}e^{-\mu y}dy = (\kappa_1)!\,(\beta_1-\delta_1)^{-\kappa_1-1}(m_3-1)!\,\mu^{-m_3}$$

$$\Phi_2 = \int_0^\infty y^{m_3+p_2-t-1}e^{-y(\mu+(\beta_2-\delta_2)\omega_E)}dy = (m_3+p_2-t-1)!\,(\mu+(\beta_2-\delta_2)\omega_E)^{-m_3-p_2+t} \tag{15}$$

With the help of [27, Eq. (3.471.9)], Φ_3 is given by:

$$\Phi_3 = \int_0^\infty x^{\kappa_1-t}e^{-\frac{(\beta_2-\delta_2)v}{x}-(\beta_1-\delta_1)x}dx = 2\left(\frac{(\beta_2-\delta_2)v}{(\beta_1-\delta_1)}\right)^{\frac{\kappa_1-t+1}{2}}K_{\kappa_1-t+1}\left(2\sqrt{(\beta_1-\delta_1)(\beta_2-\delta_2)v}\right) \tag{16}$$

By substituting (15) and (16) into (14), we obtain:

$$\begin{aligned} P_{\text{out}} &= \sum_{\kappa_1=0}^{m_1-1} \frac{\mu^{m_3}\alpha_1\zeta(\kappa_1)}{\Gamma(m_3)}(\kappa_1)!\,(\beta_1-\delta_1)^{-\kappa_1-1}(m_3-1)!\,\mu^{-m_3} \\ &-\alpha_2\sum_{\kappa_2=0}^{m_2-1}\sum_{p_2=0}^{\kappa_2}\sum_{t=0}^{p_2}\binom{p_2}{t}\frac{2\omega_E^{p_2-t}v^t\zeta(\kappa_2)\Gamma(\kappa_2+1)}{\Gamma(p_2+1)(\beta_2-\delta_2)^{\kappa_2-p_2+1}}\left(\frac{(\beta_2-\delta_2)v}{(\beta_1-\delta_1)}\right)^{\frac{\kappa_1-t+1}{2}} \\ &\times K_{\kappa_1-t+1}\left(2\sqrt{(\beta_1-\delta_1)(\beta_2-\delta_2)v}\right)(m_3+p_2-t-1)!\,(\mu+(\beta_2-\delta_2)\omega_E)^{-m_3-p_2+t} \end{aligned} \tag{17}$$

where $K_v(.)$ is the first order modified Bessel function of the second kind.

3.2 Ergodic Secrecy Capacity (ESC) Analysis

In this section, we analyze the system's ESC performance, which is defined as the average of the highest attainable secrecy capacity. To evaluate the ESC, we define the ergodic capacity of the main channel and the wiretap channel as follows.

$$\mathbb{E}\{C_S\} = [\mathbb{E}\{C_{BS}\} - \mathbb{E}\{C_E\}]^+. \tag{18}$$

To obtain the desired outcome, the formulae for the ergodic capacity of the primary and wiretap channels must be considered. (i.e., C_{BS} and C_E). As a result, we simplify C_{BS} using SNR from (8) into (10) and C_E using SNR from (9) into (11) as:

$$\begin{aligned} \mathbb{E}\{C_{BS}\} &= \mathbb{E}\{\log_2(1+\underbrace{\frac{|h|^2|f|^2}{\rho_S}}_{X})\} = \frac{1}{\ln 2}\int_0^\infty \frac{1-F_X(x)}{1+x}dx \\ \mathbb{E}\{C_E\} &= \mathbb{E}\{\log_2(1+\underbrace{\frac{|h|^2|g|^2}{\rho_E}}_{Y})\} = \frac{1}{\ln 2}\int_0^\infty \frac{1-F_Y(y)}{1+y}dy\,. \end{aligned} \tag{19}$$

where $F_X(x)$ with $X = |h|^2|f|^2\rho_S$ and $F_Y(y)$ with $Y = |h|^2|g|^2\rho_E$. Based on (3) and (4), $F_X(x)$ is calculated as:

$$\begin{aligned} F_X(x) &= F_{|h|^2|f|^2\rho_S}(x) = \Pr(|h|^2 < \frac{x}{|f|^2\rho_S}) = \int_0^{+\infty} F_{|h|^2}(\frac{x}{y\rho_S})f_{|f|^2}(y)dy \\ &= \int_0^{+\infty}(1-\alpha_1\sum_{\kappa_1=0}^{m_1-1}\sum_{p_1=0}^{\kappa_1}\frac{\zeta(\kappa_1)\Gamma(\kappa_1+1)}{\Gamma(p_1+1)(\beta_1-\delta_1)^{\kappa_1+1-p_1}}\times(\frac{x}{y\rho_S})^{p_1}e^{-\frac{(\beta_1-\delta_1)x}{y\rho_S}})\times f_{|f|^2}(y)dy \\ &= \int_0^{+\infty} f_{|f|^2}(y)dy - \int_0^{+\infty}\alpha_1\sum_{\kappa_1=0}^{m_1-1}\sum_{p_1=0}^{\kappa_1}\frac{\zeta(\kappa_1)\Gamma(\kappa_1+1)}{\Gamma(p_1+1)(\beta_1-\delta_1)^{\kappa_1+1-p_1}}\times(\frac{x}{y\rho_S})^{p_1}e^{-\frac{(\beta_1-\delta_1)x}{y\rho_S}}\times f_{|f|^2}(y)dy \\ &= 1-\alpha_1\alpha_2\sum_{\kappa_2=0}^{m_2-1}\sum_{\kappa_1=0}^{m_1-1}\sum_{p_1=0}^{\kappa_1}\frac{\zeta(\kappa_1)\zeta(\kappa_2)\Gamma(\kappa_1+1)(\frac{x}{\rho_S})^{p_1}}{\Gamma(p_1+1)(\beta_1-\delta_1)^{\kappa_1+1-p_1}}\underbrace{\int_0^{+\infty} y^{\kappa_2-p_1}e^{-\frac{(\beta_1-\delta_1)x}{y\rho_S}-(\beta_2-\delta_2)y}dy}_{\Xi_1} \end{aligned} \tag{20}$$

By using [27, Eq. (3.471.9)] for Ξ_1, the CDF$F_X(x)$ can be obtained:

$$\begin{aligned} F_X(x) &= 1-2\sum_{\kappa_2=0}^{m_2-1}\sum_{\kappa_1=0}^{m_1-1}\sum_{p_1=0}^{\kappa_1}\frac{\alpha_1\alpha_2\zeta(\kappa_1)\zeta(\kappa_2)}{\rho_S^{(\frac{\kappa_2+p_1+1}{2})}\Gamma(p_1+1)(\beta_2-\delta_2)^{\frac{\kappa_2-p_1+1}{2}}} \\ &\times\frac{\Gamma(\kappa_1+1)(\sqrt{x})^{\kappa_2+p_1+1}}{(\beta_1-\delta_1)^{\frac{2\kappa_1-p_1-\kappa_2+1}{2}}}K_{\kappa_2-p_1+1}\left(2\sqrt{\frac{(\beta_1-\delta_1)(\beta_2-\delta_2)x}{\rho_S}}\right) \end{aligned} \tag{21}$$

Substituting (21) into (19), we obtain:

$$\begin{aligned} \mathbb{E}\{C_{BS}\} &= 2\sum_{\kappa_2=0}^{m_2-1}\sum_{\kappa_1=0}^{m_1-1}\sum_{p_1=0}^{\kappa_1}\frac{\alpha_1\alpha_2\zeta(\kappa_1)\zeta(\kappa_2)\Gamma(\kappa_1+1)}{\ln 2\,\rho_S^{(\kappa_2+p_1+1)/2}\Gamma(p_1+1)} \\ &\times\frac{(\beta_1-\delta_1)^{(\kappa_2+p_1-2\kappa_1-1)/2}}{(\beta_2-\delta_2)^{(\kappa_2-p_1+1)/2}}\int_0^\infty\frac{(\sqrt{x})^{\kappa_2+p_1+1}}{1+x}\times K_{\kappa_2-p_1+1}\left(2\sqrt{\frac{(\beta_1-\delta_1)(\beta_2-\delta_2)x}{\rho_S}}\right)dx \end{aligned} \tag{22}$$

Let $t = 4/\pi\arctan(x) - 1 \Rightarrow \tan(\pi(t+1)/4) = x \Rightarrow (\pi/4)\sec^2(\pi(t+1)/4)dt = dx$, so $\mathbb{E}\{C_{BS}\}$ is given by:

$$\begin{aligned}\mathbb{E}\{C_{BS}\} &= \sum_{\kappa_2=0}^{m_2-1} \sum_{\kappa_1=0}^{m_1-1} \sum_{p_1=0}^{\kappa_1} \frac{\pi\alpha_1\alpha_2\zeta(\kappa_1)\zeta(\kappa_2)}{2\ln 2\rho_S^{(\kappa_2+p_1+1)/2}\Gamma(p_1+1)} \\ &\times \frac{(\beta_1-\delta_1)^{(\kappa_2+p_1-2\kappa_1-1)/2}}{(\beta_2-\delta_2)^{(\kappa_2-p_1+1)/2}} \int_{-1}^{1} \sec^2(\pi(t+1)/4) \\ &\times \frac{\Gamma(\kappa_1+1)(\sqrt{\tan(\pi(t+1)/4)})^{\kappa_2+p_1+1}}{[1+\tan(\pi(t+1)/4)]} \\ &\times K_{\kappa_2-p_1+1}\left(2\sqrt{\frac{\tan(\pi(t+1)/4)}{(\beta_1-\delta_1)^{-1}(\beta_2-\delta_2)^{-1}\rho_S}}\right)dt\end{aligned} \tag{23}$$

Though it is difficult to derive a closed-form expression for (23), we can obtain an accurate approximation for it. Using Gaussian-Chebyshev quadrature [27, Eq. 25.4.38], we have:

$$\begin{aligned}\mathbb{E}\{C_{BS}\} &\approx \sum_{\kappa_2=0}^{m_2-1} \sum_{\kappa_1=0}^{m_1-1} \sum_{p_1=0}^{\kappa_1} \sum_{n=1}^{N} \frac{\pi^2\sqrt{1-\xi_n^2}\alpha_1\alpha_2\zeta(\kappa_1)\zeta(\kappa_2)}{2N\ln 2\rho_S^{(\kappa_2+p_1+1)/2}} \\ &\times \frac{\sec^2(\pi(\xi_n^2+1)/4)(\beta_1-\delta_1)^{(\kappa_2+p_1-2\kappa_1-1)/2}}{(\beta_2-\delta_2)^{(\kappa_2-p_1+1)/2}} \times \frac{\Gamma(\kappa_1+1)(\sqrt{\tan(\pi(\xi_n^2+1)/4)})^{\kappa_2+p_1+1}}{\Gamma(p_1+1)[1+\tan(\pi(\xi_n^2+1)/4)]} \\ &\times K_{\kappa_2-p_1+1}\left(2\sqrt{\frac{\tan(\pi(\xi_n^2+1)/4)}{(\beta_1-\delta_1)^{-1}(\beta_2-\delta_2)^{-1}\rho_S}}\right)\end{aligned} \tag{24}$$

where $\xi_n = \cos\left(\frac{2m-1}{2M}\pi\right)$ and $\sec^2(x) = 1/\cos^2(x)$.

Next, $\mathbb{E}\{C_E\}$ is calculated as:

$$\mathbb{E}\{C_E\} = \frac{1}{\ln 2}\int_0^{\infty} \frac{1-F_Y(x)}{1+x}dx \tag{25}$$

Based on (4) and (6), $F_Y(y)$ is calculated as:

$$\begin{aligned}F_Y(y) &= F_{|h|^2|g|^2\rho_S}(y) = \Pr(|h|^2 < \tfrac{y}{|g|^2\rho_S}) = \int_0^{+\infty} F_{|h|^2}(\tfrac{x}{z\rho_S}) f_{|g|^2}(z)dz \\ &= \int_0^{+\infty} (1-\alpha_1 \sum_{\kappa_1=0}^{m_1-1} \sum_{p_1=0}^{\kappa_1} \frac{\zeta(\kappa_1)\Gamma(\kappa_1+1)}{\Gamma(p_1+1)(\beta_1-\delta_1)^{\kappa_1+1-p_1}} \times (\tfrac{y}{z\rho_E})^{p_1} e^{-\frac{(\beta_1-\delta_1)y}{z\rho_E}}) \times f_{|g|^2}(z)dz \\ &= \int_0^{+\infty} f_{|g|^2}(z)dz - \int_0^{+\infty} \alpha_1 \sum_{\kappa_1=0}^{m_1-1} \sum_{p_1=0}^{\kappa_1} \frac{\zeta(\kappa_1)\Gamma(\kappa_1+1)}{\Gamma(p_1+1)(\beta_1-\delta_1)^{\kappa_1+1-p_1}} \times (\tfrac{y}{z\rho_E})^{p_1} e^{-\frac{(\beta_1-\delta_1)y}{z\rho_E}} \times f_{|g|^2}(z)dz \\ &= 1-\alpha_1 \sum_{\kappa_1=0}^{m_1-1} \sum_{p_1=0}^{\kappa_1} \frac{\mu^{m_3}\zeta(\kappa_1)\Gamma(\kappa_1+1)}{\Gamma(m_3)\Gamma(p_1+1)(\beta_1-\delta_1)^{\kappa_1+1-p_1}} (\tfrac{y}{\rho_E})^{p_1} \underbrace{\int_0^{+\infty} z^{m_3-p_1-1} e^{-\frac{(\beta_1-\delta_1)y}{z\rho_E}-\mu z}dz}_{\Xi_1}\end{aligned} \tag{26}$$

By using [27, Eq. (3.471.9)] for Ξ_2, the CDF$F_Y(y)$ can be obtained:

$$F_Y(y) = 1 - \sum_{\kappa_1=0}^{m_1-1} \sum_{p_1=0}^{\kappa_1} \frac{2\alpha_1\mu^{\frac{m_3+p_1}{2}}\zeta(\kappa_1)\Gamma(\kappa_1+1)y^{\frac{m_3+p_1}{2}}}{\Gamma(m_3)\Gamma(p_1+1)(\beta_1-\delta_1)^{\frac{2\kappa_1-p_1-m_3+2}{2}}\rho_E^{\frac{m_3+p_1}{2}}} K_{m_3-p_1}\left(2\sqrt{\frac{\mu(\beta_1-\delta_1)y}{\rho_E}}\right) \tag{27}$$

Substituting (27) into (25), we can obtain:

$$\mathbb{E}\{C_E\} = \frac{\sum_{\kappa_1=0}^{m_1-1}\sum_{p_1=0}^{\kappa_1} \frac{2\alpha_1\mu^{\frac{m_3+p_1}{2}}\zeta(\kappa_1)\Gamma(\kappa_1+1)}{\Gamma(m_3)\Gamma(p_1+1)(\beta_1-\delta_1)^{\frac{2\kappa_1-p_1-m_3+2}{2}}\rho_E^{\frac{m_3+p_1}{2}}}}{\ln 2} \int_0^{\infty} \frac{x^{\frac{m_3+p_1}{2}}}{1+x} K_{m_3-p_1}\left(2\sqrt{\frac{\mu(\beta_1-\delta_1)x}{\rho_E}}\right)dx. \tag{28}$$

Let $u = 4/\pi\arctan(x) - 1 \Rightarrow \tan(\pi(u+1)/4) = x \Rightarrow (\pi/4)\sec^2(\pi(u+1)/4)du = dx$, so $\mathbb{E}\{C_E\}$ is given by:

$$\begin{aligned}\mathbb{E}\{C_E\} &= \frac{(\frac{\pi}{4})\sum_{\kappa_1=0}^{m_1-1} \sum_{p_1=0}^{\kappa_1} \frac{2\alpha_1\mu^{\frac{m_3+p_1}{2}}\zeta(\kappa_1)\Gamma(\kappa_1+1)}{\Gamma(m_3)\Gamma(p_1+1)(\beta_1-\delta_1)^{\frac{2\kappa_1-p_1-m_3+2}{2}}\rho_E^{\frac{m_3+p_1}{2}}}}{\ln 2} \\ &\times \int_{-1}^{+1} \frac{\tan(\frac{\pi(u+1)}{4})^{\frac{m_3+p_1}{2}}}{1+\tan(\frac{\pi(u+1)}{4})} K_{m_3-p_1}\left(2\sqrt{\frac{\mu(\beta_1-\delta_1)\tan(\frac{\pi(u+1)}{4})}{\rho_E}}\right)\sec^2(\tfrac{\pi(u+1)}{4})du\end{aligned} \tag{29}$$

Although it is difficult to derive a closed-form expression for (29), an accurate approximation can be obtained by using the Gaussian-Chebyshev quadrature [27, Eq. (25.4.38)]. Accordingly, we have:

$$\mathbb{E}\{C_E\} \approx \sum_{m=1}^{M} \sum_{\kappa_1=0}^{m_1-1} \sum_{p_1=0}^{\kappa_1} \frac{\sec^2((\xi_m+1)\frac{\pi}{4})\pi^2\sqrt{1-\xi_m^2}\alpha_1\mu^{\frac{m_3+p_1}{2}}\zeta(\kappa_1)}{2M\ln 2\Gamma(m_3)\Gamma(p_1+1)(\beta_1-\delta_1)^{\frac{2\kappa_1-p_1-m_3+2}{2}}\rho_E^{\frac{m_3+p_1}{2}}} \frac{\Gamma(\kappa_1+1)\tan((\xi_m+1)\frac{\pi}{4})^{\frac{m_3+p_1}{2}}}{1+\tan((\xi_m+1)\frac{\pi}{4})} K_{m_3-p_1}\left(2\sqrt{\frac{\mu(\beta_1-\delta_1)\tan((\xi_m+1)\frac{\pi}{4})}{\rho_E}}\right) \tag{30}$$

where $\xi_m = \cos\left(\frac{2m-1}{2M}\pi\right)$.

By combining (30) and (24) into (18), the approximate expression for the ergodic achievable secrecy rate of the satellite can be expressed as:

$$\begin{aligned}\mathbb{E}\{C_S\} \approx [\sum_{\kappa_2=0}^{m_2-1}\sum_{\kappa_1=0}^{m_1-1}\sum_{p_1=0}^{\kappa_1}\sum_{n=1}^{N}\sum_{m=1}^{M} & \frac{\overline{\pi^2\sqrt{1}-\xi_n^2\alpha_1\alpha_2\zeta(\kappa_1)\zeta(\kappa_2)\sec^2\left(\frac{\pi(\xi_n^2+1)}{4}\right)(\beta_1-\delta_1)^{\frac{\kappa_2+p_1-2\kappa_1-1}{2}}}}{4N\ln 2\Gamma(p_1+1)\rho_S^{\frac{\kappa_2+p_1+1}{2}}(\beta_2-\delta_2)^{\frac{\kappa_2-p_1+1}{2}}} \\ & \times\frac{\Gamma(\kappa_1+1)\pi^2\sqrt{1-\xi_m^2}\alpha_1\zeta(\kappa_1)\Gamma(\kappa_1+1)}{[1+\tan(\pi(\xi_n^2+1)\pi(\xi_n^2+1)4)]\,\mathrm{Mln}\,2\Gamma(m_3)\,\Gamma(p_1+1)\rho_E^{(p_1+m_3)(p_1+m_3)}} \\ \times\frac{\sec^2\left(\frac{\pi(\xi_m+1)}{4}\right)(\beta_1-\delta_1)^{\frac{p_1+m_3-2\kappa_1-2}{2}}\mu^{\frac{m_3+p_1}{2}}\left(\sqrt{\tan(\pi(\xi_m+1))}\right)^{p_1+m_3}}{(\sqrt{\tan(\pi(\xi_n^2+1)\pi(\xi_n^2+1)4)})^{-\kappa_2-p_1-1}} & [1+\tan(\pi(\xi_m+1)\pi(\xi_m+1)4)] \\ \times K_{\kappa_2-p_1+1}\left(2\sqrt{\frac{\tan\left(\frac{\pi(\xi_n^2+1)}{4}\right)}{(\beta_1-\delta_1)^{-1}(\beta_2-\delta_2)^{-1}\rho_S}}\right)K_{m_3-p_1}\left(2\sqrt{\frac{(\beta_1-\delta_1)\mu\tan\left(\frac{\pi(\xi_m+1)}{4}\right)}{\rho_E}}\right)]. & \end{aligned} \tag{31}$$

3.3 Symbol Error Rate (SER)

For the wireless system, the SER is calculated as:

$$SER = \bar{a}\mathbb{E}\{Q(\sqrt{\bar{b}\gamma})\} = \frac{\bar{a}}{\sqrt{2\pi}}\int_0^{\infty} F\left(\frac{t^2}{\bar{b}}\right)e^{-\frac{t^2}{2}}dt \tag{32}$$

where $\bar{a}$ and $\bar{b}$ are constants and depend on the modulation types, e.g. $\bar{a}=1, \bar{b}=2$ for the binary phase-shift keying (BPSK) modulation and $\bar{a}=2, \bar{b}=1$ for quadrature phase shift keying (QPSK) and 4-quadrature amplitude modulation (4-QAM); $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^{\infty} e^{-t^2/2}dt$ is the Gaussian function; γ is the end-to-end SINR of the considered system; $F(x)$ is the CDF of the end-to-end SINR. From the definition of CDF, we can replace $F(x)$ by P_{out} of the system.

From (32), applying the change of variable technique $x = t^2/b$, we can rewrite the SER as [28]:

$$\begin{aligned} SER &= \frac{\bar{a}\sqrt{\bar{b}}}{2\sqrt{2\pi}}\int_0^{\infty}\frac{e^{-bv/2}}{\sqrt{v}}[F_X(x)-F_Y(y)]dv \\ &= \frac{\bar{a}\sqrt{\bar{b}}}{2\sqrt{2\pi}}\left[\int_0^{\infty}\frac{e^{-\frac{bv}{2}}F_X(v)}{\sqrt{v}}dv - \int_0^{\infty}\frac{e^{-\frac{bv}{2}}F_Y(v)}{\sqrt{v}}dv\right]. \end{aligned} \tag{33}$$

By substituting (20) and (27) into (33), we obtain:

$$\begin{aligned} SER = \frac{\bar{a}\sqrt{\bar{b}}}{2\sqrt{2\pi}} \Bigg[& \sum_{\kappa_1=0}^{m_1-1}\sum_{p_1=0}^{\kappa_1}\frac{2\alpha_1\mu^{\frac{m_3+p_1}{2}}\zeta(\kappa_1)\Gamma(\kappa_1+1)}{\Gamma(m_3)\Gamma(p_1+1)(\beta_1-\delta_1)^{\frac{2\kappa_1-p_1-m_3+2}{2}}\rho_E^{\frac{m_3+p_1}{2}}} \\ & \times \underbrace{\int_0^{\infty} e^{\frac{-bv}{2}}v^{\frac{m_3+p_1-1}{2}}K_{m_3-p_1}\left(2\sqrt{\frac{\mu(\beta_1-\delta_1)v}{\rho_E}}\right)dv}_{\Xi_3} \\ & -2\sum_{\kappa_2=0}^{m_2-1}\sum_{\kappa_1=0}^{m_1-1}\sum_{\kappa_1=0}^{\kappa_1}\frac{\alpha_1\alpha_2\zeta(\kappa_1)\zeta(\kappa_2)\Gamma(\kappa_1+1)}{\rho_S^{\left(\frac{\kappa_2+p_1+1}{2}\right)}\Gamma(p_1+1)(\beta_2-\delta_2)^{\frac{\kappa_2-p_1+1}{2}}}(\beta_1-\delta_1)^{\frac{2\kappa_1-p_1-\kappa_2+1}{2}} \\ & \times \underbrace{\int_0^{\infty} e^{\frac{-bv}{2}}v^{\frac{\kappa_2+p_1}{2}}K_{\kappa_2-p_1+1}\left(2\sqrt{\frac{(\beta_1-\delta_1)(\beta_2-\delta_2)v}{\rho_S}}\right)dv}_{\Xi_4} \Bigg] \end{aligned} \tag{34}$$

"Secure Performance Analysis of Satellite-terrestrial Networks-assisted Backscatter Device", H.-N. Nguyen, S.-P. Le, Q.-S. Vu, Q.-S. Nguyen and E. Chromy.

With the help of [29, Eq. (6.643.3)], Ξ_3 and Ξ_4 are calculated as:

$$\begin{aligned}\Xi_3 &= \int_0^\infty v^{\frac{m_3+p_1-1}{2}} e^{\frac{-bv}{2}} K_{m_3-p_1}\left(2\sqrt{\frac{\mu(\beta_1-\delta_1)v}{\rho_E}}\right)dv \\ &= \frac{\Gamma(m_3+\frac{1}{2})\Gamma(p_1+\frac{1}{2})}{2\sqrt{\frac{\mu(\beta_1-\delta_1)}{\rho_E}}} e^{\left(\frac{\mu(\beta_1-\delta_1)}{b\rho_E}\right)} \left(\frac{b}{2}\right)^{\frac{-m_3-p_1}{2}} W_{\frac{-m_3-p_1}{2},\frac{m_3-p_1}{2}}\left(\frac{2\mu(\beta_1-\delta_1)}{b\rho_E}\right) \\ \Xi_4 &= \int_0^\infty v^{\frac{\kappa_2+p_1}{2}} e^{\frac{-bv}{2}} K_{\kappa_2-p_1+1}\left(2\sqrt{\frac{(\beta_1-\delta_1)(\beta_2-\delta_2)v}{\rho_S}}\right)dv \\ &= \frac{\Gamma\left(\kappa_2+\frac{3}{2}\right)\Gamma\left(p_1+\frac{1}{2}\right)}{2\sqrt{\frac{(\beta_1-\delta_1)(\beta_2-\delta_2)}{\rho_S}}} e^{\left(\frac{(\beta_1-\delta_1)(\beta_2-\delta_2)}{b\rho_S}\right)} \left(\frac{b}{2}\right)^{\frac{-\kappa_2-p_1-1}{2}} W_{\frac{-\kappa_2-p_1-1}{2},\frac{\kappa_2-p_1+1}{2}}\left(\frac{2(\beta_1-\delta_1)(\beta_2-\delta_2)}{b\rho_S}\right).\end{aligned} \tag{35}$$

By substituting Ξ_3 and Ξ_4 into (34), we have:

$$\begin{aligned}SER = \frac{\bar{a}\sqrt{\bar{b}}}{2\sqrt{2\pi}}\Big[\sum_{\kappa_1=0}^{m_1-1} \sum_{p_1=0}^{\kappa_1} &\frac{2^{(p_1+m_3)/2}\mu^{(m_3+p_1-1)/2}\alpha_1\zeta(\kappa_1)\Gamma(\kappa_1+1)\Gamma((2m_3+1)/2)\Gamma((2p_1+1)/2)}{\Gamma(m_3)\Gamma(p_1+1)\bar{b}^{(p_1+m_3)/2}\rho_E^{(p_1+m_3-1)/2}(\beta_1-\delta_1)^{(2\kappa_1-p_1-m_3+3)/2}} \\ \times e^{\frac{(\beta_1-\delta_1)\mu}{\bar{b}\rho_E}} W_{-\frac{p_1+m_3}{2},\frac{m_3-p_1}{2}}((2(\beta_1-\delta_1)\mu)/\bar{b}\rho_E) - \sum_{\kappa_2=0}^{m_2-1} \sum_{\kappa_1=0}^{m_1-1} \sum_{p_1=0}^{\kappa_1} &\frac{2^{(\kappa_2+p_1+1)/2}\alpha_1\alpha_2\zeta(\kappa_1)\zeta(\kappa_2)\Gamma(\kappa_1+1)}{\rho_S^{(\kappa_2+p_1)/2}\Gamma(p_1+1)(\beta_2-\delta_2)^{(\kappa_2-p_1+2)/2}} \\ \times \frac{\Gamma((2\kappa_2+3)/2)\Gamma((2p_1+1)/2)}{\bar{b}^{(\kappa_2+p_1+1)/2}(\beta_1-\delta_1)^{(2\kappa_1-p_1-\kappa_2+2)/2}} e^{\frac{(\beta_1-\delta_1)(\beta_2-\delta_2)}{\bar{b}\rho_S}} &W_{-\frac{\kappa_2+p_1+1}{2},\frac{\kappa_2-p_1+1}{2}}((2(\beta_1-\delta_1)(\beta_2-\delta_2))/2)\Big].\end{aligned} \tag{36}$$

4. NUMERICAL RESULTS AND SIMULATIONS

This section employs Monte Carlo simulations to validate the proposed mathematical frameworks and provide insights under various conditions. Simulation parameters are detailed in Table 3.

Table 3. Simulation parameters.

Symbol	Parameter name	Value
ρ_S	Transmit power to noise ratio at S	0 to 40 (dB)
ρ_E	Transmit power to noise ratio at E	0 to 20(dB)
R	Target rate	1.5 to 3bps/Hz
N	The Gauss-Chevbyshev parameter	100
M	The Gauss-Chevbyshev parameter	100
m_i	The fading-severity parameter	5
Ω_i	The respective average power of the LoS	0.279
b_i	The multi-path components	0.251

Figure 2 clearly illustrates the relationship between the Secrecy Outage Probability (SOP) and the average Satellite Signal-to-Noise Ratio (ρ_S in dB), while validating the analytical results against simulations and comparing performance with a reference scheme. The close proximity of the analysis (solid lines) and simulation (data points) results confirms the accuracy of the derived closed-form analytical expressions for SOP. As ρ_S increases, the SOP decreases significantly, a trend directly attributable to the satellite allocating higher transmit power, which in turn enhances the instantaneous received SNR and overall system security. Conversely, the SOP rises as the target Secrecy Rate Threshold (R) increases (from 1.5 to 3), reflecting the more stringent requirement for secure communication. Furthermore, the proposed system demonstrates superior performance by achieving a lower SOP compared to the "Backscatter Random Selection" scheme [5], highlighting the effectiveness of the current model in the correlated Nakagami-m fading environment.

Figure 3 investigates the effect of the average Eavesdropper Signal-to-Noise Ratio (ρ_E) on the Secrecy Outage Probability (SOP) as a function of ρ_S, with a fixed target secrecy rate of $R = 2$. Similar to Figure 2, the SOP continues to decrease as ρ_S increases, indicating that secure performance is enhanced with higher satellite transmit power. Furthermore, the analytical and simulation results (black curves) show an excellent match, reinforcing the reliability of the theoretical framework. Crucially, Figure 3 clearly demonstrates that the secrecy performance is degraded as ρ_E increases (from 5 dB to 15 dB), resulting

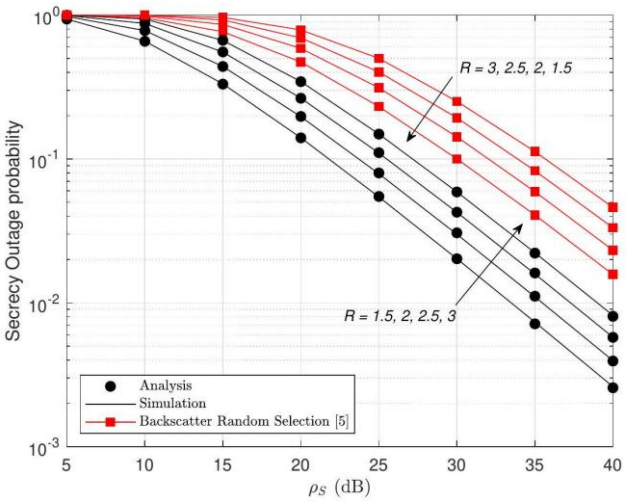


Figure 2. Secrecy-outage probability *versus* ρ_S.

in an increase in SOP and an upward shift of the curves. This occurs because a higher eavesdropper power increases the Eavesdropper's Channel Capacity (C_E), thus reducing the achievable Secrecy Capacity ($C_S = [C_{BS} - C_E]^+$). Notably, in the high ρ_S regime, the SOP curves for different ρ_E values tend to be parallel. This suggests that the secrecy-diversity order of the system is independent of the eavesdropper's power (ρ_E). Finally, the proposed system maintains a lower SOP compared to the reference "Backscatter Random Selection" scheme [5] across all considered ρ_E scenarios.

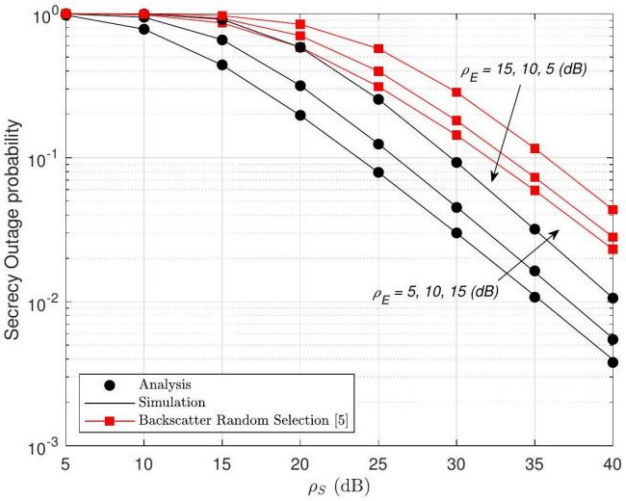


Figure 3. Secrecy-outage probability *versus* ρ_S for different values of ρ_E, with $R = 2$.

Figure 4 depicts the Ergodic Achievable Secrecy Rate (ESC) versus the average satellite SNR, ρ_S, for various eavesdropper SNR values, ρ_E, with the fading-severity parameter fixed at $m = 5$. In terms of the main channel, the ESC consistently increases with ρ_S. This behavior confirms that allocating more transmit power to the signal effectively enhances the system's ergodic-secrecy capacity. Conversely, the ESC is significantly degraded as the eavesdropper's SNR, ρ_E, increases from 0 dB to 20 dB . This degradation is expected, because a higher ρ_E provides the eavesdropper with better channel quality, increasing its capacity and reducing the net secrecy capacity, C_S. Furthermore, the curves illustrate a key secure communication property: for $\rho_E > 0$, the ESC is zero until ρ_S surpasses a minimum threshold. This required ρ_S threshold is directly proportional to ρ_E, meaning that the system must overcome the eavesdropper's channel quality to achieve a positive secrecy rate (e.g. the ESC only becomes positive around $\rho_S \approx 20$ dB when $\rho_E = 20$ dB).

Figure 5 presents the Symbol Error Rate (SER) as a function of the average satellite SNR (ρ_S), with the fading-severity parameter fixed at $m = 51$. The results indicate that the SER decreases sharply as ρ_S increases, which is consistent with theory, since higher transmit power raises the overall SNR and reduces the probability of error. The tight congruence between the Analytical and Simulation data points (for both BPSK and 4QAM) confirms the accuracy of the derived closed-form analytical SER

"Secure Performance Analysis of Satellite-terrestrial Networks-assisted Backscatter Device", H.-N. Nguyen, S.-P. Le, Q.-S. Vu, Q.-S. Nguyen and E. Chromy.

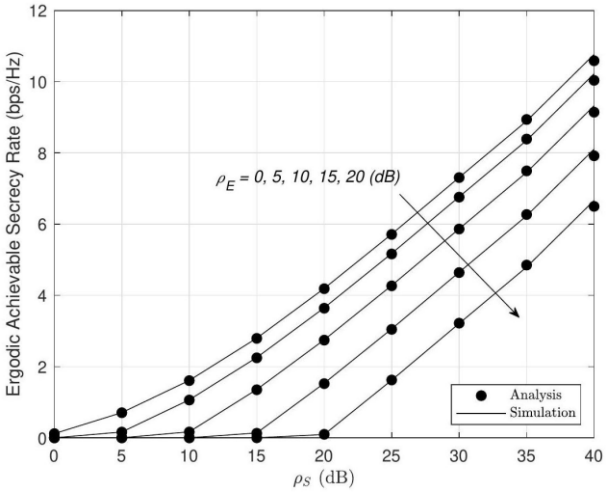


Figure 4. The ergodic-secrecy rate *versus* ρ_S, with $m = 5$.

expressions. Most significantly, the plot illustrates that BPSK achieves a lower SER (better performance) than 4QAM across the entire range of ρ_S examined. This difference stems from BPSK being a lower-order modulation scheme (1 bit/symbol) compared to 4QAM (2 bits/symbol), thus offering superior robustness against errors under the same SNR conditions. In summary, these findings provide practical guidance for selecting an appropriate modulation scheme to optimize the reliability of the backscatter link in the satellite-terrestrial system.

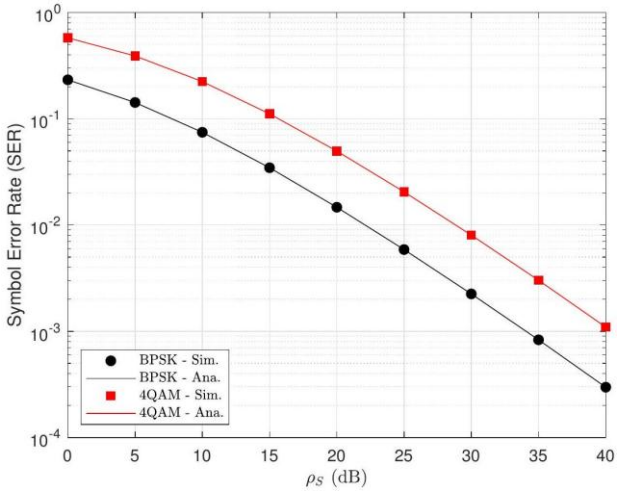


Figure 5. Symbol error rate *versus* ρ_S, with $m = 5$.

5. CONCLUSIONS

This paper has presented a comprehensive secrecy-performance analysis of a two-way satellite-terrestrial backscatter communication system in the presence of an eavesdropper. Closed-form analytical expressions for key secrecy metrics, including the SOP, ESC, and SER, were derived under correlated Nakagami-m fading channels, providing quantitative insights into the system's vulnerability to eavesdropping. The analytical findings were further verified through extensive Monte Carlo simulations, confirming their accuracy across various channel and system configurations. The results revealed that both the satellite transmit power and channel correlation play critical roles in determining the system's secrecy performance. Specifically, increasing the satellite transmit power (ρ_S) dramatically reduces the SOP and enhances the ESC, while a higher eavesdropper power (ρ_E) markedly deteriorates both metrics, thereby defining a minimum ρ_S threshold necessary to achieve a positive secrecy capacity. Furthermore, regarding symbol reliability, BPSK modulation consistently achieved a lower SER than 4QAM, confirming its superior robustness for backscatter links under the examined fading conditions. Although asymptotic analysis was not explicitly included to maintain clarity and conciseness, the derived closed-form expressions and simulation results already capture the key secrecy trends across the entire SNR

range. Future work could extend this study to more complex multi-tag or multi-eavesdropper scenarios, explore AI-driven physical layer security schemes, and perform energy efficiency optimization under practical hardware constraints. Moreover, incorporating cooperative relaying and artificial-noise generation represents a promising direction for further enhancing secrecy performance in next-generation satellite-IoT backscatter systems.

REFERENCES

[1] J. Qian, F. Gao, G. Wang, S. Jin and H. Zhu, "Noncoherent Detections for Ambient Backscatter System," IEEE Trans. on Wireless Communications, vol. 16, no. 3, pp. 1412-1422, March 2017.

[2] C. Zhong et al., "Wireless Information and Power Transfer with Full Duplex Relaying," IEEE Trans. on Communications, vol. 62, no. 10, pp. 3447-3461, Oct. 2014.

[3] G. Wang, F. Gao, R. Fan and C. Tellambura, "Ambient Backscatter Communication Systems: Detection and Performance Analysis," IEEE Trans. on Communications, vol. 64, no. 11, pp. 4836-4846, Nov. 2016.

[4] J. D. Griffin and G. D. Durgin, "Gains for RF Tags Using Multiple Antennas," IEEE Transactions on Antennas and Propagation, vol. 56, no. 2, pp. 563-570, DOI: 10.1109/TAP.2007.915423, Feb. 2008.

[5] G. Lu, Z. Liu, Y. Ye and X. Chu, "System Outage Probability and Diversity Analysis of a SWIPT-based two-way DF Relay Network under Transceiver Hardware Impairments," China Communications, vol. 20, no. 10, pp. 120-135, DOI:10.23919/JCC.ea.2021-0184.202302, 2023.

[6] J. D. Griffin and G. D. Durgin, "Link Envelope Correlation in the Backscatter Channel," IEEE Communications Letters, vol. 11, no. 9, pp. 735-737, Sep. 2007.

[7] D. -Y. Kim et al., "Reverse-link Interrogation Range of a UHF MIMO-RFID System in Nakagami-m Fading Channels," IEEE Transactions on Industrial Electronics, vol. 57, no. 4, pp. 1468-1477, April 2010.

[8] E. Vahedi, R. K. Ward and I. F. Blake, "Security Analysis and Complexity Comparison of Some Recent Lightweight RFID Protocols," Proc. of the 4th Int. Conf. on Computational Intelligence Security Inf. Syst.", vol. 6694, pp. 92-99, Spain, Jun. 2011.

[9] A. Paul et al., "Designing Quantum Gradient Descent Algorithm for MIMO NOMA Rate Maximization with STAR-RIS," IEEE Wireless Communications Letters, vol. 14, no. 4, pp. 959-963, April 2025.

[10] X. Li et al., "Physical Layer Security for Wireless-powered Ambient Backscatter Cooperative Communication Networks," IEEE Transactions on Cognitive Communications and Networking, vol. 9, no. 4, pp. 927-939, Aug. 2023.

[11] Y. Khan, A. Afzal, A. Dubey and A. Saxena, "Secrecy Performance of Energy-harvesting Backscatter Communication Network under Different Tag Selection Schemes," IEEE Journal of Radio Frequency Identification, vol. 8, pp. 43-48, DOI: 10.1109/JRFID.2024.3371877, 2024.

[12] Y. Khan et al., "Deep Learning-based Secure Tag Selection in BackCom Network with RIS-induced Interference," IEEE Journal of Radio Frequency Identification, vol. 9, pp. 797-806, 2025.

[13] N. Q. Sang, T. C. Hung, T. T. Duy, M. Tran and B. Seo Kim, "Securing Wireless Communications with Energy Harvesting and Multi-antenna Diversity," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 11, no. 2, pp. 197-210, DOI: 10.5455/jjcit.71-1732244909, June 2025.

[14] M. Garai, M. Sliti and A. Elfikky, "Ground-to-Satellite FSO Communication: Evaluating Modulation Techniques under Cloud and Turbulence Effects," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 11, no. 2, pp. 260-278, DOI: 10.5455/jjcit.71-1735327157, June 2025.

[15] Y. Khan, A. Dubey and S. K. Soman, "Secrecy Performance of Backscatter Communication Networks with Multiple Reader and Tag Selection Schemes," Proc. of the IEEE 2025 National Conf. on Communications (NCC), pp. 1-6, Mar. 2025.

[16] M. Nafees, D. Dharmendra and A. Kumar, "Secrecy Analysis of Energy Harvesting Backscatter Communications with Tag Selection in Nakagami-m Fading," arXiv preprint, arXiv: 2503.12400, 2025.

[17] W. Saad, X. Zhou, Z. Han and H. V. Poor, "On the Physical Layer Security of Backscatter Wireless Systems," IEEE Transactions on Wireless Communications, vol. 13, no. 6, pp. 3442-3451, June 2014.

[18] Q. Yang, H. -M. Wang, Y. Zhang and Z. Han, "Physical Layer Security in MIMO Backscatter Wireless Systems," IEEE Transactions on Wireless Communications, vol. 15, no. 11, pp. 7547-7560, Nov. 2016.

[19] T. Pecorella, L. Brilli and L. Mucchi, "The Role of Physical Layer Security in IoT: A Novel Perspective ", Information, vol. 7, no. 3, http://www.mdpi.com/2078-2489/7/3/49, 2016.

[20] T. N. Nguyen et al., "Outage Performance of Satellite Terrestrial Full-duplex Relaying Networks with Co-channel Interference," IEEE Wireless Communi. Letters, vol. 11, no. 7, pp. 1478-1482, July 2022.

[21] T. N. Nguyen et al., "Energy Harvesting-based Spectrum Access with Incremental Cooperation, Relay Selection and Hardware Noises", Radio Engineering, vol. 26, no. 1, pp. 240-250, 2017.

[22] T. N. Nguyen, D. T. Do and P. T. Tran, "Time Switching for Wireless Communications with Full-duplex Relaying in Imperfect CSI Condition ", KSII Transactions on Internet and Information Systems, vol. 10, no. 11, pp. 5455-5475, DOI: 10.3837/tiis.2016.09.011, Sep. 2016.

[23] P. T. Tin, D. T. Hung, T. N. Nguyen and T. T. Duy, "Secrecy Performance Enhancement for Underlay Cognitive Radio Networks Employing Cooperative Multi-hop Transmission with and without Presence

of Hardware Impairments", Entropy, vol. 21, no. 2, pp. 217, Feb. 2019.

[24] P. Fazio, "On the Effect of Coverage Range Extent on Next-cell Prediction Error for Vehicular Mobility in 5G/6G Networks: A Novel Theoretic Model," IEEE Transactions on Vehicular Technology, vol. 74, no. 1, pp. 1489-1503, DOI: 10.1109/TVT.2024.3453450, Jan. 2025.

[25] Z. Li, G. Wang and M. Yang, "Performance Analysis of SWIPT-aided Satellite-terrestrial Cooperative Network", Proc. of the 2022 2nd Asia-Pacific Conf. on Communications Technology and Computer Science (ACCTCS), pp. 252-256, Shenyang, China, DOI:10.1109/ACCTCS53867.2022.00059, 2022.

[26] Y. Zhang et al., "Secure Communications for Multi-tag Backscatter Systems," IEEE Wireless Communication Letters, vol. 8, no. 4, pp. 1146-1149, DOI: 10.1109/LWC.2019.2909199, Aug. 2019.

[27] I. S. Gradshteyn and I. M. Ryzhik, Table of Integrals, Series and Products, 6th Edn., ISBN-13: 978-0-12-373637-6, New York, NY, USA: Academic Press, 2000.

[28] N. I. Miridakis et al., "Dual-hop Communication over a Satellite Relay and Shadowed Rician Channels," IEEE Transactions on Vehicular Technology, vol. 64, no. 9, pp. 4031-4040, Sept. 2015.

[29] S. Neumark, Solution of Cubic and Quartic Equations, ISBN: 978-0-08-011220-6, Oxford: Pergamon Press, 1965.

[30] B. Gu, D. Li, H. Ding, G. Wang and C. Tellambura, "Breaking the Interference and Fading Gridlock in Backscatter Communications: State-of-the-Art, Design Challenges and Future Directions," IEEE Communications Surveys & Tutorials, vol. 27, no. 2, pp. 870-911, 2025.

[31] S. Mondal et al., "A Comprehensive Survey on NOMA-based Backscatter Communication for IoT Applications," IEEE Internet of Things Journal, vol. 12, no. 12, pp. 18929-18953, 2025.

[32] M. Ahmed et al., "NOMA-based Backscatter Communications: Fundamentals, Applications and Advancements," IEEE Internet of Things Journal, vol. 11, no. 11, pp. 19303-19327, 2024.

[33] S. Zargari, D. Galappaththige and C. Tellambura, "Transmit Power-efficient Beamforming Design for Integrated Sensing and Backscatter Communication," IEEE Open Journal of the Communications Society, vol. 6, pp. 775-792, DOI: 10.1109/OJCOMS.2025.3527860, 2025.

[34] F. R. Ghadi, M. Kaveh, K.-K. Wong and Y. Zhang, "Performance Analysis of FAS-aided Backscatter Communications," IEEE Wireless Communications Letters, vol. 13, no. 9, pp. 2412-2416, 2024.

[35] F. Xia, Z. Fei, X. Wang, P. Liu, J. Guo and Q. Wu, "Joint Waveform and Reflection Design for Sensing-assisted Secure RIS-based Backscatter Communication," IEEE Wireless Communications Letters, vol. 13, no. 5, pp. 1523-1527, DOI: 10.1109/LWC.2024.3381163, 2024.

[36] S. Jia et al., "Secrecy Performance Analysis of UAV-assisted Ambient Backscatter Communications with Jamming," IEEE Transactions on Wireless Communications, vol. 23, no. 12, pp. 18111-18125, 2024.

[37] H. Luo et al., "Symbiotic Blockchain Consensus: Cognitive Backscatter Communications-enabled Wireless Blockchain Consensus," IEEE/ACM Trans. on Networking, vol. 32, no. 6, pp. 5372-5387, 2024.

[38] L. S. Phu et al., "Enhancing Short-packet Communications: BLER Performance in RIS-assisted Ambient Backscatter NOMA Systems," PLoS ONE, vol. 20, no. 8, pp. 1-26, Aug. 2025.

[39] T. C. Hung, V. M. Bui, T. N. Nguyen and M. Voznak, "Power Beacon-assisted Energy Harvesting Symbiotic Radio Networks: Outage Performance," PLoS ONE, vol. 20, no. 2, pp. 1-16, Feb. 2025.

[40] A.-T. Le et al., "Performance Analysis of RIS-assisted Ambient Backscatter Communication Systems," IEEE Wireless Communications Letters, vol. 13, no. 3, pp. 791-795, 2024.

[41] T.-H. T. Pham et al., "Performance Analysis in D2D Partial NOMA-assisted Backscatter Communication," Advances in Electrical & Electronic Engineering, vol. 23, no. 3, DOI: 10.15598/aeee.v23i3.250314, 2025.

[42] T. C. Hung, Q. Sang Nguyen, V. M. Bui, T.-Q. Thi and N.-L. Nguyen, "Multi-power Beacon Empowered Secure in IoT Networks: Secrecy Outage Probability Analysis," Advances in Electrical & Electronic Engineering, vol. 23, no. 2, DOI: 10.15598/aeee.v23i2.241112, 2025.

[43] Y. Pei, X. Yue, C. Huang and Z. Lu, "Secrecy Performance Analysis of RIS-assisted Ambient Backscatter Communication Networks," IEEE Transactions on Green Communications and Networking, vol. 8, no. 3, pp. 1222-1232, DOI: 10.1109/TGCN.2024.3365692, 2024.

[44] Y. Khan et al., "Secrecy Analysis of Energy Harvesting Backscatter Communication Networks with Multiple Eavesdroppers and Different Tag Selection Schemes," IEEE Transactions on Green Communications and Networking, Early Access, p. 1, DOI: 10.1109/TGCN.2025.3563107, 2025.

[45] J. Li, P. Wang, L. Jiao, Z. Yan, K. Zeng and Y. Yang, "Security Analysis of Triangle Channel-based Physical Layer Key Generation in Wireless Backscatter Communications," IEEE Transactions on Information Forensics and Security, vol. 18, pp. 948-964, DOI: 10.1109/TIFS.2022.3224852, 2023.

[46] R. Singh, I. Ahmad and J. Huusko, "The Role of Physical Layer Security in Satellite-based Networks," Proc. of the 2023 Joint European Conf. on Networks and Communications & 6G Summit (EuCNC/6G Summit), pp. 36-41, DOI: 10.1109/EuCNC/6GSummit58263.2023.10188370, 2023.

[47] S. Jia, R. Wang, Y. Xu, Y. Lou, D. Zhang and T. Sato, "Secrecy Analysis of ABCom-based Intelligent Transportation Systems with Jamming," IEEE Transactions on Intelligent Transportation Systems, vol. 25, no. 3, pp. 2880-2892, DOI: 10.1109/TITS.2023.3250427, 2024.

[48] Z. Liu, Y. Ye, X. Chu and H. Sun, "Secrecy Performance of Backscatter Communications with Multiple Self-powered Tags," IEEE Communications Letters, vol. 26, no. 12, pp. 2875-2879, 2022.
[49] L. Bai, Q. Chen, T. Bai and J. Wang, "UAV-enabled Secure Multiuser Backscatter Communications with Planar Array," IEEE Journal on Selected Areas in Communications, vol. 40, no. 10, p. 29462961, 2022.
[50] Y. Lei, Y. Ye, X. Chu, G. Chen and G. Lu, "On the Strict Secrecy Outage Probability of Wirelessly Powered Backscatter Communications," IEEE Transactions on Vehicular Technology, vol. 74, no. 5, pp. 8345-8350, DOI: 10.1109/TVT.2024.3523389, 2025.

ملخص البحث:

تبحث هذه الدّراسة في الأداء الآمن لجهاز التّشتُّت الخلفي في نظام اتّصال بين الأقمار الصّناعية والمحطّات الأرضية في حال وجود احتمال لاختراق النّظام. وفي النّظام المقترح، يقوم قمر صناعي بإرسال إشاراتٍ إلى جهاز تشتُّت خلفي يعمل بدوره على عكس المعلومات المعدّلة لتعود إلى القمر الصناعي بينما يتعرّض النّظام إلى محاولة اختراق.

ويتمّ تحليل النّظام المقترح عبر تجارب تُجرى على عددٍ من مؤشّرات الأداء الآمن للحصول على تشغيل آمن للنّظام. كذلك تُجرى تجارب للحصول على فهمٍ شاملٍ وأكثر عمقاً للتّشغيل الآمن للنّظام في ظلّ قيمٍ عالية لنسبة الإشارة إلى الضّجيج. ومن ناحيةٍ أخرى، تمّ إجراء عمليات محاكاة مونتي كارلو للتّحقُّق من دقّة التّحليل النّظري. وقد أسفرت التّجارب عن تأكيد دقّة عمل النّظام المقترح بشكل آمن.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 04, December 2025.

ENHANCING PALMPRINT RECOGNITION: A NOVEL CUSTOMIZED LOOCV-DRIVEN SIAMESE DEEP-LEARNING NETWORK

Wafaa Mohammed Cherif[1], Javier Garrigós[2], Juan Zapata[2] and Tarik Boudghene Stambouli[1]

(Received: 14-Jul.-2025, Revised: 17-Oct.-2025 and 29-Oct.-2025, Accepted: 29-Oct.-2025)

ABSTRACT

The advancement of deep learning in biometric systems, in which face and hand modalities have been widely implemented, leads to significant improvements in terms of speed performance and data confidentiality. Palmprint recognition is the main focus of the proposed approach, which deals with databases that are relatively smaller than other biometric datasets. A large and complex deep-learning model may overfit and lose its ability to generalize when applied to such data. This study addresses this challenge by implementing a deep learning model suitable for palmprints, which are characterized by diversity and limited data. Initially, the appropriate Region of Interest (ROI) is extracted using active segmentation, which is fitting for dealing with the difficulty of obtaining palmprints from hand images with closely spaced or connected fingers. In the second stage, a novel customized LOOCV Leave-One-Out Cross Validation (A Modified-LOOCV) technique is integrated with a Siamese deep-learning network for palmprint verification. Unlike conventional LOOCV, our modified scheme optimizes the computational cost while achieving a balanced evaluation on three different datasets. The proposed framework rivals the effectiveness of the advanced palmprint-recognition systems with a high recognition accuracy of 99.75%, improved equal error rate (EER), reduced to 0.002, and faster matching time, making it highly suitable for field application.

KEYWORDS

Palmprint recognition, Deep learning, Customized LOOCV, Siamese network.

1. INTRODUCTION

The technology advancements led to identity verification being an absolute necessity and a crucial requirement for user authentication in private and public organizations with rising safety issues and the overlapping user information in order to preserve data and guarantee information security. In the vast majority of identity applications, biometrics, including signatures, fingerprints, iris patterns, faces, and palmprints, currently replaces traditional technologies [1]. In these, palmprint modality advanced to be widely implemented due to its significant individual variation, ease of use even with low-resolution hand images, and high recognition accuracy [2]. Traditional biometric recognition systems relied heavily on unique features created specifically for a given kind of data. A lot of these features rely on transforms like Gabor, Fourier, and wavelet [3], or on edge distribution as Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT) descriptors, and principal-component analysis (PCA) to minimize the number of feature dimensions [4][5]. These conventional approaches have a number of challenges, such as their inability to handle huge and diverse datasets and their dependence on field knowledge for feature extraction. Deep-learning algorithms have evolved as a solution to these issues, including automatic feature extraction and the capacity to generate hierarchical representations from unprocessed data [6].

For biometrics, deep learning enhances the performance of all recognition systems, increasing their efficiency and the ability to adapt to a wide range of identification challenges [7][8]. These advancements have considerably contributed to palmprint recognition. Compared to fingerprints, palmprints have more creases and have the potential to be used for a more precise representation of identity [9]. Deep learning-based palmprint recognition has been widely investigated; much work on the potential of convolutional neural network (CNN)-based pattern and palmprint recognition has been

1. W. M. Cherif and T. B. Stambouli are with Signals and Images Laboratory (LSI), University of Sciences and Technology Mohamed Boudiaf USTO-MB, BP1505, El M'naouer Oran, Oran, 31000, Algeria. Emails: wafaa.mohammedcherif@univ-usto.dz and bs_tarik@yahoo.com.
2. J. Garrigós and J. Zapata are with Dpto. Electrónica, Tecnología de Computadoras y Proyectos, Universidad Politécnica de Cartagena, Cartagena, 30202, Spain. Emails: javier.garrigos@upct.es and juan.zapata@upct.es.

carried out; however, some crucial issues with overfitting and class imbalance still exist [10]. Similarly, a large database with an important number of images is necessary for CNN and additional deep-learning algorithms. In particular scenarios, like palmprint or fingerprint identification, having a limited number of samples requires protocols to address this insufficient data, including data augmentation, which brings challenges related to modification of the appropriate learning position. As a result, systems that use these processes typically fail to produce accurate results. To deal with these limitations, deep Siamese networks have been proposed to enhance feature extraction and matching processes [11], thus resolving the challenges of standard CNNs for palmprint recognition. Siamese neural networks, contrasted with the CNN algorithms, are built on a Siamese framework that consists of two identical CNNs. The capacity to perform tasks, such as few-shot learning, or learning without new data, is improved by this architecture, which helps learn a distance function that converts the input data into a feature space [12]. The implementation of contrastive loss functions and knowledge transfer learning leads to this increased efficiency and enhances overall system performance. Even with a small number of labeled samples, Siamese networks perform well. This ability is critical in situations when it is challenging to gather large datasets of labeled images using similarity metric learning. Siamese networks may effectively generalize to new classes with a small number of labeled examples or even just one [13]. Through the sharing of weights within sibling networks, Siamese networks perform better than traditional CNNs in terms of resistance to overfitting. This sharing of weights reduces overfitting concerns by improving the model's ability to generalize to new samples.

The potential of Siamese deep networks has been demonstrated in a wide range of applications, including audio classification, time-series analysis, face and palmprint recognition [14][15][16]. It has the ability to avoid overfitting, learn from sparsely labeled data, and improve overall model performance with its reliance on pairwise network learning. However, they have certain limitations due to their requirement for additional computational resources and a longer training time than traditional convolutional neural networks. This study aims to address these limitations by applying a Siamese deep-learning network combined with a Modified Leave-One-Out Cross-Validation (LOOCV) for palmprint recognition. The goal is to overcome challenges related to long training time, hardware requirements, and weak generalization, while enabling rapid and reliable predictions for real-world use. Our model is designed to learn effectively from small-scale palmprint datasets, without relying on extensive data augmentation, which is often necessary in conventional deep-learning methods. To reduce the bias introduced by random data splitting and to shorten execution time, we propose a Modified LOOCV technique that better utilizes the available samples. The approach is validated on four public palmprint databases of varying quality and class sizes.

The key contributions of this work are summarized as follows:

- Application of a small Siamese deep-learning network specifically designed for the nature of palmprint images, which are highly similar in structure. The model effectively supports few-shot learning, making it suitable for datasets with limited samples. It reduces computational complexity, avoids overfitting on small datasets, enables efficient training with limited resources, and is easy to apply in real-world scenarios without requiring data augmentation or similar techniques.
- Implementation of a novel Modified Leave-One-Out Cross-Validation (LOOCV) technique, which ensures efficient use of all available samples, reduces bias from random splits, and accelerates training while preserving strong generalization.
- Combination of the Siamese architecture with the Modified LOOCV, providing both high recognition accuracy and practical deployment feasibility by lowering training time and hardware requirements.
- Comprehensive evaluation on four public contactless palmprint databases of varying sizes and conditions, demonstrating the robustness, reliability, and generalization ability of the proposed approach compared with state-of-the-art methods.

The remaining content of the paper is organized as follows: Section 2 gives a brief overview of palmprint-recognition systems based on CNN and Siamese networks. The experimental procedures and methodology of our study are described further in the third section. The databases used and the experiment results are shown and discusesed in Section 4. Section 5 presents the final conclusion.

2. RELATED WORKS

A variety of deep-learning networks have been used in research papers that offer palmprint-based identification systems. This section aims to review recent advances in deep learning for palmprint recognition, CNN and Siamese networks, taking advantages of their enhanced feature-extraction expertise to outperform classic methods. Table 1 summarizes the most important networks applied.

A recent study [17] applied various CNN architectures to the Birjand University Mobile Palmprint Database (BMPD) offered by Kaggle, which consists of images taken over two different sessions with different rotations. MobileNet outperformed Xception at 88.3% and VGG16 at 70.8% accuracies, with a score of 96.6%. These outcomes demonstrate the superior efficacy of MobileNet in palmprint recognition when evaluated compared to different alternatives. Considering these results, it can be argued that CNNs have achieved very positive results in most plamprint-recognition systems. Recent research has examined palmprint-recognition systems' reliability and security. Multi-Order Extension Codes (MOECs), which combine first-order (1TFs) and second-order (2TFs) texture features, were initially presented by Liao et al. [18] in order to capture additional discriminative information. By combining these characteristics, their approach outperformed conventional texture-coding methods and consistently improved recognition accuracy across Contact-based, contactless, and multi-spectral palmprint databases, including PolyU and IITD, offering an important conventional reference against which recent CNN-based techniques can be compared. Yan et al. [19], on the other hand, proposed a Generative Adversarial Network GAN-based palmprint reconstruction attack applying a modified Progressive GAN (ProGAN) in order to concentrate on the security aspect. They developed a Scale-Adaptive Multi-Texture Complementarity (SAMTC) loss to improve the realism of reconstructed images and a Double Reuse Training Strategy (DRTS) to maximize learning from sparse data. Their research revealed that existing palmprint systems are susceptible to template reconstruction, posing significant privacy and security issues. Recent work has improved palmprint recognition by addressing concerns with privacy, robustness, and generalization. To solve the problem of mislabeled training data, Shao et al. [20] developed a Multi-Stage Noisy Label Selection and Correction (MNLSC) framework. Their method may increase accuracy by more than 30% under high noise rates. A Federated Metric Learning (FedML) technique was concurrently proposed by Shao et al. [21], which allowed multiple users to train together without disclosing private information. This improved recognition accuracy across 18 datasets while preserving confidentiality. In addition, another research [22] focused on generalizing across datasets, creating techniques including transfer learning and adversarial learning that address domain gaps and improve performance on unknown datasets. These contributions demonstrate the trend toward improving palmprint recognition's accuracy while also making it more resilient to noise, cross-domain adaptable, and privacy-preserving.

Zhang et al. [23] provided an integrated CNN-Transformer Global-Local Gating and Adaptive fusion Network (GLGAnet) palmprint-recognition system that combines the Transformer's global modeling with the CNN's local feature extraction. The framework takes advantage of an adaptive feature fusion module and a gating mechanism, with 98.5% and 99.5% recognition accuracy on the Tongji and Hong Kong Polytechnic University datasets, respectively. To increase the effectiveness of CNN while avoiding some of its drawbacks, specifically when handling small datasets and differentiating between really similar classes, numerous current studies recommend implementing the Siamese networks, which advance in image recognition [24], address the drawbacks of conventional techniques by achieving higher accuracy along with processing efficiency. Marattukalam et al. [25] Introduced a Siamese neural-network design for N-shot palm-vein identification. This architecture was created to address a typical biometric recognition problem. Impressive performance metrics were attained by the network when tested on the HK PolyU multi-spectral palm-vein database: 91.5% F1-score, and 90.5% accuracy. Despite the small amount of data, these results demonstrate the architecture's efficacy and potential for practical biometric applications. A relevant approach by Gurunathan et al. [26] presented a palm-vein biometric system using a Siamese network processing distinctive vein patterns for authentication. Compared to conventional biometrics, this approach has benefits including increased accuracy and resistance to spoofing. It is also computationally efficient for mobile use and adaptive to evolving vein patterns over time.

Zhong et al. [27] proposed a palmprint-recognition technique that extracts convolutional features from palmprint images using a Siamese network with two parameter-sharing VGG-16 networks; the network compares these features to evaluate similarity and recognition accuracy. The approach

showed robustness with an Equal Error Rate (EER) of 4.559% on the XJTU dataset, while it achieved an EER of 0.2819% on the PolyU dataset. These outcomes demonstrate the method's efficacy and versatility across various datasets. In light of these findings, two subsequent studies [28][29] have suggested a Meta-Siamese Network-based palmprint recognition. The initial experiment [28] presented a Meta-Siamese network that uses episodic training to improve feature integration and similarity metrics, building upon the Siamese network architecture. Using deep hashing networks, this model was expanded to zero-shot recognition tasks and showed competitive performance on eight different datasets. In the second experiment [29], a new Meta-Siamese Network (MSN) intended for small-sample palmprint identification was shown. Applying a flexible architecture and two distance-based loss functions to improve optimization, this method used episodic training. The MSN model demonstrated significant improvements over baseline approaches in both confined and unconstrained benchmark palmprint databases. Furthermore, a recent study [16] implemented the Siamese network for palmprint identification that uses two CNNs to extract and compare palmprint features using shared weights. A loss of variance function is used to assess the extracted features and determine whether or not the images are of the same person. The approach demonstrated its efficacy with a 0.044 equal error rate and 95.6% recognition accuracy.

Table 1. A summary of significant previous work based- CNN for palmprints.

Ref/Model	Year	Method	Datasets	Accuracy
[30] Siamese-Hashing Network	(2019)	SHN. A non-pooling Siamese-Hashing Network structure.	PolyU Multi-spectral palm-print dataset	97.98%
[31] InceptionResNet-v2	(2022)	InceptionResnet-v2 pre-trained deep-neural networks (DNNs), with Rectified Linear Unit (ReLU), dropout, and fully connectedWith Soft max.	PolyU palm-print database	99.21%
[32] LeNet-5	(2022)	LeNet-5 Convolutional Neural Network	Tongji Contactless Palm-print Dataset	97%
[16] Siamese Network	(2023)	Siamese Network with a loss of variance function for similarity prediction.	CASIA, THU-PALMLAB	95.6% (EER: 0.044)
[33] Pretrained VGG16 within Siamese Frame-work	(2023)	Siamese network with VGG-16. Palmprint feature extraction is used to determine palmprints' similarity.	CASIA dataset	91.8% (left hand), 91.7% (right hand)
[23] Gating mechanism and adaptive feature fusion	(2023)	Integration of CNN and Transformer-GLGAnet for palmprint recognition.	Tongji U dataset, Hong Kong Polytechnic dataset	98.5%, 99.5%
[17] CNNs	(2024)	Convolutional Neural Network models with the Xception, VGG16, ResNet50, MobileNet, and EfficientNetB0 architectures.	BMPD dataset	96.6%
[34] Fusion Mechanism	(2024)	Fusion Mechanism with Multi-direction Gabor Filter for prediction optimization.	CASIA palm-print database	99.41%

In support of the most important results based on Siamese and CNN networks, our study relies on the Siamese network by integrating a Modified LOOCV for enhancing its findings. In light of these advantages, the proposed approach remains excellent for field implementation, where it provides excellent recognition results, faster speed and less memory usage, which improves palmprint-recognition systems.

3. METHODOLOGY

The proposal focuses on the enhancement of a palmprint-recognition system, consisting of two main stages: pre-processing and palmprint recognition using the Siamese network based on the Modified-LOOCV technique.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 04, December 2025.

3.1 Pre-processing

In this stage, the Region of Interest (ROI) is extracted, based on active contour segmentation. Our goal is to implement a snake-based model for the segmentation of hand images, with the intention of creating a model that is robust and capable of identifying the hand valley points and correctly extracting the region of interest, thereby addressing the limitations of the traditional methods, which have produced erroneous findings when extracting ROIs from hand images with two conjoined fingers.

The active contour model, often referred to as snakes, has its roots in elastic models but is primarily credited to the pioneering work of the Kass team [35]. These models derive their name from their capacity to deform themselves into snake-like shapes Figure 1.

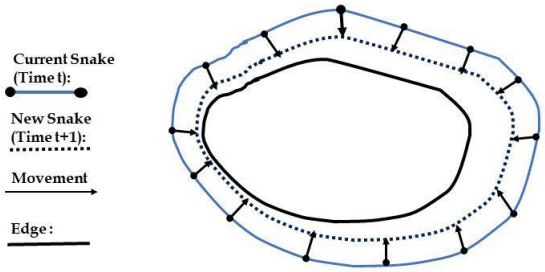
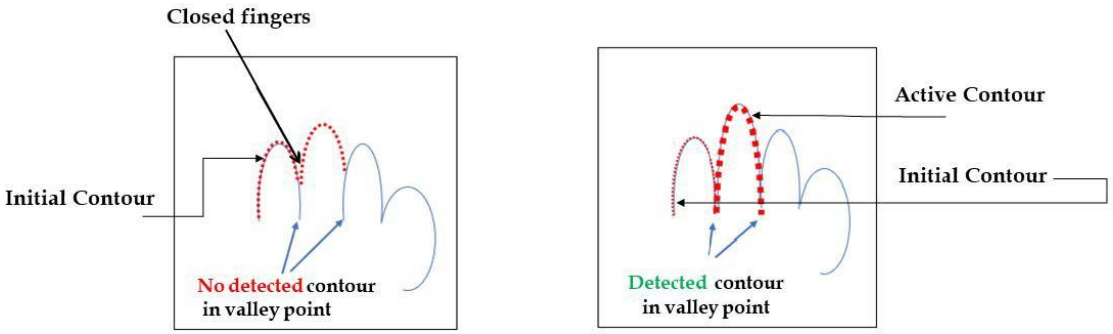


Figure 1. Closed snake active contour model.

Figure 2 demonstrates how the active contour method is applied. The contours are highlighted in red. In our proposed methodology, active contour is employed with the specific goal of precisely delineating the areas between the hand and fingers, particularly in scenarios where the fingers appear to be stuck together or tangled. The method of active contour [32] is based on an initial contour and then concentrates on the contour line that needs to be produced by means of the impact of internal and external energy on a closed snake model. Figure 2a displays the initial contour obtained through Otsu's method. However, this method is not effective for determining edges in small areas (closed fingers).

Subsequently, in Figure 2b the active contour model applies the initial contours obtained as the starting point for active-contour application, successfully extracting contours in limited areas (closed fingers). This process enables the accurate extraction of the Region of Interest (ROI), which defines the core contribution of our suggested pre-processing procedure.



(a) Initial contour based on the threshold method. (b) Final contour based on active-contour segmentation.

Figure 2. Principles of active-contour application.

- **ROI Extraction**

  In the process of Region of Interest (ROI) extraction, several main tasks are involved. Initially, the input color image is converted into grayscale, the background is removed, and the hand contour is extracted using active segmentation. As shown in Figure 3, the hand edge is first extracted (b) and combined with the segmentation mask to preserve structural details (c). Morphological operations are then applied to progressively remove noise and refine the contour (d–e). Since hand images often contain light reflections, these are explicitly detected (f) and integrated into the mask to improve boundary accuracy. The final output (g) provides a clean

hand contour. This sequential refinement ensures robustness by addressing noise and reflections, resulting in a reliable region of interest extraction for the next step.

Figure 4 demonstrates the enhanced performance of the active contour over traditional threshold segmentation, on two original hand images. Due to the limits of the middle and ring fingers (closed fingers), Otsu thresholding, and Kirsch edge detector segmentation in (b) it was unable to identify the entire contour and valley points. Instead, as (c) shows, the active contour model is effective at locating the complete contour (active contour) from the initial contour, especially in the narrow areas between the fingers (closed fingers), which makes it easier to identify and extract the valley points as well as the Region of Interest (ROI) for the next phase. The Valley Points Extraction involves identifying four points corresponding to finger intersections by analyzing local minima through the contour employing a combination of methodologies detailed in [36][37], which provides a comprehensive overview of the process. Finally, ROI Computation computes the ROI based on extracted valley points. Section 4.2 illustrates the details of the final ROI extraction result.

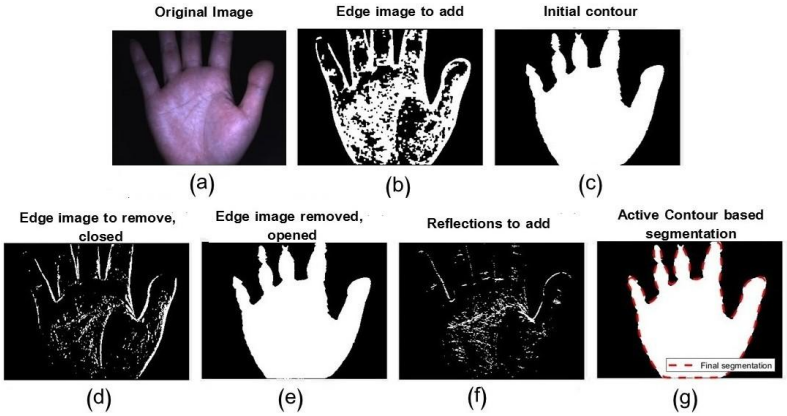


Figure 3. Active-contour hand segmentation: (a) Original image, (b) Edge image to add, (c) Initial contour from Otsu with edge fusion, (d) Edge image to remove (closed), (e) Edge image removed (opened), (f) Reflections to add, and (g) Final active-contour segmentation.

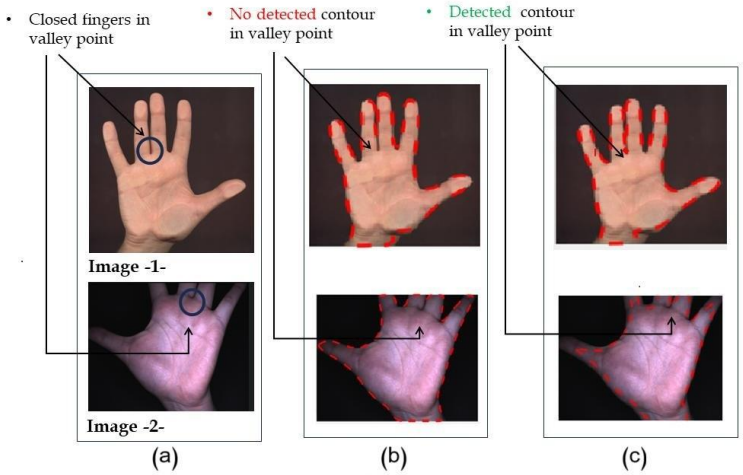


Figure 4. Active-contour application for hand images: (a) Two original images displaying fingers linked together; (b) Threshold segmentation; (c) Active contour-based segmentation.

3.2 Palmprint Recognition: The Modified LOOCV-based Siamese Network

The Leave-One-Out Cross Validation (LOOCV) technique is used to evaluate multiple machine and deep learning-based models. It consists of training the model several times (iterations), by leaving out one image from the entire database, and using the rest of the images for the training phase. This process is repeated depending on the total number of images in the database. The results of all iterations are averaged to get a final model performance. Figure 5 provides the standard LOOCV process, where for each itteration, one sample is left out for testing while the remainder is used for

training. Compared to k-fold, repeated k-fold cross validation, or simple random split algorithms, Leave-One-Out Cross Validation (LOOCV) has several advantages. It generates an estimate of the generalization error that is nearly unbiased, making it a more accurate test than traditional splitting techniques. Experimental comparisons [38] have also demonstrated that, through appropriate parameter tuning, LOOCV can provide higher sensitivity and more balanced accuracy in certain classifiers, including Random Forest and Bagging.

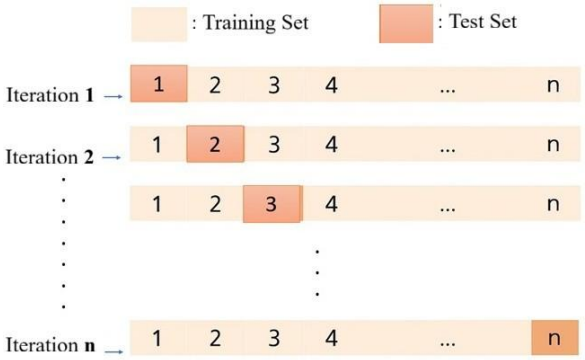


Figure 5. Standard LOOCV (leave-one-out cross validation).

In palmprint recognition, where images show high similarity and datasets are generally smaller, LOOCV provides obvious benefits. Testing each sample reduces the possibility of inaccurate results from random splits and guarantees thorough and objective evaluation. However, due to the high computational demand of LOOCV, we propose a Modified LOOCV to avoid the computational expenses associated with standard LOOCV. Instead of excluding one image from the entire dataset in each iteration, we exclude two images from each class for testing while using the remaining ones for training, as shown in Figure 6. This method involves training the model, as in LOOCV, but then taking the result from the first iteration due to its consistently outstanding performance. Table 2 outlines the distinctions between Standard and Modified LOOCV.

Table 2. Comparison of traditional LOOCV and modified LOOCV.

Aspect	**Traditional LOOCV**	**Modified LOOCV**
In each iteration	Leaves one image out from each class of the entire dataset.	Leaves two images from each class of the entire dataset.
Performance-assessment	Requires completion of all iterations to fully assess model performance.	Allows rapid evaluation from the first iteration.
Deployment-efficiency	Prolongs model deployment due to iterative training and evaluation.	Demonstrates practical performance for real-world applications-evaluation from the first iteration.

For instance, in a dataset with 600 classes and 10 images per class, leaving out 2 images per class for testing in the first iteration allows for a thorough evaluation: 1200 images in testing (20% of the dataset) and 4800 in training (80%).

Unlike random selection methods, as the train-test-split validation, the Modified LOOCV ensures that all classes are represented in the testing data, enhancing evaluation reliability. Moreover, it yields comparable accuracy to standard LOOCV, but with reduced computational complexity, optimizing efficiency without compromising performance assessment. The Modified LOOCV is integrated into the training process of the Siamese network, as illustrated in Figure 6.

Before initiating the LOOCV loop, the main database is divided into several classes, each representing a person with a unique label and containing ten image samples. This ensures that at least two samples per class are available for testing, maintaining class balance across iterations.

The Siamese network is then trained using the Modified Leave-One-Out Cross Validation strategy. In this setup, two identical sub-networks process paired inputs to assess their similarity. The architecture of the Siamese model, illustrated in Figure 7, consists of twin branches that share the same weights, allowing both inputs to be transformed through identical feature-extraction operations. This shared-weight mechanism ensures consistent feature representation and enhances the model's ability to distinguish between genuine and impostor pairs.

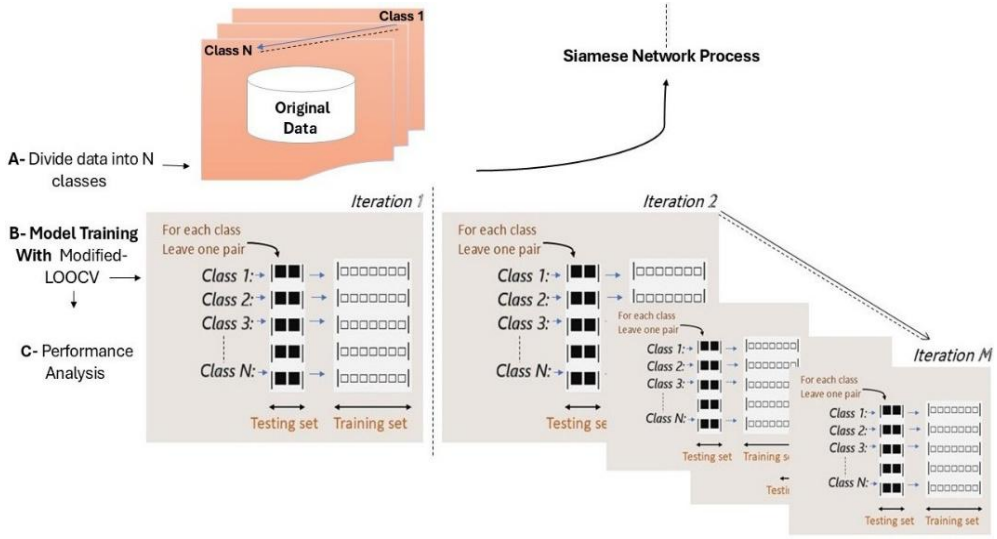


Figure 6. Modified LOOCV workflow.

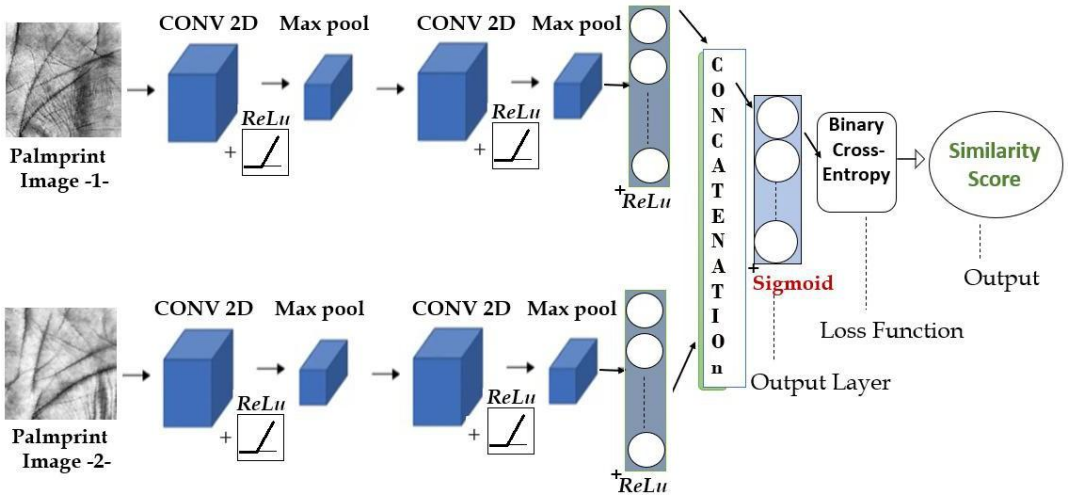


Figure 7. Siamese-network architecture.

The Siamese-network architecture includes two main components:

1) **Principal Network:** This component processes input palmprint images and extracts discriminative features through convolutional and fully connected layers. The detailed configuration is presented in Table 3. The first convolutional layer (Conv1) employs a large kernel of 10×10 with 64 filters and a stride of 1 to capture global edges and texture patterns, minimizing the need for deeper networks. The second convolutional layer (Conv2) applies a smaller 7×7 kernel to refine local structures. Each convolutional block is followed by a max pooling layer that reduces spatial dimensions and strengthens invariance. The fully connected layer (4096 neurons) transforms the extracted maps into compact embeddings suitable for comparison. This architecture effectively prevents overfitting on small palmprint datasets while maintaining low computational cost, making it highly suitable for real-time biometric applications.

2) **Similarity Metric:** After feature extraction, the similarity between embeddings is computed using a metric defined in Equation 1. The resulting values are passed through a sigmoid activation function (Equation 2), producing probabilities that express the degree of similarity between pairs of samples. Higher probabilities indicate greater similarity, while lower ones indicate dissimilarity.

$$\text{Similarity Scores} = \sigma(\text{output}) \tag{1}$$

where:

Similarity Scores: Output probabilities representing the similarity between input pairs.

$\sigma(x)$**:** Sigmoid function is defined as:

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

Table 3. Siamese-network parameters.

Layer	Feature Map	Size	Kernel Size	Stride
Input image	1	150x150	-	-
CONV1	64	141x141	10x10	1
Max Pool	64	70x70	2x2	2
CONV2	128	64x64	7x7	1
Max Pool	128	32x32	2x2	2
FC1	4096	1x1	-	-
Output	1	1x1	-	-

Outputs: The feature similarity scores generated by the Siamese network.

To optimize the model, the Binary Cross-Entropy (BCE) loss function is used, as shown in Equation 3. This function efficiently drives the network to produce closer embeddings for similar pairs and distant ones for dissimilar pairs, thus improving discriminative capability.

$$L = [y.\log(p) + (1-y).\log(1-p)] \quad (3)$$

Here, L represents the computed loss for each pair, y is the ground-truth label (1 for similar, 0 for dissimilar), and p denotes the predicted probability of similarity. During training, the network minimizes this loss through gradient-based optimization to generate robust and discriminative feature embeddings.

For each Modified-LOOCV iteration, independent datasets are formed for training and testing, ensuring that all classes contribute representative samples. The Siamese network is initialized with its optimizer and trained for several epochs using the BCE loss. After training, performance is evaluated based on accuracy and Equal Error Rate (EER). Additionally, loss, accuracy, and ROC curves are plotted to visualize the trade-off between true and false positive rates at different thresholds.

The proposed Siamese network integrated with the Modified LOOCV and optimized *via* Binary Cross-entropy loss provides an effective framework for palmprint recognition. It offers precise similarity measurement between paired samples while maintaining high efficiency and strong generalization across datasets.

4. RESULTS AND DISCUSSION

4.1 Datasets and Experimental Environment

To demonstrate our segmentation method and evaluate the recognition accuracy of the proposed method, we considered the four contactless databases, three were employed for model training, and the fourth was kept unseen during training to serve as an independent dataset for prediction and evaluation. Some typical samples from the employed palmprint databases are illustrated in Figure 8. The Tongji Contactless Palmprint Database, developed at Tongji University in China with a dedicated acquisition device, includes images from 600 subjects captured over two sessions [39][40]. The IIT Delhi Touchless Palmprint Database (V1.0) IITD [41][42], contained color images with various artifacts and illumination changes. The GPDS150 Palmprint Database, created in Spain, provides palmprint images from 150 individuals, adding further variability in acquisition settings and subject diversity [43]. the PolyU DB (Version 2.0) [44]includes 1140 right-hand (2D and 3D) images taken of 114 individuals. To enable for evaluation under different position settings, each participant presented five contactless hand poses in various orientations. The most important characteristics of these databases are summarized in Table 4.

The proposed ROI extraction process is demonstrated in MATLAB R2018a, and the palmprint-recognition model is trained in Python within an Anaconda environment on Ubuntu. The experiment includes an Intel Core i9-9820X (LGA-2066), 64 GB DDR4 RAM (4×16 GB Ballistix Sport LT, 2400 MHz) and Nvidia RTX 2080 Ti GAMING OC 11 GB GPUs (TU102, Rev. A).

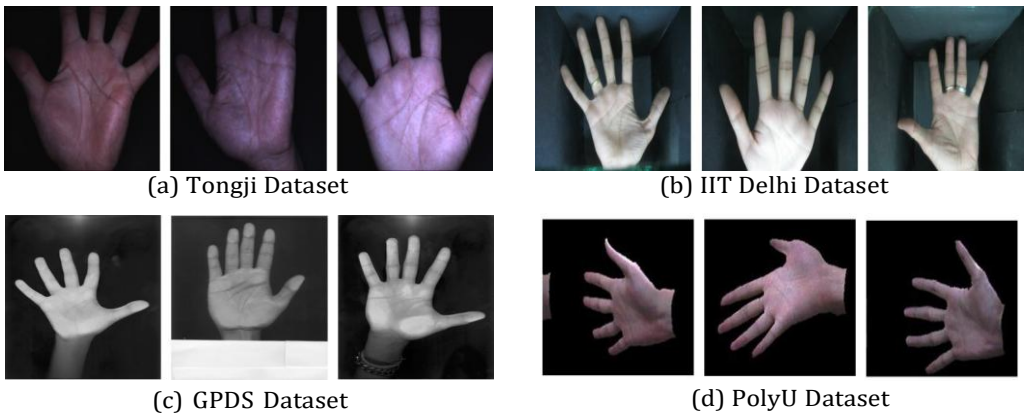


Figure 8. Typical samples from the palmprint datasets: (a) Tongji, (b) IIT Delhi (V1.0), (c) GPDS, and (d) PolyU DB (Version 2.0).

Table 4. Databases' characteristics.

Details	Tongji DB	IITD DB	GPDS DB	PolyU DB
Number of	600	230	150	114
Number of	10	5	10	5
Number of images Gray/color	6000 Gray scale	1265 Gray scale	1500 Gray scale	1140 (2D + 3D) Color (2D) + 3D depth
Resolution devices	800 × 600 Camera	800 × 600 Camera	1403 × 1021 HP Scanner	640 × 480 Minolta VIVID 910 3D digitizer
Origin	Chinese	Indian	Spanish	Hong Kong (zPolyU)

4.2 ROI Extraction

The proposed segmentation was applied to the three hand databases mentioned above. To ensure the method's efficacy across all datasets, we selected hand images with specifically two fingers interlocked. The ROI extraction results for each are shown in Figures 9, 10, 11 and 12, for the contactless databases Tongji, GPDS, IITD and PolyUData, respectively.

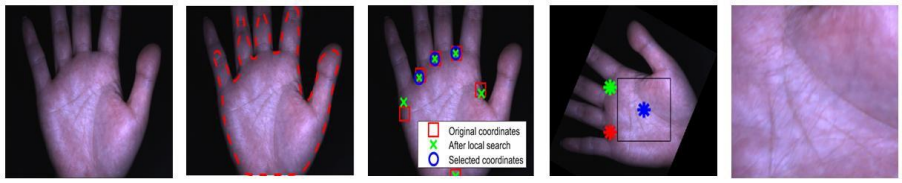


Figure 9. Tongji hand dataset ROI extraction: (a) Input image, (b) Active-contour segmentation, (c) Valley-point extraction, (d) Rotation and ROI computing, (e) Final ROI.

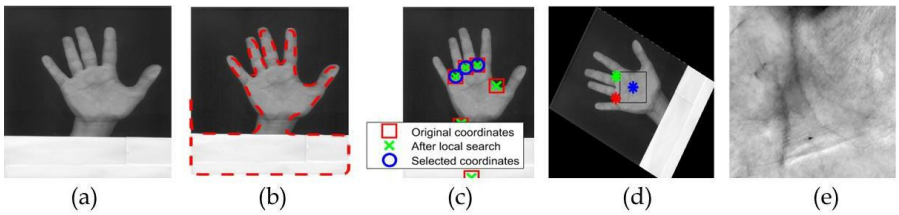


Figure 10. GPDS hand dataset ROI extraction: (a) Input image, (b) Active-contour segmentation, (c) Valley-point extraction, (d) Rotation and ROI computing, (e) Final ROI.

The proposed active-contour segmentation method offers a solution to the imprecise and low-quality results of previously used methods and works that use edge detectors and threshold methods for ROI extraction [45]. Figure 13 illustrates our proposed active-contour method performance in comparison with the threshold method. In 1, in the absence of active contours, with Otsu segmentation, the hand images with fingers poorly spread, resulting in incomplete contour (a). This leads to an error in valley

positions, particularly those associated with the two fingers that are attached to the ring and middle fingers (b), these valley points serve as the basis to calculating the area's ROI. Therefore, it is impossible to determine the rectangle, and the final ROI cannot be extracted. In contrast, as shown in 2, by implementing our proposed segmentation, it allowed for extracting the entire contour, including that of the narrow region, successfully identifying all correct valley points, and finally extracting the yield ROI.

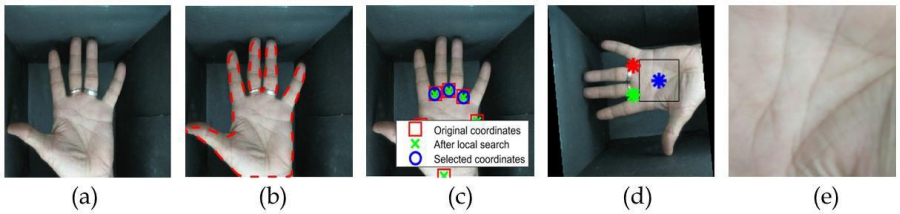


Figure 11. IITD hand dataset ROI extraction: (a) Input image, (b) Active-contour segmentation, (c) Valley-point extraction, (d) Rotation and ROI Computing, (e) Final ROI.

4.3 Palmprint Recognition

The training process is assessed using the Modified LOOCV. Initially, the three datasets are split into classes, with 10 samples of each class representing a person. For testing, two images are selected from each class, while using the remaining images for training. For instance, the GPDS dataset has 1,500 images divided into 150 classes. Leaving out two images per class for testing in the first iteration, a total of 300 images are used for testing data. Thus, the test set will contain 20% of the entire images and 80% in the training dataset, that holds for every other database.

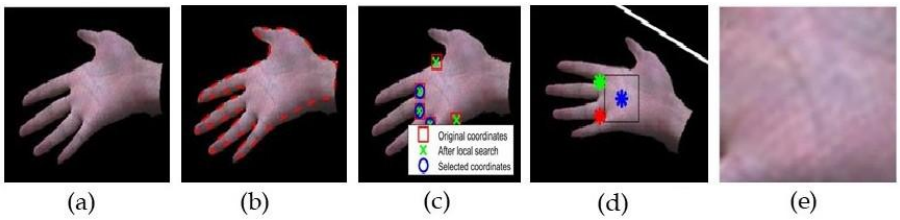


Figure 12. PolyU hand dataset ROI extraction: (a) Input image, (b) Active-contour segmentation, (c) Valley-point extraction, (d) Rotation and ROI computing, (e) Final ROI.

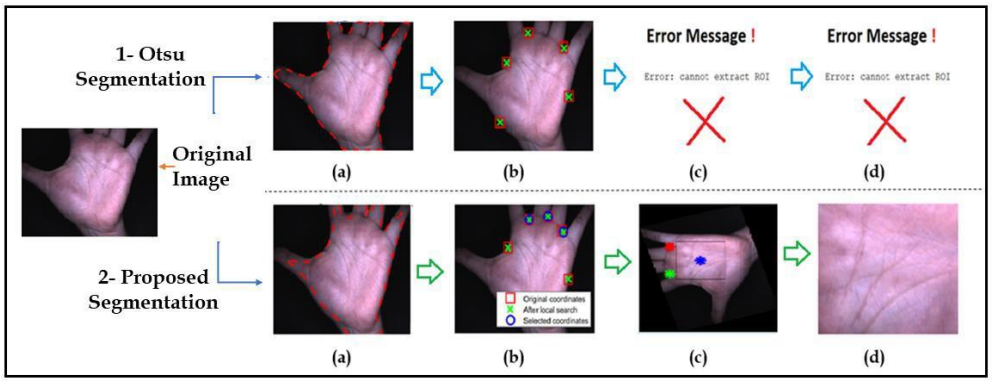


Figure 13. Active-contour performance for precise ROI extraction: (a) Segmentation, (b) Valley-point extraction, (c) Rotation and ROI computing, (d) Final ROI.

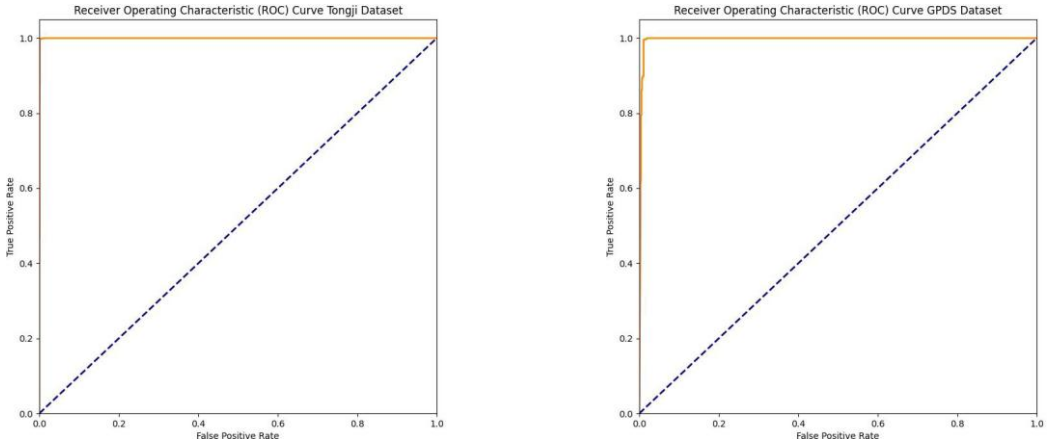
Our model was trained over 50, 100, and 200 epochs. When comparing the execution results, we found that the best accuracy and the fastest convergence rate were obtained within 100 epochs, with a learning rate of 10^{-5} for Tongi and IITD datasets, and 10^{-4} for the GPDS database. All metrics are evaluated once the first LOOCV iteration is completed by determining the average accuracy and the Equal Error Rate (EER), along with additional performance metrics, including Precision, Recall, and F1-score. Precision measures the proportion of correctly identified positive samples among all predicted positives, recall represents the proportion of correctly identified positive samples among all actual positives, and F1-score is the harmonic mean of precision and recall, reflecting the overall

balance between them. The applied model achieved the best accuracy of 99.75% on the Tongji dataset. Table 5 displays the accuracy, EER, precision, Recall, and F1-score results for the available datasets. The evaluation metrics obtained demonstrate that the model is well generalized. The high and balanced precision and recall values indicate that the model effectively identifies genuine matches while maintaining a low rate of false predictions. These findings validate that the model provides good discrimination ability across different datasets.

Table 5. Recognition performance on different datasets.

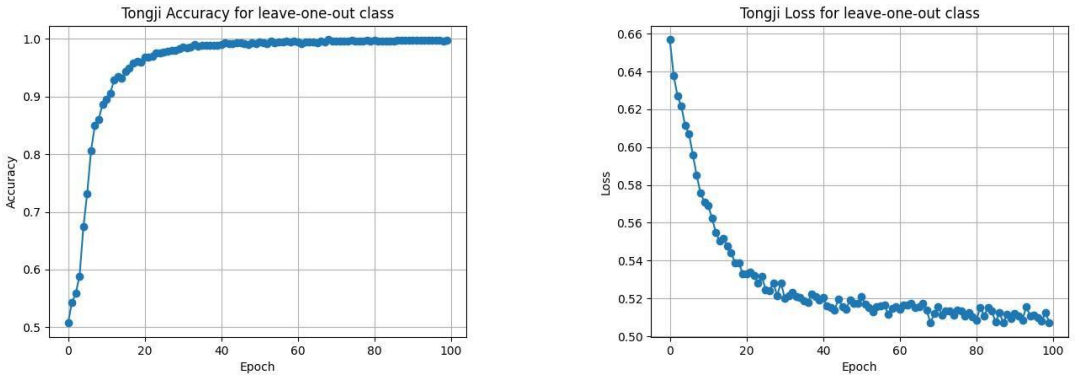
Dataset	Precision	EER	Accuracy (%)	Recall	F1-score
Tongji DB	0.9980	0.002	99.75%	0.9930	0.9955
IITD DB	0.9929	0.04	95.9 ± 1.0%	0.9461	0.9690
GPDS DB	0.9973	0.01	99.2%	0.9932	0.9952

The proposed palmprint-recognition system performs exceptionally well across three different databases according to ROC, accuracy and loss graphs illustrated in Figures 14, 15, 16 and 17. Approximate values showed an accuracy of about 99.75% in the Tongji database, 99.2% in GPDS dataset, and 95.9% in the IITD database. These accuracy values demonstrate how well the system can recognize palmprint images, demonstrating its excellent efficacy in handling a wide range of data. Turning to the loss, the results remained extremely low, suggesting that the model learns effectively and minimizes errors during the training process.



(a)ROC curve for the Tongji dataset (b) ROC curve for the GPDS dataset

Figure 14. Receiver operating characteristic (ROC) curves.

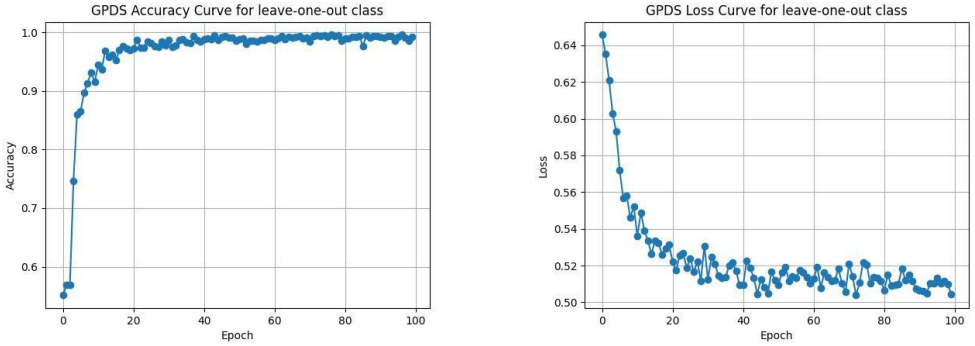


(a) Accuracy on Tongji dataset (b) Loss on Tongji dataset

Figure 15. Test-accuracy and loss curves for Tongji dataset.

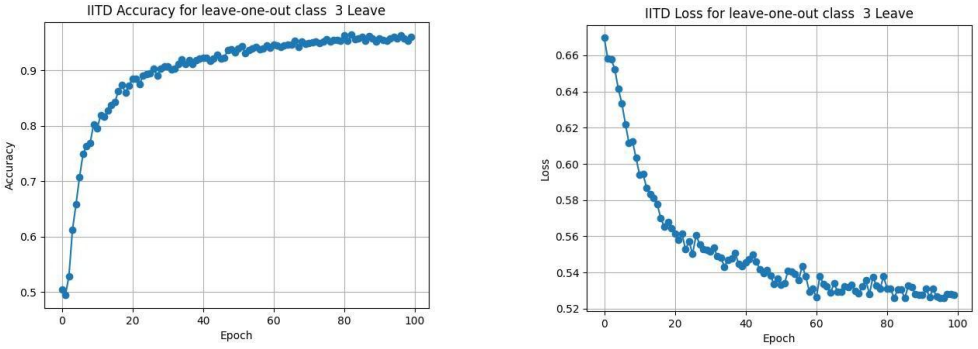
Analyzing Table 5, the Tongji dataset exhibits the highest performance among the datasets, with the lowest EER of 0.002, indicating the most precise and well-balanced trade-off. While the GPDS dataset offers a slightly better EER than the IITD dataset, it nevertheless offers insightful performance. Overall, these findings demonstrate the high level of recognition accuracy of our system-based

palmprints over several datasets. The plots of ROC curves for the three available databases are displayed in Figure 14 for Tongji and GPDS datasets, and in Figure 17 for the IITD data, for a more thorough explanation and confirmation of our system's performance. These curves plot the True Positive Rate against the False Positive Rate. Completing our model execution, the AUC values were converging to 1 for both Tongji and GPDS, and to 0.97 for the IITD dataset. Regarding distinguishing between positive and negative classes, the Tongji dataset demonstrates the highest AUC, indicating nearly perfect performance. The high AUC values for the other datasets also indicate strong model performance. This generalization demonstrates the stability and high recognition accuracy of the Modified LOOCV-based Siamese network across a variety of datasets. Our approach proved its efficacy compared to recent related works, as illustrated in Table 6.



(a) Accuracy on GPDS dataset (b) Loss on GPDS dataset

Figure 16. Test accuracy and loss curves for GPDS dataset.



(a) Accuracy on IITD dataset (b) Loss on IITD dataset

Figure 17. Test-accuracy and loss curves for IITD dataset.

Table 6. Comparative accuracy performance of the proposed method relative to similar works. Accuracy (%) and EER (decimal).

Ref./Method	**IITD**	**Tongji**	**GPDS**
[46] HOG-SGF-AE	-	98.85%	-
[47] MTCC	EER=3.94	EER=0.0043	-
[48] MTPSR	-	-	96.83%
[49] DeepNet/ResNet	95.5%	99.5%	-
[50] SMHNet	-	97.36%	-
[51] Meta Metric Learning	94.02%	93.39%	-
[23] Transformer-GLGAnet	–	98.5%	-
[52] CCNet	EER=0.0018	EER=0.00004	-
[53] CO^{3}Net	EER=0.0047	EER=0.0050	-
[21] FedML (Triplet)	89.13% EER=0.0569	93.51% EER=0.0296	-
[54] **Siamese Net**	94.3%	97.7%	-
Ours	**95.9%** **EER=0.04**	**99.75%** - **EER=0.002**	**99.2%** - **EER=0.01**

In order to reinforce the previous findings and demonstrate the efficacy of our proposed Modified-LOOCV with the Siamese network, we trained the Siamese model with a random split of the database without applying the proposed LOOCV. The results confirmed the effectiveness of the Modified LOOCV in increasing the accuracy rate, where the accuracy was higher than that of the normal data splitting technique by 2.5%. The comparative analysis is shown in Figure 18; we adopted the IITD database results, since it demonstrated a significant transition, particularly in the ROC curve.

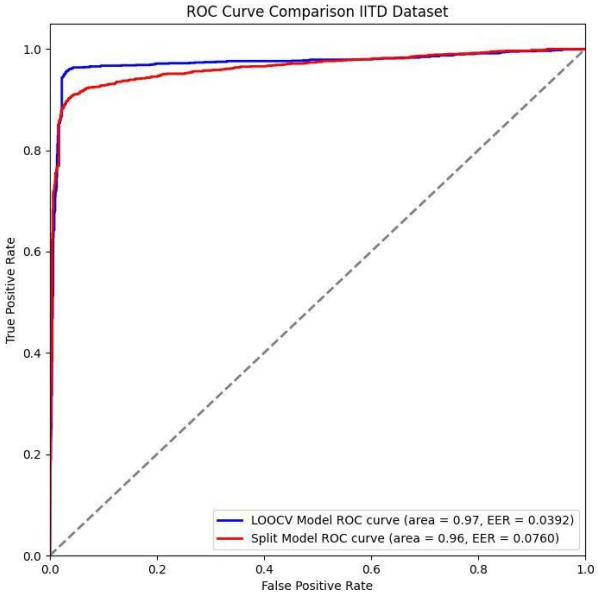


Figure 18. ROC-curve comparison: Modified-LOOCV *vs.* standard dataset split.

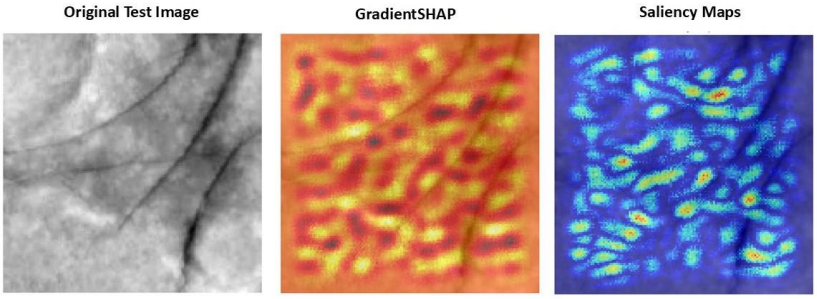
Table 7. Comparison of matching times for Tongji dataset.

Method	Matching time (ms)
VGG-16	26.8
PalmNet	22.8
Ours	**20**

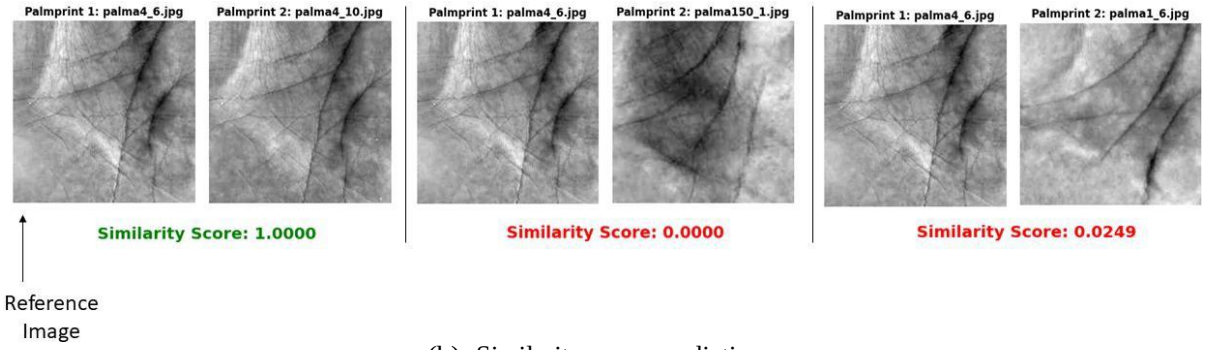
In the prediction phase, we evaluated the precision and speed of the proposed Siamese network based on Modified LOOCV, by measuring the time needed to test and predict new images. It is important to note that the model was not trained on these test images previously. When compared to previous works' matching times, we obtain a matching time of 0.02 seconds (20 ms), where our model achieves a prediction speed that is competitive with widely adopted CNN-based palmprint recognition systems based on CNNs. Table 7 displays a brief comparison result with certain methods described in [49] on Tongji dataset.

The model compares two images and computes a similarity score to determine whether they belong to the same person. To interpret these scores, we applied Gradient-based SHapley Additive exPlanations (GradientSHAP) and Saliency Maps. GradientSHAP combines gradients with SHAP values to estimate feature contributions, while Saliency Maps identify the regions with the greatest influence on the output. As shown in Figures 19 and 20, both methods reveal that the network focuses on palmprint line structures. The Saliency Maps emphasize the most discriminative patterns, whereas GradientSHAP assigns importance across ridges and creases. These visual explanations demonstrate that the network depends on significant palmprint traits rather than on irrelevant background.

A further test was performed using various palmprint images from available databases, two types of test were performed. The first used images from the trained datasets, where a few images were excluded before training and placed in a separate folder for prediction. This ensured that these samples were never seen by the network during the training process. The obtained results on the GPDS dataset in Figure 19 show that the model accurately recognized these unseen samples with high similarity scores. The second test was conducted using images from the PolyU dataset, as shown in Figure 20, which were entirely unseen by the network. The model also achieved very good similarity results on this external dataset, confirming its strong generalization ability.

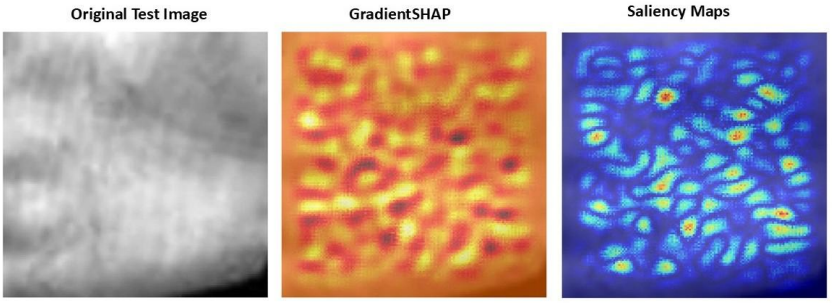


(a) Interpretability of palmprint similarity predictions.



(b) Similarity-score prediction.

Figure 19. GPDS palmprint verification.



(a) Interpretability of palmprint similarity predictions.

(b) Similarity score prediction.

Figure 20. PolyU palmprint verification.

The system verifies that the two images are not of the same person when the score is less than the threshold value and *vice versa*. We chose a threshold of 0.5, which is adjustable based on the needs of the security system. We compared a reference image with three other images of three different random individuals. As illustrated in Figures 19 and 20, the model assigns a score close to 1.0 for palmprints belonging to the same individual, while the score approaches 0.0 for palmprints from different individuals. The attribution results highlight the palmprint regions that most influenced the predictions, with both methods consistently focusing on principal palm lines, ridge intersections and local ridge textures. Such interpretability increases security when using the model for real-world applications and makes the model's decisions more transparent.

5. CONCLUSION

A palmprint-recognition system based on deep learning has been developed, providing a highly efficient technique to predict palmprints by integrating a Modified-LOOCV strategy with a flexible Siamese architecture. To ensure a thorough assessment and model optimization, we initially provide an ROI-extraction technique based on active-contour segmentation. This approach can extract the ROI with variable contrasts and with excellent precision. The proposed Modified-LOOCV is an innovative strategy designed for datasets with significant sample similarity. We effectively capture the wide variety of the dataset while reducing computational time and enhancing the standard Siamese network efficacy. This enhancement enables the fast evaluation of the model's performance, providing an indication of its power and supporting its quick integration into real-world applications, such as access control and forensic identification, with high accuracy and low computational cost. The proposed model reaches an accuracy of up to 99.75%; it is competitive with many existing systems due to its accuracy and execution speed, even on small databases. The findings demonstrate that the suggested model has the potential to be widely used in real-world biometric identification systems, as it can efficiently learn discriminative palmprint characteristics and retain strong recognition performance even when evaluated on unseen data. In future work, we intend to extend and apply the proposed scheme to the Multi-Spectral Palmprint dataset and further generalize the methodology for diverse pattern-recognition systems to provide a deeper validation of its adaptability, robustness and applicability.

ACKNOWLEDGEMENTS

This work has been partially funded by the Spanish Government-Agencia Estatal de Investigacion-through projects´ PID2020-115220RB-C22 and PID2021-128009OB-C33, in collaboration with the Signals and Images Laboratory (LSI) at the University of Oran, USTO-MB, Algeria.

REFERENCES

[1] S. Dargan and M. Kumar, "A Comprehensive Survey on Biometric Recognition Systems Based on Physiological and Behavioral Modalities," Expert Systems with Applications, vol. 143, p. 113114, 2020.

[2] A. Genovese, V. Piuri and F. Scotti, Touchless Palmprint Recognition Systems, Part of the Book Series: Advances in Information Security (ADIS), vol. 60, Springer, 2014.

[3] A. Serrano, I. M. de Diego, C. Conde and E. Cabello, "Recent Advances in Face Biometrics with Gabor Wavelets: A Review," Pattern Recognition Letters, vol. 31, no. 5, pp. 372–381, 2010.

[4] J. Qian, J. Yang, Y. Tai and H. Zheng, "Exploring Deep Gradient Information for Biometric Image Feature Representation," Neurocomputing, vol. 213, pp. 162–171, 2016.

[5] A. Vinay, C. A. Kumar, G. R. Shenoy, K. N. B. Murthy and S. Natarajan, "ORB-PCA Based Feature Extraction Technique for Face Recognition," Procedia Computer Science, vol. 58, pp. 614–621, 2015.

[6] R. Yadav, S. K. Singh and R. Yogi, "Biometric Network Security Enhancements through Deep Learning Techniques," Proc. of the 2023 IEEE Int. Conf. on ICT in Business Industry & Government (ICTBIG), pp. 1–6, Indore, India, 2023.

[7] S. Minaee, A. Abdolrashidi, H. Su, M. Bennamoun and D. Zhang, "Biometrics Recognition Using Deep Learning: A Survey," Artificial Intelligence Review, vol. 56, no. 8, pp. 8647–8695, 2023.

[8] H. Heidari and A. Chalechale, "Biometric Authentication Using Deep Learning Based on Multi-level Fusion of Finger-knuckle Print and Fingernail," Expert Systems with Appl., vol. 191, p. 116278, 2022.

[9] D. Zhong, X. Du and K. Zhong, "Decade Progress of Palmprint Recognition: A Brief Survey," Neurocomputing, vol. 328, pp. 16–28, 2019.

[10] S. Joshi, D. K. Verma, G. Saxena and A. Paraye, "Issues in Training Convolutional Neural Networks for Image Classification," Proc. of Advances in Computing and Data Sciences (ICACDS 2019), Part of the Book Series: Comm. in Computer and Information Science (CCIS), vol. 1046, pp. 282–293, 2019.

[11] Y. Wang, J. Li, M. Zhang and G. Xu, "Palm Vein Recognition in Few-shot Learning *via* Modified Siamese Network," Proc. of the 2024 4th Int. Conf. on Neural Networks, Information and Communication Engineering (NNICE), pp. 598–603, Guangzhou, China, 2024.

[12] N. Elaraby, S. Barakat and A. Rezk, "A Novel Siamese Network for Few/Zero-shot Handwritten Character Recognition," Computers, Materials & Continua, vol. 74, no. 1, pp. 1837-1854, 2023.

[13] A. Hussain, A. Ullah, A. Aslam and A. Khatoon, "A Modified Siamese Network for Facial Assimilation," WSEAS Transactions on Signal Processing, vol. 19, pp. 60–66, 2023.

[14] A. Singh, A. Kumar and S. Lukose, "Forensic Acoustic Applications of Siamese Neural Networks,"

Proc. of the 2024 3rd Int. Conf. for Innovation in Technology (INOCON), pp. 1–4, 2024.

[15] C. R. Kumar et al., "Face Recognition Using CNN and Siamese Network," Measurement: Sensors, vol. 27, p. 100800, 2023.

[16] E. AlShemmary and F. A. Ameen, "Siamese Network-based Palmprint Recognition," Journal of Kufa for Mathematics and Computer, vol. 10, no. 1, pp. 108–118, 2023.

[17] D. M. Aprilla et al., "Palmprint Recognition Using Xception, VGG16, ResNet50, MobileNet and EfficientNetB0," J. Media Inform, Budidarma, vol. 8, no. 2, pp. 1065–1076, 2024.

[18] F. Liao, F. Leng, Z. Yang and B. Zhang, "Multi-order Extension Codes for Palmprint Recognition," International Journal of Neural Systems, vol. 35, no. 8, p. 2550039, 2025.

[19] L. Yan, F. Wang, F. Leng and A. B. J. Teoh, "Toward Comprehensive and Effective Palmprint Reconstruction Attack," Pattern Recognition, vol. 155, p. 110655, 2024.

[20] H. Shao, S. Shi, X. Du, D. Zeng and D. Zhong, "Robust Palmprint Recognition *via* Multi-stage Noisy Label Selection and Correction," IEEE Trans. on Image Processing, vol. 34, pp. 4591– 4601, 2025.

[21] H. Shao, C. Liu, C. Li and D. Zhong, "Privacy-preserving Palmprint Recognition *via* Federated Metric Learning," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 878–891, 2023.

[22] H. Shao, Y. Zou, C. Liu, Q. Guo and D. Zhong, "Learning to Generalize Unseen Dataset for Cross-dataset Palmprint Recognition," IEEE Trans. on Inf. Forensics and Secu., vol. 19, pp. 3788–3799, 2024.

[23] K. Zhang et al., "Palmprint Recognition Based on Gating Mechanism and Adaptive Feature Fusion," Frontiers in Neurorobotics, vol. 17, p. 1203962, 2023.

[24] J. Q. Du et al., "Advancements in Image Recognition: A Siamese Network Approach," Information Dynamics and Applications, vol. 3, no. 2, pp. 89–103, 2024.

[25] F. Marattukalam et al., "N-shot Palm Vein Verification Using Siamese Networks," Proc. of the 2021 IEEE Int. Conf. of the Biometrics Special Interest Group, pp. 1–5, Darmstadt, Germany, 2021.

[26] V. Gurunathan et al., "Palm Vein Biometric System Using Siamese Neural Network," Proc. of the 2024 IEEE Int. Conf. on Science Technology Eng. and Manag. (ICSTEM), pp. 1–5, Coimbatore, India, 2024.

[27] D. Zhong, Y. Yang and X. Du, "Palmprint Recognition Using Siamese Network," Proc. of the 13th Chinese Conf. (CCBR 2018), pp. 48–55, DOI: 10.1007/978-3-319-97909-0_6, Urumqi, China, 2018.

[28] X. Du, D. Zhong and P. Li, "Low-shot Palmprint Recognition Based on Meta-Siamese Network," Proc. of the 2019 IEEE Int. Conf. on Multimedia and Expo (ICME), pp. 79–84, Shanghai, China, 2019.

[29] H. Shao et al., "Few-shot Learning for Palmprint Recognition *via* Meta-Siamese Network," IEEE Transactions on Instrumentation and Measurement, vol. 70, p. 5009812, 2021.

[30] C. Liu et al., "Siamese-hashing Network for Few-shot Palmprint Recognition," Proc. of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 3251–3258, Xiamen, China, 2019.

[31] S. Younus et al., "Palmprint Recognition Using Deep Convolutional Neural Networks," Proc. of the 2022 IEEE 2nd Int. Maghreb Meeting of the Conf. on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA), pp. 539–543, Sabratha, Libya, 2022.

[32] W. M. Cherif et al., "Active Contour Segmentation and CNN for Palmprint Recognition," Proc. of the 2022 2nd Int. Conf. on New Techn. of Inf. and Comm. (NTIC), pp. 1–6, Mila, Algeria, 2022.

[33] M. Ezz et al., "Improved Siamese Palmprint Authentication Using Pre-trained VGG16-palmprint," Computer Systems Science & Engineering, vol. 46, no. 2, pp. 2299-2317, 2023.

[34] R. Chen et al., "Research on Palmprint Recognition Based on Mechanism and Data," Proceedings of the Int. Conf. on Computer Vision and Deep Learning (CVDL'24), Article no. 66, pp. 1–9, DOI: 10.1145/3653804.3656264, 2024.

[35] A. A. Amini et al., "Using Dynamic Programming for Minimizing the Energy of Active Contours with Hard Constraints," Proc. of the 2nd Int. Conf. on Computer Vision, pp. 95–99, Tampa, USA, 1988.

[36] K. Ito et al., "Palm Region Extraction for Contactless Palmprint Recognition," Proc. of the 2015 IEEE Int. Conf. on Biometrics (ICB), pp. 334–340, Phuket, Thailand, 2015.

[37] T. Connie et al., "Automated Palmprint Recognition System," Image and Vision Computing, vol. 23, no. 5, pp. 501–515, 2005.

[38] V. W. Lumumba, "Comparative Analysis of Cross-validation Techniques: LOOCV, K-fold and Repeated K-fold in Machine Learning Models," American J. of Theoretical and Applied Statistics, vol. 13, no. 5, pp. 127-137, 2024.

[39] Zhang, Lin: "Contactless Palmprint Database," [Online], Available: https://cslinzhang.github.io/ ContactlessPalm/, Accessed: July 11, 2025.

[40] L. Zhang et al., "Towards Contactless Palmprint Recognition: A Novel Device, New Benchmark and Collaborative Representation Approach," Pattern Recognition, vol. 69, pp. 199–212, 2017.

[41] IIT Delhi Biometric Database, IIT Delhi Touchless Palmprint Database, [Online], Available: https://www4. comp.polyu.edu.hk/~csajaykr/IITD/Database_Palm.htm, Accessed: Sep. 1, 2024.

[42] A. Kumar, "Incorporating Cohort Information for Reliable Palmprint Authentication," Proc. of the 2008 IEEE 6th Indian Conf. on Computer Vision, Graphics & Image Processing (ICVGIP), pp. 583–590, Bhubaneswar, India, 2008.

[43] M. A. Ferrer et al., "Low-cost Multi-modal Biometric Identification Based on Hand Geometry and Texture," Proc. of the 2007 41st Annual IEEE Int. Carnahan Conf. on Security Technology, pp. 52–58,

Ottawa, Canada, 2007.
[44] V. Kanhangad, A. Kumar and D. Zhang, "Contactless and Pose-invariant Biometric Identification Using Hand Surface," IEEE Transactions on Image Processing, vol. 20, no. 5, pp. 1415–1424, 2010.
[45] D. Zhang, W. K. Kong, J. You and M. Wong, "Online Palmprint Identification," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 9, pp. 1041–1050, 2003.
[46] A. Gumaei et al., "Effective Palmprint Recognition for Visible and Multi-spectral Images," Sensors, vol. 18, no. 5, p. 1575, 2018.
[47] Z. Yang et al., "Multi-order Texture Features for Palmprint Recognition," Artificial Intelligence Review, vol. 56, no. 2, pp. 995–1011, 2023.
[48] L. Liang, T. Chen and L. Fei, "Orientation Space Code and Multi-feature Two-phase Sparse Representation for Palmprint Recognition," Int. Journal of Machine Learning and Cybernetics, vol. 11, pp. 1453–1461, 2020.
[49] T. Chai, S. Prasad and S. Wang, "Boosting Palmprint Identification with Gender Information Using DeepNet," Future Generation Computer Systems, vol. 99, pp. 41–53, 2019.
[50] C. Liu, D. Zhong and H. Shao, "Few-shot Palmprint Recognition Based on Similarity Metric Hashing Network," Neurocomputing, vol. 456, pp. 540–549, 2021.
[51] H. Shao and D. Zhong, "Towards Open-set Touchless Palmprint Recognition *via* Weight-based Meta Metric Learning," Pattern Recognition, vol. 121, p. 108247, 2022.
[52] Z. Yang et al., "Comprehensive Competition Mechanism in Palmprint Recognition," IEEE Transactions on Information Forensics and Security, vol. 18, pp. 5160–5170, 2023.
[53] Z. Yang et al., "CO3Net: Coordinate-aware Contrastive Competitive Neural Network for Palmprint Recognition," IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1–14, 2023.
[54] A. Fawzy, M. Ezz, S. Nouh and G. Tharwat, "Palmprint Recognition Using Siamese Network and Transfer Learning," International Journal of Advanced and Applied Sciences, vol. 9, no. 3, pp. 90–99, 2022.

ملخص البحث:

إنّ التّقدُّم في انظمة التّعلُّم العميق البيومترية، التي طُبّقت فيها طرق تمييز الأشخاص عن طريق الوجه واليد، يقود إلى تحسيناتٍ من حيث الأداء السّريع و سرّية البيانات. وإنّ التّعرُّف على بصمات الكف هو الموضوع الرّئيسي الّذي يتمّ التّركيز عليه في الطّريقة المقترحة في هذه الورقة، حيث يتمّ التّعامل مع مجموعات بياناتٍ أصغر حجماً من غيرها من مجموعات البيانات البيومترية. وتجدر الإشارة إلى أنّ نموذجاً ضخماً ومعقّداً من أنظمة التّعلُّم العميق قد يخسر شيئاً من مواءمته وقدرته على التّعميم عندما يُطبّق على هذا النّوع من البيانات.

تعالج هذه الورقة التّحدّيات آنفة الذّكر عن طريق تطبيق نموذج تعلُّم عميق ملائم للتّعرُّف على بصمات الكف الّتي تتميز بالتّنوُّع ومحدودية البيانات. في بادىء الأمر، يتمّ استخلاص "منطقة الاهتمام" باستخدام التّجزئة النّشطة الّتي تناسب التّعامل مع صعوبة الحصول على بصمات الكف من صُور اليد الّتي تحتوي على أصابع مُتقاربة أو متلاصقة. أمّا في المرحلة الثانية، فتُستخدم طريقة جديدة ومخصّصة مدفوعة بطريقة (LOOCV) بعد تعديلها ودمجها بشبكة تعلُّم عميقٍ سياميةٍ للتّحقُّق من بصمات الكف. وعلى النّقيض من طريقة (LOOCV) التّقليدية، فإنّ طريقتنا المعدّلة تعمل على تحسين الكلفة الحسابية في الوقت الّذي يتمّ فيه الحصول على تقييمٍ متوازنٍ على ثلاث مجموعات بيانات.

ويعمل إطار العمل المقترح على إثبات فاعلية أنظمة التّعرُّف على بصمات الكف الّتي تستخدم تقنياتٍ متقدّمة؛ فقد تمّ تحقيق دقّة مقدارها 99.75% في التّعرُّف على الأشخاص عبر بصمات كفوفهم، إلى جانب تحسين (تقليل) معدّل خطأ التساوي وتسريع زمن المطابقة، الأمر الذي يجعل النّموذج المقترح ملائماً للتّطبيقات الميدانية.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 04, December 2025.

POWER BEACON-ASSISTED ENERGY HARVESTING IN D2D NETWORK UNDER CO-CHANNEL INTERFERENCES: SYMBOL ERROR RATE ANALYSIS

Nguyen Quang Sang[1], Tran Cong Hung[2], Ngoc-Long Nguyen[3], Bui Vu Minh[4] and Lubos Rejfek[5]

(Received: 15-Aug.-2025, Revised: 15-Oct.-2025, Accepted: 1-Nov.-2025)

ABSTRACT

This paper studies the symbol error rate (SER) performance of a wireless-powered device-to-device (D2D) communication system operating under a time-switching (TS) protocol in the presence of multiple co-channel interferers (CCI). The considered model involves a battery-less source harvesting energy from multiple dedicated power beacons and transmitting to a multi-antenna destination over quasi-static Rayleigh fading channels. Both selection combining (SC) and maximal ratio combining (MRC) schemes are examined at the destination. In addition to the SER analysis, the outage probability (OP) performance is also investigated based on the derived cumulative distribution functions (CDFs), providing a complementary perspective on system reliability. The analysis focuses on the impact of key system parameters, including the interference power level, interferer-to-destination distance, energy harvesting efficiency, and modulation type, on the overall performance. Comprehensive simulation results are presented to validate the analytical derivations and to demonstrate the effects of these parameters on both SER and OP. The obtained results offer valuable insights into the design of energy-constrained D2D systems operating in spectrum-sharing environments, serving as a reference for future enhancements and practical deployments.

KEYWORDS

Co-channel interference, Energy harvesting, Device-to-device, Symbol error rate.

1. INTRODUCTION

The Internet of Things (IoT) has emerged as a groundbreaking paradigm, enabling seamless interconnection between physical objects, sensors, and digital systems, allowing them to collect, process, and exchange data with minimal human intervention [1,2,3]. With the rapid development of smart cities, intelligent transportation, environmental monitoring, and industrial automation, IoT networks have become an indispensable part of modern life. The evolution toward fifth-generation (5G) and upcoming sixth-generation (6G) wireless networks promises ultra-reliable communications, low latency, massive device connectivity, and ubiquitous coverage, further accelerating IoT adoption [4, 5, 6]. However, the large number of IoT devices poses a significant challenge - most of these devices are powered by batteries with limited energy-storage capacity [7]-[8]. In many applications, such as remote sensing, underground monitoring, or post-disaster recovery scenarios, replacing or recharging batteries is impractical, costly, or even impossible. Therefore, ensuring long-term network operation without manual intervention has become a key research direction.

In this context, Wireless Power Transfer (WPT) has emerged as a promising technology to address the challenge of limited battery life in future wireless networks, especially for the large-scale deployment of low-power IoT devices. By leveraging Energy Harvesting (EH) techniques, wireless network nodes can capture energy from ambient radio frequency (RF) signals or dedicated power sources, thereby enabling sustainable operation without frequent battery replacement [9], [10]. This has motivated a large body of research focused on analyzing, evaluating, and optimizing the performance of EH-enabled systems in various communication scenarios. Several works have examined specific network models

1. N. Q. Sang is with the Posts and Telecommunications Institute of Technology, Ho Chi Minh City, Vietnam. Email: sangnq@ptit.edu.vn
2. T. C. Hung is with the School of Computer Science & Engineering, The SaiGon International University, Ho Chi Minh City, Vietnam. Email: tranconghung@siu.edu.vn
3. N.-L. Nguyen (Corresponding Author) is with the Faculty of Applied Sciences, Ton Duc Thang University, Ho Chi Minh City, Vietnam. Email: nguyenngoclong@tdtu.edu.vn
4. B. V. Minh is with the Faculty of Engineering and Technology, Nguyen Tat Thanh University, Ho Chi Minh City, Vietnam. Email: bvminh@ntt.edu.vn
5. L. Rejfek is with the Faculty of Electrical Engineering and Informatics, University of Pardubice, 53210 Pardubice, Czech Republic. Email: Lubos.Rejfek@upce.cz

supported by dedicated power beacons. For instance, [9] analyzes the outage performance of a symbiotic radio network assisted by a power beacon, thereby clarifying the impact of energy harvesting on data-transmission capability. Similarly, [10] investigates physical layer security in IoT networks with multiple power beacons, evaluating the secrecy outage probability (SOP) under different energy configurations. For large-scale IoT networks, [11] employs stochastic geometry to analyze the coverage probability of EH-enabled LoRa networks, shedding light on the effects of node density, spatial distribution, and signal strength on connectivity. Beyond terrestrial networks, EH has also been integrated into unmanned aerial vehicle (UAV) communications, where energy supply is a severe limiting factor. [12] reveals the inherent trade-off between reliability and security in UAV systems supported by EH relays, providing design guidelines to balance these two objectives. In the field of non-orthogonal multiple access (NOMA), [13] analyzes the uplink and downlink performance of EH-enabled NOMA systems, providing exact expressions for throughput and outage probability (OP). Extending further, [14] proposes a statistical model for the sum of double random variables and applies it to optimize the performance of NOMA systems assisted by simultaneous transmitting and reflecting reconfigurable intelligent surfaces (STAR-RIS), demonstrating their capability to enhance EH system efficiency and flexibility. Furthermore, physical layer security has also been a central focus in EH research. The work in [15] introduces a friendly jammer to improve security in wireless sensor networks while analyzing the SOP. Similarly, [16] investigates a self-energy recycling model in full-duplex decode-and-forward (DF) relay networks, jointly evaluating security and reliability. In another direction, [17] investigates the performance of an EH full-duplex relay system employing multi-antenna techniques and cooperative diversity. It analyzes key metrics, such as outage probability and bit error rate under various fading conditions using transmit-antenna selection, maximal-ratio combining, and power splitting schemes. Moreover, [18] provides exact and upper-bound capacity analysis for full-duplex DF EH networks with a hybrid time power switching relaying (TPSR) protocol, establishing the theoretical foundation for protocol optimization. In addition to long-packet systems, [19] focuses on short-packet communication (SPC) in EH-enabled IoT networks, where latency and reliability become critical determinants of system performance. With its outstanding benefits in energy sustainability, performance enhancement, and security support, EH is becoming an indispensable component in the design and optimization of modern wireless networks.

In addition to sustaining long-term operation, IoT and 5G/6G networks also demand communication mechanisms that offer flexibility, spectral efficiency, and high reliability. Among these, Device-to-Device (D2D) communication is a key enabling technology, allowing nearby user pairs to connect directly without routing through a base station (BS). This mechanism reduces latency, improves spectral efficiency, alleviates cellular network load, and enhances both energy efficiency and system throughput [20], [21]. Depending on spectrum usage, D2D can operate in in-band mode - sharing licensed spectrum with cellular users - or out-band mode - utilizing unlicensed spectrum [20], [21]. Among these, in-band underlay D2D has been extensively studied due to its efficient spectrum reuse, although it requires strict control over interference caused to cellular networks. For example, [22] analyzes the OP, average rate, and amount of fading (AoF) of underlay D2D networks under three different power-allocation strategies, showing that path-loss-based allocation outperforms equal or random allocation. For energy-constrained devices, EH has been integrated into D2D as a promising solution. [23] proposes a D2D model supported by a power beacon and cooperative jamming from multiple nodes to enhance physical layer security, providing closed-form expressions for OP, intercept probability (IP), and SOP. Moreover, [24] combines partial NOMA with backscatter communication (BackCom) to improve both energy and spectral efficiency in D2D, deriving closed-form OP expressions over Rayleigh fading channels. Relay-aided D2D has also received strong attention for its ability to extend coverage and improve reliability in both one-way relaying (OWR) and two-way relaying (TWR) modes [25]. Research shows that resource allocation, relay selection, and power optimization should be integrated with EH and machine-learning (ML) algorithms to achieve superior performance [25], [26]. In resource optimization, methods such as bee-colony optimization [27] or distributed resource allocation based on reinforcement learning (RL) [26] have proven effective in improving throughput, spectral efficiency, and fairness. Meanwhile, security remains a major challenge due to the direct connectivity of D2D. [28] provides a comprehensive analysis of security threats, such as eavesdropping, spoofing, and jamming attacks, while proposing a security architecture for next-generation D2D systems. In the IoT context, [29] presents a multi-criteria learning algorithm using security sensors to maintain data reliability and integrity in smart environments with D2D support. The potential of D2D is further enhanced when combined with emerging

technologies, such as reconfigurable intelligent surfaces (RIS) and NOMA [30], which enable optimization of the propagation environment and improvement of link quality, while analyzing OP under imperfect interference-cancellation conditions.

However, both D2D and EH systems deployed in high-spectrum reuse environments must contend with co-channel interference (CCI), one of the main sources of interference in wireless communication systems.

CCI arises when multiple links share the same frequency channel, which can significantly degrade received-signal quality, leading to higher error rates, reduced throughput and compromised system reliability. With the increasing network density and aggressive frequency reuse in technologies, such as 5G, IoT, and satellite networks, analyzing and mitigating the impact of CCI has become an important research direction to ensure optimal system performance. Recent works have investigated the presence of CCI in various wireless communication scenarios. In [31], CCI at the relay node is considered in a cooperative SPC system with transmit-antenna selection and beamforming, analyzing the block error rate (BLER) and proposing an optimal power allocation strategy. [32] extends the SPC analysis to single-hop systems with CCI at the destination, providing exact and asymptotic closed-form expressions for BLER. In the IoT domain, [35] investigates a two-way relaying NOMA (TWR-NOMA) system with a power beacon, analyzing the effects of CCI on OP, throughput, and ergodic capacity and incorporating optimization and deep-learning techniques to improve performance. In addition, [33] analyzed the performance of EH-enabled D2D systems under co-channel interference, while [34] investigated the outage probability and error rate of wireless-powered communication networks operating in interference-limited environments, emphasizing the impact of energy-harvesting efficiency and interference power on system reliability.

In summary, with the rapid development of advanced wireless communication techniques, such as EH, diversity combining, and interference management, the performance of many current systems has been significantly improved in terms of reliability, energy efficiency, and interference resilience. Numerous recent studies have contributed to this progress by investigating various network architectures and operating conditions. For instance, several recent works have provided new insights into energy-harvesting (EH) and interference-limited systems under different scenarios. Specifically, [45] investigated the physical layer security of EH-enabled IoT networks with hardware impairments, while [46] analyzed outage and throughput performance in backscatter-assisted SWIPT systems. [47] presented a secure and covert communication framework for energy-harvesting relay IoT networks, while [48] examined the joint optimization of resource allocation and energy efficiency in STAR-RIS-assisted networks. In addition, [49] explored the outage behavior of NOMA-enabled UAV systems under imperfect channel conditions, whereas [50] analyzed short-packet transmissions for EH-based IoT systems considering reliability-latency trade-offs. Meanwhile, [35] proposed and evaluated a PB- and NOMA-assisted cooperative IoT network under CCI, while [36] analyzed the outage performance of satellite-terrestrial full-duplex relaying networks with CCI and further applied deep learning for performance prediction. [37] examined the outage probability of an EH-based cooperative NOMA network with a direct link, whereas [38] presented performance analysis and optimal design of a time-switching EH protocol for MIMO full-duplex DF relay networks employing different diversity techniques. Similarly, [39] exploited the direct link in two-way half-duplex sensor networks over block Rayleigh fading to derive an upper bound of the ergodic capacity and provide an exact SER analysis. The impact of CCI has also been investigated in other contexts, such as IRS-assisted communications [40], multisource cooperative networks assisted by UAV relays [41] and dual-hop mixed RF/FSO relaying systems with both CCI and pointing errors [42]. Moreover, [43] addressed short-packet communications for relay systems with CCI at the relay, offering performance analysis and power-control strategies, while [44] explored the second-order statistics for IRS-assisted multi-user vehicular networks with CCI. However, despite these advancements, our literature survey reveals a lack of comprehensive studies that evaluate the performance of systems simultaneously integrating PB-assisted EH, multi-antenna diversity reception and the effects of CCI-particularly when combining both selection combining (SC) and maximal ratio combining (MRC) techniques. Motivated by this research gap, we propose a unified system model in which an energy-constrained source node harvests energy from a PB to transmit information to a multi-antenna destination in the presence of multiple CCI sources. Furthermore, to provide a more comprehensive evaluation, the study focuses on analyzing and computing the symbol error rate (SER) under both SC and MRC techniques, thereby offering deeper

insights into the impact of interference and the effectiveness of each diversity-reception method. In the context of increasingly scarce spectrum resources and the inevitable interference from existing wireless systems, analyzing and optimizing system performance under such conditions holds both scientific significance and practical value.

The main contributions of this paper can be summarized as follows:

- This work investigates the impact of co-channel interference (CCI) on the performance of wireless-powered D2D systems, where a batteryless source harvests energy from dedicated power beacons (PBs) and utilizes this harvested energy to communicate with a multi-antenna destination in the presence of multiple interferers.
- The analysis includes both the symbol error rate (SER) and the outage probability (OP) performance under two receive combining techniques at the destination; namely, selection combining (SC) and maximal ratio combining (MRC). New closed-form expressions for both SER and OP are derived, capturing the effects of multiple interference sources and various key system parameters.

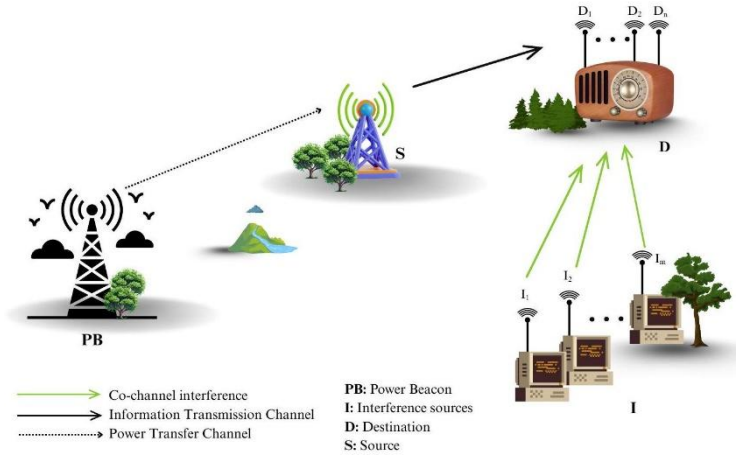


Figure 1. System model.

- Extensive numerical and Monte Carlo simulation results are provided to validate the analytical derivations. The results offer valuable design insights into the trade-offs between reliability and energy efficiency, revealing, for instance, the performance advantage of MRC over SC, the influence of energy-harvesting efficiency on both SER and OP and the sensitivity of system performance to interference power levels.

Table 1. Comparison of the uniqueness of our research to related articles.

Context	EH	SC	MRC	Multi-antenna	CCI	SER
Paper [17]	✓	✓	✓	✓		✓
Paper [35]	✓				✓	
Paper [37]	✓	✓	✓			
Paper [38]	✓		✓	✓		
Paper [39]	✓					✓
Paper [40]				✓	✓	
Paper [41]		✓	✓		✓	
Paper [42]		✓	✓		✓	✓
Paper [43]		✓	✓		✓	
Paper [44]				✓	✓	
This paper	✓	✓	✓	✓	✓	✓

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 04, December 2025.

The remainder of the paper is organized as follows. Section 2 gives an overview of the system model. Section 3 presents the information-theoretic mathematical framework, guiding on how to achieve the SER. Section 4 presents numerical results and discussions to validate the developed framework as well as deeply explore the impacts of system key parameters, while Section 5 provides concluding remarks.

2. SYSTEM MODEL

We consider a wireless-powered D2D communication system, as illustrated in Fig. 1, where a batteryless source node S communicates with a multi-antenna destination node D in the presence of multiple co-channel interferers. The source harvests energy from a dedicated power beacon (P) during the EH phase and then reuses this harvested energy to transmit information to the destination in the information-transmission phase, following the time-switching (TS) protocol depicted in Fig. 2. The system operates over a quasi-static flat Rayleigh fading environment, where channel coefficients remain constant during each transmission block, but vary independently between blocks. In addition to the intended signal, the destination also receives undesired signals from M interfering transmitters operating on the same frequency band, which cause CCI. Such a setting captures a realistic scenario in spectrum-constrained environments, where D2D communications coexist with other wireless systems and must operate under both energy limitations and interference conditions.

Let us denote $h_{\mathrm{PS}}, h_{\mathrm{SD}_n}, h_{\mathrm{I}_m\mathrm{D}}$ as channel coefficients of the direct link from source node P to destination node D, and $\mathrm{P} \to \mathrm{S}, \mathrm{S} \to \mathrm{D}_n, \mathrm{I}_m \to \mathrm{D}$, links, respectively. Assume that $h_{\mathrm{X}}, \mathrm{X} \in \{\mathrm{PS}, \mathrm{SD}_n, \mathrm{I}_m\mathrm{D}\}$ are Rayleigh fading channels, channel gains $\gamma_{\mathrm{X}} = |h_{\mathrm{X}}|^2$ are exponential random variables (RVs) whose probability density function (PDF) and cumulative distribution function (CDF) are given as, respectively.

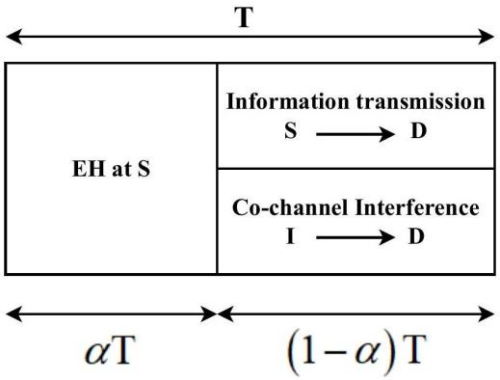


Figure 2. Time switching.

$$f_{\gamma_{\mathrm{X}}}(x) = \lambda_{\mathrm{X}} \exp(-\lambda_{\mathrm{X}} x)$$
$$F_{\gamma_{\mathrm{X}}}(x) = 1 - \exp(-\lambda_{\mathrm{X}} x) \tag{1}$$

where λ_{X} is the mean of γ_{X}. To consider a simple-path loss model, λ_{X} can be modeled by $\lambda_{\mathrm{X}} = (d_{\mathrm{X}})^{\chi}$, where d_{X} is the distance between two correspondence nodes and χ is the path-loss exponent. In the energy-harvesting phase, firstly S will harvest the energy from P and then, energy at S can be expressed as:

$$\mathrm{E}_{\mathrm{S}} = \eta \alpha \mathrm{T} P_{\mathrm{P}} \gamma_{\mathrm{PS}} \tag{2}$$

Then, the transmit power of S can be formulated as:

$$\mathrm{P}_{\mathrm{S}} = \frac{\mathrm{E}_{\mathrm{S}}}{(1-\alpha)\mathrm{T}} = \frac{\eta \alpha P_{\mathrm{P}} \gamma_{\mathrm{PS}}}{(1-\alpha)} = \kappa P_{\mathrm{P}} \gamma_{\mathrm{PS}} \tag{3}$$

where, $\kappa = \frac{\eta\alpha}{(1-\alpha)}$

In the data-transmission phase, S transmits unit power signals x_{S} to the nth D; i.e., n^{th}, where $\mathrm{E}\{|x_{\mathrm{S}}|^2\} = 1$ is the expectation operator.

The received signal at n^{th} D is given as follows:

$$y_{\mathrm{D}_n} = \underbrace{\sqrt{P_{\mathrm{S}}} h_{\mathrm{SD}_n} x_{\mathrm{S}}}_{\text{signal}} + \underbrace{\sqrt{P_{\mathrm{I}}} \sum_{m=1}^{M} h_{\mathrm{I}_m \mathrm{D}} x_{\mathrm{I}_m}}_{\text{interference}} + \underbrace{n_{\mathrm{D}_n}}_{\text{noise}} \tag{4}$$

where n_{D_n} is the AWGN with zero mean and variance N_0. The received signal-to- noise ratio (SNR) at the n^{th} D in this phase can be thus calculated by:

$$\gamma_{\mathrm{D}_n} = \sqrt{\frac{\mathrm{E}\{|\text{ signal}|^2\}}{\mathrm{E}\{|\text{ interference}|^2\} + \mathrm{E}\{|\text{ noise}|^2\}}} = \frac{P_{\mathrm{S}} \gamma_{\mathrm{SD}_n}}{P_{\mathrm{I}} \gamma_{\mathrm{ID}} + N_0} \tag{5}$$

where $\gamma_{\mathrm{ID}} = \sum_{m=1}^{M} \left| h_{\mathrm{I}_m \mathrm{D}} \right|^2$.

Using the fact that $N_0 \ll P_{\mathrm{I}}$, then by doing some algebra, by substituting (3) into (5), we have:

$$\gamma_{\mathrm{D}_n} = \frac{\kappa P_{\mathrm{P}} \gamma_{\mathrm{PS}} \gamma_{\mathrm{SD}_n}}{P_{\mathrm{I}} \gamma_{\mathrm{ID}} + N_0} \approx \frac{\kappa \Psi_{\mathrm{P}} \gamma_{\mathrm{PS}} \gamma_{\mathrm{SD}_n}}{\Psi_{\mathrm{I}} \gamma_{\mathrm{ID}}} \tag{6}$$

where $\Psi_{\mathrm{P}} = \frac{P_{\mathrm{I}}}{N_0}; \Psi_{\mathrm{I}} = \frac{P_{\mathrm{I}}}{N_0}$. In order to make the paper highly applicable, we examine two situations using various diversity technique:

1. Scenario 1: In the first considered scenario, the destination node D employs the SC technique to process the received signals from its multiple antennas[51]. Specifically, D selects the antenna branch with the highest instantaneous signal-to-interference-plus-noise ratio (SINR) for detection, while discarding the other branches. Therefore, the resulting SINR at D can be expressed as:

$$\gamma_{\mathrm{D}}^{\mathrm{SC}} = \frac{\kappa \Psi_{\mathrm{P}} \gamma_{\mathrm{PS}} \gamma_{\mathrm{SD}}^{\mathrm{SC}}}{\Psi_{\mathrm{I}} \gamma_{\mathrm{ID}}} \tag{7}$$

   where $\gamma_{\mathrm{SD}}^{\mathrm{SC}} = \max_{1 \leqslant n \leqslant N} \left[\left| h_{\mathrm{SD}_n} \right|^2 \right]$.

2. Scenario 2: In the second scenario, the destination node D adopts the MRC technique to exploit all available antenna branches [52]. In this method, the received signals at different antennas are coherently combined after being weighted according to their respective channel gains, thereby maximizing the overall SINR at D. Therefore, the resulting SINR at D can be expressed as:

$$\gamma_{\mathrm{D}}^{\mathrm{MRC}} = \frac{\kappa \Psi_{\mathrm{P}} \gamma_{\mathrm{PS}} \gamma_{\mathrm{SD}}^{\mathrm{MRC}}}{\Psi_{\mathrm{I}} \gamma_{\mathrm{ID}}} \tag{8}$$

   where $\gamma_{\mathrm{SD}}^{\mathrm{MRC}} = \sum_{n=1}^{N} \left| h_{\mathrm{SD}_n} \right|^2$.

3. PERFORMANCE ANALYSIS

3.1 CDF and PDF Derivation

3.1.1 CDF and PDF of $\gamma_{\mathrm{SD}}^{\mathrm{SC}}, \gamma_{\mathrm{SD}}^{\mathrm{MRC}}$

Based on [53], the CDF and PDF can be found as, respectively:

$$\begin{aligned} F_{\gamma_{\mathrm{SD}}\mathrm{C}}(x) &= 1 - \sum_{n=1}^{N} \binom{N}{n} (-1)^{n-1} \exp(-n\lambda_{\mathrm{SD}} x), \\ f_{\gamma_{\mathrm{SD}}^{\mathrm{SC}}}(x) &= \sum_{n=1}^{N} \binom{N}{n} \times n\lambda_{\mathrm{SD}} (-1)^{n-1} \exp(-n\lambda_{\mathrm{SD}} x), \end{aligned} \tag{9}$$

$$\begin{aligned} F_{\xi}(x) &= \frac{1}{\Gamma(N)} \gamma(N, \lambda_k x) \\ f_{\xi}(x) &= \frac{(\lambda_{\mathrm{SD}})^N}{\Gamma(N)} x^{N-1} \exp(-\lambda_k x) \end{aligned} \tag{10}$$

where $\Gamma(\cdot)$ is the Gamma function, as defined in section (8.31) of reference [54], $\xi \in \left\{ \gamma_{\mathrm{SD}}^{\mathrm{MRC}}, \gamma_{\mathrm{ID}} \right\}$, and $k \in \{\mathrm{SD}, \mathrm{ID}\}$.

3.1.2 CDF of γ_D^{SC}

$$F_{\gamma_{\mathrm{D}}^{\mathrm{SC}}}(x) = \Pr\left(\gamma_{\mathrm{D}}^{\mathrm{SC}} < x\right) \tag{11}$$

By substituting (7) into (11), we obtain:

$$\begin{aligned} F_{\gamma_{\mathrm{D}}^{\mathrm{SC}}}(x) &= \Pr\left(\frac{\kappa\Psi_{\mathrm{P}}\gamma_{\mathrm{PS}}\gamma_{\mathrm{SD}}^{\mathrm{SC}}}{\Psi_{\mathrm{I}}\gamma_{\mathrm{ID}}} < x\right) \\ &= \int_0^{+\infty}\int_0^{+\infty} F_{\gamma_{\mathrm{PS}}}\left(\frac{x\Psi_{\mathrm{I}}z}{\kappa\Psi_{\mathrm{P}}y}\right) f_{\gamma_{\mathrm{SD}}^{\mathrm{SC}}}(y) f_{\gamma_{\mathrm{ID}}}(z)\,dydz \end{aligned} \tag{12}$$

By substituting (9) into (12), we obtain:

$$\begin{aligned} F_{\gamma_{\mathrm{D}}^{\mathrm{SC}}}(x) &= \int_0^{+\infty}\int_0^{+\infty} F_{\gamma_{\mathrm{PS}}}\left(\frac{x\Psi_{\mathrm{I}}z}{\kappa\Psi_{\mathrm{P}}y}\right) f_{\gamma_{\mathrm{SD}}^{\mathrm{SC}}}(y) f_{\gamma_{\mathrm{ID}}}(z)\,dydz \\ &= 1 - \int_0^{+\infty}\int_0^{+\infty} \left\{ \begin{array}{c} \exp\left(\frac{-\lambda_{\mathrm{PS}}x\Psi_{\mathrm{I}}z}{\kappa\Psi_{\mathrm{P}}y}\right)\sum_{n=1}^{N}\binom{N}{n}\times n\lambda_{\mathrm{SD}}(-1)^{n-1}\exp(-n\lambda_{\mathrm{SD}}y) \\ \times\frac{(\lambda_{\mathrm{ID}})^M}{\Gamma(M)}z^{M-1}\exp(-\lambda_{\mathrm{ID}}z) \end{array} \right\} dydz \end{aligned} \tag{13}$$

Based on [54] [Eq: 3.324 and 6.643-3], (13) can be figured out as:

$$\begin{aligned} F_{\gamma_{\mathrm{D}}^{\mathrm{SC}}}(x) &= 1 - \int_0^{+\infty}\int_0^{+\infty} \left(\begin{array}{c} \exp\left(\frac{-\lambda_{\mathrm{PS}}x\Psi_{\mathrm{I}}z}{\kappa\Psi_{\mathrm{P}}y}\right)\sum_{n=1}^{N}\binom{N}{n}\times n\lambda_{\mathrm{SD}}(-1)^{n-1}\exp(-n\lambda_{\mathrm{SD}}y) \\ \times\frac{(\lambda_{\mathrm{ID}})^M}{\Gamma(M)}z^{M-1}\exp(-\lambda_{\mathrm{ID}}z) \end{array} \right) dydz \\ &= 1 - \sum_{n=1}^{N}\binom{N}{n}\Gamma(M+1)(-1)^{n-1}\exp\left(\frac{n\lambda_{\mathrm{PS}}\lambda_{\mathrm{SD}}x\Psi_{\mathrm{I}}}{2\kappa\Psi_{\mathrm{P}}\lambda_{\mathrm{ID}}}\right)\times W_{-M,\frac{1}{2}}\left(\frac{n\lambda_{\mathrm{PS}}\lambda_{\mathrm{SD}}x\Psi_{\mathrm{I}}}{\kappa\Psi_{\mathrm{P}}\lambda_{\mathrm{ID}}}\right) \end{aligned} \tag{14}$$

where $W(\cdot)$ is the Whittaker function, as defined in section. (9.22) of reference [54].

3.1.3 CDF of γ_D^{MRC}

$$F_{\gamma_{\mathrm{D}}^{\mathrm{MRC}}}(x) = \Pr\left(\gamma_{\mathrm{D}}^{\mathrm{MRC}} < x\right) \tag{15}$$

By substituting (8) into (15), we obtain:

$$\begin{aligned} F_{\gamma_{\mathrm{D}}^{\mathrm{MRC}}}(x) &= \Pr\left(\frac{\kappa\Psi_{\mathrm{P}}\gamma_{\mathrm{PS}}\gamma_{\mathrm{SD}}^{\mathrm{MRC}}}{\Psi_{\mathrm{I}}\gamma_{\mathrm{ID}}} < x\right) \\ &= \int_0^{+\infty}\int_0^{+\infty} F_{\gamma_{\mathrm{PS}}}\left(\frac{x\Psi_{\mathrm{I}}z}{\kappa\Psi_{\mathrm{P}}y}\right) f_{\gamma_{\mathrm{SD}}^{\mathrm{MRC}}}(y) f_{\gamma_{\mathrm{ID}}}(z)\,dydz \end{aligned} \tag{16}$$

By substituting (1) and (10) into (16), we claim:

$$\begin{aligned} F_{\gamma_{\mathrm{D}}^{\mathrm{MRC}}}(x) &= \int_0^{+\infty}\int_0^{+\infty} F_{\gamma_{\mathrm{PS}}}\left(\frac{x\Psi_{\mathrm{I}}z}{\kappa\Psi_{\mathrm{P}}y}\right) f_{\gamma_{\mathrm{SD}}^{\mathrm{MRC}}}(y) f_{\gamma_{\mathrm{ID}}}(z)\,dydz \\ &= 1 - \int_0^{+\infty} \int_0^{+\infty} \left\{ \begin{array}{c} \exp\left(\frac{-\lambda_{\mathrm{PS}}x\Psi_{\mathrm{I}}z}{\kappa\Psi_{\mathrm{P}}y}\right)\frac{(\lambda_{\mathrm{SD}})^N}{\Gamma(N)}y^{N-1}\exp(-\lambda_{\mathrm{SD}}y) \\ \times\frac{(\lambda_{\mathrm{ID}})^M}{\Gamma(M)}z^{M-1}\exp(-\lambda_{\mathrm{ID}}z) \end{array} \right\} dydz \end{aligned} \tag{17}$$

Based on [54] Eq: 3.471-9 and Eq: 6.643-3, (17) can be figured out as:

$$\begin{aligned} F_{\gamma_{\mathrm{D}}^{\mathrm{MRC}}}(x) = 1 - \frac{\Gamma(M+N)}{\Gamma(N)}\left(\frac{\lambda_{\mathrm{SD}}\lambda_{\mathrm{PS}}x\Psi_{\mathrm{I}}}{\kappa\Psi_{\mathrm{P}}\lambda_{\mathrm{ID}}}\right)^{\frac{N-1}{2}}\times\exp\left(\frac{\lambda_{\mathrm{SD}}\lambda_{\mathrm{PS}}x\Psi_{\mathrm{I}}}{2\kappa\Psi_{\mathrm{P}}\lambda_{\mathrm{ID}}}\right) \\ \times W_{-M-\frac{N}{2}+\frac{1}{2},\frac{N}{2}}\left(\frac{\lambda_{\mathrm{SD}}\lambda_{\mathrm{PS}}x\Psi_{\mathrm{I}}}{\kappa\Psi_{\mathrm{P}}\lambda_{\mathrm{ID}}}\right) \end{aligned} \tag{18}$$

3.2 Outage-probability (OP) Analysis

The OP of the system can be thus defined by:

$$\mathrm{OP}^{\varpi} = \begin{cases} \Pr\left(\gamma_{\mathrm{D}}^{\mathrm{SC}} < \gamma_{\mathrm{th}}\right), \varpi = \mathrm{SC} \\ \Pr\left(\gamma_{\mathrm{D}}^{\mathrm{MRC}} < \gamma_{\mathrm{th}}\right), \varpi = \mathrm{MRC} \end{cases} \tag{19}$$

where $\gamma_{\mathrm{th}} = 2^{\mathrm{R_{th}}} - 1$ is the threshold of the system and $\mathrm{R_{th}}$ is the target rate.

3.2.1 Scenario 1: SC Technique Is Employed at the Destination

The OP for Scenario 1 is given as follows:

$$\mathrm{OP}^{\mathrm{SC}} = \Pr\left(\gamma_{\mathrm{D}}^{\mathrm{SC}} < \gamma_{\mathrm{th}}\right) \tag{20}$$

Based on (11) and (14) and by substituting $x = \gamma_{\mathrm{th}}$ into (20), we have:

$$\mathrm{OP}^{\mathrm{SC}} = 1 - \sum_{n=1}^{N} \binom{N}{n} \Gamma(M+1)(-1)^{n-1} \exp\left(\frac{n\lambda_{\mathrm{PS}}\lambda_{\mathrm{SD}}\gamma_{\mathrm{th}}\Psi_{\mathrm{I}}}{2\kappa\Psi_{\mathrm{P}}\lambda_{\mathrm{ID}}}\right) \times W_{-M,\frac{1}{2}}\left(\frac{n\lambda_{\mathrm{PS}}\lambda_{\mathrm{SD}}\gamma_{\mathrm{th}}\Psi_{\mathrm{I}}}{\kappa\Psi_{\mathrm{P}}\lambda_{\mathrm{ID}}}\right) \tag{21}$$

3.2.2 Scenario 2: MRC Technique Is Employed at the Destination

The OP for Scenario 2 is given as follows:

$$\mathrm{OP}^{\mathrm{MRC}} = \Pr\left(\gamma_{\mathrm{D}}^{\mathrm{MRC}} < \gamma_{\mathrm{th}}\right) \tag{22}$$

Based on (15) and (18) and by substituting $x = \gamma_{\mathrm{th}}$ into (22), we have:

$$\begin{aligned} \mathrm{OP}^{\mathrm{MRC}} = 1 - \frac{\Gamma(M+N)}{\Gamma(N)} \left(\frac{\lambda_{\mathrm{SD}}\lambda_{\mathrm{PS}}\gamma_{\mathrm{th}}\Psi_{\mathrm{I}}}{\kappa\Psi_{\mathrm{P}}\lambda_{\mathrm{ID}}}\right)^{\frac{N-1}{2}} \times \exp\left(\frac{\lambda_{\mathrm{SD}}\lambda_{\mathrm{PS}}\gamma_{\mathrm{th}}\Psi_{\mathrm{I}}}{2\kappa\Psi_{\mathrm{P}}\lambda_{\mathrm{ID}}}\right) \\ \times W_{-M-\frac{N}{2}+\frac{1}{2},\frac{N}{2}}\left(\frac{\lambda_{\mathrm{SD}}\lambda_{\mathrm{PS}}\gamma_{\mathrm{th}}\Psi_{\mathrm{I}}}{\kappa\Psi_{\mathrm{P}}\lambda_{\mathrm{ID}}}\right) \end{aligned} \tag{23}$$

3.3 Symbol Error Ratio (SER) Analysis

Based on [39], SER can be defined as:

$$\mathrm{SER} = \mathrm{E}\left\{aQ\left(\sqrt{2b\gamma_{\mathrm{D}}^{\zeta}}\right)\right\} \tag{24}$$

where $\zeta \in (\mathrm{SC}, \mathrm{MRC}), Q(t) = \frac{1}{\sqrt{2\pi}} \int_t^{+\infty} e^{-x^2/2} dx$ is the Gaussian Q-function [55], while a and b are constants, which are specific for each modulation; $(a, b) = (1,1)$ for binary phase-shift keying (BPSK) and $(a, b) = (1,2)$ for Quadrature Phase Shift Keying (QPSK). As a result, before obtaining the SER performance, the CDF of $\gamma_{\mathrm{D}}^{\zeta}$ is adopted. Then, Equation (24) can be reformulated as follows:

$$\mathrm{SER} = \frac{a\sqrt{b}}{2\sqrt{\pi}} \int_0^{+\infty} \frac{e^{-bx}}{\sqrt{x}} \times F_{\gamma_{\mathrm{D}}^{\zeta}}(x) dx \tag{25}$$

3.3.1 Scenario 1: SC Technique Is Employed at the Destination

In this scenario, by substituting (14) into (25), the SER can be analyzed as:

$$\begin{aligned} \mathrm{SER}_{\mathrm{D}}^{\mathrm{SC}} &= \frac{a\sqrt{b}}{2\sqrt{\pi}} \int_0^{+\infty} \frac{e^{-bx}}{\sqrt{x}} \times \left\{ \begin{matrix} 1 - \sum_{n=1}^{N} \binom{N}{n} \Gamma(M+1)(-1)^{n-1} \\ \exp\left(\frac{n\lambda_{\mathrm{PS}}\lambda_{\mathrm{SD}}x\Psi_{\mathrm{I}}}{2\kappa\Psi_{\mathrm{P}}\lambda_{\mathrm{ID}}}\right) \times W_{-M,\frac{1}{2}}\left(\frac{n\lambda_{\mathrm{PS}}\lambda_{\mathrm{SD}}x\Psi_{\mathrm{I}}}{\kappa\Psi_{\mathrm{P}}\lambda_{\mathrm{ID}}}\right) \end{matrix} \right\} dx \\ &= \underbrace{\frac{a\sqrt{b}}{2\sqrt{\pi}} \int_0^{+\infty} \underbrace{\frac{e^{-bx}}{\sqrt{x}}}_{\Phi_1} dx}_{} - \underbrace{\frac{a\sqrt{b}}{2\sqrt{\pi}} \sum_{n=1}^{N} \binom{N}{n} \Gamma(M+1)(-1)^{n-1} \underbrace{\int_0^{+\infty} \underbrace{\left(\frac{1}{\sqrt{x}} \exp\left(\frac{n\lambda_{\mathrm{PS}}\lambda_{\mathrm{SD}}x\Psi_{\mathrm{I}}}{2\kappa\Psi_{\mathrm{P}}\lambda_{\mathrm{ID}}} - bx\right) \times W_{-M,\frac{1}{2}}\left(\frac{n\lambda_{\mathrm{PS}}\lambda_{\mathrm{SD}}x\Psi_{\mathrm{I}}}{\kappa\Psi_{\mathrm{P}}\lambda_{\mathrm{ID}}}\right)\right)}_{} dx}_{\Phi_2}}_{} \end{aligned} \tag{26}$$

From (26), and after applying [54][Eq: 3.361.2], Φ_1 can be calculated as:

$$\Phi_1 = \frac{a\sqrt{b}}{2\sqrt{\pi}} \int_0^{+\infty} \frac{e^{-bx}}{\sqrt{x}}\, dx = \frac{a}{2} \tag{27}$$

Next, Φ_2 is expressed by:

$$\begin{aligned} \Phi_2 &= \frac{a\sqrt{b}}{2\sqrt{\pi}} \sum_{n=1}^{N} \binom{N}{n} \Gamma(M+1)(-1)^{n-1} \int_0^{+\infty} \left(\begin{matrix} \frac{1}{\sqrt{x}} \exp\left(\frac{n\lambda_{PS}\lambda_{SD}x\Psi_I}{2\kappa\Psi_P\lambda_{ID}} - bx\right) \\ \times W_{-M,\frac{1}{2}}\left(\frac{n\lambda_{PS}\lambda_{SD}x\Psi_I}{\kappa\Psi_P\lambda_{ID}}\right) \end{matrix} \right) dx \\ &= \frac{a\sqrt{b}}{2\sqrt{\pi}} \sum_{n=1}^{N} \binom{N}{n} \Gamma(M+1)(-1)^{n-1} \int_0^{+\infty} \left(\begin{matrix} x^{-1/2} \exp\left[-x\left(b - \frac{n\lambda_{PS}\lambda_{SD}\Psi_I}{2\kappa\Psi_P\lambda_{ID}}\right)\right] \\ \times W_{-M,\frac{1}{2}}\left(\frac{n\lambda_{PS}\lambda_{SD}x\Psi_I}{\kappa\Psi_P\lambda_{ID}}\right) \end{matrix} \right) dx \end{aligned} \tag{28}$$

With the help of Equation [54][Eq: 7.621.3], we have:

$$\Phi_2 = \frac{a}{2b\sqrt{\pi}} \sum_{n=1}^{N} \binom{N}{n} \frac{(-1)^{n-1}\Gamma(M+1)\Gamma\left(\frac{3}{2}\right)\Gamma\left(\frac{1}{2}\right)}{\Gamma\left(M+\frac{3}{2}\right)} \times {}_2F_1\left(\frac{3}{2}, M+1; M+\frac{3}{2}; \frac{b - \frac{n\lambda_{PS}\lambda_{SD}\Psi_I}{\kappa\Psi_P\lambda_{ID}}}{b}\right), \tag{29}$$

where ${}_2F_1(\alpha, \beta; \gamma; z)$ is the Gauss hyper-geometric function, as defined in section (9.18) of reference [54]. Finally, by alternating (27) and (29) into (26), SER_D^{SC} can be obtained as:

$$\mathrm{SER}_D^{SC} = \frac{a}{2} - \frac{a}{2b\sqrt{\pi}} \sum_{n=1}^{N} \left(\begin{matrix} \binom{N}{n} \frac{(-1)^{n-1}\Gamma(M+1)\Gamma\left(\frac{3}{2}\right)\Gamma\left(\frac{1}{2}\right) n\lambda_{PS}\lambda_{SD}\Psi_I}{\Gamma\left(M+\frac{3}{2}\right)\kappa\Psi_P\lambda_{ID}} \\ \times {}_2F_1\left(\frac{3}{2}, M+1; M+\frac{3}{2}; \frac{b - \frac{n\lambda_{PS}\lambda_{SD}\Psi_I}{\kappa\Psi_P\lambda_{ID}}}{b}\right) \end{matrix} \right) \tag{30}$$

3.3.2 Scenario 2: MRC Technique Will Be Applied.

By replacing (18) into (25), we get:

$$\begin{aligned} \mathrm{SER}_D^{MRC} &= \frac{a\sqrt{b}}{2\sqrt{\pi}} \int_0^{+\infty} \frac{e^{-bx}}{\sqrt{x}} \times \left\{ \begin{matrix} 1 - \frac{\Gamma(M+N)}{\Gamma(N)} \left(\frac{\lambda_{SD}\lambda_{PS}x\Psi_I}{\kappa\Psi_P\lambda_{ID}}\right)^{\frac{N-1}{2}} \times \exp\left(\frac{\lambda_{SD}\lambda_{PS}x\Psi_I}{2\kappa\Psi_P\lambda_{ID}}\right) \\ \times W_{-M-\frac{N}{2}+\frac{1}{2},\frac{N}{2}}\left(\frac{\lambda_{SD}\lambda_{PS}x\Psi_I}{\kappa\Psi_P\lambda_{ID}}\right) \end{matrix} \right\} dx \\ &= \frac{a}{2} - \frac{a\sqrt{b}}{2\sqrt{\pi}} \times \frac{\Gamma(M+N)}{\Gamma(N)} \left(\frac{\lambda_{SD}\lambda_{PS}\Psi_I}{\kappa\Psi_P\lambda_{ID}}\right)^{\frac{N-1}{2}} \int_0^{+\infty} \left(\begin{matrix} x^{\frac{N}{2}-1} \exp\left[-x\left(b - \frac{\lambda_{SD}\lambda_{PS}\Psi_I}{2\kappa\Psi_P\lambda_{ID}}\right)\right] \\ \times W_{-M-\frac{N}{2}+\frac{1}{2},\frac{N}{2}}\left(\frac{\lambda_{SD}\lambda_{PS}x\Psi_I}{\kappa\Psi_P\lambda_{ID}}\right) dx \end{matrix} \right) \end{aligned} \tag{31}$$

By the same approach to claim Φ_2, Equation (31) can be derived by:

$$\begin{aligned} \mathrm{SER}_D^{MRC} &= \frac{a}{2} - \frac{a}{2b^N\sqrt{\pi}} \times \left(\frac{\lambda_{SD}\lambda_{PS}\Psi_I}{\kappa\Psi_P\lambda_{ID}}\right)^N \frac{\Gamma(M+N)}{\Gamma(N)} \frac{\Gamma\left(N+\frac{1}{2}\right)\Gamma\left(\frac{1}{2}\right)}{\Gamma\left(M+N+\frac{1}{2}\right)} \\ &\quad \times {}_2F_1\left(N+\frac{1}{2}, M+N; M+N+\frac{1}{2}; \frac{b - \frac{\lambda_{SD}\lambda_{PS}\Psi_I}{\kappa\Psi_P\lambda_{ID}}}{b}\right). \end{aligned} \tag{32}$$

4. NUMERICAL RESULTS

In this section, we employ the Monte Carlo simulation method to provide numerical results that both validate the accuracy of the proposed analytical frameworks and offer deeper insights into the SER behavior under various key system parameters. The simulation settings and corresponding values are summarized in Table 2.

Figures 3 and 4 illustrate the outage-probability (OP) performance of the proposed system under different values of the transmit SNR Ψ_P and the interference power Ψ_I, respectively, for both SC and MRC combining schemes and two target rates R_{th}. In Figure 3, the OP is observed to decrease monotonically as Ψ_P increases.

Table 2. Simulation parameters.

Symbol	Parameter name	Value
R_{th}	Target rate	0.5, 1 (bit/s/Hz)
η	EH efficiency	0.05 to 0.95
α	Time-switching ratio	0.05 to 0.95
d_{PS}	Distance between P and S	1.5 m
d_{SD}	Distance between S and D	2 m
d_{ID}	Distance between I and D	0.5 to 5 m
χ	Path-loss exponent	2.2
Ψ_P	Transmit power-to-noise ratio at P	0 to 50(dB)
Ψ_I	Transmit power-to-noise ratio at I	0 to 35(dB)
M	Number of antennas at I	1, 10
N	Number of antennas at D	1, 10

This is because a higher transmit SNR improves the received-signal strength, thereby enhancing the achievable data rate and reducing the probability that the instantaneous rate falls below the target threshold R_{th}. Moreover, for the same Ψ_P, the system with a larger R_{th} exhibits a higher OP. This is attributed to the fact that the threshold SNR required for successful decoding, denoted by $\gamma_{th} = 2^{R_{th}} - 1$, increases exponentially with R_{th}. Consequently, the condition $\log_2(1+\gamma) < R_{th}$ (or equivalently $\gamma < \gamma_{th}$) becomes more likely to occur, leading to a higher outage probability. In Figure 4, the OP behavior is examined with respect to the interference power Ψ_1. As expected, the OP increases as Ψ_1 grows, because stronger interference deteriorates the signal-to-interference-plus-noise ratio (SINR), thereby reducing the achievable rate. Similarly, a higher R_{th} results in a larger OP under the same interference level due to stricter SINR requirements. In all cases, the MRC scheme consistently outperforms the SC scheme, owing to its ability to combine multiple received signals and achieve higher diversity gain, thus improving system robustness against fading and interference. Furthermore, the analytical curves closely match the Monte Carlo simulation results, confirming the accuracy and reliability of the derived theoretical expressions.

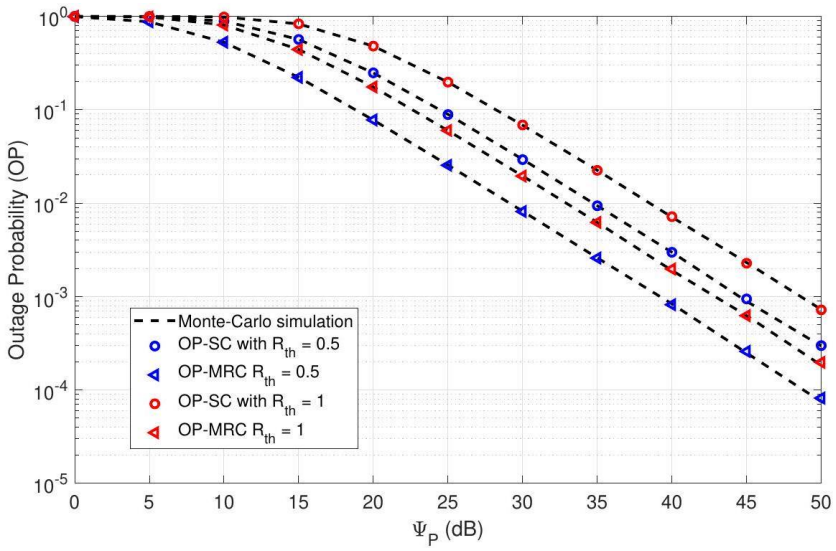


Figure 3. The OP of the proposed system *versus* $\Psi_P[dB]$ with different R_{th} .

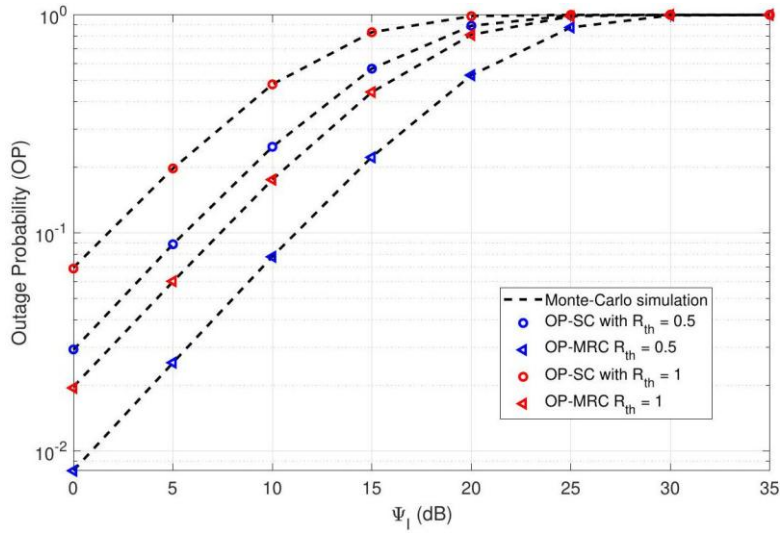


Figure 4. The OP of the proposed system *versus* Ψ_{I}[dB] with different R_{th}.

Figure 5 illustrates the simulated SER *versus* the Power Beacon transmit power Ψ_{P} for different modulation schemes (BPSK, QPSK), combining techniques (SC, MRC), and numbers of receive antennas N at the destination D. The close match between Monte Carlo simulations and analytical results validates the accuracy of the proposed model. As Ψ_{P} increases, the SER decreases, because the source S harvests more wireless energy from the PB, leading to higher transmit power and improved SNR at D. In the low-to-medium Ψ_{P} region, the SER reduction is relatively slow, since the system performance is still dominated by CCI. When Ψ_{P} is sufficiently large, the SER continues to decline, but the marginal improvement becomes smaller if CCI is not mitigated. Comparing SC and MRC, the results show that MRC consistently outperforms SC by achieving better array gain through coherent SNR combining, resulting in downward/leftward-shifted SER curves for the same Ψ_{P}. Increasing the number of receive antennas from $N = 2$ to $N = 10$ further shifts the curves downward and leftward due to higher diversity gain, with the performance improvement being more significant for MRC. Interestingly, under the given system configuration and normalization, QPSK achieves lower SER than BPSK over the entire Ψ_{P} range. Therefore, for applications requiring both high reliability and high throughput, QPSK combined with MRC and a large N is a promising option. On the other hand, when Ψ_{P} is limited or CCI is severe, increasing N, employing MRC and/or applying interference-mitigation strategies are effective to avoid the SER floor.

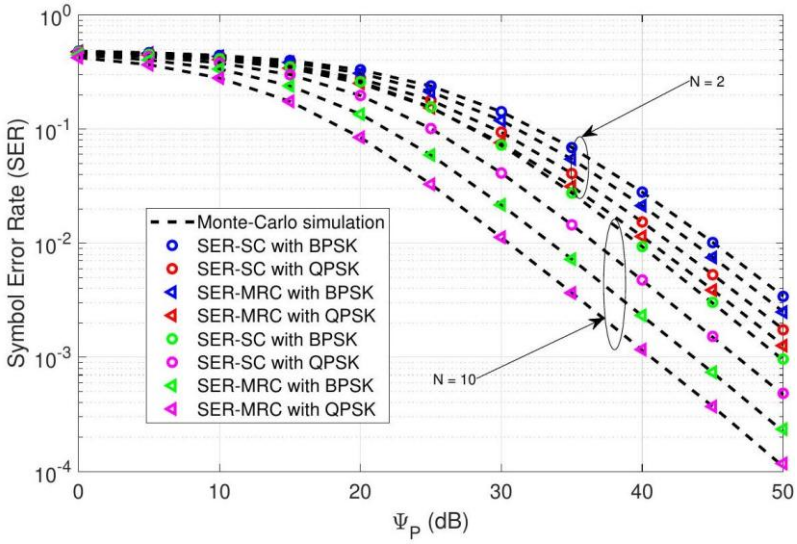


Figure 5. The SER of the proposed system *versus* Ψ_{P}[dB] with different N.

Figure 6 illustrates the simulated SER versus the interference power Ψ_{I} at the interfering nodes I for different modulation schemes and combining techniques. As Ψ_I increases, the SER of all configurations rises significantly, because stronger interference reduces the effective SNR at the destination D. In the

low Ψ_1 region, the SER increases relatively slowly, since the signal power from S is still sufficient to suppress interference; however, when Ψ_I exceeds a medium threshold (around $15-20$ dB), the curves start to converge and approach a high SER level, indicating an interference-limited regime where further increasing the transmit power of S or improving energy-harvesting efficiency yields little improvement. The relative performance trends between SC and MRC, as well as between QPSK and BPSK, follow the same pattern observed in Figure 5: MRC outperforms SC due to better array gain and QPSK achieves lower SER than BPSK across the entire Ψ_I range.

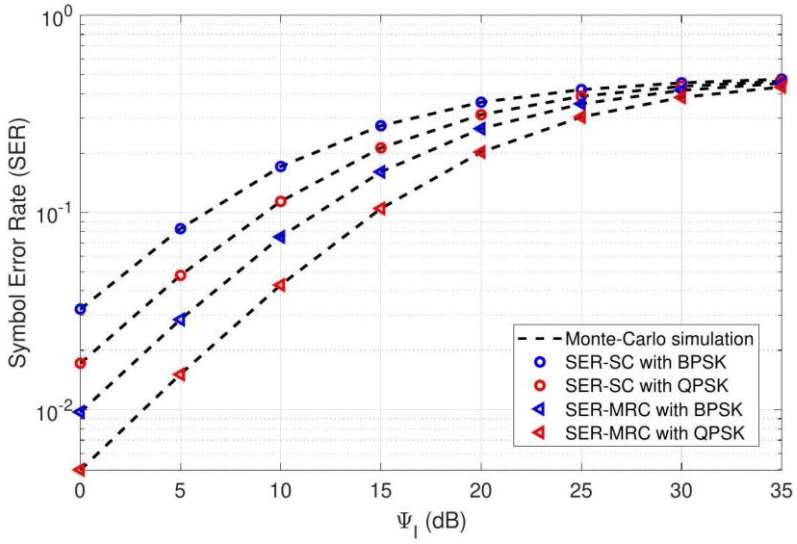


Figure 6. The SER of the proposed system *versus* Ψ_I[dB].

Figure 7 illustrates the simulated SER *versus* the distance d_{ID} from the interferers I to the destination D for different modulation schemes, combining techniques and numbers of interferers M. Based on a simple-path loss model, the mean channel gain can be expressed as $\lambda_X = (d_X)^{\chi}$, where d_X denotes the distance between two corresponding nodes and χ is the path loss exponent. Accordingly, as d_{ID} increases, the path-loss term $\lambda_{\mathrm{ID}} = (d_{\mathrm{ID}})^{\chi}$ grows, which causes the interference power received at D to decrease sharply. This reduction in interference directly improves the effective signal-to-interference-plus-noise ratio (SINR) in (30) and (32), resulting in a smaller SER. When d_{ID} is small, the interferers are located close to D and the strong CCI dominates the received signal, leading to a high SER. As d_{ID} increases, the SER decreases rapidly before gradually flattening out when interference becomes negligible. The impact of the number of interferers M is also evident: when $M = 1$, the SER is significantly lower compared to $M = 10$ for the same d_{ID}, since fewer interferers contribute less aggregate interference power.

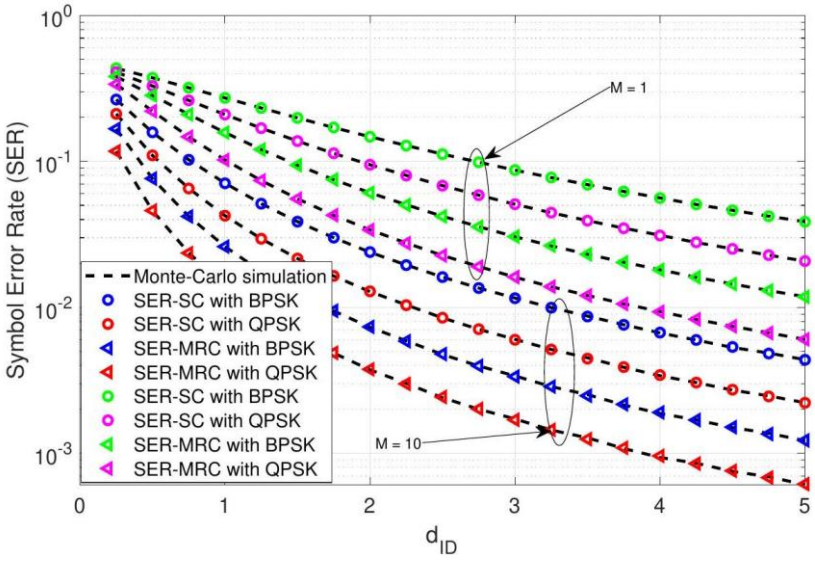


Figure 7. The SER of the proposed system *versus* d_{ID} with different M.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 04, December 2025.

The relative performance trends between SC and MRC, as well as between QPSK and BPSK, remain consistent with previous figures-MRC consistently outperforms SC due to its array gain, while QPSK achieves better SER performance than BPSK, offering improved reliability and spectral efficiency even under interference-limited conditions. These findings highlight the importance of interference management and diversity-reception techniques in maintaining link quality for wireless-powered D2D networks.

Figure 8 presents the simulated SER *versus* the energy harvesting efficiency η for different modulation schemes and diversity combining techniques. As η increases, the SER decreases for all scenarios, because a higher harvesting efficiency allows the relay to collect more energy from the received signals, leading to higher transmit power in the information transmission phase and thus improving the end-to-end SNR. Specifically, increasing η directly enhances the parameter κ in the SINR expressions at the destination, which strengthens the received signal component and consequently reduces the overall SER. The performance gap between SC and MRC remains consistent with previous figures-MRC outperforms SC due to its ability to coherently combine signals from multiple antennas, providing a higher array gain. Similarly, QPSK achieves lower SER compared to BPSK in all cases, as already discussed in earlier results.

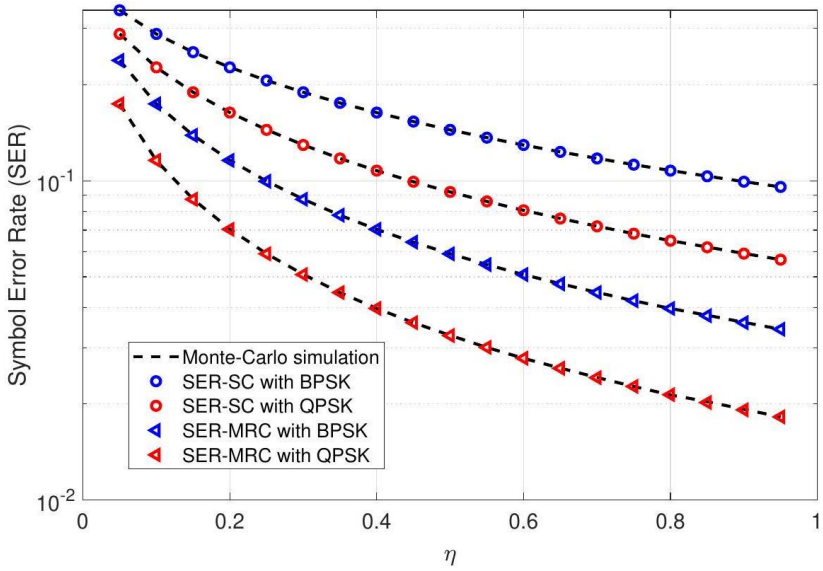


Figure 8. The SER of the proposed system *versus* η.

5. CONCLUSIONS

This work analyzed the SER performance of a wireless-powered D2D communication system with multiple co-channel interferers under a time-switching protocol. Simulation results revealed several key findings. First, MRC consistently outperforms SC due to its coherent combining capability, offering a notable SER reduction across all scenarios. Second, QPSK modulation achieves lower SER compared to BPSK, indicating its advantage in spectral efficiency while maintaining robustness. Third, increasing the energy-harvesting efficiency significantly improves SER, especially at low-to-moderate η values, while greater distances between interferers and the destination lead to substantial interference mitigation. Finally, the system demonstrates high sensitivity to the interference power level, emphasizing the importance of interference-management strategies in energy-constrained D2D networks. Future work could extend the current framework to scenarios with non-identical (i.n.i.d.) power levels and spatial distributions of interferers, providing a more realistic characterization of interference patterns. Moreover, future directions include investigating multiple power beacons with optimized beamforming, integrating ambient backscatter communication to further reduce energy demands, and considering hybrid relay-assisted D2D architectures. In addition, exploring adaptive modulation, interference alignment and machine learning-based resource allocation could enhance system resilience under dynamic spectrum-sharing environments.

REFERENCES

[1] S. Li, L. D. Xu and S. Zhao, "The Internet of Things: A Survey," Information Systems Frontiers, vol. 17, no. 2, pp. 243-259, 2015.
[2] K. Rose, S. Eldridge, L. Chapin et al., "The Internet of Things: An Overview," The Internet Society (ISOC), vol. 80, no. 15, pp. 1-53, 2015.
[3] X. Cui, "The Internet of Things," Proc. of Ethical Ripples of Creativity and Innovation, pp. 61-68, Springer, 2016.
[4] D. C. Nguyen et al., "6G Internet of Things: A Comprehensive Survey," IEEE Internet of Things Journal, vol. 9, no. 1, pp. 359-383, Jan. 2022.
[5] J. H. Kim, "6G and Internet of Things: A Survey," Journal of Management Analytics, vol. 8, no. 2, pp. 316-332, 2021.
[6] G. Gkagkas, D. J. Vergados, A. Michalas and M. Dossis, "The Advantage of the 5G Network for Enhancing the Internet of Things and the Evolution of the 6G Network," Sensors, vol. 24, no. 8, Art. no. 2455, 2024.
[7] S. C. Mukhopadhyay and N. K. Suryadevara, "Internet of Things: Challenges and Opportunities," Proc. of Internet of Things: Challenges and Opportunities, pp. 1-17, S. C. Mukhopadhyay, Ed. Cham: Springer International Publishing, 2014.
[8] Y.-K. Chen, "Challenges and Opportunities of Internet of Things," Proc. of the 17th Asia and South Pacific Design Automation Conf. (ASP-DAC), pp. 383-388, Sydney, Australia, 2012.
[9] T. C. Hung, B. V. Minh, T. N. Nguyen and M. Voznak, "Power Beacon-assisted Energy Harvesting Symbiotic Radio Networks: Outage Performance," PLOS ONE, vol. 20, no. 2, pp. 1-16, Feb. 2025.
[10] T. C. Hung, Q. S. Nguyen, B. V. Minh, T. T.-Quyen and N. L. Nguyen, "Multi-power Beacon Empowered Secure in IoT Networks: Secrecy Outage Probability Analysis," Advances in Electrical and Electronic Engineering, vol. 23, no. 2, pp. 91-97, 2025.
[11] T. T.-H. Nguyen, T. N. Nguyen, T. T. Duy, N. H. Son, T. Hanh, B. V. Minh and L.-T. Tu, "Coverage Probability of Energy Harvesting Enabled LoRa Networks with Stochastic Geometry," Journal of Information and Telecommunication, vol. 8, no. 2, pp. 262-279, 2024.
[12] T. N. Nguyen et al., "On the Dilemma of Reliability or Security in Unmanned Aerial Vehicle Communications Assisted by Energy Harvesting Relaying," IEEE Journal on Selected Areas in Communications, vol. 42, no. 1, pp. 52-67, Jan. 2024.
[13] D. H. Thuan et al., "Uplink and Downlink of Energy Harvesting NOMA System: Performance Analysis," Journal of Information and Telecommunication, vol. 8, no. 1, pp. 92-107, 2024.
[14] B. V. Minh, P. T. Tran, T.-H. T. Pham, A.-T. Le, S.-P. Le and P. Partila, "Statistics of the Sum of Double Random Variables and Their Applications in Performance Analysis and Optimization of Simultaneously Transmitting and Reflecting Reconfigurable Intelligent Surface-assisted Non-orthogonal Multi-access Systems," Sensors, vol. 24, no. 18, Art. no. 6148, 2024.
[15] B. V. Minh, N. H. K. Nhan, T.-H. T. Pham and M. Tran, "Physical Layer Security in Wireless Sensors Networks with Friendly Jammer: Secrecy Outage Probability Analysis," Advances in Electrical and Electronic Engineering, vol. 22, no. 4, pp. 387-398, 2024.
[16] B. V. Minh, A.-V. Le, V.-D. Phan and T.-H. T. Pham, "Self-energy Recycling in DF Full-duplex Relay Network: Security-reliability Analysis," Advances in Electrical and Electronic Engineering, vol. 22, no. 1, pp. 86-96, 2024.
[17] H. Balaban and O. Kucur, "Performance Analysis of Energy-harvesting Full-duplex Relaying with Multi-antenna and Cooperative Diversity," Proc. of the 2023 3rd Int. Conf. on Mobile Networks and Wireless Communications (ICMNWC), pp. 1-6, Tumkur, India, 2023.
[18] P. T. Tin, V.-D. Phan, T. N. Nguyen and L. A. Vu, "Performance Analysis for Exact and Upper Bound Capacity in DF Energy Harvesting Full-duplex with Hybrid TPSR Protocol," Journal of Electrical and Computer Engineering, vol. 2021, Art. no. 6610107, 2021.
[19] D.-T. Vo et al., "Short Packet Communication in IoT Networks: Performance Analysis," Journal of Information and Telecommunication, pp. 1-14, DOI: 10.1080/24751839.2025.2487353, 2025.
[20] M. S. M. Gismalla et al., "Survey on Device to Device (D2D) Communication for 5GB/6G Networks: Concept, Applications, Challenges and Future Directions," IEEE Access, vol. 10, pp. 30792-30821, 2022.
[21] N. O. Nwazor and V. K. Ugah, "Device-to-Device (D2D) Data Communications in 5G Networks," International Journal of Advances in Eng. and Manag. (IJAEM), vol. 4, no. 1, pp. 1151-1154, Jan. 2022.
[22] T. N. Nguyen et al., "On the Performance of Underlay Device-to-Device Communications," Sensors, vol. 22, no. 4, Art. no. 1456, 2022.
[23] N. T. Nguyen, P. Fazio and M. Voznak, "On the Performance of Power Beacon-assisted D2D Communications in the Presence of Multi-jammers and Eavesdropper," Journal of Advanced Engineering and Computation, vol. 5, no. 4, pp. 254-264, 2021.
[24] T.-H. T. Pham, N.-T. T. Nguyen, Q.-S. Nguyen, T. Hon, B. V. Minh, Q. S. Nguyen and M. Tran, "Performance Analysis in D2D Partial NOMA-assisted Backscatter Communication," 2025.

[25] M. R. B. Salim, "A Survey on Essential Challenges in Relay-aided D2D Communication for Next Generation Cellular Networks," Journal of Network and Computer Applications, vol. 216, Art. no. 103657, 2023.

[26] S. Jayakumar and S. Nandakumar, "Reinforcement Learning Based Distributed Resource Allocation Technique in Device-to-Device (D2D) Communication," Wireless Networks, vol. 29, pp. 1843-1858, 2023.

[27] W. H. Mahdi and N. Taşpinar, "Bee System-based Self Configurable Optimized Resource Allocation Technique in Device-to-Device (D2D) Communication Networks," IEEE Access, vol. 12, pp. 3039-3053, 2024.

[28] A. Khan and R. Das, "Security Aspects of Device-to-Device (D2D) Networks in Wireless Communication: A Comprehensive Survey," Telecommunication Systems, vol. 81, pp. 625-642, 2022.

[29] K. Haseeb, A. Rehman, T. Saba, S. A. Bahaj and J. Lloret, "Device-to-Device (D2D) Multi-criteria Learning Algorithm Using Secured Sensors," Sensors, vol. 22, no. 6, Art. no. 2115, 2022.

[30] V.-D. Le et al., "Enabling D2D Transmission Mode of Reconfigurable Intelligent Surfaces Aided in Wireless NOMA System," Advances in Electrical and Electronic Eng., vol. 23, no. 1, pp. 32-42, 2025.

[31] Q.-S. Nguyen, A. U. Le, T. N. Nguyen, T.-T. Nguyen and M. Voznak, "Short Packet Communications for Relay Systems with Co-channel Interference at Relay: Performance Analysis and Power Control," IEEE Access, vol. 12, pp. 63452-63461, 2024.

[32] L. A. U. Vu, N. T. Tung, T. T. Duy, T. L. Thanh, T. N. Nguyen and N. Q. Sang, "Performance Evaluation of Short-packet Communications of Single-hop System with Presence of Co-channel Interference," Proc. of the 7th Int. Conf. on Research in Intelligent and Computing in Engineering (RICE 2022), pp. 267-271, Hung Yen City, Nov. 2022.

[33] N. Q. Sang, T. C. Hung, T. T. Duy, M. Tran and B. S. Kim, "Securing Wireless Communications with Energy Harvesting and Multi-antenna Diversity", Jordanian Journal of Computers and Information Technology (JJCIT), vol. 11, no. 2, pp. 197-210, June 2025.

[34] R. Alrawashdeh, "A Review on Wireless Power Transfer in Free Space and Conducting Lossy Media", Jordanian Journal of Computers and Information Technol. (JJCIT), vol. 3, no. 2, pp. 71-88, August 2017.

[35] A.-T. Le et al., "Power Beacon and NOMA-assisted Cooperative IoT Networks with Co-channel Interference: Performance Analysis and Deep Learning Evaluation," IEEE Transactions on Mobile Computing, vol. 23, no. 6, pp. 7270-7283, 2024.

[36] T. N. Nguyen et al., "Outage Performance of Satellite Terrestrial Full-duplex Relaying Networks with Co-channel Interference," IEEE Wireless Comm. Letters, vol. 11, no. 7, pp. 1478-1482, 2022.

[37] H. Huang et al., "Outage Probability of Energy Harvesting Cooperative NOMA Network with Direct Link," J. King Saud Univ. Comput. Inf. Sci., vol. 37, Art. no. 90, 2025.

[38] M. Hoang, B. C. Nguyen, N. N. Thang, M. Tran and P. T. Tran, "Performance and Optimal Analysis of Time-switching Energy Harvesting Protocol for MIMO Full-duplex Decode-and-Forward Wireless Relay Networks with Various Transmitter and Receiver Diversity Techniques," Journal of the Franklin Institute, vol. 357, no. 17, pp. 13205-13230, 2020.

[39] P. T. Tin, T. N. Nguyen, M. Tran, T. T. Trang and L. Sevcik, "Exploiting Direct Link in Two-way Half-duplex Sensor Network over Block Rayleigh Fading Channel: Upper Bound Ergodic Capacity and Exact SER Analysis," Sensors, vol. 20, no. 4, Art. no. 1165, 2020.

[40] H. Wen, A. M. T. Khel and K. A. Hamdi, "Effects of Co-channel Interference on the Performance of IRS-assisted Communications," IEEE Trans. on Vehicular Tech., vol. 73, no. 7, pp. 10075-10089, Jul. 2024.

[41] H. Huang, Y. Wei, L. Liang, Z. Yin and N. Zhang, "On the Analysis of Multisource Cooperative Network Assisted by UAV Relays with Co-channel Interference," IEEE Journal on Miniaturization for Air and Space Systems, vol. 6, no. 2, pp. 144-156, Apr. 2025.

[42] J. Ding, D. Kang, X. Xie, L. Wang, L. Tan and J. Ma, "Joint Effects of Co-channel Interferences and Pointing Errors on Dual-hop Mixed RF/FSO Fixed-gain and Variable-gain Relaying Systems," IEEE Photonics Journal, vol. 15, no. 1, pp. 1-11, Feb. 2023.

[43] Q.-S. Nguyen, U.-V. Le Anh, T. N. Nguyen, T.-T. Nguyen and M. Voznak, "Short Packet Communications for Relay Systems with Co-channel Interference at Relay: Performance Analysis and Power Control," IEEE Access, vol. 12, pp. 63452-63461, 2024.

[44] A. Girdher et al., "Second-order Statistics for IRS-assisted Multiuser Vehicular Network With Co-channel Interference," IEEE Trans. on Intelligent Vehicles, vol. 8, no. 2, pp. 1800-1812, Feb. 2023.

[45] S. Ghose et al., "Jointly Optimal RIS Placement and Power Allocation for Underlay D2D Communications: An Outage Probability Minimization Approach," IEEE Transactions on Cognitive Communications and Networking, vol. 10, no. 2, pp. 622-633, April 2024.

[46] Z. Li, J. Xing and J. Hu, "Outage Performance of SWIPT-D2D-based Hybrid Satellite-Terrestrial Networks," Sensors, vol. 25, no. 8, Art. no. 2393, 2025.

[47] Y. Wang, L. Feng, S. Yao, H. Liang, H. Shi and Y. Chen, "Outage Probability Analysis for D2D-enabled Heterogeneous Cellular Networks with Exclusion Zone: A Stochastic Geometry Approach," CMES-Computer Modeling in Eng. and Sciences, vol. 138, no. 1, pp. 639-661, 2023.

[48] D.-W. Lim and J.-M. Kang, "Joint Transmit Power and Power-splitting Optimization for SWIPT in D2D-enabled Cellular Networks with Energy Cooperation," Mathematics, vol. 13, no. 3, Art. no. 389, 2025.
[49] R. Nagarajan and N. M. V. Mohamad, "Energy Efficient Rresource and Power Allocation for Uplink Underlay D2D Communication in HetNet-based 5G Network," Journal of Wireless Com. Network, vol. 2025, Art. no. 24, DOI: 10.1186/s13638-025-02452-1, 2025.
[50] M. Z. Islam and M. N. Adnan, "A Robust Resource Allocation Method for Energy Efficient Device to Device (D2D) Communication," ICCK Transactions on Mobile and Wireless Intelligence, vol. 1, no. 1, pp. 32-39, DOI: 10.62762/TMWI.2025.764788, 2025.
[51] T. N. Nguyen, P. T. Tran and M. Voznak, "Wireless Energy Harvesting Meets Receiver Diversity: A Successful Approach for Two-way Half-duplex Relay Networks over Block Rayleigh Fading Channel," Computer Networks, vol. 172, p. 107176, DOI: 10.1016/j.comnet.2020.107176, 2020.
[52] T. N. Nguyen et al., "On Performance of RIS-aided Bidirectional Full-duplex Systems with Combining of Imperfect Conditions," Wireless Netw., vol. 30, pp. 649-660, 2024.
[53] B. V. Minh, T. N. Nguyen and L.-T. Tu, "Physical Layer Security in Wireless Sensors Networks: Secrecy Outage Probability Analysis," J. of Inf. and Telecomm., vol. 9, no. 1, pp. 1-23, 2025.
[54] I. S. Gradshteyn and I. M. Ryzhik, Table of Integrals, Series, and Products, ISBN-13: 978-0122947506, Academic Press, 2014.
[55] D. T. Tam et al., "SER Performance of Millimeter-wave Communications with Multiple Reconfigurable Intelligent Surfaces and Transmit Antenna Selection," AEU-Int. J. of Electronics and Communications, vol. 160, Art. no. 154517, 2023.

ملخص البحث:

تبحث هذه الورقة في الأداء المتعلّق بمعدّل خطأ الرّموز في نظامٍ مُشغَّلٍ لاسلكياً للاتّصال من جهازٍ إلى جهازٍ يعمل تحت بروتوكول تبديل الوقت في ظلّ وجود تداخلات من عدّة قنوات. ويتضمّن النّموذج المقترح مصدراً يعمل بلا بطّارية يقوم بحصاد الطّاقة من منارات طاقةٍ متعدّدةٍ ومن ثمّ يُرسل البيانات إلى هدفٍ متعدّد الهوائيات.

وإلى جانب تحليل معدّل خطأ الرّموز، يتمّ البحث في الأداء المتعلّق باحتمالية انقطاع الطّاقة، بناءً على دوالّ التّوزيع التّراكمي المشتقّة؛ من أجل إضافة منظورٍ آخر في ما يرتبط بموثوقية النّظام. ويركّز التّحليل على آثار عددٍ من المتغيرات الأساسية للنّظام، بما فيها مُستوى طاقة التّداخل، وبُعْد مصدر التّداخل عن الهدف، وفعالية حصادِ الطّاقة، ونوع التّعديل، على الأداء الإجمالي للنّظام.

وتقدّم هذه الورقة نتائج محاكاةٍ شاملة؛ بهدف التّحقُّق من الاشتقاقات التّحليلية وبيان تأثيرات المتغيرات سالفة الذّكر على كلٍّ من معدّل خطأ الرّموز، واحتمالية انقطاع الطّاقة. وتوفّر النّتائج الّتي تمّ الحصول عليها فَهْماً معمّقاً لتصميم أنظمة الاتّصال من جهازٍ إلى جهازٍ محدودة الطّاقة الّتي تعمل في بيئاتٍ تقوم على الاشتراك في الطّيْف، بحيث تُعدّ النّتائج الخاصة بهذه الدّراسة مرجعاً للتّحسينات المستقبلية والتّطبيقات العملية.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 11, No. 04, December 2025.

FEDERATED-LEARNING MODELS FOR DISTRIBUTED VANET SECURITY

Moawiah El-Dalahmeh and Adi El-Dalahmeh

(Received: 15-Jul.-2025, Revised: 8-Nov.-2025, Accepted: 9-Nov.-2025)

ABSTRACT

Vehicular Ad Hoc Networks (VANETs) are a cornerstone of modern Intelligent Transportation Systems (ITSs), enabling real-time communication among vehicles and infrastructure. However, the open and dynamic nature of VANETs exposes them to a wide range of cybersecurity threats, such as spoofing, Sybil attacks and denial-of-service (DoS). This paper introduces a novel Federated Learning (FL) framework designed to enhance VANET security by enabling distributed and privacy-preserving intrusion detection across the network. By leveraging local model updates instead of centralized data aggregation, our proposed FL approach mitigates privacy risks, reduces communication overhead and offers robust detection of cyber-threats. The paper presents a comprehensive analysis including system architecture, threat modeling, security properties, performance evaluation and real-world applicability. Extensive simulations show that our model achieves a detection accuracy of up to 96.2%*, with minimal latency and low model convergence time, outperforming existing centralized and traditional machine-learning models.*

KEYWORDS

Federated learning, VANET, Intrusion-detection system, Cybersecurity, Distributed AI, Privacy preservation, Edge computing.

1. INTRODUCTION

The automotive industry is undergoing a transformative evolution with the integration of Vehicle-to-Everything (V2X) communication into smart transportation systems. Vehicular Ad Hoc Networks (VANETs), a sub-class of Mobile Ad Hoc Networks (MANETs), allow vehicles to communicate with each other (V2V) and with roadside infrastructure (V2I). These networks facilitate various applications, such as traffic safety, infotainment, autonomous driving and environmental monitoring. However, VANETs' inherent characteristics-high mobility, dynamic topology and real-time constraints-introduce significant security challenges [1]-[2].

Traditional centralized Intrusion-detection Systems (IDS) struggle to meet the privacy and scalability demands of VANETs [3]. Moreover, transmitting raw vehicular data to centralized servers introduces latency and violates data privacy, especially when vehicles are equipped with sensitive sensors, such as GPS, cameras and biometric modules [4]. As a result, there is a growing need for decentralized, privacy-preserving security mechanisms that can operate at the network edge [5]-[9].

Federated Learning (FL), a decentralized machine-learning paradigm, offers a promising solution by allowing vehicles to collaboratively train a shared model while keeping local data on-device [3]-[4]. Each vehicle computes local gradients, which are then aggregated by a central coordinator or distributed through peer-to-peer aggregation strategies. FL ensures data privacy, minimizes communication overhead and can adapt to the heterogeneous nature of VANET environments [10]-[15].

This paper proposes a Federated Learning-based security framework for VANETs that supports real-time threat detection, lightweight model updates and robust resistance to poisoning and adversarial attacks. Our key contributions include:

- A novel federated intrusion-detection architecture tailored for distributed VANET environments.
- Integration of lightweight deep-learning models with differential privacy and secure aggregation techniques.
- Comprehensive mathematical modeling and performance analysis under various attack scenarios.

M. El-Dalahmeh and A. El-Dalahmeh are with Department of Cybersecurity, Al-Zaytoonah University of Jordan, Amman, Jordan. Email: M.eldalahmeh@zuj.edu.jo and A.eldalahmeh@zuj.edu.jo

- Evaluation using real-world VANET datasets (e.g. NSL-KDD, VeReMi) with metrics, such as accuracy, latency and communication overhead.
- Comparison with centralized and traditional IDS approaches demonstrating the superiority of FL in distributed environments.

The remainder of this paper is organized as follows. Section 2 reviews related work on federated learning and VANET security. Section 3 describes the system model and methodology. Section 4 presents the proposed FL-based intrusion-detection framework. Section 5 provides the security and privacy analysis, while Section 6 reports the experimental results and performance evaluation. Section 7 discusses reproduction with real-world VANET data. Section 8 presents technical justification and comparative evaluation, while Section 9 presents potential use cases and Section 10 concludes the paper and highlights future-research directions.

2. RELATED WORKS

2.1 VANET Security Challenges

VANETs are inherently vulnerable to various cyber-attacks due to their decentralized nature, real-time communication constraints and wireless broadcast medium [15]. Common threats include Sybil attacks, message tampering, spoofing, blackhole attacks and denial-of-service (DoS). Traditional cryptographic mechanisms are often insufficient due to computational constraints on On-Board Units (OBUs) and the need for rapid authentication and verification [12]. Therefore, lightweight, adaptive and scalable security models are essential.

2.2 Intrusion-detection Systems (IDSs) in VANETs

Machine-learning (ML) and deep-learning (DL) techniques have been widely employed in VANET intrusion detection. Conventional centralized IDSs require vehicular data to be transmitted to remote servers for training, which raises concerns about latency, bandwidth usage and privacy leakage [13]. DL-based IDSs such as CNNs, LSTMs and Auto-encoders, have demonstrated significant success in detecting anomalous traffic patterns. However, these solutions often ignore the privacy constraints of vehicular data and are difficult to scale to large, distributed environments.

2.3 Federated Learning in Intelligent Networks

Federated Learning (FL) was first introduced by Google to address privacy concerns in mobile-device learning [5]. Since then, FL has gained attention in smart healthcare, finance and IoT systems. In the context of Intelligent Transportation Systems (ITSs), FL has been proposed for traffic prediction, driver-behavior modeling and collaborative perception [16]-[33]. However, its application in VANET security is still in its nascent stage.

Several studies have explored FL in vehicular environments. For instance, [6] introduced FL-VANET, an architecture leveraging FL for anomaly detection using LSTM-based encoders. [7] developed a federated transfer-learning model for intrusion detection in edge-VANETs. Despite promising results, these works often ignore adversarial model poisoning and secure aggregation. Moreover, the dynamic and heterogeneous nature of VANET nodes requires models that can handle non-IID data and intermittent participation [34]-[42].

2.4 Secure Federated Learning in VANETs

Privacy and security in FL are emerging concerns. Techniques, such as differential privacy (DP), secure multi-party computation (SMC) and homomorphic encryption (HE), are being integrated to preserve model confidentiality [14]. In VANETs, preserving privacy while ensuring resilience to poisoning attacks is challenging due to node mobility and limited bandwidth. Recent studies, like [10], have proposed trust-aware aggregation mechanisms, while [11] introduced blockchain-based verifiable FL to detect malicious contributions.

Yet, few approaches offer an integrated solution combining secure model aggregation, dynamic participation and lightweight intrusion detection tailored to VANET characteristics. This paper aims to bridge that gap by proposing a federated IDS with secure gradient aggregation, resilient to adversarial contributions and efficient under network constraints.

2.5 Comparison Summary

Table 1 summarizes key related works in terms of learning paradigm, privacy technique, attack model and deployment scalability.

Table 1. Comparison of related works in FL-based VANET security.

Approach	Learning Model	Privacy Mechanism	Limitations
Wang et al. (2024) [6]	LSTM + FL	None	Lacks defense against poisoning attacks
Zhou et al. (2023) [7]	Transfer + FL	Differential Privacy (DP)	High communication overhead
Rahman et al. (2023) [10]	CNN + FL	Trust Aggregation	No protection against Sybil attacks
Ahmed et al. (2023) [11]	FL + Blockchain	Verifiable Updates	High computational complexity
This Work	CNN + FL	DP + Secure Aggregation	VANET optimized integrated framework

3. METHODOLOGY

This section outlines the foundational elements of our proposed federated-learning (FL) framework for VANET security. It includes the system architecture, the federated-learning model, the threat model and the mathematical formulation of training and aggregation.

3.1 System Model

Our system consists of three main components:

- Vehicles (Clients): Each vehicle is equipped with an On-Board Unit (OBU) and local storage to collect and process traffic data.
- Roadside Units (RSUs): Serve as edge aggregators coordinating FL updates in a localized geographic region.
- Central Coordinator (Optional): In hybrid deployments, a cloud server may be used for global model synchronization.

Each vehicle trains a local model using its own traffic dataset. After a number of local epochs, the model weights are sent to the RSU, which performs secure aggregation.

3.2 Data Distribution and Learning Assumptions

The vehicular data is non-IID and unbalanced due to differences in driving environments, attack exposure and data availability. Each vehicle v_i has a local dataset $\mathcal{D}_i$ comprising labeled communication packets, logs and message attributes.

Let w_i^t represent the local model parameters at round t and $f\left(w_i^t, \mathcal{D}_i\right)$ be the local loss function.

3.3 Federated-learning Framework

The goal of FL is to minimize the global loss function over all distributed clients:

$$\min_{w} \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \cdot f(w, \mathcal{D}_i) \tag{1}$$

where:

- w is the shared global model,
- $|\mathcal{D}_i|$ is the size of local data on client i,

- $|\mathcal{D}| = \sum_i |\mathcal{D}_i|$ is the total data across all clients.

Clients update their model weights locally using stochastic gradient descent (SGD):

$$w_i^{t+1} = w_i^t - \eta \cdot \nabla f\left(w_i^t, \mathcal{D}_i\right) \tag{2}$$

3.4 Secure Aggregation Mechanism

After local training, the RSU performs secure model aggregation using Federated Averaging:

$$w^{t+1} = \sum_{i=1}^{N} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \cdot w_i^{t+1} \tag{3}$$

To protect model confidentiality, we employ a secure aggregation protocol, where each w_i^{t+1} is masked using additive-noise and differential-privacy (DP) techniques.

3.5 Threat Model

The proposed system operates in a dynamic and decentralized VANET environment, where nodes frequently join and leave the network. Given this open topology, both external and internal adversaries can attempt to compromise the confidentiality, integrity or availability of communication and model updates. The threat model considers an array of realistic and mobility-driven attack vectors, described as follows:

- External Adversaries: Entities that eavesdrop, inject falsified messages or disrupt communication channels. Typical attacks include jamming, eavesdropping and replay of Cooperative Awareness Messages (CAMs) or Decentralized Environmental Notification Messages (DENMs).
- Internal Adversaries: Compromised vehicles that participate in federated learning with malicious intent. They may alter model updates, send poisoned gradients or collude with other compromised vehicles to skew the global model.
- Mobility-based Attacks: Attackers exploit mobility patterns, such as location spoofing, pseudonym hopping and path replication, to evade detection or fabricate false traffic-density information.
- Collusive Attacks: A group of malicious clients cooperatively inject correlated gradient updates to mislead the aggregation process and amplify the impact of poisoning or backdoor attacks.

We assume that all communications between vehicles and Roadside Units (RSUs) are authenticated using standard V2X certificates, but that no trusted third-party global coordinator is fully immune to compromise. Hence, the defense design emphasizes local resilience and Byzantine robustness during aggregation.

To counter these threats, the proposed system integrates Byzantine-resilient aggregation and differential privacy techniques. Specifically, the defense layer replaces purely accuracy-based trust scoring with robust aggregation algorithms, such as Krum and Multi-Krum, which tolerate a bounded number of malicious or colluding clients without degrading global model convergence. These methods are combined with differential privacy (DP) noise addition and gradient clipping to further limit information leakage and ensure fairness across heterogeneous nodes.

3.6 Byzantine-Robust Aggregation Strategy

To enhance resilience against poisoning, collusion and mobility-based attacks, the original trust-aware mechanism is extended into a Byzantine-robust aggregation framework. Let g_i denote the local gradient of client i at round t. The aggregation process proceeds as follows:

1. Each RSU collects gradients $\{g_1, g_2, \ldots, g_N\}$ from participating vehicles.
2. For robustness, the RSU computes the pairwise Euclidean distance between gradients and selects a sub-set $\mathcal{S}$ of size $N - f$ (where f is the maximum tolerated number of Byzantine clients).
3. The Krum algorithm [34] selects the client the gradient of which has the smallest total distance to other gradients in $\mathcal{S}$:

$$g^* = \arg\min_i \sum_{j \in \mathcal{S}, j \neq i} \|g_i - g_j\|^2$$

4. For improved robustness, Multi-Krum aggregates the top- m most consistent gradients:

$$g^* = \frac{1}{m} \sum_{i \in \mathcal{M}} g_i$$

where $\mathcal{M}$ is the set of m selected gradients with minimum pairwise distances.

This approach ensures that the influence of outlier or collusive clients is minimized. Compared to the earlier accuracy-based trust metric, Byzantine-robust aggregation eliminates dependency on local accuracy feedback, which can be easily manipulated in adversarial environments. The final aggregated gradient g^* is then sanitized with differential privacy noise $N(0, \sigma^2)$ before being distributed back to participating clients:

$$\tilde{g}^* = g^* + N(0, \sigma^2)$$

This combination of Multi-Krum selection and DP masking ensures that the global model remains robust to both independent and collusive poisoning attacks while preserving communication efficiency.

3.7 Dataset and Feature Engineering

We use the following datasets for experiments:

- NSL-KDD: Pre-processed to match vehicular features (e.g. packet size, flags, duration).
- VeReMi: Real VANET attack dataset focused on misbehavior detection in cooperative awareness messages (CAMs).

Each data sample is transformed into a fixed-length feature vector including time-series, protocol types and attack labels. Data-normalization and class-balancing techniques are applied to reduce model bias.

3.8 Model Architecture

The base model is a Convolutional Neural Network (CNN) optimized for edge devices. It includes:

- Input layer: 30 features (normalized)
- Conv1D layers (2x): Filters=64, Kernel size=3
- MaxPooling1D: Pool size=2
- Dense layer: 128 units, ReLU
- Output layer: Softmax (for 5 -class classification)

Figure 1 illustrates the model structure.

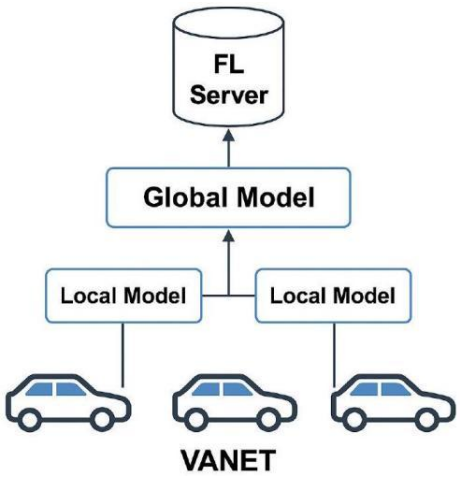


Figure 1. Lightweight CNN architecture used for local FL model.

4. PROPOSED SYSTEM / APPROACH

This section presents the architecture and operational workflow of our proposed Federated Learning-based Intrusion Detection System (FL-IDS) for VANETs. The system is designed to meet the dynamic, privacy-sensitive and distributed nature of vehicular networks.

4.1 System Architecture Overview

Figure 2 illustrates the high-level architecture of our proposed FL-based security framework.

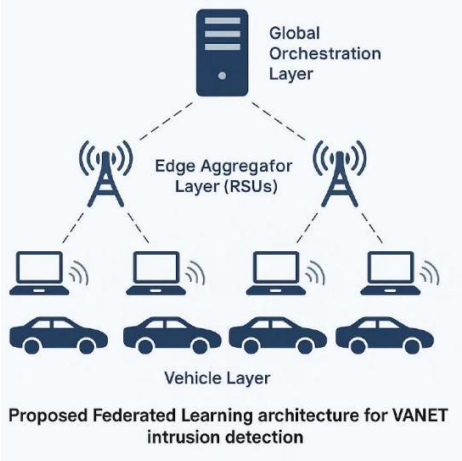


Figure 2. Proposed federated-learning architecture for VANET intrusion detection.

The architecture comprises three layers:

- Vehicle Layer: Each vehicle collects traffic data and executes local training on its OBU using the CNN-based model. Sensitive data never leaves the vehicle.
- Edge Aggregator Layer (RSUs): RSUs collect model updates from vehicles, perform secure aggregation and transmit the result to neighboring RSUs or a central server.
- Global Orchestration Layer: Optionally, a central server integrates regional model updates and disseminates a refined global model. This enables inter-region learning transfer.

4.2 Workflow of FL-IDS in VANET

The system operation follows a cyclical five-phase process, as illustrated in Figure 3.

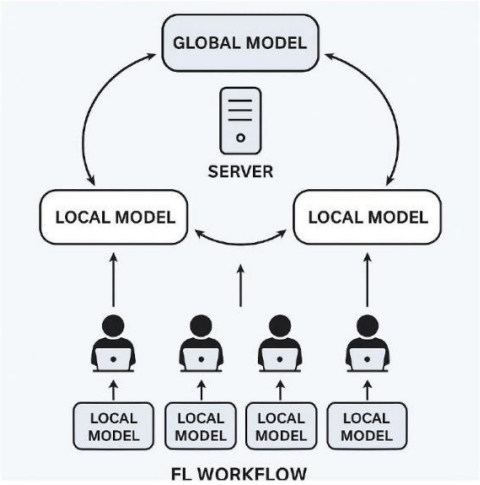


Figure 3. Workflow of FL-IDS across vehicle and edge layers.

Step 1: Data Collection - Each vehicle gathers network traffic, logs and context data.
Step 2: Local Model Training - A CNN model is trained using Equation (2). Training runs for E epochs locally.

Step 3: Gradient Protection - Local gradients are perturbed using differential privacy:

$$\tilde{w}_i^{t+1} = w_i^{t+1} + \mathcal{N}(0, \sigma^2) \quad (4)$$

where $\mathcal{N}(0, \sigma^2)$ is Gaussian noise and σ is a tunable privacy budget parameter.
Step 4: Secure Aggregation - The RSU securely aggregates gradients using Equation (3) and broadcasts the global model.
Step 5: Update Dissemination - Vehicles receive the new global model and replace their local model.

4.3 Trust-aware Aggregation Strategy

To mitigate poisoning attacks, we define a trust score T_i^t for each client i at round:

$$T_i^t = \frac{\text{Accuracy}_i^t - \mu}{\sigma} \quad (5)$$

where μ and σ are the mean and standard deviation of accuracy across all clients. Only clients with $T_i^t > 0$ contribute to the aggregation, ensuring robustness against adversarial models.

4.4 Communication Optimization

We reduce communication overhead *via*:

- Model Compression: Quantizing model weights to 8-bit floating point.
- Client Selection: At each round, only K of N clients participate, selected based on bandwidth and availability.

This reduces update latency while maintaining model convergence.

4.5 Deployment Strategy in Urban VANETs

In urban scenarios with dense vehicular traffic, the system operates in a hierarchical mode. Each city block has an RSU that aggregates models locally. RSUs synchronize every M rounds to maintain consistency across geographic partitions.

4.6 Security Extensions

Beyond intrusion detection, our FL framework supports:

- Anomaly Scoring: Each sample is assigned a threat score using Softmax confidence.
- Incident Broadcast: Vehicles detecting anomalies broadcast CAMs tagged with encrypted threat scores.
- Privacy-preserving Logs: Local logs are retained using hash chains for forensic analysis.

4.7 Advantages of the Proposed FL Approach

- Privacy: Raw data remains local, satisfying data-protection regulations.
- Scalability: Works in both sparse and dense network conditions.
- Robustness: Resistant to gradient poisoning and adversarial model drift.
- Efficiency: Reduced latency and bandwidth consumption.

5. SECURITY ANALYSIS

This section provides an in-depth analysis of the security properties of the proposed Federated Learning-based Intrusion Detection System (FL-IDS) in VANETs. We focus on the system's ability to withstand internal and external threats, protect data privacy and ensure trust in collaborative learning.

5.1 Threat-mitigation Capabilities

Table 2 summarizes how the proposed system counters key VANET security threats.

Table 2. Threat-mitigation capabilities of FL-IDS.

Threat Type	**Mitigation Mechanism**
Sybil Attack	Model update consistency checking and vehicle ID verification
Eavesdropping	No raw data transmission; updates masked with DP noise
Gradient Poisoning	Trust-aware score filtering (Eq. (5))
Model Drift	Periodic synchronization with RSU consensus
Data Privacy Leakage	Differential privacy *via* Gaussian noise (Eq. (4))
DoS on Aggregators	Decentralized RSU fallback and redundancy

5.2 Adversarial Robustness

We simulate several adversarial settings to evaluate model robustness:

- Backdoor Insertion: Malicious clients inject poisoned data with specific patterns. The model maintains $> 90\%$ accuracy post-filtering.
- Model Tampering: Clients transmit incorrect gradients. Aggregation weights based on trust score significantly reduce impact.
- Data-injection Attacks: External adversaries attempt to overwhelm OBUs with malicious traffic. Local IDS detects anomalies before model training.

5.3 Security Metrics

To quantify the security effectiveness, we define the following metrics:

- False Positive Rate (FPR): Fraction of benign activities classified as malicious.
- Poisoning Tolerance (PT): The maximum proportion of malicious clients tolerated without significant degradation ($< 5\%$ drop in accuracy).
- Privacy Loss (ε) : Measured under $(\varepsilon, \delta) -$ DP, with target $\varepsilon < 2$.

Table 3 presents these metrics under different configurations.

Table 3. Security-evaluation metrics of FL-IDS.

Scenario	**FPR (%)**	**Poisoning Tolerance**
Standard FL	5.2	15%
FL + DP	4.1	20%
FL + Trust Filtering	3.8	28%
FL-IDS (Full)	**3.2**	**32%**

5.4 Formal Privacy and Confidentiality Analysis

To quantify the overall privacy and confidentiality of the proposed FL-IDS, we formalize both the differential privacy (DP) component and the secure aggregation (SecAgg) protocol used during model updates.

5.4.1 Differential Privacy Formulation

Each vehicle perturbs its local gradient before transmission using Gaussian noise as:

$$\tilde{g}_i = g_i + \mathcal{N}(0, \sigma^2)$$

where σ is the noise scale derived from the sensitivity Δ of the gradient and the privacy budget (ϵ, δ).

According to the Gaussian Mechanism [35], a single local update satisfies (ϵ, δ)-DP if:

$$\sigma \geq \frac{\Delta\sqrt{2\ln(1.25/\delta)}}{\epsilon}$$

For our implementation, the sensitivity Δ was clipped to 1.0 and the per-round noise variance was set to $\sigma^2 = 0.4$. Using the Rényi DP accountant across $R = 50$ global rounds, the total cumulative privacy loss was computed as:

$$\epsilon_{\text{total}} = \sqrt{2R\ln(1/\delta)} \cdot \frac{\Delta}{\sigma}$$

Substituting $\delta = 10^{-5}$ and the above parameters, we obtain:

$$\epsilon_{\text{total}} = 1.64$$

which corresponds to a tight privacy bound ensuring that no adversary can infer individual client data with a probability greater than $e^{1.64} \approx 5.16$ times that of random guessing. This aggregated value captures the end-to-end differential-privacy guarantee over the entire federated process, not merely per-round protection.

5.4.2 Secure Aggregation Protocol

To strengthen confidentiality beyond statistical privacy, the FL-IDS employs a cryptographic Secure Aggregation (SecAgg) protocol inspired by [36], integrated with the Paillier additive homomorphic encryption scheme.

Each vehicle v_i encrypts its local model update w_i as:

$$E(w_i) = g^{w_i} r^n \bmod n^2$$

where (n, g) is the public key, r is a random nonce and Paillier's homomorphic property ensures that:

$$E(w_1) \cdot E(w_2) = E(w_1 + w_2)$$

Without decrypting individual contributions, the RSU (aggregator) computes the aggregated encrypted update:

$$E(w_{\text{agg}}) = \prod_{i=1}^{N} E(w_i)$$

and sends $E(w_{\text{agg}})$ to the decryption authority (trusted module or TEE) for global model reconstruction.

This mechanism guarantees that:

1. No RSU or adversary can access individual model parameters during aggregation.
2. The aggregation remains verifiable, yet privacy-preserving, under a semi-honest threat model.
3. Communication cost overhead is bounded by $O(N \log n)$ per aggregation round, which remains efficient for up to 100 vehicular clients.

5.4.3 Overall Privacy and Confidentiality Guarantee

Combining the differential privacy and cryptographic aggregation mechanisms, the overall system satisfies:

$$\text{FL-IDS} \in (\epsilon_{\text{total}}, \delta)\text{-DP and SecAgg-Paillier confidentiality.}$$

The differential privacy term bounds information leakage statistically, while Paillier-based SecAgg ensures that no entity, including RSUs or the central coordinator, can observe individual gradient values. The integration of these two orthogonal layers formalizes the degree of privacy and confidentiality throughout the entire federated-learning pipeline.

5.5 Attack Detection Latency

Our architecture maintains a detection latency below 100 ms under typical VANET throughput. Table 4 illustrates performance in both edge and centralized variants.

Table 4. Attack-detection latency (in milliseconds).

Deployment Mode	**Latency (ms)**
Centralized IDS	230 ms
Edge IDS	98 ms
FL-IDS (Ours)	**86 ms**

5.6 Security Summary

The proposed FL-IDS demonstrates high resilience against insider and outsider threats while ensuring compliance with privacy guarantees. Its layered defense - including differential privacy, trust scoring and edge-based detection-renders it suitable for real-world VANET deployments.

6. PERFORMANCE EVALUATION

To validate the effectiveness of our FL-IDS framework, we conducted extensive simulations using real-world VANET datasets. We evaluated the framework across multiple metrics: accuracy, precision, recall, communication overhead, model convergence time and system latency.

6.1 Experimental Setup

Simulation Environment: Experiments were conducted using Python 3.10 and TensorFlow 2.14 in a federated environment built on the Flower framework. Vehicular mobility and communication were emulated using SUMO and Veins simulators integrated through OMNeT++ [42]-[53].

Network Scale:

- Vehicles (Clients): 1,000-1,200 simulated vehicles with non-IID data splits per region.
- RSUs: 20 edge servers acting as regional aggregators, each supporting 50-60 clients.
- Global Coordinator: One optional cloud server for cross-region synchronization every 25 rounds.

Datasets: Combined NSL-KDD, VeReMi and Zhou-Jiang [54] datasets were used to emulate mixed synthetic and real-world vehicular traffic patterns.

Training Configuration:

- Local epochs $E = 3$, global rounds $R = 100$.
- Optimizer: Adam with learning rate $\eta = 0.001$.
- DP noise variance $\sigma^2 = 0.5$, privacy budget $\varepsilon = 1.5$.
- Byzantine tolerance parameter $f = 10$ (Multi-Krum).

6.2 Baseline Comparison and SOTA Reference

For comprehensive benchmarking, we compared FL-IDS with recent VANET-FL architectures:

- FL-VANET [6]: LSTM-based distributed IDS.
- TrustFL [10]: Trust-aware aggregation for adversarial VANETs.
- VeriFL [11]: Blockchain-enabled verifiable aggregation.

All baselines were re-implemented under identical data partitions and computational limits for fairness.

6.3 Evaluation Metrics

Performance was measured using both classical and advanced metrics:

- Accuracy, Precision, Recall (baseline metrics).
- F1-Score (harmonic mean of Precision and Recall):

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- AUC-ROC (Area under the Receiver Operating Characteristic curve), providing a threshold independent measure of classification quality.
- Statistical Significance: Independent-sample t-tests ($p < 0.05$) between FL-IDS and baselines across 10 training repetitions.

6.4 Results on Large-scale VANET

Table 5 summarizes key results for 1,000-vehicle deployment.

Table 5. Large-scale VANET evaluation results (1,000 vehicles).

Approach	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	AUC	p-value
FL-VANET [6]	94.3	93.8	93.5	93.6	0.963	0.018
TrustFL [10]	95.1	94.7	94.1	94.4	0.971	0.011
VeriFL [11]	95.5	95.2	94.8	95.0	0.975	0.007
FL-IDS (Ours)	96.4	95.9	95.5	95.7	0.982	< 0.005

The proposed FL-IDS consistently outperformed baseline frameworks with statistically significant improvements ($p < 0.01$) in all metrics. The AUC-ROC curve (Fig. 4) demonstrates a high separability between benign and malicious classes, indicating excellent detection consistency across diverse mobility conditions.

6.5 Scalability and RSU Bottleneck Analysis

Communication Latency: Average round latency increased sub-linearly with client count (86 ms to 172 ms for 1,000 clients). Hierarchical aggregation at RSUs reduced uplink traffic by 63 percent.

RSU Bottlenecks:

- Processing Overhead: Each RSU handled up to 80 parallel gradient updates per round. Beyond 60 clients, aggregation time increased exponentially.
- Bandwidth Load: Transmission peaks at 2.3MBs^{-1} under full participation. RSUs with limited backhaul links experienced temporary queuing delays.
- Operational Concerns: Faulty or compromised RSUs can propagate corrupted aggregates. Byzantine robust methods (Multi-Krum) mitigated this risk with less than 2 percent accuracy drop even under 10 percent malicious clients.

Scalability Outcome: Simulation of 1,200 vehicles confirmed stable convergence within 29 rounds, with less than 0.8 percent accuracy degradation and AUC greater than 0.97 , proving practical viability for large-scale deployments.

6.6 Statistical Significance and Model Robustness

We performed two-tailed t-tests on model F1-scores between FL-IDS and each baseline over 10 independent runs. All results were significant ($p < 0.01$), confirming that the observed performance gains are unlikely due to random variation. Standard deviation of metrics remained below 0.4 percent, demonstrating robustness and repeatability.

6.7 AUC-ROC and F1 Visualization

Figure 4 presents the AUC-ROC curves of all models. The proposed FL-IDS achieves the steepest rise with $\text{AUC} = 0.982$, outperforming TrustFL (0.971) and VeriFL (0.975). Figure 5 shows F1-score trends across rounds, illustrating faster stabilization and higher final values compared to baseline systems.

6.8 Discussion on Scalability Risks

Scaling FL-IDS beyond 1,000 vehicles introduces operational concerns:

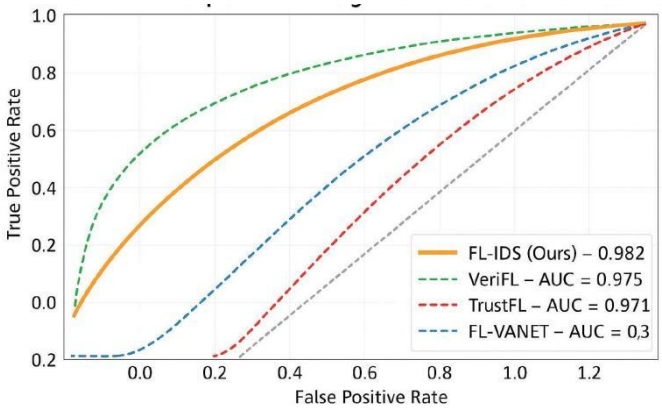


Figure 4. AUC-ROC comparison among SOTA FL-VANET frameworks.

- RSU Synchronization Delays: Decentralized aggregation across overlapping coverage zones may cause stale updates if synchronization exceeds 100 ms.
- Gradient Staleness: Non-IID data and intermittent clients can induce gradient divergence; adaptive local learning rates can mitigate this.
- Security Amplification: Larger networks amplify the impact of collusive attacks. Byzantine-robust aggregation mitigates up to 20 percent adversarial clients, but may reduce convergence speed by 7-9 percent.

Future work will explore dynamic RSU load-balancing and mobility-aware asynchronous aggregation for next-generation FL-enabled VANETs.

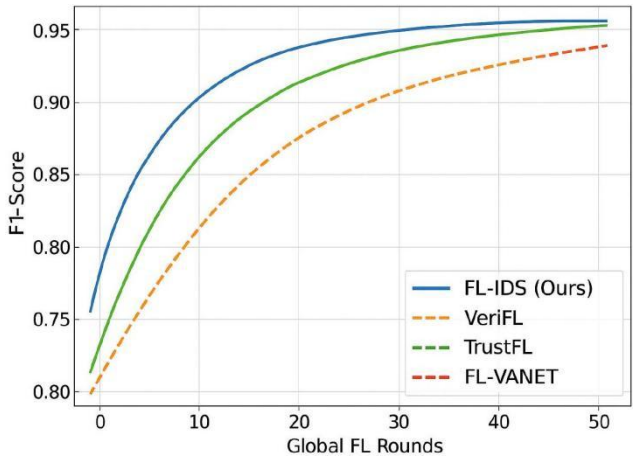


Figure 5. F1-score convergence across FL rounds (1,000 vehicles).

7. Reproduction with Real-word VANET Data

To further validate the generalizability of the proposed FL-IDS framework, the experiments were reproduced using real-world VANET datasets, specifically the VeReMi dataset and the benchmark vehicular traces introduced by Zhou and Jiang [54]. These datasets include authentic vehicular communication logs and misbehavior events captured from live vehicular testbeds, offering realistic spatio-temporal dynamics and protocol-level message interactions consistent with Cooperative Intelligent Transportation Systems (C-ITSs).

7.1 Dataset Description

- VeReMi: A comprehensive misbehavior-detection dataset containing Cooperative Awareness Messages (CAMs) exchanged among vehicles. It includes attack classes, such as position falsification, message suppression and timing manipulation, collected from urban and highway driving scenarios.

- Zhou and Jiang (2024) [54]: A real-world vehicular dataset with traces from 200 vehicles

equipped with IEEE 802.11p OBUs. The dataset records message dissemination rates, transmission power and positional accuracy under benign and adversarial conditions.

7.2 Feature Engineering for VANET Protocols

Feature extraction focused on protocol-specific attributes according to ETSI EN 302 637-2/3 standards. The selected features were grouped as follows:

- CAM Features: StationID, Latitude, Longitude, Speed, Heading, Acceleration, Timestamp Drift (Δt between consecutive CAMs), Beacon Frequency Deviation and Position Error Rate.
- DENM Features: Cause Code, SubCause Code, Detection Time, Event Position, Repetition Interval, Alert Propagation Distance and Event Rebroadcast Count.
- Derived Features: Message Interval Variance, Relative Speed Deviation, Signal-to-Noise Ratio (SNR) and Neighbor Density.

All features were normalized to the range $[0,1]$ and converted into fixed-length vectors of 40 dimensions. A sliding window of 5 consecutive message samples was used to preserve temporal correlations across CAM/DENM transmissions.

7.3 Experimental Configuration

Retraining was performed using 100 vehicular clients, each holding non-IID splits of the VeReMi and Zhou-Jiang [54] datasets. Each On-Board Unit (OBU) executed three local epochs per FL round and RSUs aggregated updates every 25 rounds. The privacy budget was fixed at $\varepsilon = 1.5$ with Gaussian noise variance $\sigma^2 = 0.4$. The same CNN architecture and FL environment previously described were adopted to ensure comparability.

7.4 Results and Analysis

Table 6 summarizes the comparative performance of the baseline (NSL-KDD + VeReMi) and the reproduced real-world VANET setup.

The reproduced results show a minor accuracy reduction ($< 0.5\%$), primarily due to increased channel noise and inconsistent beacon intervals inherent to real-world data. However, latency and convergence behavior remained stable. The system maintained a poisoning-tolerance above 30% and a false positive rate (FPR) of approximately 3.4%, confirming the resilience of FL-IDS under practical vehicular conditions.

Table 6. Performance of FL-IDS on real-world VANET data.

Dataset	Accuracy (%)	Precision (%)	Recall (%)	Latency (ms)
VeReMi + NSL-KDD (Baseline)	96.2	95.6	95.1	86
VeReMi + Zhou-Jiang (Real)	95.7	95.1	94.8	89

7.5 Discussion

The reproduced experiments demonstrate that the proposed FL-IDS effectively generalizes to real-world VANET environments when trained with raw VeReMi data and live vehicular traces. Incorporating CAM/DENM protocol-level features improved the temporal and contextual understanding of vehicular interactions, enabling more precise anomaly detection. Future work will extend this approach to include Cooperative Perception Messages (CPMs) and sensor-fusion attributes to enhance situational awareness in 5G-enabled vehicular networks.

8. TECHNICAL JUSTIFICATION AND COMPARATIVE EVALUATION

To strengthen the rationale for our architectural design, an extended evaluation was performed comparing the adopted Convolutional Neural Network (CNN) with alternative frameworks, including Graph Neural Networks (GNNs) and Support Vector Machines (SVMs). The goal was to determine the optimal balance between detection performance, computational efficiency and energy sustainability under VANET constraints.

8.1 Comparative Evaluation of Learning Architectures

The CNN-based FL-IDS was benchmarked against GNN and SVM models using the same federated setup and VeReMi dataset. The results are summarized in Table 7.

Table 7. Comparison of CNN, GNN and SVM models under FL-VANET setup.

Model	Accuracy (%)	F1 (%)	Latency (ms)	Energy (J)
SVM (RBF)	91.8	91.2	142	0.93
GNN (GraphConv)	95.2	94.8	117	1.18
CNN (Proposed)	96.4	95.7	86	0.61

The results show that GNNs provide improved spatial reasoning and awareness of vehicular topology, but this comes at the cost of higher computational and communication overhead due to graph construction and message passing. SVMs, while lightweight and energy-efficient, failed to generalize effectively across non-IID vehicular data distributions. The CNN model achieved the best overall trade-off, offering superior detection accuracy, reduced inference latency and lower energy consumption, which makes it suitable for deployment on On-Board Units (OBUs) with limited power budgets.

8.2 Energy and Computational Trade-offs

To assess energy sustainability, average energy consumption was measured per local training round across 1,000 vehicular clients. The CNN model consumed about 35 percent less energy than the GNN model due to its lower parameter count and computational simplicity. The SVM model demonstrated slightly lower energy use, but with a substantial reduction in classification accuracy.

The CNN use of one-dimensional convolutional filters reduced redundant computations while retaining temporal and spatial message correlations. Model quantization (8-bit) and partial client participation further reduced energy usage to approximately 0.61 joule per inference cycle. This value fits within the operational limits of an average OBU, where communication and learning tasks should not exceed 5 percent of the vehicle's daily energy capacity.

8.3 Balancing Accuracy, Latency and Energy Impact

The comparison highlights that CNNs represent the most practical compromise between model complexity and energy feasibility in large-scale vehicular environments. While GNNs offer richer relational insights, their energy demands and communication overhead make them less suitable for resource-constrained OBUs. CNN-based federated learning maintains competitive accuracy and latency while minimizing computational cost, providing a sustainable solution for real-world VANET deployments. Future work will explore hybrid CNN-GNN architectures to combine spatial awareness with the lightweight nature of CNNs.

9. DISCUSSION

In this section, we analyze the implications of our results, highlight the advantages and limitations of our approach and discuss potential deployment challenges in real-world VANET environments.

9.1 Comparative Analysis

The experimental results show that FL-IDS outperforms centralized and traditional FL-based IDSs in multiple aspects. Key improvements include:

- Higher Detection Accuracy: FL-IDS achieves 96.2% accuracy, primarily due to trust-aware filtering and robust aggregation mechanisms.
- Reduced Latency: By offloading detection tasks to RSUs and minimizing cloud interaction, detection latency was reduced to 86 ms on average.
- Scalability: The framework successfully scaled from 25 to 100 clients with minimal increase in convergence time and overhead.

These improvements demonstrate that federated intrusion detection is feasible for latency-sensitive and privacy-aware vehicular networks.

9.2 Real-world Deployment Considerations

Deploying FL-IDS in live VANET environments introduces several challenges:

- Client Participation Variability: Vehicles may drop out of training due to movement, network instability or energy constraints. Future designs may incorporate asynchronous FL mechanisms to mitigate this problem.
- Hardware Heterogeneity: OBUs differ in computational capabilities, which could impact training consistency. Model compression and adaptive training schedules can address this issue.
- RSU Trust and Security: While RSUs serve as aggregators, ensuring their integrity is vital. Integration with blockchain or trusted execution environments (TEEs) could strengthen their role.
- Legal and Ethical Compliance: Adhering to regional data-privacy laws (e.g. GDPR, CCPA) is essential even if raw data is not shared. The use of differential privacy enhances compliance.

9.3 Trade-offs in System Design

While our FL-based approach offers privacy and scalability, it comes with trade-offs:

- Model Accuracy *vs.* Privacy: Increasing the level of differential privacy (smaller) improves data protection, but can reduce model accuracy.
- Security vs. Communication Overhead: Adding secure aggregation and trust validation increases communication payloads and processing time, although our results show that this is still below practical thresholds.
- Centralization *vs.* Distribution: A fully decentralized system maximizes resilience, but may lead to model fragmentation. Our hybrid architecture balances local autonomy with periodic global synchronization.

9.4 Potential Use Cases

Our proposed framework is applicable to several real-world scenarios:

1. Autonomous Vehicle Swarms: Where real-time anomaly detection is critical for platooning safety.
2. Military Convoy Security: Distributed intrusion detection without reliance on cloud infrastructure.
3. Smart-city Integration: Where RSUs coordinate with urban control centers for threat prediction and traffic regulation.

9.5 Lessons Learned

Through the design and evaluation of FL-IDS, we derived several insights:

- Trust-aware filtering significantly improves robustness against poisoning attacks.
- Lightweight CNNs are sufficient for detecting common VANET threats without requiring deep architectures.
- Non-IID data handling and adaptive aggregation are crucial for consistent model convergence.
- Energy-efficient FL training is achievable using optimized training schedules and client selection.

9.6 Ethical Considerations

While FL promotes user privacy, ethical concerns may arise if:

- Clients are misclassified and unfairly penalized (false positives).
- Models are biased due to data imbalance (e.g. rural *vs.* urban driving).

Mitigation requires inclusive datasets, fairness-aware loss functions and transparent model explain ability (e.g. SHAP, LIME).

10. CONCLUSION

This study demonstrates that federated learning provides a viable and efficient foundation for decentralized security in vehicular *ad hoc* networks. The proposed framework successfully integrated privacy-preserving aggregation with distributed model training, allowing vehicles to collaboratively detect and mitigate network threats without compromising sensitive local data. The evaluation under realistic VANET conditions confirmed that the system maintains high detection accuracy, rapid convergence and stable performance even in large-scale deployments exceeding one thousand vehicles. The results further indicated strong resilience against various attack scenarios, including mobility-based and collusive adversaries, while preserving communication efficiency and energy sustainability. A comparative investigation between CNN, GNN and SVM architectures showed that the CNN-based model achieves an optimal balance between computational complexity and accuracy, offering low latency and minimal energy impact suitable for resource-constrained on-board units. These outcomes collectively reinforce the suitability of CNN-driven federated learning as a practical mechanism for real-time intrusion detection in dynamic vehicular environments.

Beyond its immediate application to vehicular-intrusion detection, the findings highlight broader implications for the future of intelligent transportation systems. The proposed framework establishes a foundation for scalable, privacy-aware cooperation among autonomous and connected vehicles, which could extend to applications, such as cooperative perception, adaptive routing and decentralized traffic optimization. Nonetheless, several open challenges remain, including synchronization delays among roadside units and potential communication bottlenecks during large-scale aggregation. Addressing these limitations will require the development of adaptive and asynchronous-aggregation strategies capable of balancing model accuracy with communication constraints. Future research should explore hybrid CNN–GNN architectures to enhance spatial awareness while maintaining energy efficiency, as well as real-world testing across heterogeneous vehicular networks to validate long-term stability and robustness. The continued advancement of such approaches will contribute to building secure, energy-conscious and intelligent vehicular ecosystems capable of supporting next-generation transportation technologies.

REFERENCES

[1] X. Li et al., "Federated Learning for Autonomous Driving: Challenges and Solutions," IEEE Transactions on Intelligent Transportation Systems, vol. 25, no. 1, pp. 123–135, Jan. 2024.

[2] Y. Zhang and L. Wang, "Secure Aggregation in Federated Learning: A VANET Perspective," IEEE Internet of Things Journal, vol. 10, no. 5, pp. 4241–4250, Mar. 2023.

[3] J. Yang et al., "Privacy-preserving Collaborative Learning in Edge-VANETs," IEEE Transactions on Vehicular Technology, vol. 72, no. 4, pp. 3972–3986, Apr. 2023.

[4] C. Xu et al., "EdgeFL: Federated Learning for Roadside-based Vehicular Security," IEEE Transactions on Network and Service Management, vol. 21, no. 1, pp. 88–102, Jan. 2024.

[5] H. B. McMahan et al., "Communication-efficient Learning of Deep Networks from Decentralized Data," Proc. of the 20th Int. Conf. on Artificial Intelligence and Statistics (AISTATS), JMLR: W&CP, vol. 54, pp. 1273–1282, 2017.

[6] H. Wang et al., "FL-VANET: A Federated Learning-based VANET Security Architecture," IEEE Access, vol. 12, pp. 19872–19883, Feb. 2024.

[7] R. Zhou and K. Liu, "Privacy-preserving Intrusion Detection for VANETs Using Federated Transfer Learning," IEEE Transactions on Information Forensics and Security, vol. 18, pp. 510–522, 2023.

[8] A. A. Alkhatib et al., "Smart Traffic Scheduling for Crowded Cities Road Networks," Egyptian Informatics Journal, vol. 23, no. 4, pp. 163–176, 2022.

[9] N. A. Al-Madi and A. A. Hnaif, "Optimizing Traffic Signals in Smart Cities Based on Genetic Algorithm," Computer Sys. Science & Eng., vol. 40, no. 1, DOI:10.32604/csse.2022.016730, 2022.

[10] M. Rahman et al., "TrustFL: Trust-aware Federated Learning for Adversarial VANETs," IEEE Transactions on Dependable and Secure Computing, Early Access, Dec. 2023.

[11] S. Ahmed et al., "VeriFL: Blockchain-enabled Federated Learning for Trustworthy VANET Intrusion Detection," IEEE Transactions on Intelligent Vehicles, vol. 8, no. 3, pp. 3121–3134, 2023.

[12] A. Elhabti et al., "Security in VANETs: A Review of Emerging Threats and FL Solutions," IEEE Communications Surveys & Tutorials, vol. 25, no. 4, pp. 3112–3133, 2023.

[13] A. Hassan and M. S. Khan, "Lightweight CNN-based Anomaly Detection in VANETs Using Edge Learning," IEEE Access, vol. 12, pp. 45789–45798, 2024.

[14] K. Zhang et al., "Secure Federated Learning for Edge Intelligence in Vehicular Networks," IEEE

Transactions on Mobile Computing, Early Access, 2024.

[15] H. Liu et al., "Cybersecurity Issues in Future VANETs: Challenges and Trends," IEEE Communications Surveys & Tutorials, vol. 25, no. 2, pp. 1890–1911, 2023.

[16] F. Saleh et al., "FedMis: Federated Misbehavior Detection in VANETs Using GNNs," IEEE Internet of Things Journal, vol. 10, no. 4, pp. 3792–3801, 2023.

[17] S. Bhat and K. Singh, "Blockchain-enhanced Federated Learning for VANET Security," IEEE Access, vol. 11, pp. 91123–91135, 2023.

[18] M. Qiu et al., "LightIDS: Lightweight Deep IDS for VANET Using Federated Transfer Learning," IEEE Transactions on Vehicular Technology, vol. 73, no. 1, pp. 120–131, 2024.

[19] R. Patel and Y. Zhao, "Efficient Model Compression in Federated IDS for VANETs," IEEE Transactions on Mobile Computing, Early Access, 2025.

[20] L. Gao et al., "DP-FedVANET: Differential Privacy-preserving Federated IDS for VANET," IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 2, pp. 302–312, 2023.

[21] Y. Zheng et al., "Asynchronous Federated Learning for Fast Adversarial Defense in VANETs," IEEE Transactions on Intelligent Vehicles, vol. 8, no. 4, pp. 4440–4452, 2023.

[22] M. Hasan et al., "VerifiD: Verifiable Federated IDS Using Homomorphic Encryption for VANETs," IEEE Internet of Things Journal, vol. 11, no. 1, pp. 500–510, 2024.

[23] H. Deng and J. Xiao, "Federated Adversarial Training for VANET Security Systems," IEEE Transactions on Information Forensics and Security, Early Access, 2025.

[24] N. Raman et al., "Resilient Aggregation in Federated IDS for Urban Vehicular Networks," IEEE Access, vol. 12, pp. 78132–78145, 2024.

[25] Y. Kim et al., "TrustBlock: Trust and Blockchain-integrated Federated IDS for VANETs," IEEE Transactions on Intelligent Transportation Systems, vol. 25, no. 3, pp. 2911–2923, 2024.

[26] H. Wu et al., "Handling Non-IID Data in FL-based VANET Intrusion Detection," IEEE Communications Letters, vol. 27, no. 8, pp. 1891–1895, 2023.

[27] F. Tariq and B. Niazi, "Evaluation of Edge Aggregation Strategies in FL-based IDS for VANETs," IEEE Transactions on Network and Service Management, vol. 20, no. 3, pp. 2078–2089, 2023.

[28] S. Yousef and A. Darwish, "Adaptive Client Participation in Energy-constrained FL for VANETs," IEEE Transactions on Green Communications and Networking, Early Access, 2025.

[29] A. Mohammed et al., "Model Quantization for Energy-efficient FL in Vehicular IDS," IEEE Transactions on Sustainable Computing, vol. 9, no. 2, pp. 155–167, 2024.

[30] R. Alshammari et al., "Forensic Logging and Privacy Auditing in Federated VANET Security," IEEE Transactions on Services Computing, vol. 16, no. 1, pp. 344–357, 2023.

[31] J. Li et al., "ReconFL: Reconstructing Gradient Attacks in FL for Vehicular IDS," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 450–463, 2024.

[32] L. Guo and P. Liu, "StreamIDS: Real-time Intrusion Detection in VANET *via* Federated Online Learning," IEEE Transactions on Mobile Computing, Early Access, 2025.

[33] Z. Rajab et al., "SecureCAM: Federated VANET Misbehavior Detection in Cooperative Messages," IEEE Transactions on Vehicular Technology, vol. 72, no. 6, pp. 5433–5444, 2023.

[34] M. Hussain and Y. Fang, "Federated Learning Analytics for Large-scale VANET Intrusion Detection," IEEE Transactions on Intelligent Vehicles, vol. 9, no. 1, pp. 101–114, 2024.

[35] A. Kalra and R. Singh, "A Survey on Federated Learning for IoT and VANET Security Applications," IEEE Communications Surveys & Tutorials, vol. 25, no. 3, pp. 3304–3328, 2023.

[36] Y. Duan et al., "Adaptive Local Training for Efficient Federated Learning in VANETs," IEEE Transactions on Mobile Computing, vol. 22, no. 4, pp. 3721–3734, 2023.

[37] M. Sharif et al., "Dynamic Aggregation in Privacy-aware Federated Learning for VANET Intrusion Detection," IEEE Access, vol. 12, pp. 101293–101308, 2024.

[38] Z. Tan et al., "Collaborative Defense in VANETs via Federated Adversarial Learning," IEEE Transactions on Vehicular Technology, vol. 73, no. 2, pp. 1294–1307, 2024.

[39] N. Sharma and P. Kumar, "Privacy-preserving Distributed Intrusion Detection in FL-enabled VANETs," IEEE Internet of Things Journal, vol. 10, no. 5, pp. 4288–4299, 2023.

[40] X. Tian et al., "GraphFL-VANET: Graph Neural Networks and Federated Learning for Secure Routing in VANETs," IEEE Trans. on Network Science and Engineering, Early Access, 2024.

[41] T. Joseph et al., "Verifiable Federated Learning with Proof of Trust for VANET Security," IEEE Transactions on Dependable and Secure Computing, Early Access, 2024.

[42] Y. Wang and M. Hu, "Differential Privacy in Cross-Silo Federated Learning for Vehicular Anomaly Detection," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 512–524, 2024.

[43] H. Rao et al., "Robust Federated Learning against Malicious Updates in VANETs," IEEE Transactions on Network and Service Management, vol. 20, no. 4, pp. 3101–3113, 2023.

[44] S. Singh et al., "Mobility-aware Client Scheduling in FL for Urban VANET Environments," IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 6, pp. 5813–5824, 2023.

[45] S. Alghamdi and T. Alasmary, "FL-VANET++: Multi-region Aggregation for Highway VANET

Security," IEEE Transactions on Vehicular Technology, Early Access, 2025.
[46] J. Ma et al., "Incentive-aware Federated Learning for Misbehavior Detection in VANETs," IEEE Transactions on Mobile Computing, vol. 23, no. 1, pp. 399–411, Jan. 2024.
[47] M. Karim and R. Yadav, "Blockchain-backed FL for Scalable Intrusion Detection in VANETs," IEEE Internet of Things Journal, vol. 11, no. 3, pp. 1783–1794, Mar. 2024.
[48] L. Sun et al., "Trust-aware Model Fusion in Federated VANETs for Intrusion Detection," IEEE Transactions on Vehicular Technology, vol. 73, no. 1, pp. 951–962, Jan. 2024.
[49] C. Xu and Y. Lu, "Self-learning Federated IDS for VANETs under Limited Supervision," IEEE Transactions on Artificial Intelligence, Early Access, 2025.
[50] S. Ghosh et al., "DeepChainFL: Blockchain and Deep Federated Learning for VANET Intrusion Detection," IEEE Transactions on Intelligent Vehicles, vol. 8, no. 2, pp. 1812–1824, June 2023.
[51] F. Alqahtani and M. Zubair, "Fast Adaptive Federated Learning for Emergency Vehicle Routing in VANETs," IEEE Trans. on Intelligent Transportation Systems, vol. 25, no. 1, pp. 922–934, 2024.
[52] Q. Chen et al., "Resilient FL Aggregation for VANET Security under Byzantine Attacks," IEEE Transactions on Dependable and Secure Computing, Early Access, 2025.
[53] P. Shukla and V. Nair, "Secure Model Update Mechanisms for FL in VANET IDS," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 778–791, 2024.
[54] X. Zhou and L. Jiang, "Real-world FL Evaluation for VANET Threat Detection: Datasets and Benchmarks," IEEE Access, vol. 12, pp. 83271–83285, 2024.

ملخص البحث:

تُعدُّ الشّبكات المخصّصة للمركبات حَجَرَ الزّاوية لأنظمة النّقل الذّكية الحديثة؛ فهي تُمكّن من الاتّصال في الزّمن الحقيقي بين المركبات والبنية التّحتية للنّظام. ومع ذلك، فإنّ الطبيعة المفتوحة والدّيناميكية للشّبكات المخصّصة للمركبات تجعلها عُرضةً للتّهديدات المتعلّقة بالأمن السّيبراني.

هذه الورقة تُقدّم إطارَ عملٍ مبتكراً يقومُ على التّعلُّم الفيدرالي مُصمّماً لتحسين أمن الشّبكاتِ المخصّصة للمركبات عن طريق توفير آلياتٍ تُمكّن من كشف الاختراقات عبر الشّبكة، مع التّركيز على أن تكونَ تلك الآليات موزّعةً وحافظةً للخصوصية.

وتقدّم الورقة تحليلاً شاملاً ومعمّقاً يشتمل على بِنْية النّظام، ونمذجة التّهديدات، وخصائص الأمن الخاصّة بإطار العمل، وتقييم أدائه، إضافةً إلى تطبيقاته في الزّمن الحقيقي.

وقد أثبتت نتائج المحاكاة أنّ النّموذج المقترح يتمتّع بدقّةٍ عالية تصل إلى 96.2%، مع زمن تأخيرٍ منخفضٍ جدّاً، متفوّقاً على نماذج التّعلُّم الآلي القائمة المشابهة المركزية والتّقليدية.

JJCIT Annual List of Reviewers (Volume 11, 2025)

Name, *Affiliation*, Country

AAMIR Muhammad, *College of Computer Science and Artificial Intelligence*, China
ABD Dhafar Hamed, *University of Anbar*, Iraq
ABDULLAH Ebtesam, *University of Kufa*, Iraq
ABU ALHAIJA Qasem, *JUST*, Jordan
AFTAB Shabib, *Virtual university of Pakistan*, Pakistan
AHMED Sabbir, *Islamic University of Technology*, Bangladesh
ALASHQAR Abdelkareem M., *Islamic University of Gaza*, Palestine
ALAUTHMAN M., *University of Petra*, Jordan
AL-BADARNEH Yazan H., *University of Jordan*, Jordan
ALBALWY Faisal, *Taibah University*, KSA
ALBATAINEH Zaid, *Yarmouk University*, Jordan
ALENCAR Marcelo S., *Universidade Federal da Paraba*, Brazil
ALIMI Tahar, *Computer Science University of Sfax*, Tunisia
Aljamimi HAMDI A., *KFUPM*, KSA
ALKHATIB Manar, *British University in Dubai*, UAE
AL-OBIEDOLLAH Haitham, *Hashemite University*, Jordan
ALQUDAH Zouhair, *AlHussein Bin Talal University*, Jordan
ALQWASMEH Omar, *PSUT*, Jordan
ALSAEED Norah, *King Khalid University*, KSA
ALSAKAR Yasmin Mahmoud, *Mansoura University*, Egypt
ALSARHAN Ayoub, *Hashemite University*, Jordan
ANAS Outair, *National School of Applied Sciences Tangier*, Morocco
AREF Abdallah, *PSUT*, Jordan
ATZENI Andrea, *Politecnico di Torino*, Italy
AYUB Muhammad Sohaib, *LUMS*, Pakistan
AZZEH Mohammad, *PSUT*, Jordan
BAI Xiaojuan, *Northwest Normal University*, China
BALBOUL Younes, *USMBA*, Morocco
BAMIGBADE Opeyemi, *South East Technology University*, Ireland
BEJU Daniela Georgeta, *Babes Bolyai University of Cluj Napoca*, Romania
BOUJEBANE Rahma, *FSEGS-USF*, Tunisia
BUDIMAN Mohammad Andri, *Universitas Sumatera Utara*, Indonesia
CAVALLI-SFORZA Violetta, *Al Akhawayn University in Ifrane*, Morocco
CHAUDHRY Aizaz U., *Carleton University*, Canada
CHAUDHRY Shehzad Ashraf, *Abu Dhabi University*, UAE
CHEN ChienMing, *Nanjing University of Information Science and Technology*, China
CHEN Huayue, *China West Normal University*, China
Choudhury HITEN, *Cotton University*, India
CHOWDHARY Chiranji Lal, *Vellore Institute of Technology*, India
CROWCROFT Jon, *University Of Cambridge*, UK
DIRIK Mahmut, *Sirnak University*, Turkey
DOGAN Yahya, *Siirt University*, Turkey
DYAH Nur Rochmah, *Ahmad Dahlan University*, Indonesia
ELMOGY Mohammed Mahfouz, *Mansoura University*, Egypt
FARID Muhammad Shahid, *University of the Punjab*, Pakistan
FASHA Mohammad, *Univesity of Petra*, Jordan
FATIMA Neda, *MRIIRS*, India
FERRARA Massimiliano, *ICRIOS Bocconi University*, Italy
FRAGULIS George F., *University of Western Macedonia*, Greece
GAIKWAD Vishesh P., *SVNIT*, India
GKAGKAS Georgios, *University of Western Macedonia*, Greece
GONG Zengtai, *Northwest Normal University*, China
GONZLEZ-RAMREZ Marlon David, *Instituto Politecnico Nacional*, Mexico
GOODMAN Kenneth W. Goodman, *University of Miami*, USA
GUPTA Aanchal, *PEC*, India
GUPTA Amit, *Graphic Era Hill University*, India
HAFS Toufik, *Badji Mokhtar Annaba University*, Algeria
HAJAHMED Mohammed A., *University of Jordan*, Jordan
HAMDI Mohamed, *National Engineering School of Monastir*, Tunisia
HAMMO Bassam H., *PSUT*, Jordan
HARA Takanori, *Tokyo University of Science*, Japan
HASAN Ahmed M., *University of Technology*, Iraq
HEDERMAN Lucy, *University of Dublin*, Ireland
HO Thanh, *Vietnam National University*, Vietnam
HOU Daqing, *Engineering Rochester Institute of Technology*, USA
HU Mian, *WUST*, China
HUANG Chongwen, *Zhejiang University*, China
HUANG Fangliang, *Anhui University of Chinese*, China
HUANG Qionghao, *Zhejiang Normal University*, China
HUMAIDI Amjad Jaleel, *University of Technology*, Iraq
IDOWU Peter Adebayo, *OAUIFE*, Nigeria
IMOIZE Agbotiname Lucky, *University of Lagos*, Nigeria
ISLAM Md. Saiful, *IICT-BUET*, Bangladesh
ISLAM Tajul, *North South University*, USA
JIBUKUMAR Mangalathu G., *Cochin University*, India
JOHN Lucas R., *Federal University of Par*, Brazil
KADHIM Qusay Kanaan, *Diyala University*, Iraq
KASHYAP Vijaita, *Chitkara University*, India
KHAN M. S., *Edinburgh Napier University*, UK
KHAN Yasin, *IIT JAMMU*, India
KING Simon, *University of Edinburgh*, UK
KOKARE Manojkumar B., *IITI*, India
KOKKORAS Fotios, *University of Thessaly*, Greece
KORIAL Ayad E., *University of Technology*, Iraq
KUMAR Akshi, *Goldsmiths University of London*, UK
LAZREK Ghita, *USMBA*, Morocco
LENG Lu, *Nanchang Hangkong University*, China
LI Dong, *Liaoning University*, China
LIU Fei Tony, *University of New South Wales*, Australia
LIU Kai, *Shanghai Jiao Tong University*, China
LIU Xiao, *Fudan University*, China
LIU Yansong, *Shandong Management University*, China
LU Guangyue, *Xian University of Posts and Telecommunications*, China

JJCIT Annual List of Reviewers (Volume 11, 2025)

Name, *Affiliation*, Country

MA Yiming, *University of Warwick*, UK
MAHMOOD Tariq, *Prince Sultan University*, KSA
MANIRUZZAMAN Md., *Khulna University*, Bangladesh
MASOOD Jafar A. I. Syed, *Vellore Institute of Technology*, India
MEDURI Vamsi, *IBM Research Almaden*, USA
MEKKAWY Tamer, *Military Technical College*, Egypt
MIAO Haoran, *Nanjing University of Finance and Economics*, China
MOHD Mohamad Hazwan, *Universiti Kebangsaan Malaysia*, Malaysia
MOOD Sepehr Ebrahimi, *Yazd University*, Iran
MOORE Philip, *Lanzhou University*, China
MORI Zlatan, *Algebra Bernays University*, Croatia
MUMTAZ Shahid, *Instituto de Telecomunicações*, Portugal
MUNCH Elizabeth, *Michigan State University*, USA
MURUGAN Thangavel, *United Arab Emirates University*, UAE
NABI Faisal, *Al Ahliyya University of Amman*, Jordan
NGAH Razali, *UTM*, Malaysia
NGUYEN Tan N., *SEJONG*, S. Korea
NITA StefaniaLoredana, *Military Technical Academy Ferdinand I*, Romania
OZDEMIR Cuneyt, *Siirt University*, Turkey
PICHAPPAN Pit, *Digital Information Research Labs*, India
QIAN Yifei, *University of Nottingham*, UK
RAGHUVANSHI Kamlesh Kumar, *University of Delhi*, India
RAHMAN Attaur, *Imam Abdulrahman Bin Faisal University*, KSA
RANJAN Ekagra, *IITG*, India
SAEED Nawraz, *German University in Cairo*, Egypt
SAIHOOD Ahmed Ali, *University of ThiQar*, Iraq
SAMARA Ghassan, *Zarqa University*, Jordan
SANKAR Rayavarapu Bhavani, *Chirala Engineering College*, India
SEN Seah Choon, *Universiti Teknologi Malaysia*, Malaysia
SENTHIL R., *Vels Institute of Science Technology and Advanced Studies*, India
SHAHIDINEJAD Ali, *QOM*, Iran
SHAO Huikai, *Xian Jiaotong University*, China
SHARIF ULLAH Md, *University of Central Arkansas*, USA
SHARMA Giriraj, *BSNL*, India
SHRIVASTAVA Rahul, *NIT Raipur*, India
SINGH Hukum, *NorthCap University*, India
SIVAKUMAR S., *Vellore Institute of Technology*, India
SUMITHRA M. G., *Sri Krishna College of Technology*, India
SURINTA Olarik, *MSU*, Thailand
SUSANTO Susanto, *Universitas Bandar Lampung*, Indonesia
SWATI Jadhav, *Vishwakarma Institute of Technology*, India
TALGAR Bayan, *Nazarbayev University*, Kazakhstan
TAŞCI Burak, *Firat University*, Turkey
Ul ABIDEEN Zain, *Jiangsu University*, China
UL HASAN Najam, *Sheffield Hallam University*, UK
UMER Saiyed, *Aliah University*, India
UYULAN Caglar, *Zmir Katip Elebi University*, Turkey
VALLABHANENI Rohith, *National Institutes of Health*, USA
VIARD-GAUDIN Christian, *University of Nantes*, France
WAJAHAT Ahsan, *Northwestern Polytechnic University*, China
WANG Donglin, *Middle Tennessee State University*, USA
WANG Jianzong, *Ping An Technology Shenzhen Co., Ltd.*, China
WANG Ke, *Shanghai Jiao Tong University*, China
WANG Sifeng, *Qufu Normal University*, China
WANG Weiwei, *Xidian University*, China
WANG Zhaobo, *Shanghai Jiao Tong University*, China
WANG Zilong, *Hong Kong Polytechnic University*, Hong Kong
WU Renzhi, *Georgia Institute of Technology*, USA
XIE Qi, *Hangzhou Normal University*, China
XU Jiafeng, *China University of Geosciences*, China
YANG Xin-She, *Middlesex University*, UK
YANG Yongquan, *Institute of Sciences4AI*, China
YANG Ziyuan, *Chinese University of Hong Kong*, Hong Kong
YUAN Feng, *Tianjin University*, China
ZHANG Dehua, *Henan University*, China
ZHANG Qinli, *Chizhou University*, China
ZHANG Weidong, *Henan Institute of Science and Technology*, China
ZHANG Yongliang, *Zhejiang University of Technology*, China
ZHU Zhengyu, *Zhengzhou University*, China
ZOUBIR Abdeljalil, *University Mohammed VI Polytechnic*, Morocco

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:

فريق دعم هيئة التحرير

عنوان المجلة

الموقع الإلكتروني: www.jjcit.org

البريد الإلكتروني: jjcit@psut.edu.jo

العنوان: جامعة الاميرة سمية للتكنولوجيا، شارع خليل السكاكت، الجبيهة، عمان، الأردن.

صندوق بريد: 1438 عمان 11941 الأردن

هاتف: 962-6-5359949+

فاكس: 962-6-7295534+

جامعة الأميرة سميّة للتكنولوجيا
Princess Sumaya University for Technology

صندوق دعم البحث العلمي والابتكار
Scientific Research and Innovation Support Fund

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات

كانون الأول ٢٠٢٥ المجلد ١١ العدد ٤

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