**In the Name of God**

# Journal of
## Information Systems & Telecommunication
**Vol. 14, No.1, January-March 2026, Serial Number 53**

**Research Institute for Information and Communication Technology**
**Iranian Association of Information and Communication Technology**
**Affiliated with: Academic Center for Education, Culture and Research (ACECR)**

**Manager-in-Charge:** Dr. Ali Mokhtarani, ACECR, Iran
**Editor-in-Chief:** Dr. Masoud Shafiee, Amir Kabir University of Technology, Iran

**Editorial Board**
Dr. Abdolali Abdipour, Professor, Amirkabir University of Technology, Iran
Dr. Ali Akbar Jalali, Professor, Iran University of Science and Technology, Iran
Dr. Alireza Montazemi, Professor, McMaster University, Canada
Dr. Ali Mohammad-Djafari, Associate Professor, Le Centre National de la Recherche Scientifique (CNRS), France
Dr. Hamid Reza Sadegh Mohammadi, Associate Professor, ACECR, Iran
Dr. Mahmoud Moghavvemi, Professor, University of Malaya (UM), Malaysia
Dr. Mehrnoush Shamsfard, Associate Professor, Shahid Beheshti University, Iran
Dr. Omid Mahdi Ebadati, Associate Professor, Kharazmi University, Iran
Dr. Rahim Saeidi, Assistant Professor, Aalto University, Finland
Dr. Ramezan Ali Sadeghzadeh, Professor, Khajeh Nasireddin Toosi University of Technology, Iran
Dr. Sha'ban Elahi, Professor, Vali-e-asr University of Rafsanjan, Iran
Dr. Shohreh Kasaei, Professor, Sharif University of Technology, Iran
Dr. Habibollah Asghari, Associate Professor, ACECR, Iran
Dr. Zabih Ghasemlooy, Professor, Northumbria University, UK
Dr. Hadi Aliakbarian, Associate Professor, Khajeh Nasireddin Toosi University of Technology, Iran

**Executive Editor:** Dr. Fatemeh Kheirkhah
**Executive Manager:** Mahdokht Ghahari
**Print ISSN**: 2322-1437
**Online ISSN**: 2345-2773
**Publication License:** 91/13216
**Editorial Office Address:** No.5, Saeedi Alley, Kalej Intersection., Enghelab Ave., Tehran, Iran,
P.O.Box: 13145-799 Tel: (+9821) 88930150 Fax: (+9821) 88930157
E-mail: info@jist.ir , infojist@gmail.com
URL: jist.acecr.org

## Indexed by:
- SCOPUS www. Scopus.com
- Islamic World Science Citation Center (ISC) www.isc.gov.ir
- Directory of open Access Journals (DOAJ) www.Doaj.org
- Scientific Information Database (SID) www.sid.ir
- Regional Information Center for Science and Technology (RICeST) www.ricest.ac.ir
- Magiran www.magiran.com

**Publisher:**
Iranian Academic Center for Education, Culture and Research **(ACECR)**

This Journal is published under scientific support of
Advanced Information Systems (AIS) Research Group and
Telecommunication Research Group, ICTRC

## Acknowledgement

JIST Editorial-Board would like to gratefully appreciate the following distinguished referees for spending their valuable time and expertise in reviewing the manuscripts and their constructive suggestions, which had a great impact on the enhancement of this issue of the JIST Journal.

**(A-Z)**

- Alaeiyan, Mohammad Hadi, K.N. Toosi University of Technology, Tehran, Iran
- Asghari, Seyed Amir, Kharazmi University, Tehran, Iran
- Azarkasb, Seyed Omid, K.N. Toosi University of Technology, Tehran, Iran
- Azadimotlagh, Mehdi, Persian Gulf University, Bushehr, Iran
- Al-Qurabat, Ali Kadhum, Al-Mustaqbal University, Iraq
- Behmanesh, Ali, Iran University of Medical Sciences, Tehran, Iran.
- Ebadati, Omid Mahdi, Kharazmi University,Tehran, Iran
- Faili, Heshaam, Tehran University, Tehran, Iran
- Farsi, Hassan, University of Birjand, South Khorasan, Iran
- Kashef, Seyed Sadra, Urmia University, Urmia, Iran
- Kheirkhah, Fatemeh, ACECR, Tehran, Iran
- Kuchaki Rafsanjani, Marjan, Shahid Bahonar University, Kerman, Iran
- Kolahkaj, Maral, Islamic Azad University, Karaj Branch, Iran
- Moradi, Gholamreza, Amirkabir University,Tehran, Iran
- Mohammadi, Mohsen, University in Esfarayen, North Khorasan, Iran
- Mirroshandel, Seyed Abolghasem, University of Guilan, Rasht, Iran
- Razavi, Seyyed Mohammad, University of Birjand, South Khorasan Province, Iran
- Rastegar, Abbasali, Semnan University, Semnan, Iran
- Shamsi, Mahboubeh, University of Qom, Iran
- Soleimanian Gharehchopogh, Farhad, Islamic Azad University Urmia, Iran
- Shahbahrami, Asadollah, Guilan University, Guilan, Iran
- Tourani, Mahdi, University of Birjand, South Khorasan, Iran
- Vahabie, Abdol-Hossein, Tehran University, Tehran, Iran
- Verij Kazemi, Mohammad, Institute of Higher Education Pooyandegan Danesh., chalous, Mazandaran, Iran

# Table of Contents

Journal of Information Systems and Telecommunication
Vol.14, No.1, January-March 2026, 1-15
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

# Robust Multi-Source Deep Transfer Learning for IoT Unknown Intrusion Detection under Data Scarcity

Amirhossein Hojjatinia [1], Ali Maroosi[1*] , Arash Deldari[1]

[1].Department of Computer Engineering, University of Torbat Heydarieh, Torbat Heydarieh, Iran.

Received: 05 Aug 2025/ Revised: 04 Apr 2026/ Accepted: 13 May 2026

**Abstract**

The rapid expansion of the internet of things (IoT) has heightened the need for robust intrusion detection systems that can identify previously unseen cyber threats. Traditional approaches often struggle with novel attack patterns, leading to decreased detection rates and increased vulnerability. To address this limitation, we propose an innovative framework that combines multi-source transfer learning with autoencoders to detect unlabeled and unknown attack types with good accuracy. Unlike prior methods that rely on single-source transfer learning or basic feature fusion, our approach introduces two novel techniques: the Concurrent Feature Fusion Model (CoFFM) and the Cascading Feature Fusion Model (CaFFM). In CoFFM approach information of the different sources transferred in concurrent manner and in CaFFM information of sources is transferred as cascade manner to the target domain. These models, along with an enhanced Unified Feature Fusion Model (UFFM), utilize autoencoders to enhance adaptability across diverse feature domains. Experimental results on the datasets demonstrate that CoFFM achieves 98.13% accuracy, outperforming non-transfer learning methods (92%) and the best single-source transfer learning (94%). CoFFM achieves a 12.24% performance gain over baseline methods when trained on only 10% of available data (random sampling), demonstrating strong robustness under data scarcity conditions.

**Keywords:** Internet of Things; Intrusion Detection; Unknown Attacks; Transfer Learning; Multi-ource; Autoencoder.

## 1- Introduction

The internet of things (IoT) is rapidly expanding, and the number of connected devices is continually increasing. As the number of networked devices grows, Ensuring the security of these networks against attackers has become a critical concern [1]. To counter network threats, intrusion detection systems (IDS) have been developed, capable of detecting attacks using various techniques. One of the most critical challenges in this field is identifying unlabeled or unknown attacks [2, 3]. Unknown attacks are evolving each year, it takes many days to fully determine the pattern of a new and unknown attack [4]. Most researchers have focused on detecting known attacks, while unknown attacks have received less attention. When a network is prepared to handle a previously identified attack, new data with an unseen pattern can suddenly enter the network. Consequently, the IDS may fail to recognize this new data, putting the network at risk of being compromised by the new attack. Currently, detecting such attacks primarily relies on clustering methods [3].

This paper proposes an approach that leverages deep learning and autoencoders to significantly enhance intrusion detection systems (IDS) in accurately identifying unknown attacks. In this study, unknown attacks refer to malicious network activities whose patterns are not present during the training phase and have not been previously labeled or observed by the intrusion detection system. For our work we used different structure of autoencoder in our system. Thus, we used normal samples to trained autoencoders and not used attack samples for training or validation. From detection system view point all attacks are unknown attacks. During testing both normal and abnormal (unknown attack) are given to system, Trained autoencoders that trained to reconstruct just normal samples appropriately can reconstruct normal samples with low error while attacks are reconstructed with high error. The system detect samples that reconstructed with high error reconstruction as attack (unknown attack). This approach uses transfer learning from three source domains, where knowledge from each domain is transferred to and utilized in the target domain.

The contributions of this research are as follows:

✉ **Ali Maroosi**
ali.maroosi@torbath.ac.ir

1. In the field of transfer learning, this study is the only one that uses a multi-source approach with autoencoders to detect unknown attacks.
2. The study employs a multi-source transfer learning strategy.
3. Three methods, CaFFM, CoFFM, and UFFM are proposed to identify unknown attacks.
4. The proposed approaches demonstrate improved accuracy even with limited data.

The rest of this paper is organized as follows: Section 2 presents related work, and Section 3 outlines the proposed methods. Also, Section 4 describes the simulation results. Finally, the conclusion section is given in section 5.

## 2- Related work

The identification of unknown attacks has been examined in various domains, including advanced, network-connected vehicles. Ensuring the security of data in these vehicles is crucial for the safety of drivers and passengers, making it one of the most important aspects of a connected car. An anomaly detection system based on neural networks is presented in [2]. This system uses Long Short-Term Memory (LSTM) and single-source transfer learning to create a model that can detect both known and unknown attacks. This method primarily focuses on identifying attacks that were not used during model training to enhance the approach for detecting unknown attacks. However, multi-source transfer learning could be employed in this work to achieve further improvement.

Further highlighting the importance of attack detection in vehicles, the study [5] introduces a model capable of identifying unknown attacks. This method implements a deep learning-based model trained exclusively on normal data. To develop an effective model against cyber-attacks, the training was conducted using synthetic and random data. However, transfer learning could be integrated into this study to improve the work.

A zero-shot learning method based on an autoencoder is presented in [2] using the NSL-KDD dataset to identify unknown attacks. This work achieves an accuracy of 80.30%. Zero-shot learning identifies an unknown attack solely based on its semantic description. An autoencoder is utilized to improve this method. When unknown data enters the network, this method uses the established known attack patterns to identify the new pattern as an unknown attack.

The approach in [6] presents a convolution model for an intrusion detection system implemented using transfer learning. In this method, two ConvNet models are utilized. The first is considered the base model, and the second is the target model. After training the dataset with the first ConvNet and acquiring the relevant knowledge, the data is transferred to the target model and trained with the second ConvNet. This method has achieved an acceptable level of accuracy on the NSL-KDD dataset.

Kang and Shen [2] introduce a method for detecting abnormal anonymous messages in vehicles, which combines a Long Short-Term Memory (LSTM) network and a Generative Adversarial Network (GAN) to generate fake abnormal messages. The knowledge is then transferred using transfer learning to the target model, which is composed of an LSTM network. In the target model, only a small amount of data is available, and the transferred knowledge can help this small amount of data detect abnormal messages.

Given the significant growth in data exchange within the network, it is essential to provide an intrusion detection system that can handle the vast amounts of data being generated. Yang et al. [7] proposed a method that introduces a multi-classification model for intrusion detection based on feature reconstruction and adaptation. In this method, computations are performed on smaller scales at edge nodes, resulting in highly accurate multi-class classification.

In the field of the internet of vehicles, a deep learning-based intrusion detection system has been effectively implemented in [8] that ensures the security and privacy of vehicles.

Lilhore et al. [9] analyze the security issues concerning internet-connected systems in the industry and propose a method to address these problems by combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) with multi-source transfer learning. The Grey Wolf Optimization algorithm is used for fine-tuning the parameters of the CNN. The knowledge gained from the training in the previous model is then transferred to the main model, where it is combined with LSTM.

Yutao et al. [10] propose a multi-source transfer learning-based method that has significantly improved the detection accuracy for identifying damaging attacks and has managed to reduce the computational resource scarcity and the detection time.

Further similar works [11] combine multi-source transfer learning with five CNN models using two datasets to achieve optimal results. In this research, numerical data is transformed into image sizes to achieve better results.

Zhao et al. [12] conducted research on unknown attacks on the network, using unlabeled data instead of labeled data, which requires higher processing speeds. They developed a model based on transfer learning, named Hierarchical Transfer Learning, which can find relationships between known and unknown attacks and identify patterns of unknown attacks from known ones. This work uses optimization equations to align source-target domains, but struggles with dynamic IoT attacks and data scarcity. Deep learning approaches automatically learn domain-invariant features, enabling robust adaptation to unknown threats.

To ensure network system security in vehicles, Khatri et al. [13] introduced a transfer learning-based method that enhances network resilience against various attacks. This approach enables the detection system to effectively distinguish between abnormal and normal messages and issue warnings in the event of intrusions. The model also achieves a reduction in training and testing time compared to other related methods.

Tien et al. [13] propose a support vector machine (SVM) and deep learning method using autoencoders for anomaly detection and transfer learning in the IoT. One of the key positive aspects of this approach is that it provides acceptable feedback and performance for identifying anomalies in factories. However, it is not considered cross-domain information to improve performance.

Elubeyd et al. [14] present a method for real-time detection of various vulnerabilities and attacks using an innovative multi-faceted deep transfer learning approach in Software-Defined Networks (SDNs). This method offers flexibility and scalability, enabling it to maintain high performance in the data-intensive environments of SDNs.

Wang et al. [15] present a groundbreaking concept centred on zero-shot learning utilizing single-source transfer learning. The majority of intrusion detection methods today rely on labeled data, posing a challenge in identifying unknown attacks, especially when information is limited. The refined approach proposed in this paper harnesses zero-shot learning and autoencoders to address this challenge. However, one drawback of zero-shot learning is domain shift, where changes in the domain disrupt the algorithm's performance and lead to declining accuracy rates. To mitigate this issue, the researchers in this paper have incorporated an autoencoder structure.

Zachos et al. [16] developed an anomaly-based intrusion detection system specifically for internet of medical things environments utilizing One-Class support vector machines. While their approach demonstrated effective real-time detection capabilities, the system was constrained by its reliance on single-source data, consequently failing to leverage potential benefits from cross-domain knowledge transfer through techniques such as transfer learning.

The work by Logeswari et al. [17] proposed a quantum-inspired particle swarm optimization combined with an adaptive neuro-fuzzy inference system, followed by a multi-stage classification pipeline using capsule networks (CapsNets) and attention-augmented recurrent neural networks (RNNs) for known and unknown attacks. [18] employed attention-based Transformer architectures but relied solely on single-source training, overlooking the potential benefits of multi-source feature fusion.

The study by Gao et al. [19] reduced false alarms using a memory-enhanced autoencoder, yet their model operated on isolated smart grid data without leveraging multi-source correlations. Similarly, [20] employed long short-term memory (LSTM) and GRU in software-defined networking for unknown attacks but did not explore transfer learning.

While Hernandez-Jaimes et al. [21] improved detection via attention mechanisms, their single-source training limited adaptability to heterogeneous IoT environments. The survey by Walling and Lodh [22] confirmed that most machine learning-based intrusion detection systems neglect multi-source transfer learning.

Reviewing previous work in this research domain, as depicted in Table 1, it becomes apparent that none of the studies have delved into the utilization of multi-source transfer learning and autoencoders for identifying unknown attacks. Hence, exploring this subject within the realm of unknown attack detection appears highly promising. Consequently, our research focuses on the aspects as mentioned earlier.

Table 1: Comparison of Existing Work in Transfer Learning for Identifying Unknown Attacks

| Reference | year | Network type | Transfer learning | Multi-source | Detection of Unknown Attacks |
|---|---|---|---|---|---|
| [6] | 2019 | CNN | Yes | No | No |
| [12] | 2019 | Clusters | Yes | No | Yes |
| [5] | 2019 | CNN & DNN | No | No | Yes |
| [23] | 2020 | CNN | Yes | No | Yes |
| [3] | 2020 | Autoencoder | No | No | Yes |
| [2] | 2021 | LSTM | Yes | No | Yes |
| [8] | 2021 | CNN | Yes | No | No |
| [13] | 2021 | Autoencoder | Yes | No | No |
| [15] | 2021 | Autoencoder | Yes | No | Yes |
| [11] | 2022 | CNN | Yes | Yes | No |
| [10] | 2022 | CNN | Yes | Yes | No |
| [9] | 2023 | LSTM & CNN | Yes | Yes | No |
| [14] | 2023 | LSTM | Yes | Yes | No |
| [24] | 2023 | Hybrid LSTM & CNN | Yes | No | No |
| [25] | 2024 | CNN | Yes | No | No |
| [7] | 2024 | LSTM, CNN & Autoencoder | Yes | Yes | No |
| [16] | 2025 | One-Class SVM | No | No | Yes |
| [17] | 2025 | CapsNets + Attention RNNs | No | No | Yes |
| [18] | 2025 | Transformer | Yes | No | No |
| [19] | 2025 | Autoencoder | No | No | Yes |

| | | | | | |
|---|---|---|---|---|---|
| [20] | 2025 | LSTM+GRU | No | No | Yes |
| [21] | 2025 | Attention-based | No | No | Yes |
| **Proposed method** | **-** | **Autoencoder** | **Yes** | **Yes** | **Yes** |

## 3- Proposed Method

This section provides the proposed approach. In this approach, transfer learning is implemented using three source domains. Each source domain is trained with normal data, and its knowledge is ultimately transferred to the target domains. Each source completes its training using autoencoder structures with a small number of nodes. To identify unknown attacks and optimize the proposed approach, no non-normal data is used for training in either the source domains or the target domain.

In the proposed experimental setup, unknown attacks are strictly excluded from both the training and validation phases. The autoencoder models in the source and target domains are trained and validated exclusively using normal traffic data to prevent information leakage and ensure unbiased learning. Unknown attacks are introduced only during the testing phase, where the trained models are evaluated on a mixture of normal traffic and all attack samples. This design reflects realistic deployment scenarios, where intrusion detection systems encounter previously unseen attacks only during operation.

Finally, to test the system, a mix of normal data and all attack data is fed into the network to evaluate accuracy.

This research introduces three methods—CaFFM, CoFFM, and UFFM—for identifying unknown attacks. A key component of these methods is the autoencoder, which plays a crucial role in feature extraction and anomaly detection, as described below.

An autoencoder is a type of neural network designed for unsupervised learning, primarily used for feature extraction and anomaly detection. It consists of three main components: an encoder (E), a latent space (L), and a decoder (D), as shown in Fig. 1.

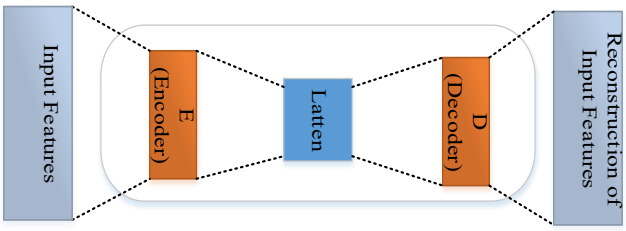



Fig 1. The basic structure of autoencoder

The encoder compresses input data into a lower-dimensional representation, capturing essential features. The decoder then reconstructs the data from this compressed form. By comparing the reconstructed output with the original input, the autoencoder calculates the reconstruction error or loss, typically using the Mean Squared Error (MSE). High reconstruction error indicates significant deviation from normal data, signalling potential anomalies or unknown attacks. In this study, the autoencoder learns patterns from normal network traffic, enabling effective detection of deviations without prior knowledge of attack signatures. This capability makes autoencoders ideal for intrusion detection systems (IDS), where unknown threats must be identified swiftly and accurately.

In this study, anomaly detection is performed based on the reconstruction error computed using the Mean Squared Error (MSE). The decision threshold is determined using a validation set that contains only normal traffic. Specifically, the threshold is defined as the mean reconstruction error plus a scaled standard deviation obtained from the validation data. Any sample whose reconstruction error exceeds this threshold is classified as an attack. The threshold remains fixed during the testing phase and is determined separately for each dataset to account for domain-specific characteristics.

In our study, each proposed method (CaFFM, CoFFM, and UFFM) determines its reconstruction error threshold independently. Specifically, for each method, the threshold is computed using its own validation set containing only normal samples. This ensures that model-specific characteristics and feature distributions are appropriately considered. The thresholds remain fixed during testing and are not shared across models.

The autoencoder is trained only on normal network traffic to learn a compressed representation of typical patterns. During testing, each input sample is passed through the autoencoder, and the reconstruction error (Mean Squared Error, MSE) is computed between the original input and the reconstructed output.

- If the reconstruction error is below a predefined threshold, the sample is classified as normal.
- If the reconstruction error is above the threshold, the sample is classified as an anomaly, which may correspond to an unknown attack. This decision rule allows the system to detect deviations from normal behavior without requiring prior knowledge of specific attack signatures, making it particularly suitable for unknown attack detection.

The threshold for classifying a sample as normal or anomalous is determined empirically using the reconstruction errors of normal samples in the training or validation set. In this study, the threshold was set asthreshod equal to$\mu_{\text{train}} + k \cdot \sigma_{\text{train}}$. Where $\mu_{\text{train}}$and $\sigma_{\text{train}}$are the mean and standard deviation of the reconstruction errors of the

normal training data (validation data during training), and $k = 1$ is a scaling factor chosen to balance the trade-off between false positives and false negatives. This strategy ensures that most normal samples fall below the threshold, while samples with unusually high reconstruction errors—likely corresponding to unknown attacks—are flagged as anomalies.

The presented methods are designed for *n* sources and one target, and the layers of the autoencoders can be adjusted according to the number of input features of the target autoencoder, making the approach generalizable. However, for simplicity and without loss of generality, we have used the IDSAI dataset [26], which is the target data in this study, as previous approaches could not achieve high accuracy rates in detection.

The details of the three proposed methods are provided in the following subsections. The fundamental setup for each source domain is standardized to ensure consistent simulation environments. The simulation parameters utilized in the study are presented in Table 2, that are obtained experimentally.

Table 2: Simulation parameters

| Parameters | Description |
|---|---|
| **Learning rate** | 5e-4 |
| **Optimizer** | Adam |
| **Loss function** | Mean Square Error (MSE) |
| **Metrics** | MSE |
| Training data ratio | 0.7 of normal data |
| Validation data ratio | 0.1 of training data |
| test data ratio | 0.3 of normal data+ attacks |
| **Batch size** | One twentieth of the data |
| **Maximum Epochs** | 500 |
| **Number of datasets utilized** | 4 |

In our experiments, we used Adam optimizer with a learning rate of 5e-4, and the loss function was MSE. The number of epochs was set to 500. To prevent overfitting, early stopping based on the validation loss was employed with a patience of 20 epochs. No additional dropout or weight decay was applied, as preliminary experiments showed that early stopping provided sufficient regularization.

Hyper-parameters, including learning rate, batch size, and number of nodes in each autoencoder layer, were tuned empirically to achieve the best performance on the validation set.

## 3-1- Cascading Feature Fusion Model

In the cascading feature fusion model (CaFFM) method, *n* sources are individually normalized and trained using autoencoders. Each source is trained exclusively with normal data. The resulting models are then transferred to the target domain and frozen. Subsequently, these models are fed into the network in series, and the new model is tested with 30% normal data and all attack data.

Fig. 2 illustrates the architecture of the CaFFM. In this approach, the dataset considered as the source domain is first pre-processed. All data are normalized between 0 and 1, and missing data are assigned using the median method. The data is then trained using a network composed of an autoencoder, which includes encoder layers, a latent layer, and decoder layers.

In Fig. 2, E represents the encoder, D represents the decoder, and L represents the latent layer.$D_j^{si}$ denotes the *j-th* layer of the decoder from source *i*, and $E_j^{si}$ denotes the *j-th* layer *(j = 1, 2, 3)* of the encoder from source *i (i = 1, …, n).* The structure of the autoencoder networks is symmetric between the encoder and decoder sections, meaning the number of nodes in corresponding layers is equal. For instance, the number of nodes in layer $D_k^{si}$ equals the number of nodes in layer $E_j^{si}$ when $k = j$. Additionally, $D^t$ and $E^t$ are the first layers of the encoder and decoder for the target domain, respectively.

As shown in Fig. 2, n pre-trained models, labeled *pre 1, pre 2,…, pre n* represent the source models. For source *i (i = 1, 2, …, n),* the number of nodes in layers $D_1^{si}$, $D_2^{si}$ and $D_3^{si}$ is equal to the number of nodes in layers $E_1^{si}$, $E_2^{si}$ and $E_3^{si}$, respectively. The autoencoder structure is identical for all sources except for the first and last layers. In other words, the number of nodes in source *i* for layers $D_2^{si}$ and $D_3^{si}$ is equal to the number of nodes in source *j* (for layers $D_2^{sj}$ and $D_3^{sj}$). For the encoder part, datasets with different feature numbers require that the first layer, E1si, and the last layer, D1si, match the number of features of each source, differing from those of other sources.

After training each source domain with 70% of normal data, the knowledge is transferred to the target domain. The same process and layer equivalence apply to the other source domains, and their knowledge is transferred to the target domain. In the target domain, normal data is input into the network and normalized like the source domains.

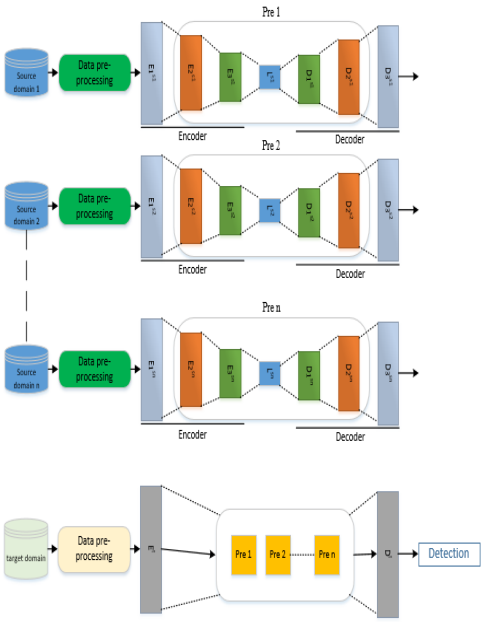



Fig 1. The architecture of CaFFM

Then, in cascade, the pre-trained models (*pre 1 to pre n*) are incorporated into the new network. This network, corresponding to the target domain, consists of an encoder $E^t$ and a decoder $D^t$, each with several nodes equal to the number of features in the target domain dataset. According to the standard autoencoder setup, the number of nodes in $D^t$ equals the number of nodes in $E^t$. Finally, the newly created model is used by the intrusion detection system to identify unknown attacks. In this method, the test data comprises 30% normal data and all attack data from the target domain dataset.

## 3-2- Concurrent Feature Fusion Model

In the concurrent feature fusion model (CoFFM) method, similar to the CaFFM, *n* models in the source domain are trained using autoencoders. The training structure in the source domain is the same as in the CaFFM, so it will not be repeated here. Instead of arranging the pre-trained source models in a cascade approach, they are placed in a concurrent approach. Data from the first layer $E^t$ of the target domain is fed into all these models in a concurrent approach. The outputs of the CoFFMs then feed into the final layer $D^t$ in the target domain.

In the target domain, the proposed model enables the use of each pre-trained source model to the extent that it best matches the features of the target data. For example, if source model *i* aligns most closely with the target domain data, the weights connecting the target domain input to model *i* will be greater. This approach is expected to enhance the performance of the target domain model.

Fig. 3 illustrates the architecture of the CoFFM. The concurrent configuration enables the integration of knowledge from multiple pre-trained models simultaneously, which can be especially beneficial if different source models excel in other aspects relevant to the target domain. Consequently, this method aims to improve the accuracy and robustness of the target domain model in identifying unknown attacks.

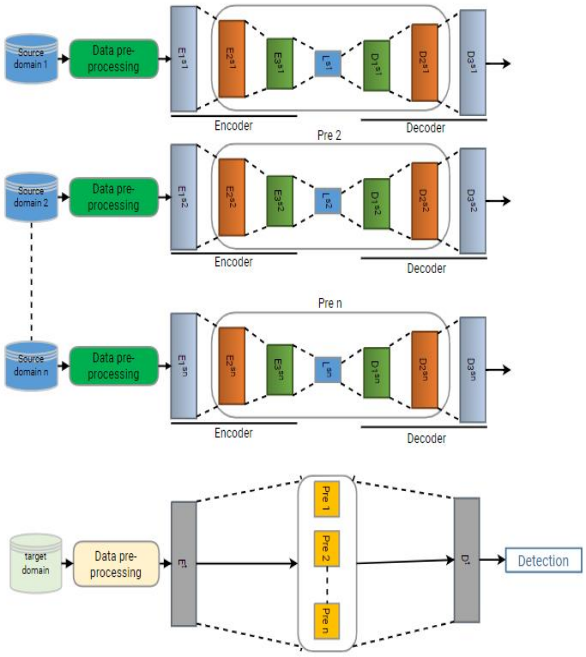



Fig 2. The architecture of the CoFFM

## 3-3- Unified Feature Fusion Model

The training structure for the autoencoder in the unified feature fusion model (UFFM) is similar to that in other methods. However, the key difference is that a single model handles the core training of all n source domains called the UFFM. In each iteration loop, the weights of the UFFM are updated using the dataset from one of the source domains. The process then moves to the next source domain, and the same UFFM is trained again. This cycle repeats until all source domains have updated the UFFM, at which point the process starts over with the first source domain. This loop continues until the UFFM is sufficiently updated. Finally, the UFFM, which has been refined through several iterations by all *n* sources, is saved.

The resulting model is then transferred to the target domain and frozen for use. In the target domain, the model is trained on normal data while distinguishing between

attacks and non-attacks. For testing, the remaining 30% of the normal data and all attack data are used. The pseudo code of this approach is as follows:

Pseudo code Overview of UFFM model Training

Consider a constant *Sub_Max_Iteration*
(we empirically consider *Sub_Max_Iteration* = 10)
Set *Sub_max_epochs =max_epochs/Sub_Max_Iteration*
Inputs: Source datasets $S_1$, $S_2$, ..., $S_n$
Initialize: UFFM model weights
*n_epoch*=0
**repeat** until stopping criterion (e.g., convergence or n_epoch >= max_epochs):
  **for** each source dataset $S_i$ in {$S_1$, ..., $S_n$}:
    Train UFFM using normal data from $S_i$ for *Sub_max_epochs* epochs.
    Update UFFM weights
  **end for**
  *n_epoch=n_epoch+ Sub_max_epochs*
**end repeat**
Save the refined UFFM model
Transfer the UFFM to the target domain and freeze weights
Train on target normal data for anomaly detection

Fig. 4 shows the architecture of the UFFM method. The descriptions of the layers in the UFFM are the same as those in the CaFFM, explained in previous sections. The difference is that in this model, the layers $D_3^{si}$, $E_3^{si,}$ and $L^{si}$ are trained jointly across all sources within the UFFM. After completing the training and finishing the iteration loop over the *n* models, the "com" part, which is common to all domains, is extracted from the loop and saved. This final model is then transferred to the target domain and frozen as shown in Fig. 4.

.

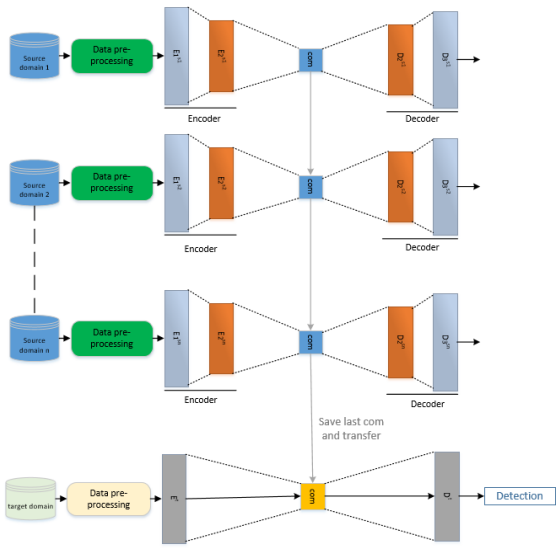



Fig 3. The architecture of UFFM

## 4- Result and Discussion

The proposed approach was implemented using Python on an Intel® Core™ i5-8250U CPU @ 1.60 GHz, 1.80 GHz system with 32 GB of RAM. This simulation utilized the Jupyter Notebook environment and Python deep learning libraries, including Keras, Scikit-learn, and TensorFlow.

Four datasets were used for the target and source domains to evaluate the transfer learning model using autoencoders: IDSAI [26], UNSW-NB15[27], NSL KDD-train [28], and TON IoT train-test network [29]. The number of samples for these datasets can be seen in Table 3.

Table 3. Samples number of different datasets

| Dataset | Samples number | Normal samples number | Attack samples number | Features number |
|---|---|---|---|---|
| UNSW-NB15 (20% of data) | 82332 | 37000 | 45332 | 43 |
| NSL KDD | 125973 | 67343 | 58530 | 21 |
| TON IoT | 461043 | 30000 | 161043 | 43 |
| IDSAI | 1000000 | 500000 | 500000 | 26 |

The autoencoder consists of several layers: encoder layers, a latent layer, and decoder layers. This method considers three source domains. Based on various experiments and considering the feature dimensions of the datasets, we concluded that the number of nodes in the layers for source *i (i = 1, 2, 3)* for all methods should be as follows: $E_2^{si}$, $E_3^{si}$, *L*, $D_3^{si}$, $D_2^{s}$ have *20, 15, 8, 15* and *20* nodes respectively. Additionally, the number of nodes in layers $E_1^{si}$ and $D_1^{si}$ (where the number of nodes in the encoder and decoder parts is equal for each source) is equal to the number of features in the dataset for each source. Specifically, the number of nodes in $E_1^{s3}$, $E_1^{s2,}$ and $E_1^{s1}$ is *41*, *44*, and *44,* respectively.

To outline the structure of each method and the details of its neural network layers, we will describe each one in this section. For the CaFFM, Table 4 shows the structure of the source domain, including three models: *model_1*, *model_2*, and *model_3*. Note that these models encompass everything between the input and output, i.e., from *encoder_2* to *decoder_2*. Table 5 specifies the structure of the target domain.

Table 4: Details of the source domain models' structure for the CaFFM and CoFFMs

| Parameters | The output structure of the first pre-trained model Pre 1 (model_1) | The output structure of the second pre-trained model Pre 2 (model_2) | The output structure of the third pre-trained model Pre 3 (model_3) |
|---|---|---|---|
| encoder_1 input (Dense) | (None, 44) | (None, 44) | (None, 41) |
| encoder_2 (Dense) | (None, 20) | (None, 20) | (None, 20) |
| encoder_3 (Dense) | (None, 15) | (None, 15) | (None, 15) |
| Latten (Dense) | (None, 8) | (None, 8) | (None, 8) |
| decoder_3 (Dense) | (None, 15) | (None, 15) | (None, 15) |
| decoder_2 (Dense) | (None, 20) | (None, 20) | (None, 20) |
| decoder_1 output (Dense) | (None, 44) | (None, 44) | (None, 44) |

Table 5: The structure of the target domain in the CaFFM method

| Parameters | Output structure |
|---|---|
| encoder_1 input (Dense) | (None, 20) |
| model_1 (Functional) | (None, 20) |
| model_2 (Functional) | (None, 20) |
| model_3 (Functional) | (None, 20) |
| decoder_1 output (Dense) | (None, 20) |

For the neural network layers in the CoFFM, Table 6 details the source domain, while Table 5 outlines the structure of the target domain. It is important to note that the source domains for both the CaFFM and CoFFMs are identical.

Table 6: The structure of the target domain in the CoFFM

| Parameters | Output structure |
|---|---|
| encoder_1 input (Dense) | (None, 20) |
| model_1 (Functional) | (None, 20) |
| model_2 (Functional) | (None, 20) |
| model_3 (Functional) | (None, 20) |
| Concatenating all Pre-trained models | (None, 60) |
| decoder_1 output (Dense) | (None, 20) |

In the final approach, known as the UFFM method, three structures are considered. The first structure, detailed in Table 7, is the UFFM, which updates its weights during training with each source domain. The second structure, shown in Table 8, provides details of the source domain. The third structure presents the details of the target domain. Table 9 illustrates the structure of the target domain for the UFFM.

Table 7: Structural details of the pre-trained UFFM for the source domain in the UFFM

| Parameters | Output structure |
|---|---|
| encoder_1 (Dense) | (None, 15) |
| Latten (Dense) | (None, 8) |
| decoder_1 (Dense) | (None, 15) |

Table 8: Structural details of the encoder model for the source domain in the UFFM

| Parameters | Output structure |
|---|---|
| (Input source 1 with 44 features) encoder_1 input_1 (Dense) | (None, 44) |
| encoder_2 (Dense) | (None, 15) |
| (Unified core) pre-trained-com (Functional) | (None, 15) |
| decoder_2 (Dense) | (None, 15) |
| (Output source 1 with 44 features) decoder_1 output_1 (Dense) | (None, 44) |
| (Input source 2 with 44 features) encoder_1 input_2 (Dense) | (None, 44) |
| encoder_2 (Dense) | (None, 15) |
| Unified core pre-trained-com (Functional) | (None, 15) |
| decoder_2 (Dense) | (None, 15) |
| (Output source 2 with 44 features) decoder_1 (Dense) | (None, 44) |
| (Input source 3 with 41 features) encoder_1 input_3 (Dense) | (None, 41) |
| encoder_2 (Dense) | (None, 15) |
| (Unified core) pre-trained-com (Functional) | (None, 15) |
| encoder_2 (Dense) | (None, 15) |
| (Output source 3 with 41 features) decoder_1 output_3 (Dense) | (None, 41) |

Table 9: The architecture of the target domain in the UFFM

| Parameters | Output structure |
|---|---|
| (In the dataset target with 20 features) encoder_1 input (Dense) | (None, 20) |
| (Unified core) pre-trained-com (Functional) | (None, 15) |
| decoder_1 output (Dense) | (None, 20) |

In assessing the proposed approach, parameters such as *accuracy*, *recall*, *F-score*, and *precision* have been utilized [25, 30-32]. Table 10 presents the evaluation results for the CoFFM and UFFMs, demonstrating a superior improvement rate compared to other approaches such as [23], [33], [34], and [35] that we simulated these approaches for our datasets.

The proposed approach demonstrates the ability to achieve better accuracy with limited data from the IDSAI

dataset, for unknown attacks, compared to when the approach is not utilized.

Table 10: Evaluation parameters for the CoFFM, the UFFM, and the approach without transfer learning

| Evaluation parameters | Recall | F-score | Precision | Accuracy |
|---|---|---|---|---|
| CoFFM | %98.69 | %98.09 | %97.51 | %98.13 |
| UFFM | %94.41 | %93.86 | %93.33 | %94.13 |
| Fan et al. [23] | %92.89 | %93.09 | %93.30 | %94.10 |
| Wang et al. [33] | %93.10 | %92.04 | %91.01 | %92.51 |
| Alrayes et al. [34] | %93.5 | %93.3 | %93.2 | %93.27 |
| Zha et al. [35] | %90.2 | %90.6 | %91.2 | %90.4 |

The proposed method was tested on the IDSAI dataset under five different conditions: using 10%, 30%, 50%, 75%, and 100% of the data. It is important to note that the data were randomly selected from the entire dataset, emphasizing the effectiveness of the proposed methods when training data is scarce. The improvement can be examined in detail in Fig. 5, 6, 7, and 8, which are separated by accuracy, precision, recall, and F-score parameters.

The evaluation across precision, recall, and F-score shows that CoFFMs consistently outperform other models, maintaining stable performance even under reduced data volumes. CaFFMs exhibit lower performance in all metrics, with more noticeable declines as the available data decreases. UFFMs perform reasonably well but experience significant drops in all metrics with smaller datasets due to limitations in learning complex features. The Best Single Model achieves good results with larger datasets but deteriorates substantially when the data volume is limited. Models trained without transfer learning generally have the lowest metrics, with declines particularly evident under scarce data conditions. These observations reinforce the effectiveness of multi-source transfer learning combined with autoencoder structures, highlighting CoFFM's ability to maintain high and stable performance across all evaluation measures even with limited training data.

While the proposed methods (CaFFM, CoFFM, UFFM) achieve high accuracy in detecting unknown attacks, there are several limitations to consider.

1. Binary Classification: Currently, the models classify data only as normal or attack, without distinguishing between different attack types. This limits the system's ability to provide fine-grained insights about specific attack categories. Future work could extend the framework to multi-class classification, differentiating between normal traffic, known attack types, and unknown attacks.
2. Incremental Learning: The present approach does not dynamically update the model when new attack patterns appear after deployment. Implementing incremental or continual learning strategies would enable the IDS to adapt to evolving threats without retraining from scratch.
3. Data Scarcity & Domain Shift: Although multi-source transfer learning improves performance with limited data, extreme scarcity or significant differences between source and target domains may still impact detection accuracy. Future research could explore adaptive thresholding, few-shot learning, or domain adaptation techniques to further enhance robustness.
4. Integration with Security Frameworks: Integrating the proposed methods with other technologies, such as blockchain, could enhance data integrity, reduce susceptibility to adversarial attacks, and improve trust in transferred knowledge.

Overall, addressing these limitations will make the intrusion detection framework more flexible, adaptive, and capable of handling complex real-world IoT environments.

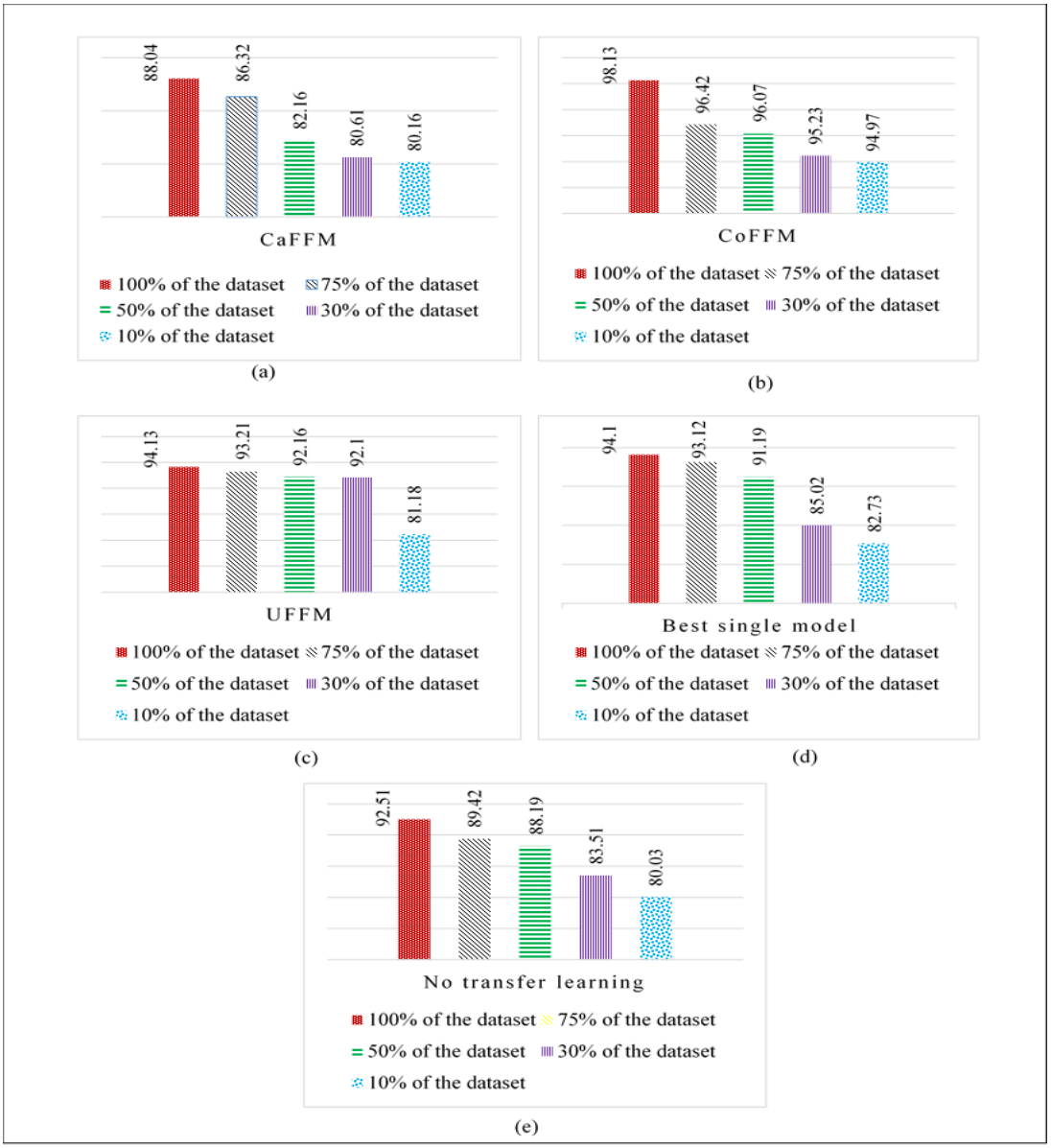



Fig. 4. Accuracy rates of the proposed methods on the IDSAI dataset for detecting unknown attacks

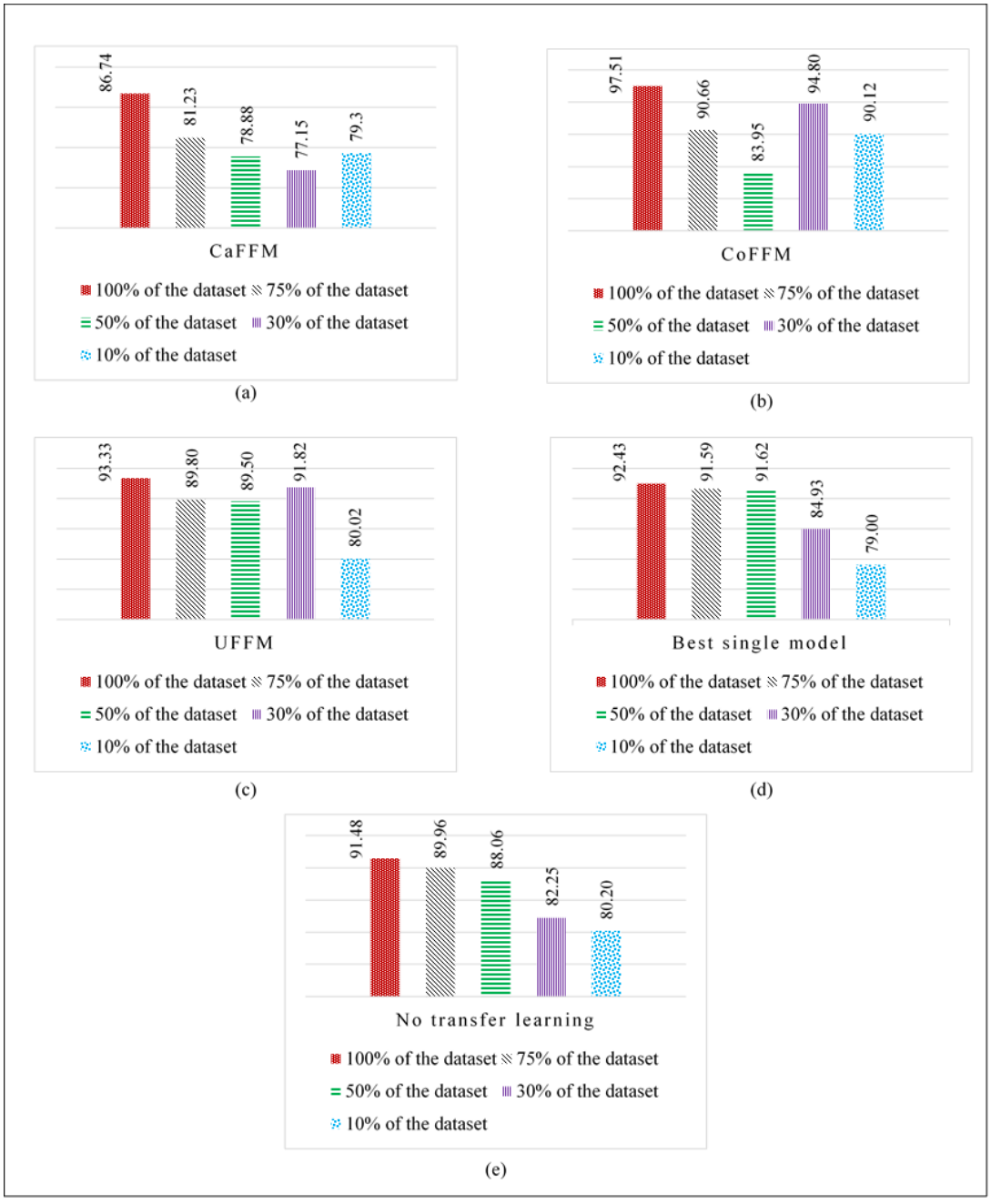



Fig 6. Precision rates of the proposed methods on the IDSAI dataset for detecting unknown attack

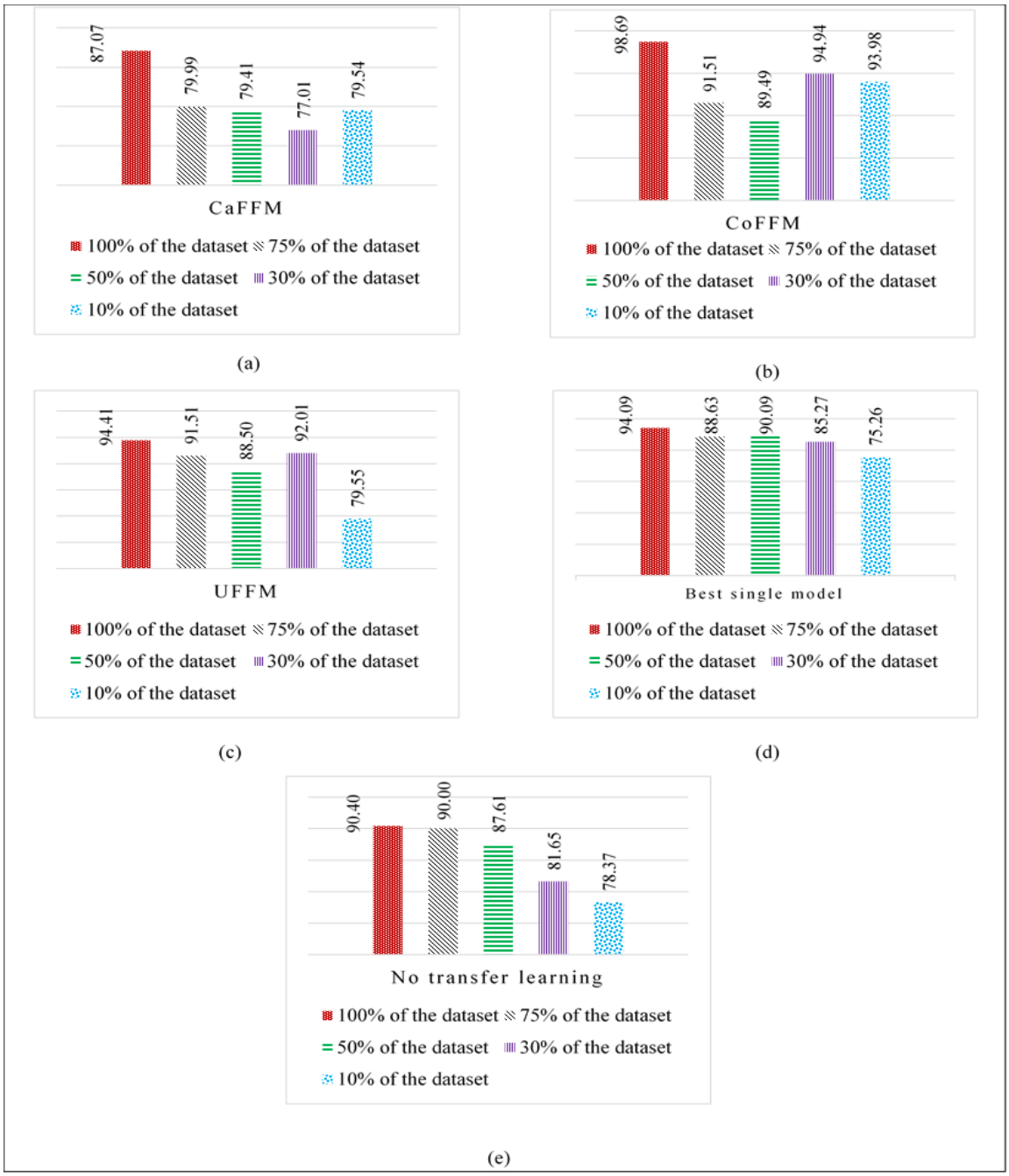



Fig 7. Recall rates of the proposed methods on the IDSAI dataset for detecting unknown attacks

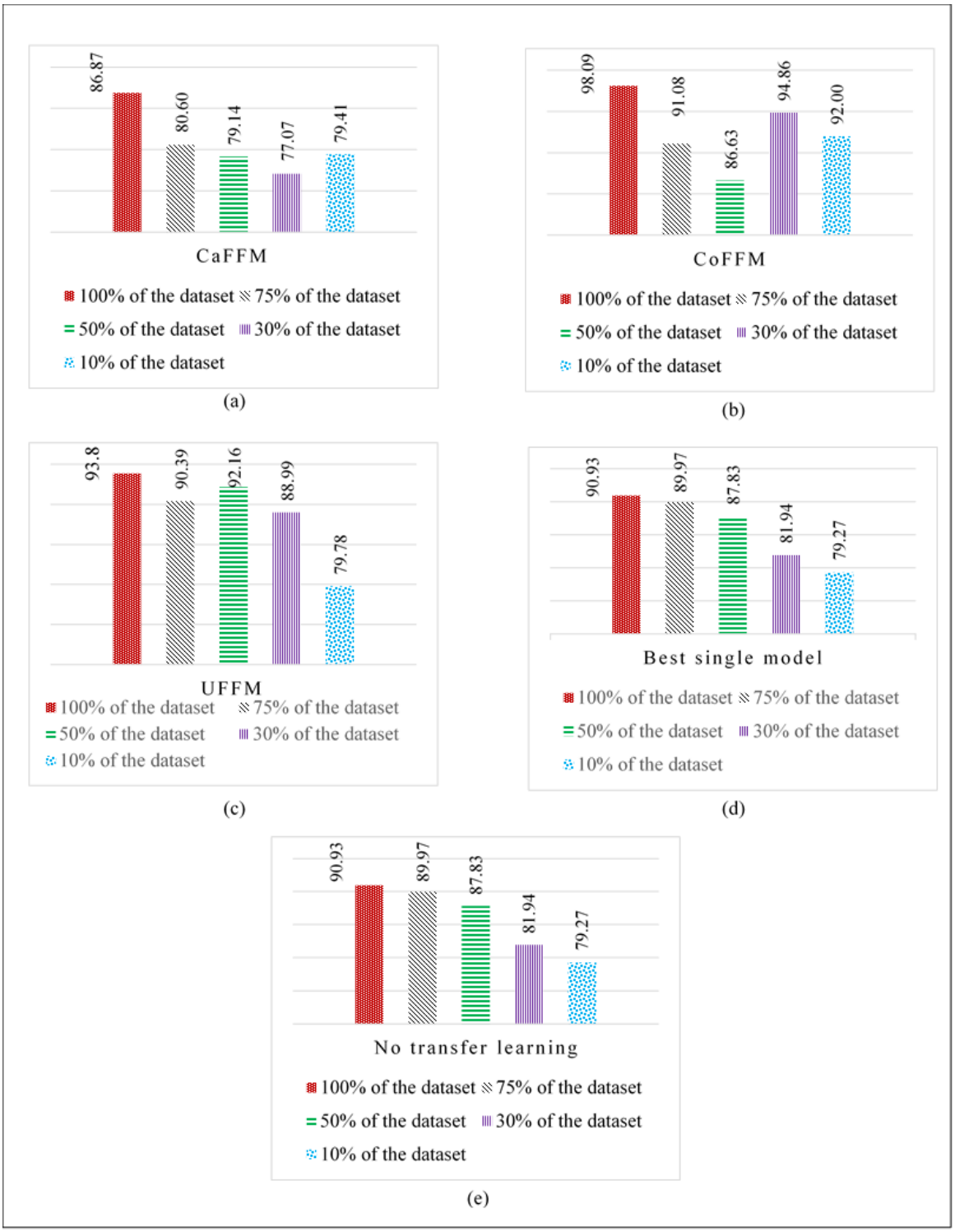



Fig 8. F-score rates of the proposed methods on the IDSAI dataset for detecting unknown attacks

Overall, despite this limitation, our proposed method effectively detected normal traffic during testing and achieved commendable accuracy for identifying unknown attacks. Moreover, it demonstrated adaptability to the varied characteristics of the source domains and exhibited satisfactory performance even with less data.

## 5- Conclusions and Future Work

As the landscape of IoT usage becomes increasingly complex, the sophistication of network intrusion methods grows in tandem. Among the critical challenges in this realm is the prevention of cyber-attacks, a task for which current network systems lack established patterns. Put simply, intrusion detection systems struggle to identify unknown attacks. Addressing this security challenge, multi-source transfer learning emerges as one of the most effective techniques. This paper introduces three approaches – CaFFM, CoFFM, and UFFM – which leverage multi-source transfer learning methodologies to notably enhance the accuracy of identifying unknown attacks compared to conventional methods. In all three methods, there are three source domains. Each domain is transferred to the target domain after being trained on normal data, at which point the target neural network is formed. The resulting model is tested with 30% of the normal data and all the attack data. Notably, the CoFFM, integrating transfer learning and autoencoders, achieved an impressive accuracy rate of 98.13%. Furthermore, it displayed performance enhancements even with limited data from the IDSAI dataset, showcasing its adaptability to diverse dataset features. The expansion of IoT technology has led to an increase in the scope and prevalence of cyberattacks. Consequently, deep learning techniques must deliver higher accuracy than before. Intrusion detection systems currently face a significant challenge that future research must address more thoroughly. This challenge arises because most models proposed for intrusion detection systems are trained with large volumes of data, which are often unlabeled and lack specific features. Additionally, the majority of data generated in real-world environments consists of normal traffic, which can cause transfer learning models to become potentially biased towards normal data. Under these circumstances, attackers can directly target the learning model, replace it with their manipulated version, and transfer it to the target domain, thereby disrupting the intrusion detection system.

One solution to this challenge in future research is to combine blockchain technology with transfer learning techniques. Blockchain possesses a strong capability to analyze data quickly and cost-effectively while maintaining high security, thereby preventing exploitation by cyber attackers. Researchers can enhance the accuracy of their models by integrating the methods presented in this study with blockchain technology. This approach necessitates the development of more precise algorithms with reduced complexity.

## References

[1] M. Moudi, A. Soleimani, and A. Hojjatinia, "A Survey of Intrusion Detection Systems Based On Deep Learning for IoT Data," Journal of Information Systems and Telecommunication (JIST), vol. 3, p. 197, 2024.

[2] L. Kang and H. Shen, "A transfer learning based abnormal can bus message detection system," in 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS), Denver, CO, USA, 2021, pp. 545-553.

[3] Z. Zhang, Q. Liu, S. Qiu, S. Zhou, and C. Zhang, "Unknown attack detection based on zero-shot learning," IEEE Access, vol. 8, pp. 193981-193991, 2020.

[4] L. Bilge and T. Dumitraş, "Before we knew it: an empirical study of zero-day attacks in the real world," in Proceedings of the 2012 ACM conference on Computer and communications security, New York, NY, USA, 2012, pp. 833-844.

[5] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based intrusion detection system for in-vehicle network," in 2018 16th Annual Conference on Privacy, Security and Trust (PST), Belfast, Ireland, 2018, pp. 1-6.

[6] P. Wu, H. Guo, and R. Buckland, "A transfer learning approach for network intrusion detection," in 2019 IEEE 4th international conference on big data analytics (ICBDA), Suzhou, China, 2019, pp. 281-285.

[7] Y. Yang, J. Cheng, Z. Liu, H. Li, and G. Xu, "A multi-classification detection model for imbalanced data in NIDS based on reconstruction and feature matching," Journal of Cloud Computing, vol. 13, p. 31, 2024.

[8] I. Ahmed, G. Jeon, and A. Ahmad, "Deep learning-based intrusion detection system for internet of vehicles," IEEE Consumer Electronics Magazine, vol. 12, pp. 117-123, 2021.

[9] U. K. Lilhore, P. Manoharan, S. Simaiya, R. Alroobaea, M. Alsafyani, A. M. Baqasah, et al., "HIDM: Hybrid Intrusion Detection Model for Industry 4.0 Networks Using an Optimized CNN-LSTM with Transfer Learning," Sensors, vol. 23, p. 7856, 2023.

[10] W. Yutao, L. Zhongtian, B. Yi, L. Jie, X. Fangzheng, and B. Yu, "Internet of Things Intrusion Detection System based on Transfer Learning," in 2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI), Changchun, China, 2022, pp. 25-30.

[11] O. D. Okey, D. C. Melgarejo, M. Saadi, R. L. Rosa, J. H. Kleinschmidt, and D. Z. Rodríguez, "Transfer learning approach to IDS on cloud IoT devices using optimized CNN," IEEE Access, vol. 11, pp. 1023-1038, 2023.

[12] J. Zhao, S. Shetty, J. W. Pan, C. Kamhoua, and K. Kwiat, "Transfer learning for detecting unknown network attacks," EURASIP Journal on Information Security, vol. 2019, pp. 1-13, 2019.

[13] C.-W. Tien, T.-Y. Huang, P.-C. Chen, and J.-H. Wang, "Using autoencoders for anomaly detection and transfer learning in IoT," Computers, vol. 10, p. 88, 2021.

[14] H. Elubeyd, D. Yiltas-Kaplan, and Ş. Bahtıyar, "A Multi-Modal Deep Transfer Learning Framework for Attack Detection in Software-Defined Networks," IEEE Access, vol. 11, pp. 114128-114145, 2023.

[15] H. Wang, Y. Wang, and Y. Guo, "A Novel Approach of Unknown Network Attack Detection Based on Zero-Shot Learning," in 2021 IEEE International Conference on Data Science and Computer Application (ICDSCA), Dalian, China, 2021, pp. 312-318.

[16] G. Zachos, G. Mantas, K. Porfyrakis, J. M. C. S. d. Bastos, and J. Rodriguez, "Anomaly-Based Intrusion Detection for IoMT Networks: Design, Implementation, Dataset Generation, and ML Algorithms Evaluation," IEEE Access, vol. 13, pp. 41994-42028, 2025.

[17] G. Logeswari, J. D. Roselind, K. Tamilarasi, and V. Nivethitha, "A Comprehensive Approach to Intrusion Detection in IoT Environments Using Hybrid Feature Selection and Multi-Stage Classification Techniques," IEEE Access, vol. 13, pp. 24970-24987, 2025.

[18] U. C. Akuthota and L. Bhargava, "Transformer-Based Intrusion Detection for IoT Networks," IEEE Internet of Things Journal, vol. 12, pp. 6062-6067, 2025.

[19] J. Gao, M. Fan, Y. He, D. Han, Y. Lu, and Y. Qiao, "MACAE: memory module-assisted convolutional autoencoder for intrusion detection in IoT networks," The Journal of Supercomputing, vol. 81, p. 231, 2024/12/02 2024.

[20] Z. Alwaeli, O. A. Fadare, and F. Al-Turjman, "Developing Deep Learning-Based Network Intrusion Detection Systems (NIDS) for Iot Networks," in Smart Infrastructures in the IoT Era, F. Al-Turjman, Ed., ed Cham: Springer Nature Switzerland, 2025, pp. 1105-1113.

[21] M. L. Hernandez-Jaimes, A. Martinez-Cruz, K. A. Ramírez-Gutiérrez, and A. Morales-Reyes, "Network traffic inspection to enhance anomaly detection in the Internet of Things using attention-driven Deep Learning," Integration, vol. 103, p. 102398, 2025/07/01/ 2025.

[22] S. Walling and S. Lodh, "An Extensive Review of Machine Learning and Deep Learning Techniques on Network Intrusion Detection for IoT," Transactions on Emerging Telecommunications Technologies, vol. 36, p. e70064, 2025.

[23] Y. Fan, Y. Li, M. Zhan, H. Cui, and Y. Zhang, "Iotdefender: A federated transfer learning intrusion detection framework for 5g iot," in 2020 IEEE 14th international conference on big data science and engineering (BigDataSE), Guangzhou, China, 2020, pp. 88-95.

[24] N. Khatri, S. Lee, and S. Y. Nam, "Transfer Learning-based Intrusion Detection System for a Controller Area Network," IEEE Access, vol. 11, pp. 120963-120982, 2023.

[25] Ü. Çavuşoğlu, D. Akgun, and S. Hizal, "A novel cyber security model using deep transfer learning," Arabian Journal for Science and Engineering, vol. 49, pp. 3623-3632, 2024.

[26] H. B. Arteaga. (2023). Intrusion Detection System using Machine Learning. Available: https://github.com/BioAITeam/Intrusion-Detection-System-using-Machine-Learning/tree/main/DBs

[27] U. S. N. Australia. (2021). The UNSW-NB15 Dataset. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset

[28] C. I. f. Cybersecurity. (2023). ISCX NSL-KDD dataset 2009 Available: https://www.unb.ca/cic/datasets/nsl.html

[29] cloudstor. (2019). Download Ton-IoT Dataset. Available: https://cloudstor.aarnet.edu.au/plus/s/ds5zW91vdgjEj9i?path=%2FTrain_Test_datasets

[30] I. Idrissi, M. Azizi, and O. Moussaoui, "Accelerating the update of a DL-based IDS for IoT using deep transfer learning," Indones. J. Electr. Eng. Comput. Sci, vol. 23, pp. 1059-1067, 2021.

[31] G.-P. Fernando, A.-A. H. Brayan, A. M. Florina, C.-B. Liliana, A.-M. Héctor-Gabriel, and T.-S. Reinel, "Enhancing Intrusion Detection in IoT Communications through ML Model Generalization with a New Dataset (IDSAI)," IEEE Access, vol. 11, pp. 70542-70559, 2023.

[32] Y. Wang, Y. Lai, Y. Chen, J. Wei, and Z. Zhang, "Transfer learning-based self-learning intrusion detection system for in-vehicle networks," Neural Computing and Applications, vol. 35, pp. 10257–10273, 2023.

[33] J. Wang, P. Li, W. Kong, and R. An, "Unknown Security Attack Detection of Industrial Control System by Deep Learning," Mathematics, vol. 10, p. 2872, 2022.

[34] F. S. Alrayes, M. Zakariah, S. U. Amin, Z. I. Khan, and M. Helal, "Intrusion detection in IoT systems using denoising autoencoder," IEEE Access, 2024.

[35] C. Zha, Z. Wang, Y. Fan, X. Zhang, B. Bai, Y. Zhang, et al., "SKT-IDS: Unknown attack detection method based on Sigmoid Kernel Transformation and encoder–decoder architecture," Computers & Security, vol. 146, p. 104056, 2024.

Journal of Information Systems and Telecommunication
Vol.14, No.1, January-March 2026, 16-27
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

# Re-CAC: A Re-Engineered Call Admission Control for LTE Downlink Networks Using Stepwise Bandwidth Degradation Concept

Vitalis I. Onyeke[1], Udora N. Nwawelu[1*], Bonaventure O. Ekengwu[1], Nnaemeka C. Asiegbu[1], Benjamin O. Ezurike[2], Dumtochukwu O. Oyeka[1], Chukwudi M. Chukwudozie[1], Chimdalu P. Okide[1]

[1].Department of Electronic and Computer Engineering, University of Nigeria, Nsukka, Enugu State, Nigeria
[2].Department of Mechatronics Engineering, Alex Ekwueme Federal University, Ndufu-Alike, Ebonyi State, Nigeria

Received: 24 Sep 2024/ Revised: 04 Feb 2026/ Accepted: 09 Mar 2026

## Abstract

In Long Term Evolution (LTE) networks, bandwidth degradation is a concept that Call Admission Control (CAC) schemes employ to improve the Quality of Service (QoS) of admitted real-time (RT) calls. However, it has led to noticeable resource wastage due to the inappropriate degradation method employed. In this paper, a Re-engineered Call Admission Control (Re-CAC) scheme that uses stepwise bandwidth degradation is proposed. This contribution to improving resource management allows sequential bandwidth degradation in a stepwise manner. The Re-CAC scheme was tested in MATLAB using the Video and File Transfer Protocol (FTP), representing RT and non-real-time (NRT) classes of service standardized by 3GPP for LTE networks. The performance of Re-CAC was evaluated using throughput, call blocking probability (CBP), call dropping probability (CDP), and spectral efficiency metrics. The Re-CAC scheme was further compared with quality of service-aware CAC (QA-CAC), adaptive CAC (ACAC), and enhanced adaptive CAC (EA-CAC) schemes with reference to the ACAC scheme: Re-CAC, EA-CAC, and QA-CAC schemes achieved respective throughput increase of RT calls by 19.55%, 16.84%, and 12.30% while Re-CAC, QA-CAC, and EA-CAC schemes achieved throughput reduction of NRT calls by 3.25%, 5.38%, and 6.80%, respectively. The Re-CAC, EA-CAC, and QA-CAC schemes achieved corresponding CBP reduction of RT calls by 27.72%, 24.69%, and 12.78% while the Re-CAC, QA-CAC, and EA-CAC schemes achieved corresponding CBP increase of 1.23%, 3.02% and 3.75% for NRT calls. Furthermore, the Re-CAC, EA-CAC, and QA-CAC schemes achieved corresponding CDP reductions of 27.21%, 19.27%, and 12.41% for RT calls, whereas the Re-CAC, QA-CAC, and EA-CAC schemes achieved corresponding CDP increases of 3.01%, 5.13%, and 6.57% for NRT calls. At the same time, the Re-CAC, EA-CAC, and QA-CAC achieved increases in spectral efficiency of 19.34%, 16.84%, and 12.17%, respectively, for RT calls. In contrast, the Re-CAC, QA-CAC, and EA-CAC schemes achieved respective percentage reductions in spectral efficiency of 3.05%, 5.58%, and 6.60% for NRT calls. These results demonstrate the superiority of the Re-CAC scheme over the benchmark CAC schemes in handling RT services.

**Keywords:** Call Admission Control; Long Term Evolution; Bandwidth Degradation; QoS; RT and NRT Calls.

## 1- Introduction

Long Term Evolution (LTE) is a fourth generation network standard designed to support broadband connectivity. The Third Generation Partnership Project (3GPP), the group that developed this standard, has introduced enhanced methodologies in its specifications to improve Quality of Service (QoS) [1][2]. LTE network supports Multiple Input Multiple Output (MIMO) technology to achieve high peak data rates of up to 300 Mbps, improve spectral efficiency, and provide wide coverage [3].

The radio access of the LTE network employs Orthogonal Frequency Division Multiple Access (OFDMA) for downlink and Single Carrier Frequency Division Multiple Access (SC-FDMA) for uplink transmissions [1][4].

In LTE downlink transmissions, radio resources are partitioned across both the frequency and time domains

✉ **Udora N. Nwawelu**
nwabuoku.nwawelu@unn.edu.ng

using OFDMA [5][6][7]. This partitioning results in a structured radio resource grid, as depicted in Figure 1.

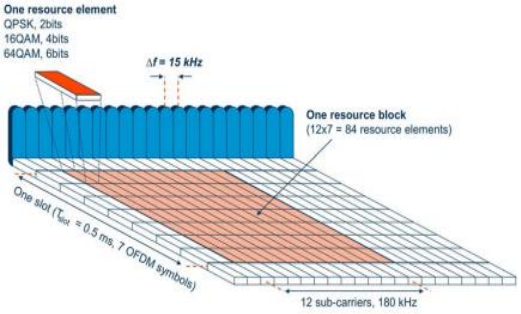



Fig. 1 LTE Downlink Resource Grid Based on OFDM [8]

The resource grid in Figure 1 comprises a matrix of subcarriers in the frequency domain and time slots in the time domain, allowing for efficient allocation and management of radio resources to meet the varying demands of users and applications. These resources are grouped into ten sub-frames. Each sub-frame consists of two slots. Each slot counts 6 or 7 OFDM symbols for a normal or extended cyclic prefix. Assuming an extended cyclic prefix, a resource block (RB) is made up of 7 OFDM symbols and 12 sub-carriers in the time-frequency domain. In the frequency domain, RB has a bandwidth of 180 kHz; thus, the spacing between the sub-carriers is 15 kHz. In the time domain, a slot is 0.5 ms. This shows that each RB contains 84 resource elements (REs) for a normal cyclic prefix. For the extended cyclic prefix, each RB contains 72 resource elements [8]. The LTE downlink physical parameters, including channel bandwidth, subcarrier spacing, the number of occupied subcarriers, and the number of resource blocks, are presented in Table 1.

Table 1: LTE Downlink Physical Parameters [5]

| Channel Bandwidth (MHz) | 1.4 | 3 | 5 | 10 | 15 | 20 |
|---|---|---|---|---|---|---|
| Subcarrier Spacing (kHz) | 15 | | | | | |
| No of occupied Sub-carriers | 72 | 180 | 300 | 600 | 900 | 1200 |
| Number of Resource Blocks | 6 | 15 | 25 | 50 | 75 | 100 |

The information presented in Table 1 was obtained using OFDM symbols with a normal cyclic prefix. As observed, there is a direct relationship among the system channel bandwidth, the number of resource blocks, and the number of occupied sub-carriers.

In the LTE network standard, one aspect that is not fully addressed is CAC strategies. It is one of the distinctive features in the Medium Access Control (MAC) sublayer of a network [9]. CAC is one of the radio resource management (RRM) techniques that can be implemented at the network's admission point. It is among the essential innovations for improving network QoS [10]. CAC is employed by a router or switch to accept or reject a flow according to a predefined flow specification. Before accepting any new flow for processing, the CAC checks the flow specifications to ensure that its total resource, which includes its previous commitments to other flows, is sufficient to accommodate the additional new flow.

In recent times, research interests have focused on CAC schemes, as they have shown significant potential for effectively addressing radio resource wastage in communication networks [1][11][12]. Consequently, relevant contributions to improving resource management using CAC have been proposed. This paper focuses on various CAC schemes proposed for LTE networks.

Mamman et al. suggested an adaptive CAC (ACAC) with bandwidth reservation for downlink transmission in an LTE network [1]. At the admission point into the network, the ACAC scheme allocates the maximum and minimum required bandwidths to the respective RT and NRT calls. When there is insufficient bandwidth to admit a call upon arrival, it degrades the bandwidth of the admitted RT calls to their minimum bandwidth requirements. The bandwidth recouped is added to the system's available bandwidth and used to admit NRT call if its bandwidth requirement is less than or equal to the available bandwidth; otherwise, the scheme rejects the call. On the other hand, RT call is accepted if the needed bandwidth is less than or equal to the available system bandwidth; otherwise, the scheme rejects the call. The ACAC scheme achieved increased throughput and reduced the NRT call blocking rate, though at the expense of RT calls. This approach wastes bandwidth because it lacks a mechanism to determine whether the available bandwidth can accommodate additional calls before degrading the bandwidth.

A Quality of Service aware call admission control (QA-CAC) scheme was used to guarantee QoS for RT calls [11]. The RT and NRT calls are initially assigned maximum bandwidth upon arrival in the network. Upon the subsequent arrival of RT call and there is insufficient resource to accommodate it, the scheme degrades the bandwidth of the admitted NRT calls to their minimum requirement bandwidth. The recouped bandwidth is added to the available system bandwidth. The RT call is accepted if the requested bandwidth is less than or equal to the new available bandwidth; otherwise, the call is denied access. On the other hand, NRT call is accepted if the requested bandwidth is less than or equal to the available system bandwidth; otherwise, the call is rejected. This scheme also wastes resources when the bandwidth recouped from the degradation action is insufficient to handle the newly arrived RT call. This is because the scheme lacks a

mechanism to determine whether the reduced bandwidth is sufficient to accept a new call before the reduction.
Meanwhile, the new call blocking probability is reduced using an adaptive Markov-based CAC scheme [13]. The scheme dynamically reserves physical resource blocks (PRBs) for new calls, and the remaining available PRBs are used to admit other call request types. Under heavy traffic, the scheme degrades the bandwidth of lower priority calls so long as the resources to admit new higher priority call requests are insufficient. This scheme reduced the probability that higher priority calls would be starved during an influx of call requests. This leads to starvation of low priority calls and is attributed to the bandwidth degradation strategy the scheme employs.

In another study, Ali et al. [14] presented a CAC scheme that increased resource usage and reduced the probability of different call requests being dropped. Calls were classified either as a handoff call (HC) or a new call (NC). Higher priority is given to HC and lower priority to (NC). This scheme first checks for the availability of PRB to admit NC or HC call. However, NC can only be accepted if the available PRB exceeds the PRB requested by the new call; otherwise, the call request is denied access. This scheme reduces the probability of dropping the HC call. However, the reserved resources are wasted when no frequent arrivals of HC. Also, the scheme wastes resources if the demand does not match the allocation.

The CAC algorithm for mobility management in the LTE network was suggested [15]. The aim was to reduce the handoff call drop rate, delay and network traffic congestion. The scheme was divided into two sections: to handle handoff calls in the queue, the first module was built using a limited-queue mechanism. The second module was designed specifically to prevent incoming calls from exceeding the base station's threshold. An intelligent fuzzy logic controller was employed to control the capacity of the base station queue. However, the scheme did not apply any form of bandwidth degradation to the admitted call with the maximum bandwidth requirement.

Maharazu et al. presented a CAC scheme for vehicular LTE networks [16]. The scheme prioritized the RT handoff call over the NRT new call. The scheme computed the threshold value and divided the traffic intensity into low and high categories. When the threshold value was less than or equal to the traffic intensity, a new or handoff call was accepted into the network. If the threshold value is higher than the traffic intensity, the suggested strategy rejects handoff calls or new ones. However, the scheme did not apply any form of bandwidth degradation to the admitted call with the maximum bandwidth requirement.

An Enhanced Adaptive CAC (EA-CAC) scheme with bandwidth reservation is suggested in [12]. Initially, maximum bandwidth is assigned to RT and NRT calls. The scheme implemented two mechanisms: adaptive degradation and pre-check mechanisms. The former mechanism first degrades NRT calls before RT calls, and the latter mechanism is used on the admitted RT calls to ascertain if the bandwidth recouped is sufficient to admit new calls. The RT call is accepted if the requested bandwidth is less than or equal to the available system bandwidth; otherwise, the admitted NRT calls are degraded to their minimum required bandwidth. The bandwidth recouped is added to the available bandwidth. Next, the RT call is admitted if the requested bandwidth is less than or equal to the available system bandwidth; otherwise, the admitted RT calls are degraded. Pre-check action is vital to ensure that the degradable bandwidth is enough to accept RT calls. The EA-CAC still wastes RT resources and increases the rate at which NRT calls are blocked or dropped due to the inappropriate degradation mechanism applied.

The reviewed CAC schemes that employ bandwidth degradation are unable to address the problem of resource wastage effectively. As observed, when there is insufficient bandwidth to admit a new call request, the schemes always degrade the admitted RT call bandwidths to their minimum straight away. Afterwards, the new calls are assigned their minimum required bandwidth, and any remaining degraded bandwidth is wasted. An inappropriate bandwidth degradation strategy causes this problem. This paper aims to address this issue by employing a stepwise bandwidth degradation mechanism. This proposal will not only minimize resource wastage but also improve other network QoS parameters.

The remaining parts of this paper are structured as follows: Section 2 presents the methods for achieving the proposed CAC scheme; Section 3 presents the results and discussion; and Section 4 provides the concluding remarks and future research directions.

## 2- Methodology

The CAC conceptual model is presented in Figure 2. The user equipment UEs are wirelessly connected to an eNodeB. The CAC scheme is implemented at the MAC layer domiciled at the eNodeB.

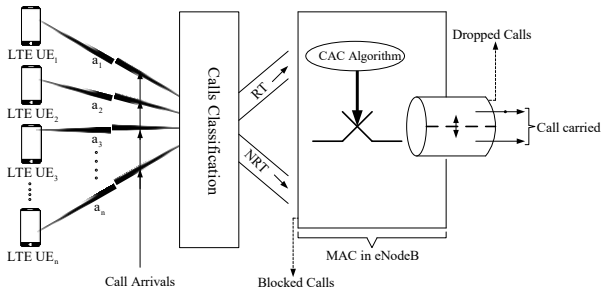



Fig. 2 Conceptual View of an LTE CAC Model

For the admissions process, the UEs transmit calls generated by the call generating module to the eNodeB. Each generated traffic has features such as call type, packet size, arrival time, among others. The traffic is routed to the call prioritizing and classification module, whose function is to categorize the generated calls into two QoS class types, namely RT and NRT calls. The higher priority calls are sent to the CAC module, which is responsible for admissions. Each time a call arrives, the modules verify the traffic specification, which includes the call type, QoS class, arrival time, required bandwidth, and available system bandwidth. It first verifies these parameters, then checks the requirements for admitting the class type and call type. If the requirements are met, the call is approved; otherwise, it is denied. When the bandwidth requested by a call is unavailable, the call is routed to the module responsible for bandwidth degradation. If the degradation requirements are met, the respective accepted calls are degraded.

## 2-1- Proposed Re-CAC Scheme Development

The Re-CAC concept is conceived because EA-CAC, QA-CAC, and ACAC schemes are unable to effectively address resource wastage in the LTE network. For a deeper understanding, readers of this work are encouraged to read the complete work on each scheme [1][11][12].

### Stepwise Bandwidth Degradation Concept

This concept refers to the gradual reduction of the bandwidth assigned to the admitted call. The reduction is in a discrete manner. The assigned bandwidth is gradually degraded in response to a new call service request. The stepwise bandwidth degradation process is achieved following the listed steps:

1. **Initial Bandwidth Allocation:** The scheme allocates a defined amount of bandwidth to each service class.
2. **Monitoring for New Call Requests:** The new call request is monitored, and the call type is classified for resource allocation.
3. **Apply Stepwise Degradation Policy:** The scheme reduces the allocated bandwidth in a stepwise manner.
4. **Further Degradation:** The scheme continues gradual degradation of the allocated bandwidth if the minimum required bandwidth for the admitted calls has not been reached, and the minimum bandwidth needed for the new call request has not been recouped.

At the network admission point, the Re-CAC scheme assigns the maximum bandwidth requirement to RT and NRT calls. Eq. (2.1) and (2.2) are the mathematical operations of the maximum bandwidth allocated to the respective RT calls and NRT calls.

$$\mathrm{RT_{calls}} = \mathrm{Max_{BW\_RT}} \tag{2.1}$$

$$\mathrm{NRT_{calls}} = \mathrm{Max_{BW\_NRT}} \tag{2.2}$$

Where $\mathrm{RT_{calls}}$ and $\mathrm{NRT_{calls}}$ indicate respective real time and non-real time calls. Whereas, $\mathrm{Max_{BW\_RT}}$ and $\mathrm{Max_{BW\_NRT}}$ denote the respective maximum bandwidth requirement allocated to RT and NRT calls.

When bandwidth is not enough to admit a new call, the scheme checks whether bandwidth can be degraded to accept the call. The RT call, being delay sensitive, is prioritized over NRT calls. Thus, it checks the availability of the admitted NRT bandwidth first before checking the availability of the admitted RT bandwidth. This is expressed as in Eq. (2.3).

$$\mathrm{NRT\ Call_{BW_{avail}}} > 0 \tag{2.3}$$

Where $\mathrm{NRT\ Call_{BW_{avail}}}$ indicates the availability of the admitted NRT bandwidth.

After checking that the NRT call's bandwidth is available for degradation, the scheme further checks whether the maximum degradable bandwidth of NRT calls is greater than the bandwidth required to admit a new call. The mathematical operation to that effect is presented in Eq. (2.4).

$$\mathrm{NRT\ Call_{Max_{BW_{deg}}}} > \mathrm{BW_{req}} \tag{2.4}$$

Where $\mathrm{NRT\ Call_{Max_{BW_{deg}}}}$ indicates the maximum degradable NRT bandwidth and $\mathrm{BW_{req}}$ indicates the bandwidth request for a call.

If the maximum degradable NRT bandwidth is sufficient, the Re-CAC scheme uses a stepwise degradation on the admitted NRT calls. The stepwise degradation is achieved using a generalized mathematical operation described by Eq. (2.5).

$$\mathrm{Deg_{BW_n}} = \mathrm{Max_{NRT\_BW}} - \mathrm{Min_{BW}} \tag{2.5}$$

$\mathrm{Min_{BW}}$ is obtained using an expression in Eq. (2.6)

$$\mathrm{Min_{BW}} = \mathrm{Max_{NRT\_BW}} - \mathrm{h} \quad (2.6)$$

Where $\mathrm{Deg_{BW_n}}$ indicates the bandwidth obtained at n = {1, 2, ….. , i} degradation steps. $\mathrm{Max_{NRT\_BW}}$ indicates the maximum allowable admitted NRT bandwidth, $\mathrm{Min_{BW}}$ indicates the minimum degradable bandwidth at each step, and h is the step decrement size.

For clarity, given $\mathrm{Max_{NRT\_BW}}$ = 1450 and $h$ = 50, the Re-CAC degrades the maximum allowable admitted NRT call bandwidth in a stepwise manner to recoup available degradable bandwidth using Eq. (2.7):

$$\mathrm{Deg_{BW_1}} = \mathrm{Max_{NRT\_BW}} - \mathrm{Min_{BW}} \quad (2.7)$$

While $\mathrm{Max_{NRT\_BW}}$ and $\mathrm{Min_{BW}}$ retain their original meanings, $\mathrm{Deg_{BW_1}}$ indicates the bandwidth obtained after the first phase of the stepwise degradation of the admitted NRT calls. The bandwidth recouped is added to the available system bandwidth, and a call request is given access if there are sufficient resources, as shown in Eq. (2.8). Minimum bandwidth requirement is allocated to calls admitted after degradation at the point of admission.

$$\mathrm{Avail_{BW}} + \mathrm{Deg_{BW_1}} \geq \mathrm{Req_{BW}} \quad (2.8)$$

Where $\mathrm{Avail_{BW}}$ indicates the available bandwidth, $\mathrm{Deg_{BW_1}}$ retains its original meaning, and $\mathrm{Req_{BW}}$ indicates the requested bandwidth for a call.

When the obtained bandwidth is insufficient, the second phase of stepwise degradation is performed using the mathematical operation in Eq. (2.9).

$$\mathrm{Deg_{BW_2}} = \mathrm{Max_{NRT\_BW}} - \mathrm{Min_{BW}} \quad (2.9)$$

Where $\mathrm{Deg_{BW_2}}$ indicates the bandwidth obtained after the second phase of stepwise degradation of the admitted calls. The bandwidth recouped after the second phase of the stepwise degradation is added to the available system bandwidth, and a call is accepted only if there is enough bandwidth, as shown in Eq. (2.10).

$$\mathrm{Avail_{BW}} + \mathrm{Deg_{BW_2}} \geq \mathrm{Req_{BW}} \quad (2.10)$$

Where $\mathrm{Avail_{BW}}$ and $\mathrm{Req_{BW}}$ retained their original meanings and $\mathrm{Deg_{BW_2}}$ indicates the bandwidth obtained after the second phase of the stepwise degradation of the admitted NRT calls.

The process of stepwise bandwidth degradation continues in an iterative and sequential manner up to the last allowable degradation step. At this point, the Re-CAC perform the final degradation on the admitted NRT calls. The mathematical operation employed is shown in Eq. (2.11).

$$\mathrm{Deg_{BW_n}} = \mathrm{Max_{NRT\_BW}} - \mathrm{Min_{BW}} \quad (2.11)$$

Where $\mathrm{Deg_{BW_n}}$ indicates the bandwidth obtained after the last phase of the stepwise degradation of the admitted NRT calls.$\mathrm{Max_{NRT\_BW}}$ and $\mathrm{Min_{BW}}$ retain their original meanings.

The recouped bandwidth after the final stepwise bandwidth degradation is added to the available system bandwidth. A call is accepted provided there is enough bandwidth, as shown in Eq. (2.12), and the minimum bandwidth requirement is allocated to calls admitted at the network admission point.

$$\mathrm{Avail_{BW}} + \mathrm{Deg_{BW_n}} \geq \mathrm{Req_{BW}} \quad (2.12)$$

Where $\mathrm{Avail_{BW}}$, $\mathrm{Deg_{BW_n}}$, and $\mathrm{Req_{BW}}$ retain their original meanings

If the available bandwidth and the one recouped after the final stepwise degradation of the NRT calls are added up, and the bandwidth is still insufficient to handle the RT calls, the Re-CAC scheme now checks the availability of the admitted RT bandwidth for degradation using the expression in Eq. (2.13).

$$\mathrm{RT\ Call_{BW_{avail}}} > 0 \quad (2.13)$$

Where $\mathrm{RT\ Call_{BW_{avail}}}$ indicates the availability of admitted RT bandwidth.

If the admitted RT bandwidth is available, the Re-CAC pre-checks the admitted RT calls. This is to ascertain if the recouped bandwidth after degradation would be enough to admit a call before performing degradation. In this case, Eq. (2.14) is used.

$$\mathrm{Avail_{BW}} + \sum \mathrm{BW_{RT_{deg}}} \geq \mathrm{Req_{BW}} \quad (2.14)$$

Where $\mathrm{Avail_{BW}}$ and $\mathrm{Req_{BW}}$ retain their former meanings and $\sum \mathrm{BW_{RT_{deg}}}$ indicates the total of the RT degradable bandwidth**.**

If the condition is met, then the RT call bandwidth is degraded using the proposed stepwise degradation and the call request is accepted; else, the call request is rejected. The aim of employing stepwise degradation on RT calls is to ensure that the bandwidth degraded from RT is used, thereby reducing network resource waste. The flow process for achieving the Re-CAC scheme is presented in Figure 3, while its pseudocode is shown in Algorithm 1

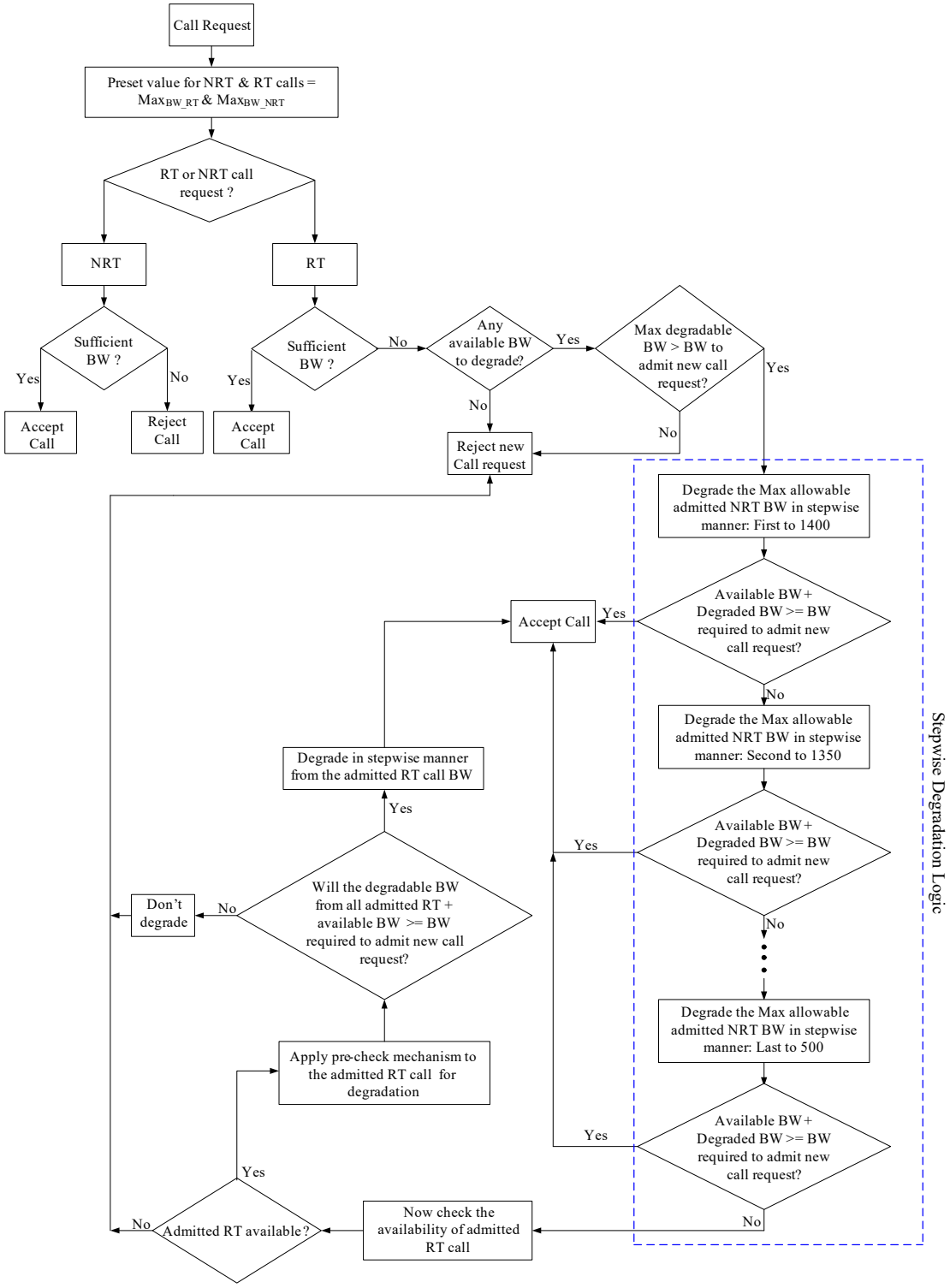



Fig. 3 Flowchart Illustrating the Suggested Re-CAC Scheme

✉ **Udora N. Nwawelu**
nwabuoku.nwawelu@unn.edu.ng

**Algorithm 1:** Re-CAC Scheme Pseudo Code

**1: Input Parameters:**

- **RT:** Real time call
- **NRT:** Non-real time call
- **$\mathbf{Max_{BW\_RT}}$:** Maximum allowable RT bandwidth
- **$\mathbf{Max_{BW\_NRT}}$:** Maximum allowable NRT bandwidth
- **$\mathbf{RT_{BW_{avail}}}$:** Available RT bandwidth
- **$\mathbf{NRT_{BW_{avail}}}$:** Available NRT bandwidth
- **$\mathbf{Req_{BW}}$:** Request bandwidth for a call
- **$\mathbf{Avail_{BW}}$:** Available bandwidth in the network
- **NRT $\mathbf{Call_{Max_{deg}}}$:** Maximum allowable degraded NRT bandwidth
- **SMT:** Simulation time
- **TTI:** Transmission Time Interval
- **h: S**tep decrement size

**2: Initializations**
**3: While** TTI is within SMT **do**
**4: Begin**
**5: Input** Call Request
**6: If** Call Request is **NRT Then**
**7:** Check $NRT_{BW_{avail}}$
**8:** **If** Sufficient $NRT_{BW_{avail}} >= Req_{BW}$ **Then**
**9:** Accept Call
**10:** **Else**
**11:** Reject Call
**12:** **End If**
**13:** **Else If** Call Request is **RT, Then**
**14:** Check $RT_{BW_{avail}}$
**15:** **If** Sufficient$RT_{BW_{avail}} >= Req_{BW}$ **Then**
**16:** Accept Call
**17:** **Else**
**18:** Check if any bandwidth is available for degradation
**19:** **If** No Available Bandwidth to degrade **Then**
**20:** Reject Call Request
**21:** **Else**
**22:** Check if Max Degradable Bandwidth > $Req_{BW}$
**23:** **If** Yes **Then**
**24:** Accept Call
**25:** **Else**
**26:** Reject Call Request
**27:** **End If**
**28:** **End If**
**29:** **End If**
**30:** **End If**
**31:** While $Avail_{BW} <$ Required BW & Steps Avail for Deg
**32:** **Degrade Bandwidth** in stepwise manner
**33:** **If** $(Avail_{BW} +$ Degraded BW$) >= Req_{BW}$ **Then**
**34:** Accept Call
**35:** **End If**
**36:** **End While**
**37:** **If** $(Avail_{BW} +$ Degraded BW$) < Req_{BW}$**Then**
**38:** Check Availability of Degradable RT bandwidth
**39:** **If** $(\sum RT_{BW_{deg}} + Avail_{BW}) \geq Req_{BW}$ **Then**
**40:** Degrade Admitted RT
**41:** Accept Call
**42:** **Else**
**43:** Don't Degrade Admitted Call
**44:** Reject Call
**45:** **End If**
**46:** **End If**
**47:** **End**
**48: End While**

## 2-2- Implementation of the Re-CAC Scheme

The Re-CAC scheme is implemented in MATLAB. LTE supports nine (9) traffic classes, and each traffic class is categorized as either RT or NRT [17][18]. In this paper, the RT traffic class (Video Streaming) and the NRT traffic class (File Transfer Protocol, FTP) are the two traffic service classes considered.

Video traffic is generated using the networkTrafficVideoConference object. This object specifies the configuration to create a video application pattern. It was achieved using object-oriented programming in MATLAB. The video traffic generated follows Weibull distribution pattern. On the other hand, the FTP traffic was generated using the networkTrafficFTP object in MATLAB. The networkTrafficFTP object specifies the configuration to generate FTP application pattern. The FTP traffic generated follows a Lognormal distribution pattern. The generated traffic is transmitted to the network's Access Point, and call arrivals follow a Poisson process.

For LTE networks with a typical Maximum Transmission Unit (MTU) of 1500 bytes, the maximum video streaming packet size is around 1400–1450 bytes, and the minimum is around 200–500 bytes for medium-bitrate video. On the other hand, for networks with an MTU of 1500 bytes, the maximum FTP packet size is around 1400 – 1450 bytes, and the minimum FTP packet size is around 50 – 100. The packet size supported by an LTE network can vary depending on the specific network configuration, device capability, and QoS settings. In this paper, respective maximum and minimum packet sizes of 1450 and 500 bytes were used for video streaming, while respective maximum and minimum packet sizes of 1450 and 100 bytes were used for FTP. Table 2 presents all the non-default parameters used in this paper. These parameters are the same for both the Re-CAC and benchmark CAC schemes, thus giving a common ground for simulation and analysis.

Table 2: Simulation Parameters

| Parameters | Value |
|---|---|
| System Bandwidth | 5 MHz |
| Number of Resource Blocks | 25 |
| Sub-carrier Spacing | 15 kHz |
| Number of eNB | 1 |
| Maximum UE in eNB | 120 |
| TTI Duration | 1ms |
| Services Classes Tested | Video and FTP |
| Simulation Time | 1000s |
| Packet Size [Video] | Maximum = 1450 Bytes; Minimum = 500 Bytes |
| Packet Size [FTP] | Maximum = 1450 Bytes; Minimum = 100 Bytes |
| Call Arrival | Poisson Process |
| Transmission Scheme | 2×2 MIMO, OLSM Antennas |
| Considered Cyclic Prefix | 4.7 µS Normal Cyclic Prefix |
| UE Distribution | Uniform |

## 3- Result and Discussions

The results achieved by simulating the Re-CAC scheme and benchmark CAC schemes (QA-CAC, ACAC, and EA-CAC) are discussed. These schemes are evaluated based on Throughput, Call Blocking and Call Dropping Probabilities, and Spectral Efficiency metrics for both RT and NRT calls.

In the result analysis, the following are worth noting: the scheme that recorded the highest throughput is graded better than others in performance, the scheme that recorded the lowest CBP and CDP is graded better than others in performance, and the scheme that achieved the highest spectral efficiency value is rated to outperform others.

### 3-1- Throughput Results

Figure 4 presents the throughputs of RT calls with Re-CAC and benchmark CAC schemes. As observed, when the call request is low, Re-CAC, together with the benchmark CAC schemes, achieved equal throughputs. This is because the available bandwidth was enough to handle all call requests. As call requests increase, Re-CAC outperformed all benchmark schemes in admitting more RT calls. The Re-CAC achieved an average throughput of 0.1657 Mbps, whereas EA-CAC, QA-CAC and ACAC schemes achieved throughputs of 0.1603, 0.1520, and 0.1333 Mbps, respectively. With reference to the ACAC scheme, further analysis shows that Re-CAC, EA-CAC, and QA-CAC achieved throughput increases of 19.55%, 16.84%, and 12.30%, respectively. The results are as expected, since Re-CAC prioritized RT calls over NRT calls. Most importantly, unlike the benchmark schemes which when there is insufficient bandwidth to admit a new call upon arrival, degrades the bandwidth of the admitted RT calls to their minimum bandwidth requirements straight away, the Re-CAC scheme degrades in a stepwise manner (i.e., gradual degradation until it gets to the minimum bandwidth required by the admitted calls) on active NRT calls before using it on the RT calls. The Re-CAC scheme also adopted a pre-check mechanism and employed it at every degradation step. This ensures that when the active call's bandwidth is degraded and added to the unused bandwidth, the total bandwidth remains neither insufficient nor oversufficient for the new RT calls. EA-CAC, together with the QA-CAC scheme, increases RT call throughput compared to ACAC. The reason is that the EA-CAC scheme first degrades NRT calls before degrading RT calls.

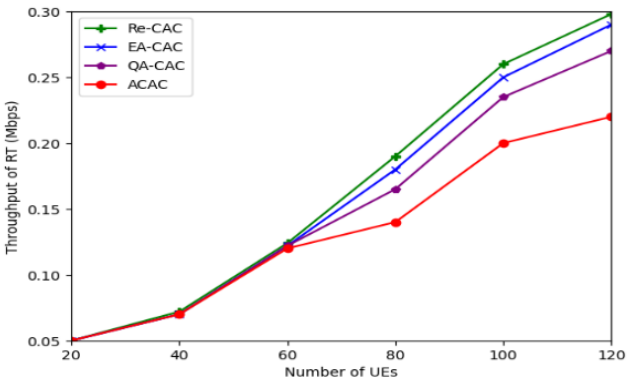


Fig. 4 Throughput of RT Calls with Re-CAC and Benchmark CAC Schemes

Figure 5 shows the throughputs achieved by NRT calls with the Re-CAC scheme, along with those of the benchmark CAC schemes. As observed, as the number of calls requesting admission increases, the ACAC scheme achieves a slightly higher throughput than the Re-CAC, QA-CAC, and EA-CAC schemes. Descriptively, the ACAC achieved an average throughput of 0.0985 Mbps, whereas the Re-CAC, QA-CAC, and EA-CAC schemes achieved individual throughputs of 0.0953, 0.0932, and 0.0918 Mbps, respectively. With reference to the ACAC scheme, further analysis shows that the Re-CAC, QA-CAC, and EA-CAC schemes achieved throughput reduction of 3.25%, 5.38%, and 6.80%, respectively. The reason is that the ACAC scheme degrades RT calls to their minimum bandwidth requirement instantly if there is insufficient bandwidth to admit a new call. The new call is admitted using the recouped bandwidth and the unused system bandwidth. The Re-CAC scheme admitted slightly more NRT calls than the QA-CAC and EA-CAC schemes due to the gradual degradation of NRT calls when the required bandwidth to admit a new call is insufficient. Calls admitted using the QA-CAC scheme are fewer than those admitted under both the ACAC and Re-CAC schemes. This is due to the degradation approach it uses for admitted NRT calls when the bandwidth to admit a new call is insufficient.

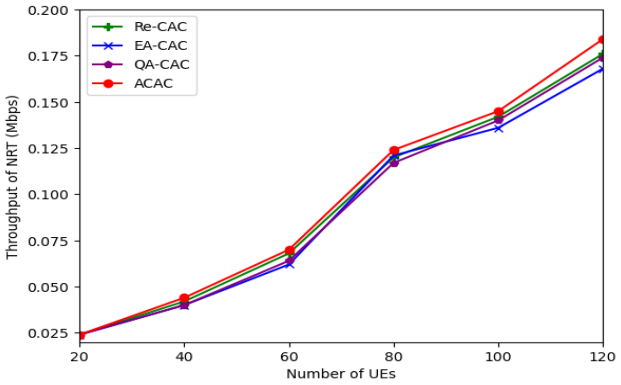



Fig. 5 Throughput of NRT Calls with Re-CAC and Benchmark CAC Schemes

## 3-2- Throughput Results

The CBP achieved by RT calls with Re-CAC and benchmark CAC schemes are presented in Figure 6. As the call request increases, the Re-CAC, QA-CAC, and EA-CAC schemes reject fewer calls than the ACAC scheme. The Re-CAC scheme has the lowest RT call blocking. The CBP for RT calls with Re-CAC, EA-CAC, QA-CAC, and ACAC schemes are 0.0837, 0.0872, 0.1010, and 0.1158, respectively. With reference to the ACAC scheme, further analysis shows that the Re-CAC, EA-CAC, and QA-CAC schemes achieved CBP reductions of 27.72%, 24.69%, and 12.78%, respectively. The Re-CAC scheme achieves the lowest CBP reduction because it allocates sufficient resources to RT calls at the network admission point. Also, due to the stepwise degradation applied to NRT calls when there is insufficient bandwidth to accept a new call, the recouped bandwidth is used to admit more RT calls. The Re-CAC scheme also employed a pre-check mechanism for RT calls to ascertain their availability before degradation. Furthermore, the EA-CAC scheme blocks fewer RT calls than the QA-CAC scheme, and the QA-CAC scheme rejects fewer calls than the ACAC scheme.

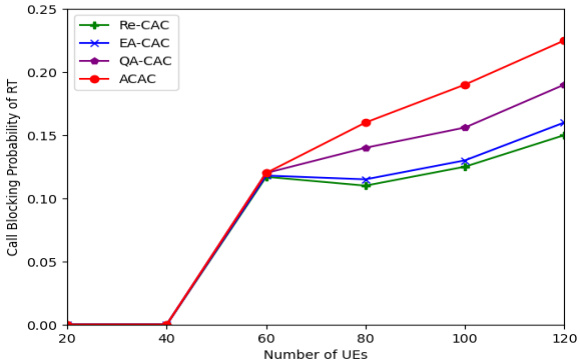



**Fig. 6** CBP of RT Calls with Re-CAC and Benchmark CAC Schemes

Meanwhile, Figure 7 presents the CBP achieved by NRT calls with Re-CAC and benchmark CAC schemes. From the observation, when call arrivals into the network are low, both the Re-CAC and the benchmark schemes block no NRT calls. This is because at that moment, enough resources are available to accommodate all calls requesting admission to the network. As call requests keep increasing, the ACAC scheme has experienced the lowest NRT call blocking. The CBP values for NRT calls with the ACAC, Re-CAC, QA-CAC, and EA-CAC schemes are 0.0642, 0.0650, 0.0662, and 0.0667, respectively. With reference to the ACAC scheme, further analysis shows that the Re-CAC, QA-CAC, and EA-CAC schemes achieved corresponding CBP increases of 1.23%, 3.02%, and 3.75%, respectively. The ACAC scheme achieves the lowest CBP for NRT calls by prioritizing NRT calls over RT calls. Thus, the admitted RT calls were degraded so that the bandwidth obtained was used to admit NRT calls. On the other hand, the Re-CAC blocks fewer NRT calls than the QA-CAC and EA-CAC schemes.

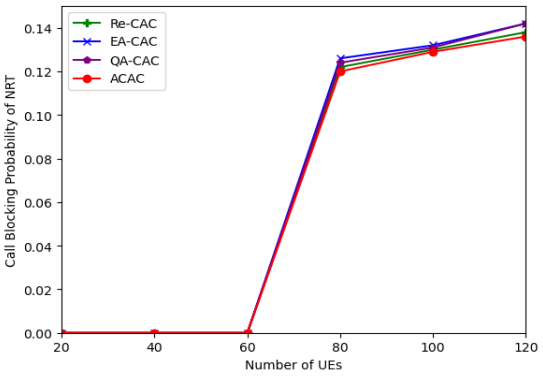



Fig. 7 CBP of NRT Calls with Re-CAC and Benchmark CAC Schemes

## 3-3- Call Dropping Probability Results

The CDP experienced by RT calls with Re-CAC and benchmark CAC schemes is shown in Figure 8. It is observed that all the schemes drop no RT calls when the call arrival is low. This is because, at that point, sufficient resources were available for call admissions. As the number of call requests increases, the Re-CAC scheme drops fewer RT calls than the EA-CAC, QA-CAC, and ACAC schemes. The CDP for RT calls with Re-CAC, EA-CAC, QA-CAC, and ACAC schemes are 0.0733, 0.0813, 0.0882, and 0.1007, respectively. With reference to the ACAC scheme, further analysis shows that Re-CAC, EA-CAC, and QA-CAC schemes achieved CBP reduction of 27.21%, 19.27%, and 12.41%, respectively. The Re-CAC scheme achieves the lowest CDP reduction because it assigns sufficient resources to RT calls at their point of admission. Additionally, the scheme applies stepwise bandwidth degradation to NRT calls when the bandwidth required to admit a new call is insufficient. Thereafter, the bandwidth obtained at each step decrement is used to admit RT calls. The Re-CAC scheme also employed a pre-check mechanism for RT calls. The EA-CAC scheme drops fewer RT calls than the QA-CAC and ACAC schemes, while the QA-CAC scheme rejects fewer RT calls than the ACAC scheme.

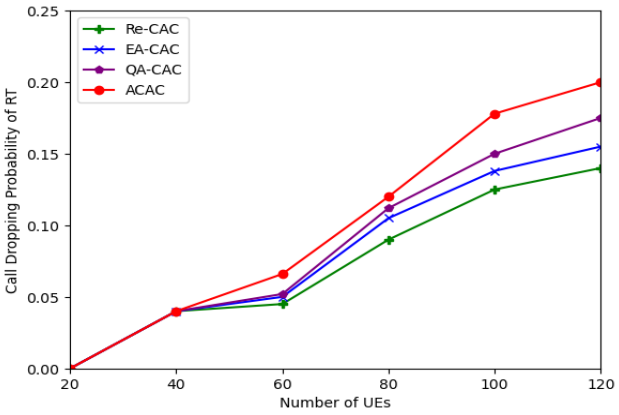



Fig. 8 CDP of RT Calls with Re-CAC and Benchmark CAC Schemes

Conversely, Figure 9 presents a plot of the CDP experienced by NRT calls under the Re-CAC and benchmark CAC schemes. It can be seen that when the number of call requests for access to network resources is low, no NRT calls are dropped by the Re-CAC or other benchmark schemes. This is because there are sufficient resources to handle all admission requests. As the number of call requests continues to increase, the ACAC scheme rejects slightly fewer NRT calls than the Re-CAC, EA-CAC, and QA-CAC schemes. The CDP of NRT calls with ACAC, Re-CAC, QA-CAC, and EA-CAC schemes are 0.0740, 0.0763, 0.0780, and 0.0792, respectively. With reference to the ACAC scheme, further analysis shows that the Re-CAC, QA-CAC, and EA-CAC schemes achieved corresponding CBP increases of 3.01%, 5.13% and 6.57% for NRT calls. The ACAC scheme achieves the lowest CDP for NRT calls by prioritizing NRT calls over RT calls. Thus, when bandwidth is insufficient, the scheme degrades RT calls, and the recouped bandwidth is used to admit more of NRT calls. Also, the CDP with the Re-CAC scheme is slightly lower than the EA-CAC and QA-CAC schemes. The reason is that the Re-CAC scheme gradually reduced the admitted NRT call bandwidth in steps. The QA-CAC scheme rejects slightly fewer NRT calls than the EA-CAC scheme, due to the degradation mechanism it applies to admitted NRT calls.

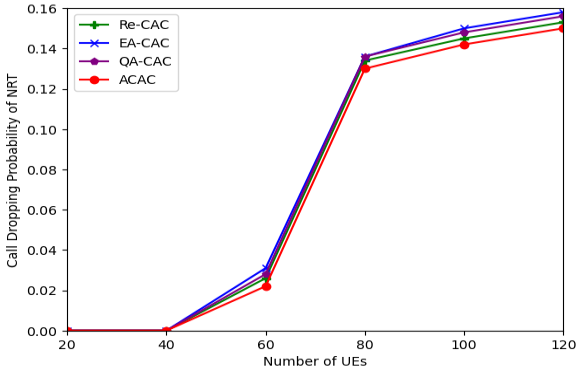


Fig. 9 CDP of NRT Calls with Re-CAC and Benchmark CAC Schemes

## 3-4- Spectral Efficiency Results

The efficacy of the Re-CAC scheme in reducing bandwidth wastage is further demonstrated through the spectral efficiency metric.

Figure 10 presents the spectral efficiency achieved by RT calls with the Re-CAC and benchmark schemes. It can be seen that when the number of calls requesting network access is low, the Re-CAC and benchmark schemes achieve equal spectral efficiency. As call requests keep increasing, the spectral efficiencies achieved with the schemes are increasing, and the Re-CAC scheme outperforms EA-CAC, QA-CAC, and ACAC in spectrum utilization for RT calls. The results of the Re-CAC scheme showed that a 0.0331 bps/Hz spectrum is used on average. On the other hand, the average of 0.0321, 0.0304, and 0.0267, all in bps/Hz, is used by the EA-CAC, QA-CAC, and ACAC schemes, respectively. With reference to the ACAC scheme, further analysis shows that Re-CAC, EA-CAC, and QA-CAC schemes achieved spectral efficiency increases of 19.34%, 16.84%, and 12.17%, respectively. The superiority of the Re-CAC scheme over EA-CAC, QA-CAC, and ACAC is attributed to the stepwise degradation mechanism it employs.

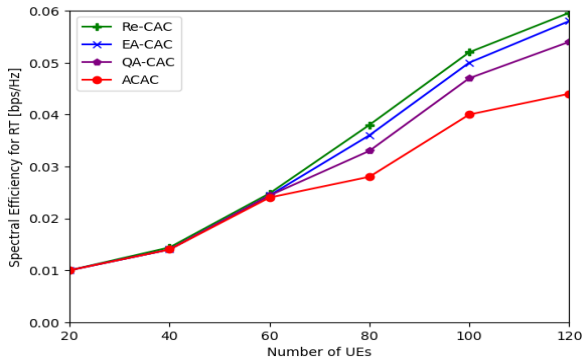



Fig. 10 Spectral Efficiency of RT Calls with Re-CAC and Benchmark CAC Schemes

Similarly, the spectral efficiency achieved by NRT calls with Re-CAC and benchmark CAC schemes is presented in Figure 11. As observed, when call requests are low, both the Re-CAC and benchmark schemes achieve equal spectral efficiency for NRT calls. As the number of calls requesting admission increases, the ACAC scheme achieves slightly better spectrum efficiency than the Re-CAC, QA-CAC, and EA-CAC schemes. The ACAC achieved an average spectral efficiency of 0.0197 bps/Hz, whereas the Re-CAC, QA-CAC, and EA-CAC schemes achieved spectral efficiencies of 0.0191, 0.0186, and 0.0184 bps/Hz, respectively. With reference to the ACAC scheme, further analysis shows that the Re-CAC, QA-CAC, and EA-CAC schemes achieved reductions in spectral efficiency of 3.05%, 5.58%, and 6.60%, respectively. The reason is that the ACAC scheme gave NRT calls higher priority. Thus, when there is a need to carry out degradation action, RT calls are degraded to their minimum bandwidth requirement, and the bandwidth obtained is added to the available system bandwidth to admit new NRT calls. This leads to increased spectrum utilization for NRT calls. The Re-CAC scheme slightly admitted more NRT calls than the QA-CAC and EA-CAC schemes. Calls requests admitted using the QA-CAC scheme are fewer than those admitted under both the ACAC and the Re-CAC schemes.

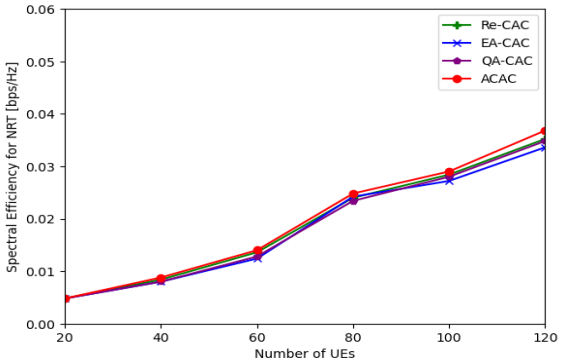



**Fig. 11** Spectral Efficiency of NRT Calls with Re-CAC and Benchmark CAC Schemes

## 4- Conclusions

Stepwise bandwidth degradation is recommended when gradual bandwidth degradation is desired. In this paper, the stepwise bandwidth degradation concept is employed in the formulation of the Re-CAC scheme for the LTE downlink network. It was observed that resource wastage is associated with the bandwidth degradation approach considered by the bandwidth degradation-based CAC scheme. The Re-CAC significantly reduced bandwidth wastage. This is demonstrated by the obtained spectral efficiency value and other performance metrics, namely throughput, CBP, and CDP. The improvement is achieved by addressing the fundamental limitation of EC-CAC, QA-CAC, and ACAC schemes (i.e., the use of an inappropriate bandwidth degradation mechanism). The Re-CAC scheme introduced a stepwise bandwidth degradation instead. This ensures that the degradable bandwidth from the active calls is not surplus to admit new calls into a network. The reason is that calls admitted after degrading active calls should be allocated their minimum required bandwidth at the network admission point. The Re-CAC scheme can be beneficial to network and traffic managers whose target is on resource optimization.

The Re-CAC scheme was implemented and further compared with ACAC, QA-CAC, and EA-CAC schemes in

MATLAB. This comparison is based on throughput, CBP and CDP, and spectral efficiency metrics. The superiority of the Re-CAC over the ACAC, QA-CAC, and EA-CAC schemes is demonstrated by the results. The Re-CAC scheme achieved higher throughput, reduced CBP and CDP, and higher spectral efficiency for RT calls without sacrificing the expected performance of NRT calls.

Estimating an exact bandwidth to degrade without resorting to resource waste is a future research interest. We intend to explore machine learning capabilities for estimating the exact bandwidth degradable step to employ when a call requests admission. This concept will further reduce to the barest minimum the resource wastage with the existing CAC schemes.

**Acknowledgments**

The authors remain grateful to Prof. C. I. Ani for his role in laying the foundation for this work.

## References

[1] M. Mamman, Z. M. Hanapi, A. Abdullah and A. Muhammed, "An Adaptive Call Admission Control with Bandwidth Reservation for Downlink LTE Networks," *IEEE Access*, vol. 5, pp. 10986–10994, 2017.

[2] U. N. Nwawelu, C. I. Ani, "Improving Exp-Rule Scheduler for Real Time Services in LTE Downlink Networks," vol. 29, no. 9, pp. 7398–7406, 2020.

[3] M. A. Akajewole, D. U. Onyishi, "A Reliable Downlink MIMO Algorithm for Mitigating the Effect of User Equipment Mobility in Multi-User MIMO in Fifth-Generation and Beyond Networks" *Nigerian Journal of Technology*, vol. 43, no. 2, pp.328 – 337, 2024.

[4] Navita and Amandeep, "Performance Analysis of OFDMA, MIMO, and SC-FDMA Technology in 4G LTE networks" 6th International Conference – Cloud System and Big Data Engineering, Noida, India, pp. 554-558, 2016.

[5] J. Zyren, "Overview of the 3GPP Long Term Evolution Physical Layer," 2007.

[6] C. O. Nnamani, C. L. Anioke, C. I Ani, "Improved MLWDF Scheduler for LTE Downlink Transmission," *International Journal of Electronics*, vol. 103, no. 11, pp. 1857 – 1867, 2016.

[7] C. Wu, W. Xu, "Key Technologies in 4G/LTE Network. Encyclopedia of Wireless Networks" Springer, Cham, pp. 695 – 698, 2020.

[8] O. Liberg, M. Sundberg, Y. P. Eric-Wang, J. Bergman, J. Sachs, G. Wikstrom, "Cellular Internet of Things" Second Edition, Academic Press, ScienceDirect, pp. 155 – 254, 2020

[9] G. O. Ugwu, U. N. Nwawelu, M. A. Ahaneku, C. I. Ani, "Effect of Service Differentiation on QoS in IEEE 802.11e Enhanced Distributed Channel Access: A Simulation Approach," *Journal of Engineering and Applied Science*, vol. 69, no. 1, pp. 1–18, 2022.

[10] B. A. Forouzan, "Data Communications and Networking," McGraw-Hill International Edition, 4th Edition, New York, 2007.

[11] M. M. Umar, A. Mohammed, A. Roko, A. Y. Tambuwal, A. Abdulazeez, "QoS-Aware Call Admission Control (QA-CAC) Scheme for LTE Networks," 15th International Conference on Electronics, Computer and Computation, Abuja, Nigeria, pp. 1-5, 2019.

[12] M. M. Umar, A. Mohammed, A. Roko, A. Y. Tambuwal, A. Abdulazeez, "Enhanced adaptive call admission control scheme with bandwidth reservation for LTE networks," *International Journal of Mobile Computing and Multimedia Communications*, vol. 12, no. 1, pp. 23–42, 2021.

[13] K. B. Ali, M. S. Obaidat, F. Zarai, L. Kamoun, "Markov Model-based Adaptive CAC Scheme for 3GPP LTE Femtocell Networks," IEEE ICC- Communications Software, Services and Multimedia Applications Symposium, pp. 6924 – 6928, 2015.

[14] R. Khdhir, K. Mnif, A. Belghith, L. Kamoun, "An Efficient Call Admission Control Scheme for LTE and LTE-A Networks," International Symposium on Networks, Computers and Communications, Yasmine, Hammamet, Tunisia, pp. 1–5, 2016.

[15] E. E. Ekechukwu, O. C. Nosiri, M. I. Ajumuka, "Efficient Call Admission Control Algorithm for Mobility Management in LTE Networks," International Journal of Networks and Communications, vol. 12, no. 1, pp. 28-38, 2022.

[16] M. Maharazu, M. H. Zurina, A. Azizol, M. Abdullah, "Call Admission Control for Real-Time and Non-real-time traffic for Vehicular LTE downlink networks," Lecture Notes in Electrical Engineering, 425, pp. 46-53, 2018.

[17] D. Pratiwi, I. J. Matheus Edward, "Analysis Efficiency Network Performance of 4G LTE in Video Conference Application," 16th International Conference on Telecommunication Systems, Services, and Applications, Lombok, Indonesia, pp. 1 – 6, 2022.

[18] S. Tabbane, "LTE Planning and Dimensioning" ITU PITA Workshop on Mobile Network Planning and Security, Nadi, Fiji Islands, 23-25 October 2019.

Journal of Information Systems and Telecommunication
Vol.14, No.1, January-March 2026, 28-39
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

# Modeling SOLOMO Marketing Based on Technological Development in the Tourism Industry

Meysam Bayat [1], Elham Fazeli Veisari [2*] , Mohammad Javad Taghipourian [3]

[1]. Department of Management, Ro.C., Islamic Azad University, Roudehen, Iran
[2]. Department of Management and accounting, To.c., Islamic Azad University, Tonekabon, Iran
[3]. Department of Management, cha.c., Islamic Azad University, Chalous, Iran

Received: 05 Feb 2025/ Revised: 04 Jan 2026/ Accepted: 18 Feb 2026

**Abstract**

The objective of this research is to identify the components and develop the internal relationships of the SOLOMO marketing components in the tourism industry. This study makes a significant contribution by presenting a robust framework derived from both qualitative and quantitative analyses, offering practical insights for enhanced marketing management in the tourism sector. The statistical population of the research in the qualitative part includes 15 experts in the field of tourism, using the purposive sampling method, and in the quantitative part includes 420 active participants in the tourism sector. In the qualitative part, the techniques of word association, sentence completion, and dream exercises were used in the form of in-depth interviews and using the MAXQDA software, and in the quantitative part, confirmatory factor analysis was used to verify the validity of the constructs and the PLS software. The findings in the qualitative part, based on the analysis performed in the open coding stage, identified 145 codes, and then 18 core codes and finally 3 selective codes (social media marketing, mobile marketing, and local marketing) were categorized. Also, in the quantitative part of the research, for the overall model fit, the GOF criterion was used, which resulted in a value of 0.830, indicating a strong overall fit of the model in the present research. The results of this research have provided the tourism sector with the opportunity to better access, interact with, and analyze customers, and have improved marketing management in this industry. Especially for the dissemination of attractive and valuable content, it can help attract and convert potential customers. This validated framework warrants further investigation for its applicability across other consumer-facing industries.

**Keywords:** SOLOMO Marketing; Social Media Marketing; Mobile Marketing; Local Marketing; Content Analysis; Projective Techniques.

## 1- Introduction

SOLOMO marketing is actually a type of modern marketing that can be described as a combination of three methods: social media marketing, local marketing, and mobile marketing. SOLOMO is an emerging concept in marketing that can leverage modern digital marketing tools and explore the convergence of social media, the execution capability of social media marketing, and mobile connectivity. SOLOMO has the potential to set new standards for service personalization and establish a new paradigm in customer-centric marketing. One new type of marketing that can be particularly useful in the tourism sector is solo marketing [1].

The rapid development of information technology and the widespread use of various social media have brought about great changes in the current tourism industry and have also put forward more new requirements for tourism management strategy and marketing management tools. The deep integration of tourism with technologies such as mobile Internet devices, online sharing platforms, and geographic positioning information systems has also given the tourism industry a richer concept and development trend [2].

In developing countries, the tourism industry can be considered one of the fastest and fastest growing industries, serving as an important source of income and foreign exchange reserves that drives growth and development. This industry can definitely pave the way for sustainable development and create job opportunities in these countries.

✉ **Elham Fazeli Veisari**
elhamfv@iau.ac.ir

In this regard, attention should be paid to all factors affecting the improvement of the tourism industry, among which it is essential to pay attention to the role of appropriate marketing and, in particular, to turn to digital marketing elements and methods to anticipate needs and satisfy tourists [3].

Today, in many countries, tourism is considered the main force of economic development and growth, and it is so important in the economic and social growth of countries that economists have called it an invisible export. In a situation where there is a lot of competition among tourism product suppliers to attract travelers, companies will be successful that distinguish themselves from competitors and create a favorable and unique position in the minds of their consumers [4].

Marketing in the tourism industry involves anticipating the needs and satisfying current and future tourists, which is the basis for travel companies and suppliers to compete with each other. In the tourism industry, marketing has been done in different periods based on the media that was available to the industry's practitioners, and with the emergence of new media, this type of marketing activity had to change. The new media of our era, especially social networks, are a great opportunity to present a new way in the field of marketing with multimedia and sharing capabilities (Fahmi and colleagues, 2022 and Tussyadiah).

Social media has become the most popular communication tool for service providers and consumers [5].

For a country like Iran, which is dependent on oil revenues and this is considered one of the main weaknesses of its economy, tourism development moves the country out of a single-product economic system and brings the country abundant income. Tourism and its latent capacities can be one of the strategies used in implementing the idea of resistance economy. Due to its very high potential for the country's economic development and the need to diversify income sources and reduce the country's dependence on a single source such as oil, and to create employment and development, this industry can be referred to as a hidden wealth that the country has not yet been able to bring its various parts from potential to reality [4].

In today's complex, dynamic and highly variable environment, companies need to design and adopt strategies that can help them improve their performance day by day. Because in such a competitive environment, only companies that are able to survive can keep up with the changing and dynamic conditions of the competitive market [6].

Companies must resort to unconventional marketing methods to survive in the competitive arena. One of these unconventional methods is solo marketing, which has attracted the attention of many companies in recent years, but not much research has been done in this field, especially in the tourism industry, and the effects of this type of marketing on consumer behavior are still not completely clear. Therefore, it can be said that one of the obstacles to the success of the tourism industry in Iran is the reliance on traditional marketing tools and communication channels and the lack of sufficient attention to the use of new technologies in this field. Despite the importance and significant role of media, social networks, and tools such as mobile phones in influencing consumer behavior and the increasing growth of this communication tool, it seems that this issue has been somewhat neglected in the literature and research in the field of tourism in the country, as well as in the planning of tourist attraction programs, and there is a research gap in this field. According to the presented materials, the main goal of the research is to develop the SOLOMO marketing scale in the tourism industry. The novelty of this study lies in employing an integrated mixed-method approach to comprehensively identify and model the key components of SoLoMo marketing. Unlike most previous studies that examined social, mobile, or location-based marketing separately, this research unifies these dimensions within a single conceptual framework. By combining qualitative content analysis with quantitative validation through confirmatory factor analysis, the study provides a localized and empirically tested model of SoLoMo marketing. This framework offers valuable insights for businesses seeking to design more effective, context-aware digital marketing strategies that align with mobile consumer behavior and local engagement dynamics.

## 2- Theoretical Framework

The tourism industry, much like many other economic sectors, has recently witnessed profound transformations driven by digital transformation. The emergence of new technologies, shifting traveler expectations, and the increasing prevalence of online platforms have completely altered the nature of marketing and service delivery in tourism. In this era, the success of businesses is no longer solely dependent on the allure of destinations or the quality of traditional services, but rather on their ability to integrate smart marketing strategies, personalize customer experiences, and effectively utilize big data. Identifying and understanding the key components of this new marketing ecosystem, including social media marketing, mobile marketing, and local marketing, is crucial for maintaining a competitive advantage and reaching the new generation of customers. This research endeavors to build a comprehensive model for mastering these digital marketing components within the dynamic tourism landscape. In this section, the basic idea used in this research is defined. in the section, definitions of SOLOMO marketing are examined, and then the marketing framework of the SOLOMO model in tourism destinations is explained.

## 2-1- SOLOMO Marketing

SOLOMO Marketing is a type of advertising strategy that integrates the best of both worlds to transform technology, e-commerce, digital marketing, media and public relations. SOLOMO revolves around the idea of today's consumers consuming more content on their mobile phones and trusting the opinions of their social media friends around their geographic location [2].

The SOLOMO model analysis framework was first proposed by American investment researcher John Dwyer (2011), which is based on the current process of information and communication technology and media communication theory. The SOLOMO-based marketing model is recognized as the future development trend of marketing management. By better explaining this model, it can provide a useful reference for the development of tourism destination marketing in the new Internet era [2].

First of all, it is for "social", which is mainly due to the development of Internet science and technology, such as the introduction of high-speed 4G and 5G networks into people's lives, the efficiency of information transmission has been greatly improved. People can use video phones, cameras, videos, audio, and quickly upload what they have to all kinds of social media platforms, and a wide range of sharing. The points raised are typical of the "social" feature, which forms the foundation of marketing in the current Internet era. Secondly, with the increasing intelligence of mobile terminals such as mobile phones, tablets, and smart watches, the performance of LBS service (Location-based service) on most mobile terminals, it enables the service to determine local location more accurately. Finally, for "mobile", it emphasizes the importance of mobility and flexibility of mobile terminals in the Internet era. It creates almost an information portal for users to receive and give feedback on all kinds of information at all times, which enables different people to communicate effectively through the Internet platform [3].

Figure (1) describes the SOLOMO model analysis framework and its operating mechanism in marketing management.

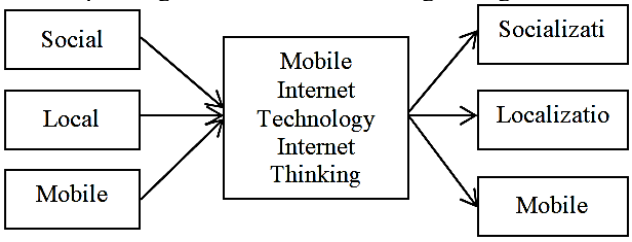



Fig 1. The main framework of the SOLOMO model

Social development creates a marketing environment where everyone participates, while local mainly makes the positioning of the marketing target more accurate and reliable, and provides more local and personalized services. Mobile works by making the carrier of information transfer more flexible and effective, so that marketing behavior can receive more appropriate feedback [4].

The Internet and digitalization have brought about many changes in today's world, both positive and negative. Iran has not been and will not be an exception to these changes. With the improvement of the Internet infrastructure in the country, it has become possible for businesses to operate on this platform [7].

With these opportunities increasing, a number of challenges arise that businesses looking to adopt SOLOMO marketing should consider in their strategy [8].

Many challenges arise from the existing literature on social media marketing, context-aware marketing, and mobile marketing. One of the main challenges to address is privacy concerns, including who will manage this information and how. Collecting aggregated contextual information about an individual raises questions about how to do this. The technique is becoming a powerful tool for marketers. Smartphones are rapidly becoming the remote control of people's lives. With the growing adoption of smart mobile devices and their proliferation into consumers' daily lives, SOLOMO marketing will play an important role in the near future. In particular, smartphones appear to be benefiting from a series of innovations through applications that provide highly valuable services [9].

## 2-2- Social Media Marketing Path

Social media marketing is the most specific part of the SOLOMO model of marketing management. With the rapid growth of internet users, the download of various social media applications has increased, which makes it possible for tourism destinations to carry out social media-based marketing [10].

According to the daily statistics of magazines, out of Iran's approximately 82 million population, about 73 million people, or 89 percent of the total population, are using the internet. 77 percent of the Iranian population uses mobile internet, and about 57 percent of Iranians, or 47 million people, use social media. 3 million people are added to the number of internet users in Iran every year, which is a staggering number. Therefore, from the above data, it can be seen that Iran basically has a good social media market, and given these statistics, businesses in the tourism industry can make good use of this opportunity.

In this environment, tourism destination marketing can also take advantage of the rapid development of online social media, considering social media as the main port for tourism marketing [11] and implement different destination marketing strategies according to the characteristics of different ports [12].

## 2-3- Local Offline Marketing Path

Online marketing is based on the focus and characteristics of social media, and local offline marketing of tourism destinations is an effective online marketing action on social media. Otherwise, online marketing activities without the support of offline institutions are mostly rootless, which cannot have a practical effect on improving the tourism environment and tourism quality of tourism destinations. Different from online social media marketing, it can only attract the target group in the virtual visual or auditory perception. Offline marketing activities are more involved in the actual shaping experience, through their own experience related to tourism and cultural activities, with the social media platform to achieve a good contrast with the sensory experience, enhance the useful experience. In addition, participants can feedback their experiences and activities back to the social media platform, which can form an effective iteration and effectively create a comprehensive positive feedback marketing approach of "online perception, offline experience, experience feedback, online secondary perception" integrated with online and offline [13].

In terms of specific measures, offline tourism destination marketing activities can mainly adopt the following ways and strategies. First, it is necessary to understand the user information data and the needs of the tourism group audience according to different tourism seasons to determine the main direction of tourism destination marketing. For example, in the off-season of each year, attract nearby tourists to frequent games through monthly tickets or special tickets. In the peak season, popular activities such as "cultural festivals", "trade fairs" and "special concerts" can be held to increase the popularity of tourism destinations. The second is to optimize and improve the marketing mode of tourism destinations, which mainly focuses on exploring and extracting local tourism resources. Although tourism destination marketing is very important, its basic core is that the tourism destination itself must have valuable tourism resources and services that can attract tourists. Third, the integration of local tourism resources is the formation of tourism brand. An important goal of marketing activities is to create a well-known tourism brand. Under the support of brand tools, it will be easier to carry out marketing activities. Therefore, tourism destinations should actively integrate a variety of local tourism resources, take advantage of their localization advantages, explore local tourism resources, and launch various tourism brand marketing activities under the support of local governments [14].

## 2-4- Mobile Marketing Path

Mobile is an effective link between online and offline marketing tools. At present, mobile tools have become the most important gateway for customers. When formulating online and offline marketing channels, we must also rely on information technology to deeply explore the content of mobile marketing management, firmly grasp the mobile portal, and add its foundation to the marketing of tourism destinations [12].

According to the above theoretical foundations, it can be seen that, relying on the SOLOMO model, tourism destination marketing should take online social media marketing as the main tool, combine various offline marketing activities as an effective auxiliary tool, and use information technology support to obtain tools. With the joint efforts of these three components, a multi-level and comprehensive tourism destination marketing path is formed, and the tourism destination marketing promotion strategy is realized in the current Internet era. This process is shown in Figure (2).

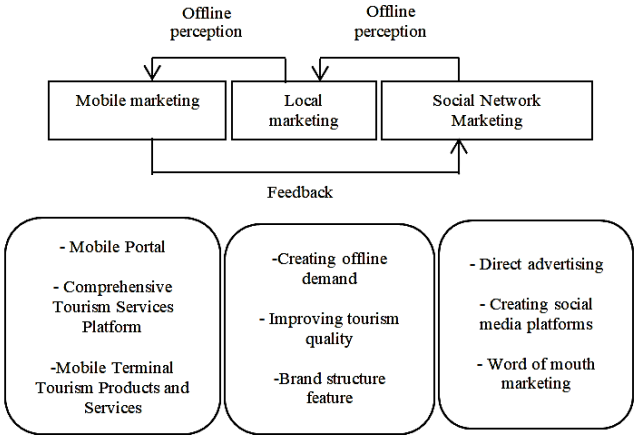



Fig 2. SOLOM0 Destination Marketing Framework

Review and analysis of the background of the subject of this study shows the expansion of studies, especially foreign studies, in this field, but little research has been conducted in the country and no research has been conducted to identify new components and present a model in this field. Accordingly, the following domestic and foreign research can be mentioned in this regard: In a study to design and validate the electronic marketing model in the tourism industry, 12 main categories were achieved using a mixed method, which was followed by the results of structural equation modeling that the effect of all independent variables on the development of electronic marketing in the tourism industry of Yazd province was positive and significant [15].

In a study aimed at investigating the role of using social networks and electronic customer relationship management in improving marketing performance, the statistical population included managers and employees of insurance companies, and using a structural equation model, the results show that using social networks has a positive effect

on the success of implementing electronic customer relationship management. Success in implementing electronic customer relationship management also has a mediating role in the relationship between using social networks and marketing performance of insurance companies [16].

In another study aimed at investigating the effect of SOLOMO marketing on tourists' behavioral intentions and destination brand equity, tourists in the last 5 years were considered as the statistical population in Isfahan, Shiraz or Susa. The data were available using an online questionnaire and sampling. Finally, the research findings show that SOLOMO marketing has a positive and significant effect on tourist destination brand equity and tourists' behavioral intentions. Also, brand equity plays a mediating role in the effect of SOLOMO marketing on tourists' behavioral intentions [17].

In a study aimed at studying the content analysis of bank profiles on social networks, the social pages of 14 foreign and Iranian banks on 4 well-known social networks were studied using the content analysis method. Ultimately, the findings showed that most banks use their social pages to display bank information about their mission/goals, to provide product information about the added product, to show organizational identity about the company image, to introduce bank events about conferences, to display bank videos about the music style, to display bank photos about the lifestyle, to support customers about information support, to display advertising slogans about the intimate/conversational/rhythmic/poetic style, to display bank albums about their campaigns, to present issues related to the bank's social responsibility, to use sports and environment-related posts, to send bank marketing messages about services, to publish bank information about the bank, to publicize the bank's social responsibility and public information and advice, and also to engage customers interactively [18].

An external study on sustainable development in digital marketing found that digital marketing strategies promote sustainable development by encouraging sustainable consumption patterns. This scenario stated that by understanding consumer behavior, communicating key messages through the best channels and green marketing campaigns, it influences consumer attitudes and purchasing decisions [3].

In a study that analyzed the most buzzwords in advertising in 2022, it attempts to examine buzzwords, buzz marketing, and the main reason behind the spread of these marketing strategies in the business world. In conclusion, the study showed that such a type of marketing strategy has the potential to increase the speed of advertising messages to end consumers as well as increase brand sales [2].

In another study, they examined the moderating role of gender for SOLOMO-based product recommendations on consumers' acceptance intention and finally found that consumers' acceptance of SOLOMO-based product recommendations is determined by source credibility and recommendation usefulness; perceived source credibility, accuracy, and perceived benefit have a positive effect on recommendation usefulness. In addition, gender is an effective moderator [19].

In a study, SOLOMO examined the financial behavior of platform users during the COVID-19 pandemic and found that defining financial behavior will help companies adapt to the current situation. They concluded that the results obtained from the findings can be applied across different business sectors [1].

In an article, location services and marketing communications were studied from a global perspective, and the aim was to examine the characteristics associated with the localization of portable electronic devices in space and related efforts to improve the level of personalized communications in order to correctly time the wireless distribution of advertising content to the recipient. This further helps the findings of this study to identify different types of localization technologies that convey information in SOLOMO marketing to the target audience [20].

In a study, how SOLOMO-based product recommendations affect consumers' acceptance intention with the moderating role of gender was tested. According to the information acceptance model, this study examined how source credibility, perceived accuracy, and perceived benefit affect consumers' acceptance intention towards SOLOMO-based product recommendations through recommendation usefulness, with gender as a moderator [21].

In a study, the university library's WeChat knowledge service system was investigated based on SOLOMO. Referring to the concept of the SOLOMO Internet boundary, this study designs a new mobile smart service system, including system architecture design, content design, and data communication design [22].

To strengthen the theoretical foundation of the model, three key frameworks are utilized that directly overlap with technology-driven and interactive components:

Technology Acceptance Theories (TAM and UTAUT) for Explaining the Use of Mobile and Online Technologies

Since the Mobile Marketing component and the use of online intelligent tools play a central role in your model, technology acceptance theories are essential for explaining how and why individuals (both tourists and businesses) adopt these tools.

Technology Acceptance Model (TAM)

As the most fundamental model, TAM predicts an individual’s intention to use a system based on two core constructs:

- Perceived Usefulness (PU): The degree to which a user believes that using this technology (e.g., a booking application, mobile marketing platforms) will enhance their performance.

- Perceived Ease of Use (PEOU): The degree to which a user believes that using this technology is free of cognitive effort.

Relevance to SOLOMO: To explain the adoption of Mobile Marketing and the use of Online Intelligence tools by tourists, PU and PEOU must first be measured.
Reference: Systematic literature reviews (such as the literature review in the tourism field) have shown that TAM remains a robust tool for assessing technology acceptance in the hospitality and tourism industry ([1], [3]).
Unified Theory of Acceptance and Use of Technology (UTAUT)
As a more comprehensive extension of TAM, UTAUT incorporates other factors that are vital for networked environments:

- Social Influence: The importance of peer recommendations and communities in social networks on user decision-making.
- Facilitating Conditions: The availability of necessary infrastructure and support for technology use.

Relevance to SOLOMO: UTAUT is directly ideal for explaining the success of Social Media Marketing (due to the presence of "Social Influence") as well as the success of Online Support Services (due to the presence of "Facilitating Conditions"). Newer advancements like UTAUT2 are also applicable in explaining the adoption of sustainable technologies (which may relate to Green Marketing).
Reference: The literature suggests that the UTAUT model can be integrated with other theories to provide greater explanatory power in studies of Information and Communication Technology (ICT) in tourism ([4]).
Consumer Engagement Theory for Explaining Communication
The components of Social Media Marketing and Viral Marketing (which appear frequently in your data) necessitate a framework beyond mere technology adoption; this framework must focus on the level of cognitive and behavioral involvement of the user.
Relevance to SOLOMO: Consumer Engagement Theory explains how two-way interactions activate emotions, cognitions, and brand-related behaviors (such as sharing, commenting, and loyalty) in active users. This theory can well explain how "User-Generated Content (UGC)" activities (seen in your attached data) lead to co-creation of value and Smart Loyalty.
Media Richness Theory (MRT) for Local Marketing
While previous theories focus on digital aspects, Local Marketing sometimes requires face-to-face communication or media capable of rapid feedback transfer and handling ambiguities.
Relevance to SOLOMO: MRT posits that "rich" media (such as face-to-face communication or video conferencing) are better suited for transmitting complex and ambiguous messages, whereas lean media (such as simple SMS or email) are better for transmitting routine and transparent data. This helps justify the strategic selection of the appropriate tool for Local Marketing and the transmission of Transparent Information [23,24,25,26].

## 3- Research Methodology

Given the purpose of the research, which is to develop the SOLOMO marketing scale in the tourism industry, a mixed method (qualitative and quantitative) was used. Given the quantitative research conducted in the field of SOLOMO marketing and the lack of SOLOMO marketing scales and components, the qualitative part of this research attempts to identify new components of SOLOMO marketing. Among the methods of conducting qualitative research, the projection technique was used. The main idea in this method is that people are more willing to project their feelings onto others than to attribute them to themselves. This method helps the person to express things that are difficult to say directly, indirectly and is considered less threatening and revealing to the person [27].
Projection techniques, originally developed in clinical psychology to explore individuals' subconscious thoughts and emotions, were later adapted by marketing researchers to gain deeper insights into consumer perceptions, motivations, and attitudes. Given that the present study focuses on SoLoMo (Social–Local–Mobile) marketing—where social influence, trust, and authenticity play crucial roles—this approach is particularly useful. In social and influencer marketing contexts, consumers tend to rely more on the opinions and recommendations of friends, family members, and trusted influencers than on traditional advertising messages. As a result, projection methods provide a powerful means to access consumers' implicit beliefs and emotional responses, yielding data that are both richer and closer to their actual decision-making behavior.
Projection techniques are generally classified into three main categories based on the nature and depth of responses they elicit: word association, sentence completion, and dream or imaginative exercises. In the word association technique, participants are presented with a sequence of stimulus words—usually unrelated to reduce bias—and are asked to immediately state the first word or thought that comes to mind. The sentence completion method extends this idea by requiring respondents to complete an unfinished sentence, producing more elaborate and spontaneous associations that reveal underlying attitudes or perceptions. Finally, in dream or imaginative exercises, participants are encouraged to imagine or describe hypothetical scenarios, leveraging the creativity and symbolic expressions that emerge when logical constraints are relaxed. These exercises can uncover deep-seated emotions and motivations that might not be readily articulated in direct questioning.
In the present study, semi-structured interviews constituted the main instrument for qualitative data collection. The interview protocol was designed using projection-based questions and prompts, allowing respondents to express their perceptions freely and indirectly. The qualitative population consisted of experts and professionals active in

the tourism industry—a field where SoLoMo marketing is rapidly evolving. Participants were purposefully selected based on their expertise and engagement with digital marketing practices in tourism, and 15 in-depth interviews were conducted until theoretical saturation was achieved.
The collected data were analyzed using MAXQDA software. To extract and conceptualize the key components of SoLoMo marketing in tourism, qualitative content analysis was employed. Both manifest (explicit) and latent (implicit) content analysis approaches were applied—meaning that, in addition to direct textual meanings, indirect and symbolic interpretations embedded in participants' responses were also considered. The analysis followed a structured process including: data transcription, coding of meaningful units, categorization of themes, and interpretation to identify the fundamental dimensions underlying SoLoMo marketing behavior.

- Implementation of interviews: The recorded interviews were implemented in Word software format.
- Data summary: The findings from each interview were summarized and coded in the form of tables in MaxQDA software.

Data classification: The findings from each interview were placed in a separate table. A similar general table was completed for the codes assigned to the concepts, which resulted in 145 items that were provided to 12 experts who had worked in this field or had empirical competence. Following the qualitative interview phase, the resulting findings and concepts were systematically organized into a comprehensive Conceptual Code Matrix, which yielded 145 final codes (sub-categories). To enhance the Content Validity and Reliability of this conceptual framework, an Expert Review process was implemented. The final code matrix was distributed to 12 field experts possessing relevant empirical competence and specialized backgrounds in areas such as digital marketing and technology. Experts were asked to evaluate each of the 145 codes based on predefined criteria (including direct relevance to the SOLOMO conceptual model, clarity, and comprehensiveness). Acceptance of the codes was determined by a Consensus Criterion. Only those codes agreed upon by a minimum threshold of 80% of the experts were adopted into the final model. This two-stage validation (researcher coding followed by expert ratification) ensures the effective conversion of qualitative data into quantifiable structure for the mixed-methods approach.
In the following, in the quantitative part of the present study, factor analysis is used to find out the underlying variables of a phenomenon or summarize a set of data. The primary data for factor analysis is the correlation matrix between variables. Since in this study, the relationships between the constructs were previously identified through qualitative analysis, a confirmatory factor analysis approach is used in this part. To confirm the factors extracted from the research questionnaires, confirmatory factor analysis method based on PLS software was used. One of the prerequisites and very important things in structural equation modeling and in general testing research hypotheses is to examine the validity and quality of the measurement tool, or questionnaire (in most cases). In general, the concept of construct validity deals with whether the questions designed for a construct or latent variable are related to that construct or not?
The verification and validation of validity and reliability in qualitative research generally includes four criteria: validity, transferability, reliability, and confirmability, according to Guba and Lincoln in 2000. The present study is valid due to continuous engagement, the use of integration in the research, and the researcher's review. Reliability is very similar to reliability. In this study, since the findings are close and related to each other, the reader will be able to evaluate the adequacy of the analysis by following the researcher's decision-making process. One of the criteria for the reliability of data is confirmability. Confirmability is a gradual and continuous process criterion. Recording data step by step and the time sequence of the data collection process are very important in confirmability. To describe the demographic characteristics of the study population, descriptive statistics indicators were used to categorize data related to gender, age, and education level. Among them, about 60% were men and 40% were women. The lowest number was under 30 years old, and the highest number was between 40 and 50 years old, with about 40%. In terms of education, bachelor's degrees accounted for the largest number, with about 48%.

## 4- Research Findings

In order to extract the components of SOLOMO marketing in the field of tourism, qualitative content analysis and projection technique were used. There is no specific formula for determining the sample size in qualitative research, and the main criterion for this is that we select a sample of experts, experienced and skilled in the field under study to meet our research needs. The main criterion for sampling in qualitative research is quality, not quantity. For sampling, it is recommended to use the purposive sampling method, and the condition for sampling was to reach theoretical saturation. The number of interviews conducted was 15 interviews. In this method, specific participants are consciously selected by the researcher. Regarding saturation, it should be noted that the sample size of N=15 was achieved after the data collection was stabilized, meaning that subsequent interviews did not provide any new and significant concepts, codes, or relationships related to the SOLOMO model. This confirmed the operational achievement of theoretical saturation within the scope of our research questions and

therefore, justified the end of the data collection phase at this stage, as is standard practice in rigorous qualitative research. The indicators and components affecting SOLOMO marketing in the field of tourism can be categorized as follows. For classification, higher-level concepts are placed as categories and lower-level concepts as subcategories. The method of describing subcategories forms the categories. As can be seen in Table 1, the open codes identified through the interviews include 145 components, which were categorized into 18 axial codes and 3 Greenwich codes based on semantic affinity and family affiliation. All selective codes are also categorized into two categories: visible and invisible. In Table 2, we classify the characteristics of the concepts and axial anopen codes extracted from the analysis of the interviews

Table 1: Characteristics of axial and open codes extracted from the analysis of interviews

| Components | Subcategory of conceptualization in the axial code | codes | Open code |
|---|---|---|---|
| Social Media Marketing | Online Services | 8 | 48 |
| | Online Intelligence | 20 | |
| | Online Data Processing | 11 | |
| | Online Advertising | 9 | |
| | Online Trust | 8 | 36 |
| | Online Promotion | 17 | |
| | Online Viral Marketing | 11 | |
| Local marketing | Smart Advertising | 8 | 31 |
| | Smart Space Design | 7 | |
| | Smart Branding | 4 | |
| | Smart Marketing | 12 | |
| | Smart Customer Orientation | 6 | 15 |
| | Smart Loyalty | 3 | |
| | Smart Innovation | 6 | |
| Mobile marketing | Online Support | 7 | 10 |
| | Complaint Support | 3 | |
| | Advertising Support | 2 | 5 |
| | Data Support | 3 | |
| **Open and axial code** | | **18** | **145** |

The identified components of SOLOMO marketing in a network are shown in Figure 3 based on the output of MAXQDA software:

In the second part of the study, which is the quantitative part, the concept of construct validity will be used to examine whether the questions designed for a construct or latent variable are related to that construct. The construct validity of SOLOMO's marketing indicators and the fit of the overall research model are shown below.

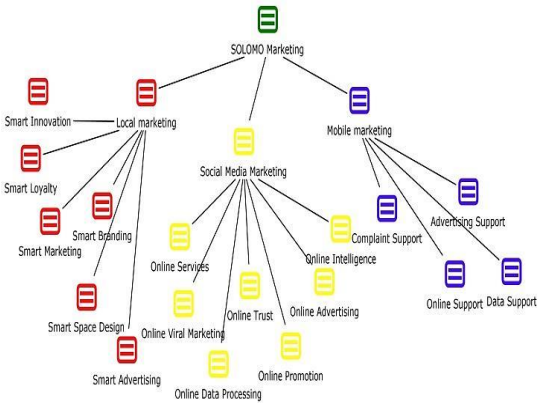



Fig 3. Solomo Marketing Components Network in the Tourism Field

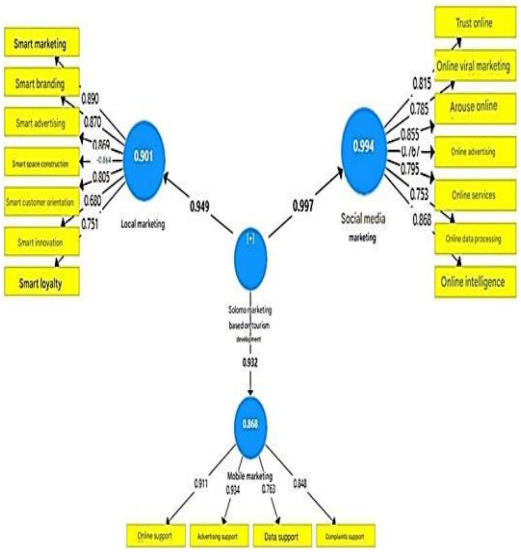



Fig 4. General research measurement model (Path coefficient)

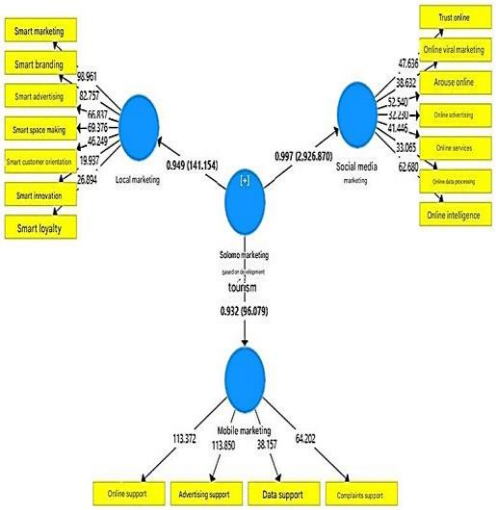


Fig 5. General research measurement model (t-significance coefficients)

Table 2: Cronbach's alpha, Composite Reliability and Convergent Validity of General realization model

| Variables | Cronbach alpha (Alpha>0.7) | Composite Reliability (Cr>0.7) | AVE (AVE>0.5) |
|---|---|---|---|
| Mobile Marketing | 0.877 | 0.923 | 0.751 |
| Social Media Marketing | 0.910 | 0.928 | 0.650 |
| SOLOMO Marketing Based on Tourism Development | 0.956 | 0.972 | 0.920 |
| Local Marketing | 0.918 | 0.935 | 0.675 |

Given that Cronbach's alpha, composite reliability (internal consistency), and average extracted variance are all within the relevant range, it can be confirmed that the overall model's reliability and convergent validity are appropriate.
According to the matrix in the table 3, the principal diameter values of all constructs are greater than their correlation values with other constructs, which indicates appropriate divergent validity and good fit of the overall measurement model.
Given that the calculated value of the t-statistic for the paths is not in the range [1.96, -1.96], it can be concluded that the constituent indices are capable of measuring this construct.

## 4-1- Checking the Predictive Fitness of the Overall Model

The value of the Q^2 (Stone-Geisser) statistic, which determines the predictive power of the model in endogenous constructs. Models that have acceptable structural fit should be able to predict the endogenous variables of the model. This means that if the relationships between constructs are correctly defined in a model, the constructs have sufficient influence on each other and thus the hypotheses are correctly confirmed. Hensler and colleagues (2009) have determined three values of 0.02, 0.15 and 0.35 as low, medium and strong predictive power.

## 4-2- Overall Model Fit (GOF Criterion)

The overall model includes both the measurement and structural model parts, and by confirming its fit, the fit check in a model is complete. To check the fit of the overall model, a single criterion called GOF is used:
This criterion is calculated through the following formula:

$$GOF = \sqrt{\overline{Communalities} \times \overline{R^2}}$$

Thus, the value of the GOF criterion for the above research is equal to:

$$GOF = \sqrt{\overline{Communaty} \times \overline{R^2}} = \sqrt{0.749 \times 0.921} = 0.830$$

Table 3: Fornell and Larker matrix to examine the divergent validity of the research model

| Independent Dependent Path | Coefficient Value | Standard Deviation | t-Test | p-value |
|---|---|---|---|---|
| SOLOMO Marketing Based on Tourism Development Mobile marketing | 0.923 | 0.010 | 96.079 | 0.0001 |
| SOLOMO Marketing Based on Tourism Development Social Media Marketing | 0.997 | 0.000 | 2926.870 | 0.0001 |
| SOLOMO Marketing Based on Tourism Development Local marketing | 0.949 | 0.007 | 141.154 | 0.0001 |

Table 4: T-statistic, path coefficient and standard deviation of the research model

| Situation | $Q^2$ Stone-Geisser $Q^2$(=1-SSE/SSO) | SSE | SSO | Variables |
|---|---|---|---|---|
| Strong predictive fit | 0.628 | 625.399 | 1680.000 | Mobile Marketing |
| Strong predictive fit | 0.624 | 1106.647 | 2940.000 | Social Media Marketing |
| Strong predictive fit | 0.584 | 1222.865 | 2940.000 | Local Marketing |
| Situation | $Q^2$ Stone-Geisser $Q^2$ (=1-SSE/SSO) | SSE | SSO | Variables |

Table 5: Stone-Geisser statistic values of research variables

| | Mobile Marketing | Social Media Marketing | SOLOMO Marketing Based on Tourism Development | Local Marketing |
|---|---|---|---|---|
| Mobile Marketing | 0.867 | | | |
| Social Media Marketing | 0.804 | 0.806 | | |
| SOLOMO Marketing Based on Tourism Development | 0.832 | 0.803 | 0.959 | |
| Local Marketing | 0.774 | 0.801 | 0.819 | 0.821 |

Table 6: Communality and Observable Variable R2

| Observable Variable | *Communality* | $R^2$ |
|---|---|---|
| Mobile Marketing | 0.751 | 0.868 |
| Social Media Marketing | 0.650 | 0.994 |
| SOLOMO Marketing Based on Tourism Development | 0.920 | - |
| Local Marketing | 0.675 | 0.901 |
| Average of the above values | 0.794 | 0.921 |

Considering three values of 0.01; 0.25 and 0.36 as weak, medium and strong values for GOF (Mohsinin and Esfidani, 2014: 73), the value of 0.830 for GOF indicates a strong overall fit of the model in the present study.

## 5- Discussion and Conclusion

In this study, based on the main objective of conceptualizing and categorizing SOLOMO marketing components in the field of tourism, 145 open codes and 14 core codes were extracted from the content analysis of the interviews, which included 376 sentences, and the new SOLOMO marketing components were classified into 3 categories. The first category is social media marketing, which is itself classified into two visible and invisible parts. In the visible part, the core components and codes include online services, online intelligence, online data processing, and online advertising. The intangible part of the core codes includes online trust, online stimulation, and online viral marketing. The second category of local marketing is also classified into two visible and invisible parts. In the visible part, the core components and codes include smart advertising, smart space building, smart branding, and smart marketing. The intangible part of the core codes includes smart customer orientation, smart loyalty, and smart innovation. Finally, the third category is mobile marketing, which in the visible part includes online support components, complaint support, and in the intangible part includes advertising support, data support. To benefit from the present research model, we first draw the identified components in the form of Figure (5).

Also, in the quantitative part of the research, considering the Cronbach's alpha numbers and the composite reliability (internal consistency) and the average variance extracted are all in the relevant range, it can be confirmed that the reliability and convergent validity of the overall model are appropriate. Also, based on the Fornell and Locker matrix of the divergent validity of the research model, the values of the main diameter of all the structures are greater than their correlation values with other structures, which indicates appropriate divergent validity and good fit of the overall measurement model. Considering that the calculated value of the t-statistic for the paths is not in the range [1.96, -1.96], it can be concluded that the constituent indicators are able to measure this structure. Finally, considering the value of 0.830 for GOF, it indicates a strong overall fit of the model in the present study.

Table 1 comprehensively outlines the complete set of identified components, illustrating the clear hierarchical relationship between the sub-dimensions and their overarching primary dimensions.

Based on the findings of the research, it was generally shown that the components of SOLOMO marketing were placed in three parts, so it can be said that the present study is in line with Shir Shamsi (2023) who, in his research using a mixed method and a mixed approach in the tourism industry, reached 12 new categories that are consistent in some components, for example, electronic and smart marketing and also branding, which ultimately all of these things reduce costs, and in both studies, electronic marketing is efficient in the country's tourism industry [15]. Also, Mohammadi (2022) in his research, which aims to influence mobile marketing on tourist behavior, is in line with the present study in a main component, which is mobile marketing. The results of the two studies confirm that mobile marketing can be used as a new and effective tool to promote tourism destinations [24]. Asgarnejad (2022), who in his research sought to examine the role of social networks and emphasized the impact of these networks on customer relationships, is in line with the present study and is in line with it, explaining that cyberspace has an impact on businesses and causes easy communication, increased customer power, increased competition, word-of-mouth advertising, and as a result, reduced marketing and advertising costs, and leads to brand loyalty [16]. Rosario (2023) is in line with the present study in that it is the same as one of the main and most frequent components and codes of green marketing campaigns, because both studies emphasize that green marketing has a positive effect on consumers' attitudes and purchasing decisions [3]. Zhou and Chang (2021) are also in line with the present study because SOLOMO affects consumers' acceptance intentions, and the present study has analyzed all dimensions of SOLOMO [19].

Practical suggestions: One of the basic points that should be considered in this context is the way of looking at SOLOMO marketing because, as mentioned in the theoretical foundations section, SOLOMO marketing is a new concept in marketing that can use completely new and modern digital tools and creates convergence between social media, the ability to implement social media marketing and mobile connectivity. SOLOMO marketing is a combination of three types of marketing including social media marketing, local marketing and mobile marketing, in fact, the interaction of social networks, local information and the mobile network at the right time and place has created a new service called SOLOMO. Therefore, it is suggested that considering that one of the main components of social media marketing is and the new components identified in this category are online intelligence, online advertising and online viral marketing, the results of this research can be useful for tourism businesses and marketing managers in this industry so that SOLOMO can reduce costs in the advertising sector in large volumes by using online viral marketing and adopt the right strategy and attract more customers with the lowest advertising costs. Another point that can be mentioned in this section is the issue of trust, this component can create conditions in which the customer steps towards this business or company in question without

worries and with complete trust and benefits from its services. Also, with the smart marketing and the expansion of smart social networks, most have taken a broad trend towards using the new marketing paradigm. In the second section, which is local marketing and the discussion of smart advertising, smart branding and smart customer orientation is raised, in fact, this strategy is best suited to businesses in the tourism industry such as restaurants, professional services, retail stores that need a local marketing strategy and are of great help, so that they target audiences close to the business location and here smart advertising can be used correctly and content production can be used to attract more customers. Another new component is smart customer orientation. New customers are always found in the surrounding areas, and with smart advertising you can reach potential customers in your area who do not yet know about your business or do not support it. The last section is mobile marketing and the identified components include online support and since it is mentioned in the theoretical foundations that today about 80% of Internet users have smartphones, we must engage users or in other words create more interaction and use this tool to create mobile campaigns. In mobile marketing, information can be transferred quickly and the customer or audience can be informed about different campaigns. It should be noted that the effectiveness of text messages is greater than messages sent in email. The next and very important point is the low and affordable costs of mobile marketing.

Research limitations: The present study, like other studies, has limitations, and removing these limitations will pave the way for new studies. This study is almost the first study in the field of SOLOMO marketing that attempts to examine its dimensions and components in the field of tourism in Iran and also to identify the components using interviews. The main limitation is related to theoretical foundations and literature. The researcher has tried to search all relevant keywords both inside and outside Iran correctly, but the theoretical foundations that directly interpret and explain the main components of SOLOMO, especially inside the field of tourism, are very few and limited. For future research, it is suggested that based on the components identified in this study, fuzzy DEMATEL be used for cause-and-effect relationships and ranking criteria. Identify the importance of the elements and use multi-criteria decision-making methods. It is also suggested that the present study be compared in the studied population among the three generations X, Y, and Z. Presenting a comprehensive model for SOLOMO marketing success using Q methodology.

## References

[1] A. Zaušková, M. Kubovics, and M. Vanko, "Financial Behaviour of Users of SOLOMO Platforms during the COVID-19 Pandemic," SHS Web of Conferences, Vol. 92, 2021, pp. 03032. https://doi.org/10.1051/shsconf/20219203032.

[2] A. S. Sethi, V. Bisht, B. S. Bisht, and J. Singh, "MARVELS OF ADVERTISING: TALES OF TRIUMPH OF THE MOST BUZZING ADVERTISING WORDS IN 2022," The Journal of Contemporary Issues in Business and Government, Vol. 29, No. 1, 2023, pp. 679–687. https://forms.gle/1PsASopxVotMHyj96.

[3] A. T. Rosário, P. R. Lopes, and F. S. Rosário, "The Digital Marketing for Sustainable Development," JHASS, Vol. 2023, DOI: 10.26855/jhass.2023.01.033.

[4] S. Zhengyi, "Mobile Marketing Strategies in Omnichannel Context. " http://edoc.bseu.by:8080/handle/edoc/100092.

[5] R. K. Brathwaite, "Social Media and Tourism Development in Grenada: A Phenomenological Study," Doctoral Dissertation, 2019. http://researchrepository.napier.ac.uk/Output/2005971.

[6] M. Kubovics and A. Zaušková, "Possibilities for Data Collection and Example of Visualization of Environmental Activities of Businesses in the SOLOMO Environment," Economics Business Organization Research, Vol. 3, No. 2, 2021, pp. 132–154. https://dergipark.org.tr/en/pub/ebor/issue/65701/878673.

[7] A. Tarabasz, N. Patwa, K. Kkanzadee, A. Chaudhary, N. Jain, S. Basu, and S. Deepadhar, "Factors Affecting Customer Engagement in Shopping Malls through SOLOMO (Social, Local, Mobile) Application," The International Journal of Management, Vol. 6, No. 4, 2017, pp. 23–39.

[8] M. Bayat, E. Fazeli Visari, and M. J. Taghi Pourian, "A New Approach to Online Marketing in Iran: Presenting a Qualitative Model of SOLOMO Marketing with Projection Technique," Intelligent Marketing Management, Vol. 5, No. 4, 2024, pp. 69–89, DOI: JABM.3.2.15564.35997779559.

[9] D. Buhalis and M. Foerste, "SoCoMo Marketing for Travel and Tourism: Empowering Co-Creation of Value," Journal of Destination Marketing & Management, Vol. 4, No. 3, 2015, pp. 151–161. https://doi.org/10.1016/j.jdmm.2015.04.001.

[10] L. Xia, "Countermeasure Research on Transformation of Local Retail Industry Based on SOLOMO Model-Take Wuhan Hundred Group as an Example," Industrial Economy Review, 2017. www.hillpublisher.com/UpFile/202302/20230210182137.

[11] E. Fazeli Visari, M. J. Taghi Pourian, G. Ghanbarzadeh, and R. Taveli, "Developing a Viral Marketing Model in Online Businesses Using a Mixed Approach," Intelligent Business Management Studies, Vol. 9, No. 36, 2021, pp. 1–37. https://doi.org/10.22054/ims.2021.53590.1763.

[12] W. Wei, H. Peng, L. Huang, et al., "Interactive Interface Design of Online Travel Products under SOLOMO Mode," Idea & Design, No. 7, 2018, pp. 44–45.

[13] C. Huo, J. Hameed, M. W. Sadiq, et al., "Tourism, Environment and Hotel Management: An Innovative Perspective to Address Modern Trends in Contemporary Tourism Management [RETRACTED]," Molecular Human Reproduction, Vol. 27, No. 7, 2021, pp. 27. https://doi.org/10.1108/BPMJ-12-2020-0543.

[14] P. C. Udomraksasup, C. Panyadee, H. Thaveeseng, et al., "Biodiversity-Based Tourism Management for Community Enterprise Groups, Mae Chaem District, Chiang Mai Province," Allied Business Academies, No. 7, 2021, pp. 31–39.

[15] A. Shirshamsi, V. Mirabi, E. Hassanpour, and M. Ranjbar, "Design and Validation of an Electronic Marketing Model in the Tourism Industry (Case Study: Yazd Province)," Tourism

and Development, Vol. 12, No. 2, 2013, pp. 67–85. https://doi.org/10.22034/jtd.2022.321714.2543.

[16] B. Asgarnejad Nouri, A. Gholipour, and B. Firouzi, "The Role of Using Social Networks and Electronic Customer Relationship Management in Improving the Marketing Performance of Insurance Companies," Iranian Rubber Industry, Vol. 26, No. 105, 2014, pp. 57–74. https://doi.org/10.22034/irm.2022.152945.

[17] S. Mohammadi, A. Darzian Azizi, and N. Hadian, "Location-based Services as Marketing Promotional Tools to Provide Value-added in E-tourism," International Journal of Digital Content Management, Vol. 2, No. 3, 2021, pp. 189–215. https://doi.org/10.22054/dcm.2021.13683.

[18] M. Abdolmohammad Sagha, M. Handijanifard, and A. Koushki Jahromi, "Banks and Social Media Marketing: A Research Based on Content Analysis," Smart Business Management Studies, Vol. 9, No. 35, 2021, pp. 35–76. https://sid.ir/paper/1030656/fa.

[19] D. H. Zhu, Q. He, and Y. P. Chang, "How SOLOMO-Based Product Recommendations Influence Consumers' Acceptance Intention: The Moderating Role of Gender," International Journal of Services Technology and Management, Vol. 27, No. 1–2, 2021, pp. 129–142. https://doi.org/10.1504/IJSTM.2021.113579.

[20] A. Madleňák, "Geolocation Services and Marketing Communication from a Global Point of View," SHS Web of Conferences, Vol. 92, 2021, pp. 02040. https://doi.org/10.1051/shsconf/20219202040.

[21] K. Fahmi, M. Sihotang, R. H. Hadinegoro, et al., "Health Care SMEs Products Marketing Strategy: How the Role of Digital Marketing Technology through Social Media?" UJoST-Universal Journal of Science and Technology, Vol. 1, No. 1, 2022, pp. 16–22. https://doi.org/10.11111/ujost.v1i1.55.

[22] M. Chen and W. Zhang, "WeChat Knowledge Service System of University Library Based on SOLOMO: A Holistic Design Framework," Journal of Information Science, Vol. 46, No. 5, 2020, pp. 616–629. https://doi.org/10.1177/0165551519860045.

[23] Ali MB, Tuhin R, Alim MA, Rokonuzzaman M, Rahman SM, Nuruzzaman M (2024), "Acceptance and use of ICT in tourism: the modified UTAUT model". Journal of Tourism Futures, Vol. 10 No. 2 pp. 334–349, doi: https://doi.org/10.1108/JTF-06-2021-0137

[24] Li, F. (Sam), Zhu, D., Lin, M.-T. (Brian), & Kim, P. B. (2024). The Technology Acceptance Model and Hospitality and Tourism Consumers' Intention to Use Mobile Technologies: Meta-Analysis and Structural Equation Modeling. Cornell Hospitality Quarterly, 65(4), 461-477. https://doi.org/10.1177/19389655241226558

[25] C. Neves, T. Oliveira, F. Cruz-Jesus, V. Venkatesh, Extending the unified theory of acceptance and use of technology for sustainable technologies context, International Journal of Information Management, Volume 80, 2025, 102838, ISSN 0268-4012. https://doi.org/10.1016/j.ijinfomgt.2024.102838.

[26] El Archi, Y., Benbba, B. (2023). The Applications of Technology Acceptance Models in Tourism and Hospitality Research: A Systematic Literature Review. Journal of Environmental Management and Tourism, (Volume XIV, Spring), 2(66): 379 - 391. DOI:10.14505/jemt.v14.2(66).0

[27] L. G. R. Antunes, R. de Freitas Souza, A. C. Ferreira, et al., "Projective Techniques: In Search of an Alternative to Validity and Reliability," ReMark-Revista Brasileira de Marketing, Vol. 23, No. 3, 2024, pp. 1277–1314.

[28] S. Mohammadi, A. Darzian Azizi, and N. Hadianfar, "The Effect of Mobile Marketing on Tourists' Behavioral Intentions: An Analysis of the Role of Tourism Destination Brand Equity," Tourism and Development, Vol. 11, No. 2, 2012, pp. 231–247. https://doi.org/10.22034/jtd.2021.288073.2357.

Journal of Information Systems and Telecommunication
Vol.14, No.1, January-March 2026, 40-49
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

# Explainable AI for Enhanced Anomaly Detection in Fraud Detection

Reza Amiri[1*], Mohammad Hadi Zahedi[2], Mehdi Azadimotlagh[3]

[1]. Faculty member of the Advanced Information System Research Group, ICTRC, ACECR
[2]. Faculty of Industrial Engineering, K. N. Toosi University of Technology, Tehran, Iran
[3]. Department of Computer Engineering of Jam, Persian Gulf University, Jam, IRAN

Received: 14 Sep 2025/ Revised: 04 May 2026/ Accepted: 04 Jun 2026

**Abstract**

The application of machine learning has become indispensable in the critical domain of financial fraud detection. However, a major limitation of traditional models is their "black box" nature, which obscures the reasoning behind a flagged transaction. This lack of transparency often leads to many false positives, which can undermine customer trust and incur substantial operational expenses. To address this challenge, this paper proposes a novel framework for Explainable Anomaly Detection in financial fraud, using advanced Explainable AI (XAI) techniques to provide clear insights into the model's predictive processes. Our approach is designed to move beyond a simplistic binary output of "fraud/no fraud." Our framework combines advanced anomaly detection models (e.g., Isolation Forests and Deep Autoencoders) with model-agnostic explanation methods such as SHAP and LIME, to clearly show which features contribute to a transaction's anomaly score. The efficacy of our framework has been evaluated using a financial transaction benchmark dataset. The results show that integrating XAI not only makes the system more transparent and trustworthy, but also improves the efficiency of fraud investigations. Based on these results, our method reduces the time and resources needed for manual reviews, while still maintaining high accuracy in detecting fraudulent activities.

**Keywords:** Explainable Artificial Intelligence; Anomaly Detection; Fraud Detection; Interpretable Models; Machine Learning.

## 1- Introduction

Financial fraud has grown rapidly in the digital era, creating complex and ongoing challenges for both individuals and organizations. While machine learning has emerged as a crucial tool in this ongoing struggle, the efficacy of many advanced models is often hampered by their "black box" nature, which conceals the rationale behind a flagged transaction. This inherent lack of transparency often leads to many false positives, which can not only erode customer confidence but also impose significant operational and financial burdens on a company [1]. This paper, proposes a new framework for Explainable Anomaly Detection in financial fraud. This paper aims to transcend the simplistic binary output of "fraud/no fraud" by providing profound and actionable insights into the model's predictive reasoning. Our methodology combine robust anomaly detection models, such as Isolation Forests and Deep Autoencoders, with leading Explainable AI (XAI) techniques, specifically SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME).

The core contribution of this work is the development of a system that delivers a detailed, human-readable explanation for each flagged transaction. For an analyst, this means being able to swiftly and precisely identify the key contributing factors that triggered the alert. This feature is paramount for enhancing the efficiency of the fraud investigation process, enabling analysts to prioritize genuinely suspicious activities and substantially reduce the time dedicated to manual reviews. The framework's effectiveness is empirically validated using a real-world financial dataset, demonstrating that the strategic integration of XAI not only improves system transparency and trust but also significantly enhances the overall effectiveness and efficiency of fraud detection.

✉ **Reza Amiri**
amirish60@yahoo.com

### 1-1- Novelty and Motivation

This research is motivated by the critical need to address the "black box" problem in fraud detection. While machine learning models are effective, their lack of transparency leads to high false positives and significant operational costs. What makes our framework different is its two-phase design that combines unsupervised anomaly detection with a post-hoc explainability engine. This can provide a transparent, and actionable solution that can rebuild trust in AI-driven financial security systems.

The novelty of this work lies in the design of a unified, two-phase framework that explicitly separates anomaly detection from explainability in financial fraud detection. Unlike existing approaches that either rely on supervised classifiers with limited interpretability or apply explainability methods as an afterthought, this framework enables powerful unsupervised models such as Isolation Forests and Deep Auto-encoders to operate independently of the explanation mechanism. This modular design allows the system to maintain high detection performances in highly imbalanced and evolving fraud scenarios. In addition, it provides meaningful, feature-level explanations through model-agnostic techniques. In addition, the proposed approach emphasizes operational relevance by linking explanations to human decision-making efficiency, cost sensitivity, and analyst workload, rather than focusing solely on predictive accuracy.

### 1-2- Paper Structure

The organization of this paper described as follows: Section 2, Related Work, establishes the foundational context by surveying the current landscape of fraud detection and explainable AI, highlighting the existing challenges and knowledge gaps that motivate our research. This sets the stage for Section 3, Proposed Methods, where we detail our two-phase, hybrid framework for Explainable Anomaly Detection. We then provide empirical evidence to validate our approach in Section 4, Experimental Results, presenting a structured evaluation of its performance against established baselines. Following this, Section 5, Limitations, and Section 6, Discussion, provide a critical self-assessment of our work, interpreting the broader implications of our findings and acknowledging areas for improvement. Finally, Section 7, Conclusion and Future Directions, summarizes the paper's key contributions and outlines directions for subsequent research in this domain.

## 2- Related Works

The field of fraud detection has seen a significant shift from traditional rule-based systems to sophisticated machine learning models. These models, especially deep learning, reach high levels of accuracy but often act like 'black boxes,' making them hard to understand and trust. This section reviews five recent, state-of-the-art publications that address this critical challenge.

Ying Zhou et al. [2] introduces a user-centered XAI framework for financial fraud detection. Using XGBoost as the core classifier, explanations are generated via SHAP, offering both global feature importance and local case-by-case interpretability. The framework emphasizes usability for fraud investigators, aiming to reduce investigation time by providing meaningful, human-interpretable insights. Chen, Li et al. [3] integrates Federated Learning (FL) with XAI for fraud detection, allowing decentralized model training while maintaining data privacy. The framework incorporates SHAP-based explanations for interpretability, aiming to build trust with banking institutions while preserving security. The system introduces computational complexity due to FL infrastructure and has not been widely tested across heterogeneous financial institutions.

Jiaqi Liu et al. [3] proposed an Unsupervised Continual Anomaly Detection (UCAD) framework to address the challenge of incremental learning in unsupervised anomaly detection settings where labeled data are unavailable. Their approach introduces a Continual Prompting Module that uses a compact key–prompt–knowledge memory bank to guide task-invariant anomaly detection using task-specific normal knowledge, thereby mitigating catastrophic forgetting. In addition, they incorporate structure-based contrastive learning with the Segment Anything Model to enhance feature representation and anomaly segmentation by exploiting structural mask information. Extensive experiments demonstrate that UCAD significantly outperforms existing unsupervised anomaly detection methods, including rehearsal-based approaches, in continual learning scenarios.

Li, Zhong et al. [4] proposes a stacking ensemble (XGBoost, LightGBM, CatBoost) enhanced with interpretability via SHAP, LIME, Partial Dependence Plots (PDP), and Permutation Feature Importance (PFI). On the IEEE-CIS fraud dataset, the framework achieves ~99% accuracy and strong AUC-ROC while providing interpretability.

Fahad Almalki and Mehedi Masud [4] proposed a fraud detection framework that addresses the trade-off between predictive accuracy and model interpretability often seen in traditional machine learning models. Traditional models frequently prioritize accuracy at the expense of transparency, making it challenging for organizations to comply with regulations and gain stakeholder trust. Their framework combines a stacking ensemble of well-known

gradient boosting models: XGBoost, LightGBM, and CatBoost. To enhance transparency and interpretability, explainable artificial intelligence (XAI) techniques were employed. SHAP (SHapley Additive Explanations) was used for feature selection to identify the most influential features, while Local Interpretable Model-Agnostic Explanations (LIME), Partial Dependence Plots (PDP), and Permutation Feature Importance (PFI) were applied to explain the model's predictions. The IEEE-CIS Fraud Detection dataset, comprising more than 590,000 real transaction records, was used to evaluate the proposed approach. The framework achieved high performance, attaining 99% accuracy and an AUC-ROC score of 0.99, outperforming several recent related methods. These results demonstrate that it is possible to combine high predictive performance with transparent interpretability, offering a more ethical and trustworthy solution for financial fraud detection.

Amjad Iqbal and Rashid Amin [5] present an innovative anomaly detection framework tailored for time-series financial data particularly credit card fraud by combining transformer and graph neural network (GNN) architectures with ensemble approaches. The models are made interpretable using SHAP and LIME, which help highlight how specific features contribute to prediction outcomes.

Table 1: Summary table of related works

| Authors | Year | Key Techniques & Highlights | Limitation |
|---|---|---|---|
| Zhou, Ying; Li, Haoran; Xiao, Zhi; Qiu, Jing | 2023 | User-centered XAI: XGBoost + SHAP for local/global interpretability in fraud detection | Focused on XGBoost; may lack generalizability across other modeling approaches |
| Jiaqi Liu et al. | 2024 | Unsupervised Anomaly Detection (UAD) with incremental training | Limited real-world validation across diverse fraud types |
| Fahad Almalki, Mehedi Masud | 2025 | Financial Fraud Detection Using Explainable AI and Stacking Ensemble Methods | Real-world applicability remains untested; lacks complete performance evaluation in fraud contexts |
| Amjad Iqbal and Rashid Amin | 2025 | Transformer + GNN ensemble; SHAP & LIME interpretability; near-perfect accuracy | Extremely high performance based on experiments; real-world applicability remains untested |

# 3- The Proposed Framework

This paper introduces a new framework for Explainable Anomaly Detection in financial fraud, designed to address the limitations of opaque "black-box" systems. Our approach uses two-phase architecture that integrates (i) a high-performance anomaly detection core, responsible for learning and identifying suspicious transactions, with (ii) a post-hoc explainability engine, dedicated to generating human-interpretable justifications. This integration ensures that flagged transactions are not only detected with high accuracy but also accompanied by transparent rationales, essential for compliance, trust, and human-in-the-loop decision making.

## 3-1- The Anomaly Detection Core

The anomaly detection core is the first phase of the framework. Its objective is to assign an anomaly score to each financial transaction and classify it as either normal or potentially fraudulent. It uses unsupervised learning methods, especially Isolation Forest (IF) and Deep Autoencoders (DAE).

### 3-1-1- Isolation Forest

Isolation Forest operates on the principle that anomalies are "few and different" and can be isolated more easily than normal points. For a given transaction dataset $X \in \mathscr{R}^{n\times d}$ with n transactions and d features, the algorithm constructs an ensemble of t binary trees by recursively partitioning the dataset.

The anomaly score for a transaction x is computed as:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \quad (1)$$

Where E(h(x)) is the expected path length of x in the forest, c(n) is the average path length of unsuccessful searches in Binary Search Trees, defined as:

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n} \quad (2)$$

with H(n) being the n-th harmonic number. A score close to 1 indicates high anomaly likelihood [6].

### 3-1-2- Deep Autoencoder

A Deep Autoencoder learns a compressed latent representation of normal transactions and reconstructs them. Given an input transaction $X \in \mathscr{R}^d$, the encoder maps it to a latent representation z:

$$z = f_\theta(x) \quad (3)$$

and the decoder reconstructs:

$$\hat{x} = g_\phi(z) \quad (4)$$

The reconstruction error, often measured using Mean Squared Error (MSE), is used as the anomaly score:

$$\text{AnomalyScore}(x) = \| x - \hat{x} \|_2^2 \quad (5)$$

A higher reconstruction error suggests anomalous behavior [7].

## 3-2- The Explainability Engine

The second phase, the explainability engine, provides insights into the model's decisions. This engine is powered by two leading model-agnostic XAI techniques: SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME). For

each flagged transaction, these methods are used to determine the individual contribution of each feature to the final fraud score.

### 3-2-1- SHAP (Global Explanations)

SHAP values provide a unified measure of feature importance rooted in cooperative game theory. The SHAP value $\phi_i$ for a given feature i is defined by the following equation, which represents the average marginal contribution of that feature across all possible feature subsets:

$$\phi_i(v, x) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N|-|S|-1)!}{|N|!} [v(S \cup \{i\}) - v(S)] \quad (6)$$

Here, N is the set of all features, S is a subset of features, and $v(S)$ is the model's prediction for the features in set S. This calculation provides a robust, global view of feature importance that remains consistent across various model types [8].

### 3-2-2- LIME (Local Explanations)

LIME, offers a simplified, local explanation. It functions by creating a small, perturbed dataset around the flagged transaction, then uses a simple, interpretable model (like a linear model) to approximate the complex model's behavior within that localized region. This offers a different yet equally valuable perspective on the factors driving the decision for a specific transaction.

$$g(z) = w^T z + b \quad (7)$$

where z is a binary vector indicating sampled feature presence/absence [9].

This dual-phase approach ensures scalability for large financial datasets while maintaining transparency and interpretability. Isolation Forest provides efficiency, Autoencoders capture complex fraud patterns, and SHAP/LIME explanations satisfy regulatory and operational needs. The insights generated by the Explainability Engine empower human analysts to quickly validate alerts, distinguish between true fraud and harmless anomalies, and ultimately reduce the significant costs associated with false positives.

## 3-3- Algorithm of the Proposed Framework

Let $X \in \mathscr{R}^{n \times d}$ be the transaction matrix with n transactions and d features. Let $\tau \in (0,1)$ be the anomaly threshold, and let $\hat{y}(x) \in \mathcal{R}$ denote the anomaly score for transaction x. The detection core is instantiated as either Isolation Forest (IF) with t trees (subsample size $m \leq n$) or a Deep Autoencoder (DAE) trained for E epochs. The explainability engine generates per-case explanations via SHAP and LIME; let k be the number of LIME perturbations per explanation and r the number of flagged transactions (with $0 \leq r \leq n$). Algorithm 1 indicates our proposed framework in details.

**Algorithm 1:** Two-Phase Explainable Anomaly Detection

**Input:** $\mathbf{X} \in \mathscr{R}^{n \times d}$; $\mathbf{method} \in \{\mathbf{IF}, \mathbf{DAE}\}$; parameters $\mathbf{t}, \mathbf{m}, \mathbf{E}, \mathbf{k}, \boldsymbol{\tau}$.
**Output:** Anomaly scores $\mathbf{S} \in \mathbf{R_n}$; explanations $\mathbf{E} = \{\mathbf{E(x)}\}_{\mathbf{x} \in \mathbf{X_{flag}}}$.

1. **Training: Detection Core**
   If $\mathbf{method} == \mathbf{IF}$: build $\mathbf{t}$ isolation trees on subsamples of size $\mathbf{m}$.
   If $\mathbf{method} == \mathbf{DAE}$: train encoder–decoder on $\mathbf{X}$ for $\mathbf{E}$ epochs.
2. **Inference: Scoring**
   For each $\mathbf{x} \in \mathbf{X}$: compute score $\hat{\mathbf{y}}(\mathbf{x})$ via the trained core (expected path length for **IF**; reconstruction error for **DAE**).
   Collect $= (\hat{\mathbf{y}}(\mathbf{x}))_{\mathbf{x=1}}^{\mathbf{n}}$.
3. **Flagging**
   Define $\mathbf{X_{flag}} = \{\mathbf{x} \in \mathbf{X}: \hat{\mathbf{y}}(\mathbf{x}) \geq \boldsymbol{\tau}\}$; $\mathbf{let} \mid \mathbf{X_{flag}} \mid = \mathbf{r}$.
4. **Explainability** For each $x \in \mathbf{X_{flag}}$:
   SHAP: compute approximate Shapley values $\boldsymbol{\phi}(\mathbf{x}) \in \mathbf{R_d}$.
   LIME: generate k perturbed samples around x, evaluate the core, fit a sparse local linear model, and extract feature weights $w(x) \in \mathbf{R_d}$;.
   Set $\mathbf{E(x)} \leftarrow (\boldsymbol{\phi}(\mathbf{x}), \mathbf{w(x)})$.
5. **Return S and E**

Algorithm 1 outlines the proposed two-phase explainable anomaly detection framework. First, the detection core is trained using either Isolation Forest, which builds t trees on subsamples of size mm, or a Deep Autoencoder trained for E epochs. Second, each transaction x is then scored using expected path length (IF) or reconstruction error (DAE), producing anomaly scores S . Transactions exceeding threshold $\tau$ form the flagged set $X_{flag}$. For each flagged instance, explanations are generated using SHAP, which estimates feature contributions, and LIME, which fits interpretable local surrogates via perturbed samples. The framework outputs anomaly scores and explanations.

# 4- Experimental Results and Evaluation

## 4-1- Dataset Description

To evaluate the performance of our proposed framework, we conducted a series of experiments on a publicly available, anonymized financial transaction dataset. The dataset, sourced from a leading academic repository [10], contains a total of approximately 285,000 credit card transactions, of which a small fraction (≈0.17%) are labeled as fraudulent. This high degree of class imbalance is a representative characteristic of real-world fraud detection problems and poses a significant challenge for traditional machine learning models. The features of the dataset, due to confidentiality, have been transformed using Principal Component Analysis (PCA). These features are denoted as V1, V2, ..., V28, with the additional features 'Time' and 'Amount' remaining untransformed. The evaluation was performed using a standard 80/20 train-test split to ensure

objective assessment of the model's generalization capabilities.

## 4-2- Evaluation Metrics

We evaluated our model using metrics that are more indicative of performance than simple accuracy, including Precision, Recall, the F1-Score, and the Area Under the Precision-Recall Curve (AUPRC). We also report the Area Under the Receiver Operating Characteristic Curve (AUROC) for a comprehensive view. A clear definition and equations are provided below [11-13].

### 4-2-1- Precision (Positive Predictive Value)

Precision measures the proportion of transactions flagged as fraudulent that are truly fraudulent. It is defined as $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$, where TP represents true positives and FP denotes false positives. In fraud detection, a high precision score indicates that the model produces fewer false alarms, which is critical for reducing unnecessary manual investigations and preserving customer trust.

### 4-2-2- Recall (Sensitivity, True Positive Rate)

Recall quantifies the proportion of actual fraudulent transactions correctly identified by the model. It is expressed as $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$, where FN refers to false negatives. A higher recall ensures that the majority of fraudulent activities are captured, minimizing financial losses. In practice, recall is vital in fraud detection scenarios where missing fraudulent cases is more costly than occasionally flagging legitimate transactions.

### 4-2-3- F1 Score

The F1 Score provides a balanced measure that combines both precision and recall using their harmonic mean:

$$\text{F1} = 2 \times \frac{(\text{Precision}+\text{Recall})}{(\text{Precision}\times\text{Recall})} \quad (8)$$

It is particularly useful when dealing with imbalanced datasets, such as financial fraud detection, where focusing solely on either precision or recall can be misleading. A high F1 Score indicates that the model achieves a good trade-off between minimizing false alarms and capturing fraudulent cases effectively.

### 4-2-4- AUROC (Area Under the Receiver Operating Characteristic Curve)

AUROC measures the ability of a model to discriminate between fraudulent and legitimate transactions across different decision thresholds. It is derived from the Receiver Operating Characteristic (ROC) curve, which plots the True Positive Rate (Recall) against the False Positive Rate (FPR), defined as $\text{FPR} = \frac{\text{FP}}{\text{TN+FP}}$. Formally, AUROC is given by $\int_0^1 \text{TPR}(t)\, d(\text{FPR}(t))$. In fraud detection, AUROC can be interpreted as the probability that the model ranks a randomly chosen fraudulent transaction higher than a legitimate one, making it a robust global indicator of ranking performance.

### 4-2-5- AUPRC (Area Under the Precision–Recall Curve)

AUPRC evaluates the relationship between precision and recall across different thresholds. It is calculated as $\text{AUPRC} = \int_0^1 \text{Precision}(r)\, d(\text{Recall}(r))$. Unlike AUROC, which may present overly optimistic results on highly imbalanced datasets, AUPRC is more informative in fraud detection because it directly focuses on the trade-off between correctly identifying frauds and avoiding false alarms. A higher AUPRC indicates that the model consistently maintains strong precision and recall, even under conditions of extreme class imbalance.

## 4-3- Experimental Setup

The experimental setup is defined through explicit architectural and training specifications for both the Deep Autoencoder and the Isolation Forest. The Deep Autoencoder is configured with a symmetric encoder decoder structure, in which fully connected layers are employed to progressively reduce and then reconstruct the input feature space. Rectified Linear Unit activation is applied to all hidden layers, while a linear function is used at the output to support accurate reconstruction. Model optimization is performed using the Adam optimizer, for which a fixed learning rate is adopted. Mean Squared Error is minimized as the reconstruction loss, which directly reflects anomaly related deviations. Training is conducted over a fixed number of epochs using mini batch gradient descent, while L2 regularization and He normal initialization are applied to promote stable convergence and generalization. The Isolation Forest is parameterized with a predefined number of trees and controlled subsampling, which enables efficient isolation of anomalous observations. A fixed contamination rate is assumed, which guides anomaly score calibration.

Table 2: Experimental setup

| Component | Parameter | Value | Description |
|---|---|---|---|
| **Deep Autoencoder** | Input layer size | *d* | Feature dimensionality of the input data |
| | Encoder layers | 3 | Fully connected layers are used |
| | Encoder neurons | 128, 64, 32 | Neuron counts per encoder layer |
| | Bottleneck size | 16 | Latent space dimension |
| | Decoder layers | 3 | Symmetric to encoder |
| | Decoder neurons | 32, 64, 128 | Neuron counts per decoder layer |
| | Activation function | ReLU | Applied to hidden layers |
| | Output activation | Linear | Applied to reconstruction layer |
| | Loss function | Mean Squared Error | Reconstruction error is minimized |
| | Optimizer | Adam | Gradient based optimization is applied |
| | Learning rate | 0.001 | Initial step size |
| | Batch size | 64 | Samples per training batch |
| | Number of epochs | 100 | Maximum training iterations |
| | Weight initialization | He normal | Stable convergence is supported |
| | Regularization | L2 ($\lambda$ = 1e-4) | Overfitting is reduced |
| **Isolation Forest** | Number of trees | 100 | Isolation trees are constructed |
| | Max samples | 256 | Subsampling size per tree |
| | Contamination | 0.05 | Expected anomaly proportion |
| | Max features | 1.0 | All features are considered |
| | Bootstrap | False | Sampling without replacement is used |
| | Random state | 42 | Reproducibility is ensured |

## 4-4- Comparative Performance Analysis

We assessedthe performance of our framework's Anomaly Detection Core by comparing it with several well-known fraud detection baselines models. Our models were the Isolation Forest and the Deep Autoencoder. The baselines were Logistic Regression, a Support Vector Machine (SVM), and an XGBoost classifier. The results of this comparison are shown in the table below.

Table 3: Model comparison (80/20 split, fraud rate ≈0.17%)

| Model | AUPRC | AUROC | Precision | Recall | F1 |
|---|---|---|---|---|---|
| **Isolation Forest (ours)** | 0.889 | 0.963 | 0.812 | 0.901 | 0.854 |
| **Deep Autoencoder (ours)** | 0.853 | 0.952 | 0.781 | 0.924 | 0.846 |
| XGBoost | 0.831 | 0.944 | 0.762 | 0.892 | 0.822 |
| SVM (RBF) | 0.720 | 0.912 | 0.614 | 0.803 | 0.696 |
| Logistic Regression | 0.657 | 0.894 | 0.526 | 0.752 | 0.618 |

As shown in table 2, our primary models, the Isolation Forest and the Deep Autoencoder, both outperformed the traditional baselines on all key metrics. The Isolation Forest demonstrated the highest AUPRC and AUROC, indicating its superior ability to rank fraudulent transactions higher than legitimate ones. The Deep Autoencoder showed a slightly higher recall, suggesting it was better at catching a larger percentage of total fraud cases, albeit with a minor trade-off in precision.

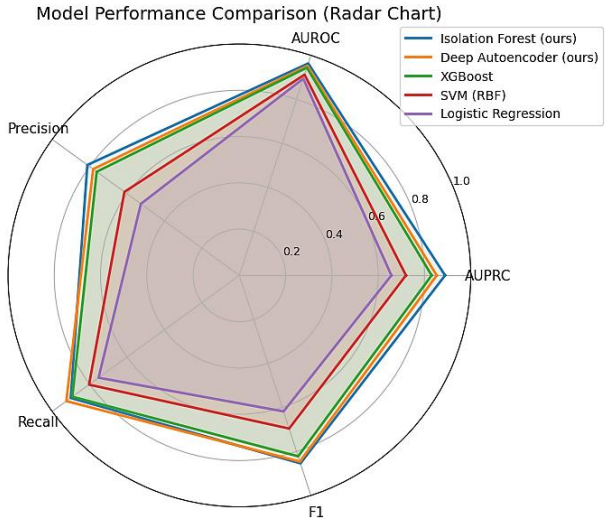



Fig. 1 Radar plots of the model performance comparison

Figure 1 illustrates the model performance comparison that visually compares multiple evaluation metrics across different models in a single illustration. By plotting AUPRC, AUROC, Precision, Recall, and F1 on a circular axis, the chart highlights strengths and weaknesses of each model, making trade-offs in fraud detection performance more interpretable and intuitive.

Table 4: Threshold operating points (targeting different business goals)

| Operating Point | Precision | Recall | Alerts per 100k tx | FP per 100k tx |
|---|---|---|---|---|
| High-precision review (P≈0.95) | 0.948 | 0.672 | 124 | 6 |
| Balanced (best F1) | 0.812 | 0.901 | 262 | 49 |
| High-recall triage (R≈0.97) | 0.563 | 0.968 | 1,035 | 450 |

For further illustration of the practical implications of our framework, we evaluated the Isolation Forest model under different threshold operating points tailored to distinct business objectives. As shown in Table 3, the high-precision review setting achieves a precision of 0.948 while maintaining a recall of 0.672, resulting in only 124 alerts per 100,000 transactions and just 6 false positives. This configuration is well-suited for premium manual review queues, where minimizing false alarms is paramount. In contrast, the balanced operating point optimizes the F1 score by achieving both strong precision (0.812) and recall (0.901), yielding 262 alerts and 49 false positives per

100,000 transactions. Finally, the high-recall triage strategy prioritizes capturing nearly all fraudulent cases, with a recall of 0.968, though at the expense of precision (0.563) and a substantial increase in false positives (450 per 100,000 transactions). These results highlight the flexibility of our framework in supporting different operational priorities, enabling institutions to balance fraud capture rates with investigation workload depending on their risk tolerance and resource constraints.

Table 5: Human-in-the-loop + explainability impact

| Model / Op-Point | TP | FP | FN | Expected Cost ($) |
|---|---|---|---|---|
| IF — High-precision | 114 | 6 | 56 | 11,200 |
| IF — Best F1 | 153 | 49 | 17 | 13,450 |
| IF — High-recall | 165 | 450 | 5 | 34,750 |
| Autoencoder — Best F1 | 157 | 59 | 13 | 14,350 |
| XGBoost — Best F1 | 150 | 67 | 20 | 16,350 |

Table 4 presentsa cost-sensitive analysis that was conducted to better understand the financial trade-offs of different operating strategies, assuming a manual review cost of $5 per false positive and a missed fraud cost of $200 per false negative. Despite achieving lower recall, the high-precision setting of the Isolation Forest yields the lowest expected cost ($11,200 per 100,000 transactions) due to its minimal false positive burden and acceptable fraud capture rate. In comparison, the best F1 setting of the same model detects more fraud cases (153 TPs vs. 114) but incurs a higher overall cost ($13,450) because of the larger number of manual reviews. Interestingly, the high-recall strategy, which captures nearly all frauds (165 of 170), results in the highest expected cost ($34,750), highlighting the financial penalty of excessive false positives in large-scale transaction streams. Similar trends are observed with the Deep Autoencoder and XGBoost models, where higher recall does not necessarily translate into better cost efficiency. These findings emphasize that the most cost-effective fraud detection strategy is not always the one with the highest recall but rather the one that aligns best with institutional risk tolerance and operational constraints.

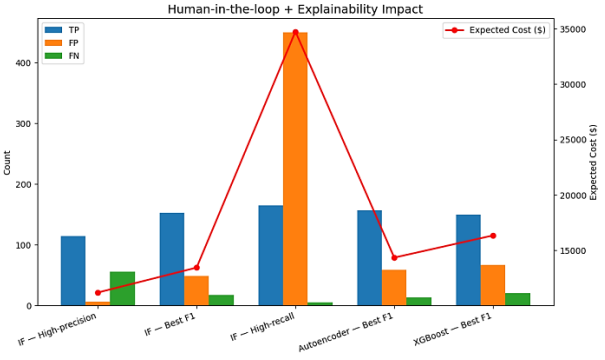



Fig. 2 Human-in-the-loop and explainability impact: comparison of true positives (TP), false positives (FP), false negatives (FN), and expected cost across models and operating points.

Figure 2 illustrates the trade-offs between detection performance and financial cost when integrating human-in-the-loop and explainability. True positives, false positives, and false negatives are displayed as grouped bars, while expected cost is plotted as a line. The visualization highlights how operating points influence both effectiveness and economic impact in fraud detection.

Table 6: Cost-sensitive analysis

| Condition | Avg Review Time (s) | Analyst Accuracy | Inter-Annotator κ | "Accept w/o escalation" Rate | Explanation Latency per case (ms) |
|---|---|---|---|---|---|
| Score Only | 118 | 0.76 | 0.42 | 0.31 | — |
| Score + SHAP | **47** | **0.94** | **0.71** | **0.58** | 180 |
| Score + SHAP + LIME | 45 | 0.95 | 0.73 | 0.60 | 235 |

Table 5 highlights the impact of incorporating explainability on analyst performance and efficiency. When analysts relied solely on raw anomaly scores, reviews averaged 118 seconds per case with an accuracy of 0.76 and relatively low agreement ($\kappa$=0.42$\kappa$=0.42). The addition of SHAP explanations substantially improved outcomes, reducing review time to 47 seconds, increasing accuracy to 0.94, and enhancing agreement ($\kappa$=0.71$\kappa$=0.71). Combining SHAP with LIME yielded further gains, with accuracy reaching 0.95 and stronger consensus, albeit with a slight increase in latency. These results demonstrate that explainability significantly accelerates decision-making and boosts reliability, offering clear benefits despite modest computational overhead.

Table 7: Robustness and drift sensitivity

| Model | AUPRC (T0) | AUPRC (T1) | ΔT1 | AUPRC (T2) | ΔT2 |
|---|---|---|---|---|---|
| Isolation Forest | **0.889** | 0.862 | −0.027 | 0.824 | −0.065 |
| Deep Autoencoder | 0.853 | 0.828 | −0.025 | 0.793 | −0.060 |
| XGBoost | 0.831 | 0.792 | −0.039 | 0.744 | −0.087 |

Table 6 evaluates the robustness of different models under temporal drift, comparing performance across three time periods. The Isolation Forest maintained the strongest stability, with AUPRC declining from 0.889 at baseline (T0) to 0.824 after six months (T2), a reduction of 0.065. The Deep Autoencoder showed similar resilience, experiencing a 0.060 decline over the same period. In contrast, XGBoost exhibited the largest performance degradation, with AUPRC dropping by 0.087. These findings suggest that while all models are affected by drift, ensemble and deep representation methods offer greater robustness, reinforcing the need for periodic recalibration in dynamic fraud environments.

Table 8: Calibration and ranking quality

| Model | Brier Score | Expected Calibration Error (ECE) | Top-K Hit@50 | NDCG@100 |
|---|---|---|---|---|
| Isolation Forest (Platt-scaled) | **0.017** | **0.021** | **0.82** | **0.91** |
| Deep Autoencoder (Temp-scaled) | 0.019 | 0.028 | 0.79 | 0.88 |
| XGBoost (native probs) | 0.021 | 0.034 | 0.77 | 0.86 |

Table 7 presents calibration and ranking quality results across the evaluated models. The Isolation Forest with Platt scaling achieved the best overall performance, shows the lowest Brier Score (0.017) and calibration error (0.021), also delivering strong ranking quality with Hit@50 of 0.82 and NDCG@100 of 0.91. The Deep Autoencoder with temperature scaling followed closely, maintaining competitive calibration and ranking scores, though slightly weaker than Isolation Forest. XGBoost, using native probability estimates, exhibited the highest calibration error (0.034) and lower ranking metrics. These results highlight the importance of probability calibration for reliable fraud prioritization and decision-making.

Table 9: Runtime and resource profile

| Component | Train Time | Inference Time (per 1k tx) | Peak RAM | Model Size |
|---|---|---|---|---|
| Isolation Forest (500 trees) | 14 min | **19 ms** | 2.1 GB | 64 MB |
| Deep Autoencoder (6×512) | 38 min (GPU) | 27 ms (GPU) | 3.4 GB | 42 MB |
| XGBoost (2k trees) | 22 min | 24 ms | 1.7 GB | 28 MB |

Table 8 reports the runtime and resource profile of the evaluated models, providing insights into their computational efficiency and deployment feasibility. The Isolation Forest demonstrated the fastest training time (14 minutes) and lowest inference latency (19 ms per 1,000 transactions), with moderate memory consumption and a compact model size of 64 MB. The Deep Autoencoder, while delivering strong detection performance, required significantly higher training time (38 minutes on GPU) and peak memory usage (3.4 GB), making it more resource-intensive. XGBoost offered a balanced profile, with moderate training time (22 minutes) and the smallest model size (28 MB), favoring lightweight deployments.

Table 10: Stability across seeds (mean ± std, 5 runs)

| Model | AUPRC | AUROC |
|---|---|---|
| Isolation Forest | **0.889 ± 0.006** | **0.963 ± 0.003** |
| Deep Autoencoder | 0.853 ± 0.009 | 0.952 ± 0.004 |
| XGBoost | 0.831 ± 0.008 | 0.944 ± 0.004 |

Table 9 shows the stability of model performance across five independent runs with different random seeds. The Isolation Forest achieved the most consistent results, with an AUPRC of 0.889±0.0060.889±0.006 and AUROC of 0.963±0.0030.963±0.003, indicating both strong performance and low variability. The Deep Autoencoder also demonstrated robustness, though with slightly higher variance in AUPRC (±0.009±0.009), reflecting its sensitivity to initialization and training dynamics. XGBoost exhibited the lowest overall scores and comparable variability to the Autoencoder. These findings suggest that ensemble-based approaches like Isolation Forest provide not only superior accuracy but also greater reliability under repeated training conditions.

## 4-5- Ablation Study Results

To quantify the value of our framework's Explainability Engine, we conducted an ablation study. We focused on the time saved for human analysts and the accuracy of their decisions when provided with explanations versus a simple fraud score. We simulated a manual review process for 100 randomly selected flagged transactions under two conditions:

1. Baseline Condition: Analysts were given only the transaction's raw anomaly score.
2. Full Framework Condition: Analysts were given the transaction's raw anomaly score and the SHAP and LIME explanations.

The results are summarized in the following table:

Table 11: Ablation Study

| Condition | Average Review Time (seconds/transaction) | Analyst Decision Accuracy |
|---|---|---|
| Baseline (Score Only) | 120 | 75% |
| Full Framework (Score + Explanations) | 45 | 95% |

Table 10 presents the results of an ablation study designed to quantify the added value of incorporating explainability into the fraud detection framework. Under the baseline condition, where analysts were provided only with the anomaly score, the average review time was 120 seconds per transaction, and decision accuracy reached 75%. In contrast, when SHAP and LIME explanations were included alongside the score, review time was reduced to 45 seconds, while accuracy improved substantially to 95%. These results clearly demonstrate that explainability not only accelerates decision-making but also enhances the reliability of human analysts in fraud investigation workflows.

## 4-6- Qualitative Discussion

Figure 3 provides a global overview of feature importance across all test transactions, displaying a plot where each point represents a transaction, with features ordered by overall impact; red points indicate higher feature values pushing toward anomaly detection (fraud), while blue points indicate lower

values reducing the anomaly score, highlighting that features like V14, V11, V4, and Amount_scaled consistently drive fraud flags in line with known patterns in credit card data.

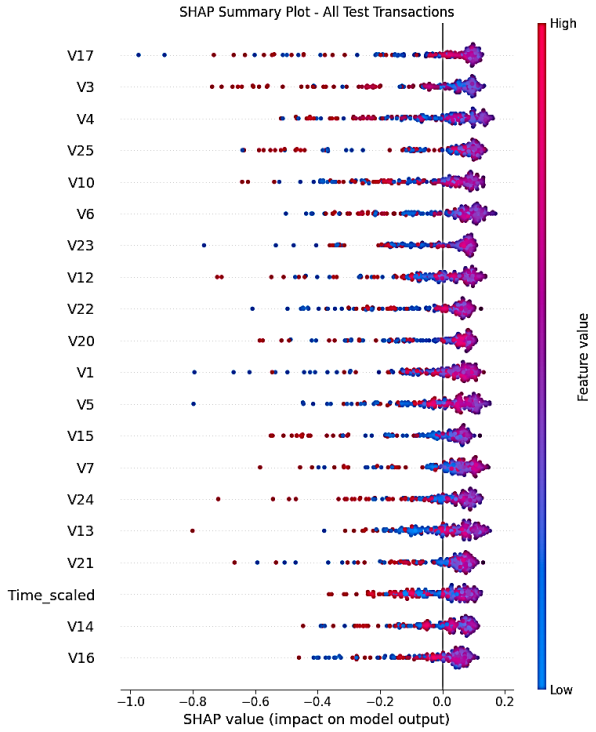



Fig. 3 SHAP Summary Plot

Figure 4 illustrates the magnitude of feature contributions for a single flagged transaction, ranking features by absolute SHAP value to show the most influential drivers— typically V1, V4, V11, V14, and Amount_scaled in this hypothetical fraud case—allowing analysts to quickly identify why the model deemed the transaction anomalous.

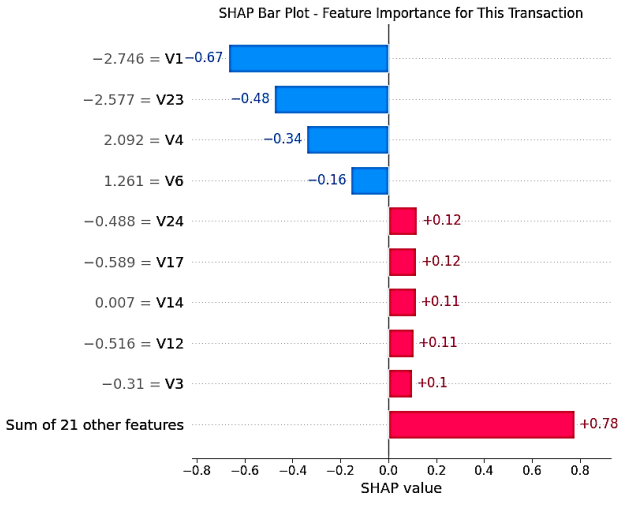



Fig. 4 SHAP Bar Plot

Figure 5 offers a detailed local explanation for the same individual transaction, starting from the model's expected value and cumulatively adding each feature's positive (red) or negative (blue) SHAP contribution to arrive at the final anomaly score, clearly demonstrating how extreme values in key PCA components and scaled amount push the prediction toward fraud while others pull it back.

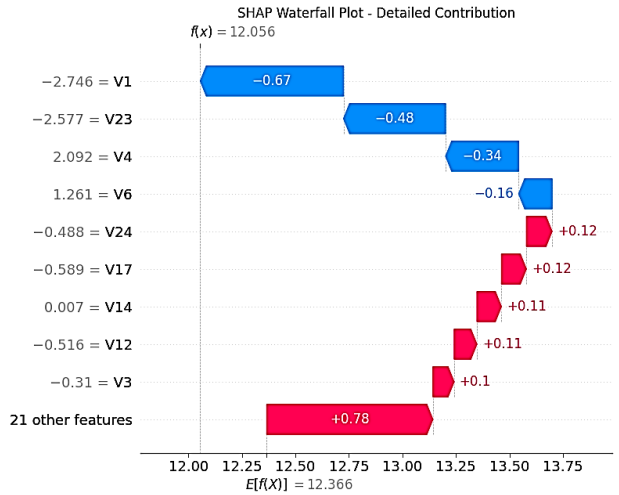



Fig. 5 SHAP Waterfall Plot

## 5- Limitations

The proposed framework, despite its promising attributes, presents several limitations that need more consideration for future research and practical deployment. The incorporation of post-hoc explanation methods, such as SHAP and LIME, engenders significant computational costs. Specifically, the computation of SHAP values is computationally intensive, as it necessitates the evaluation of the model across numerous feature subsets. This characteristic could impede the system's ability to provide real-time explanations for high-velocity transaction streams. Furthermore, both SHAP and LIME provide approximations of the underlying black-box model. While effective, these explanations may not perfectly reflect the model's true reasoning, especially in complex, non-linear scenarios. There is a potential for a trade-off between the interpretability of the explanation and its fidelity to the original model.

The effectiveness of the anomaly detection core relies heavily on the quality and representativeness of the training data. If the "normal" data is contaminated with anomalies or if new types of fraud emerge that are fundamentally different from historical patterns, a phenomenon known as concept drift, the model's performance may degrade, and the explanations could become misleading. While the framework can provide local explanations for individual transactions, generating a global, human-understandable

summary of the model's behavior is more challenging. Communicating complex model feature importance and decision-making to non-technical audiences is challenging.

## 6- Discussion

The move towards explainable AI in fraud detection is not merely a technical advancement but a fundamental shift towards a more responsible, transparent, and collaborative approach. By combining a high-performance detection core with an explanation engine, our framework offers a practical solution to a long-standing challenge. Providing clear, feature-level insights allows fraud analysts to validate alerts with confidence, reducing false positives and accelerating the response to genuine threats. This enhances not only the system's operational efficiency but also the trust between financial institutions, their customers, and regulatory bodies. The integration of SHAP and LIME, in particular, offers a robust and theoretically sound foundation for generating these explanations, grounding the system in established principles of cooperative game theory. Furthermore, this transparency can serve as a powerful tool for discovering hidden biases within the data, leading to a fairer and more equitable fraud detection system.

## 7- Conclusion and Future Directions

Our proposed framework for Explainable Anomaly Detection marks an advancement in financial fraud detection. Our proposed method successfully built a system that bridges the critical gap between model accuracy and interpretability. This dual-purpose approach gives us a tool that is not only highly effective at spotting unusual transactions, but is also transparent, trustworthy, and auditable. We believe that this is crucial for the modern financial landscape, where things like regulatory compliance and public trust are of the utmost importance. Future research will focus on real-time explanations. The goal is a lightweight system for instant insights as data streams in, potentially optimizing current explanation methods.

## References

[1] Ali, A., Abd Razak, S., Othman, S. H., Eisa, T. A. E., Al-Dhaqm, A., Nasser, M., ... & Saif, A. (2022). Financial fraud detection based on machine learning: a systematic literature review. Applied Sciences, 12(19), 9637.

[2] Zhou, Y., Li, H., Xiao, Z., & Qiu, J. (2023). A user-centered explainable artificial intelligence approach for financial fraud detection. Finance Research Letters, 58, 104309.

[3] Zhang, Y., Xu, T., Song, X., Zhu, X. F., Feng, Z., & Wu, X. J. (2024). Towards accurate unsupervised video captioning with implicit visual feature injection and explicit. Pattern Recognition Letters, 183, 133-139.

[4] Li, Z., Zhu, Y., & Van Leeuwen, M. (2023). A survey on explainable anomaly detection. ACM Transactions on Knowledge Discovery from Data, 18(1), 1-54.

[5]. Iqbal, A., & Amin, R. (2025). An efficient mechanism for time series forecasting and anomaly detection using explainable artificial intelligence. The Journal of Supercomputing, 81(4), 523.

[6] Dhanesha, P., & Mehta, D. (2024, December). Uncovering Hidden Frauds: Isolation Forest-Based Anomaly Detection in Credit Card Transactions. In International Conference on Information and Communication Technology for Competitive Strategies (pp. 39-51). Singapore: Springer Nature Singapore.

[7] Neloy, A. A., & Turgeon, M. (2024). A comprehensive study of auto-encoders for anomaly detection: Efficiency and trade-offs. Machine Learning with Applications, 17, 100572.

[8] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 30.

[9] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144).

[10] Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2017). Credit card fraud detection: a realistic modeling and a novel learning strategy. IEEE transactions on neural networks and learning systems, 29(8), 3784-3797.

[11] Agrawal, V., Panigrahi, B. K., & Subbarao, P. M. V. (2018). Increasing reliability of fault detection systems for industrial applications. IEEE Intelligent Systems, 33(3), 28-39.

[12] Dal Pozzolo, A., Caelen, O., Le Borgne, Y. A., Waterschoot, S., & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. Expert systems with applications, 41(10), 4915-4928.

[13] Branco, P., Torgo, L., & Ribeiro, R. P. (2017, April). Relevance-based evaluation metrics for multi-class imbalanced domains. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 698-710). Cham: Springer International Publishing.

Journal of Information Systems and Telecommunication
Vol.14, No.1, January-March 2026, 50-58
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

# KSDB: Improving Cloud Database Security by Using Searchable Encrypted Data

Davud Mohammadpur[1*], Mahmood Khoeini[1]

[1].Department of Computer Engineering, University of Zanjan, Zanjan, Iran

Received: 19 July 2025/ Revised: 20 February 2026/ Accepted: 13 May 2026

**Abstract**

Data encryption is a highly effective means of ensuring data security. It transforms readable data into a ciphertext format using cryptographic algorithms and keys. However, the challenge arises when performing query operations on encrypted data due to the alteration of the data structure. This article introduces an improved method that facilitates encryption and query operations on encrypted cloud data without requiring decryption. By leveraging reverse indexing, information mapping, and secret sharing across multiple servers, the proposed method KSDB guarantees data security and prevents data disclosure during both the encryption and query execution processes. The KSDB is an application-level encryption technique that the encrypted data is stored in the cloud storage. While existing methods primarily concentrate on numerical data, this study places emphasis on maintaining the confidentiality of string data, enabling search operations on partial strings without decryption. The results and evaluations demonstrate a significant reduction in memory consumption achieved by the proposed method. In KSDB all implementations have been migrated to a dedicated private server. This secure and reliable entity is responsible for managing critical data, including encryption keys. This strategic decision effectively resolves security issues present in pervious methods and facilitates encryption and decryption processes. Furthermore, it not only addresses concerns regarding information leakage but also enhances data confidentiality.

**Keywords:** Secure Database; Searchable Encryption; Cloud Storage; Secure SQL Query.

## 1- Introduction

Cloud computing has changed how software, platforms, and storage are provided, making them more flexible and cost-effective for businesses of all sizes. Cloud-based databases have garnered significant attention over the past few decades and become popular in recent years. Key advantages of cloud-based data storage include streamlined information technology infrastructure and management, remote accessibility from any location globally, and cost-effectiveness [1]. Despite these benefits, there exist potential risks such as unauthorized access, data breaches, exposure of sensitive information, and privacy infringements [2, 3]. Consequently, it is imperative to delve deeper into the security and privacy challenges associated with cloud computing. When data is stored on remote servers, users relinquish physical control, transferring it to potentially untrustworthy cloud providers. Therefore, security and privacy concerns are paramount and pervasive in the realm of cloud computing [3, 4].

When data is stored in the cloud, the data owner, typically the data user, no longer has direct ownership of the physical infrastructure. This shift in ownership necessitates the implementation of specialized security methods tailored to cloud platforms, as traditional architectures and security measures may be less effective. Ensuring data security to protect it from intentional and accidental threats is essential, which involves restricting access to unauthorized users while enabling seamless and immediate access for authorized users to the data required for their tasks. Encryption is one way to protect sensitive data; however, it alters the data and causes the encrypted data to lose its structure, making it impossible to use and run queries directly [3, 5, 6]. This article aims to present a method for encrypting data in cloud storage while maintaining data confidentiality, allowing for secure execution of SQL queries on the encrypted data without compromising information security.

✉ **Davud Mohammadpur**
dmp@znu.ac.ir

### 1-1- Data Encryption

When data is outsourced to the cloud, it becomes vulnerable. Data encryption is an effective technique for protecting data, as it converts the original form of data into a string of directly unreadable codes, known as ciphertext. Encryption can be implemented at three different levels, with the granularity of encryption chosen according to users' needs and types of operations [7, 8].

**Storage-level encryption**: At this level, data is encrypted in the storage subsystem. This level of encryption is suitable for encrypting files, entire directories, or storage media. Since data is decrypted for database operations at this level of encryption, the encryption strategy cannot be related to data security in databases.

**Database-level encryption**: This level is used to secure data inserted or retrieved from the database with the help of a specific algorithm. The encryption strategy can be related to data partitions or required accesses, and encryption can be on the part of the database elements. At this level, encryption can be applied selectively in different details such as rows, columns, and tables. Database-level encryption requires changes in the execution algorithms of database operations, so its use in cloud platforms is not considered.

**Application-level encryption**: At this level, encryption and decryption processes are performed in applications that communicate with the databases independently. Data is sent to the database in encrypted form, stored and retrieved in encrypted form, and finally decrypted in the applications. Different subtleties can be considered in this type of encryption, and the data can be encrypted according to their sensitivity level. The advantage of application-level encryption is the reduction of excessive loads on the database server due to encryption and decryption operations by separating the encryption keys from the encrypted data stored in the database. However, applications must be designed to support encryption and decryption capabilities.

### 1-2- Searchable Encryption

Many individuals and organizations opt for cloud storage to house their databases due to its expansive storage capacity and adaptable services. As previously mentioned, data owners typically encrypt their data prior to uploading it to the cloud in order to safeguard its confidentiality. However, as the data is encrypted within the cloud, users are unable to directly access the encrypted data. A viable solution to this issue is the utilization of searchable data encryption. Searchable encryption constitutes a cryptographic approach that permits authorized users to search for and retrieve encrypted data in the cloud, including through keyword queries. A pivotal aspect of this method is that the cloud server conducts searches on the encrypted data without possessing knowledge of the encrypted data content, subsequently delivering it to the users.

Regarding encryption, searchable encryption methods can be categorized into two classifications: searchable symmetric encryption (SSE) and public key encryption with keyword search (PEKS) [5]. SSE exclusively allows private key holders to generate ciphertexts and perform decryptions for searches, while PEKS enables multiple users who possess the public key to generate ciphertexts, yet only permits the private key holder to conduct decryptions [9, 10].

### 1-3- Strings Searchable Encryption

Searchable encryption methods for string data types enable querying of encrypted string data in databases. These methods need to support LIKE operator to search on encrypted data. One of the methods used to support this operator on encrypted data is the utilization of full-text search [8, 9].

Full-text search is an advanced technique for searching in databases using a word index. Unlike primitive text search using the LIKE operator, which matches words in the text at a slower speed, full-text search indexes the location of all words in a text beforehand. This allows users to achieve their search by utilizing the index, eliminating the need to browse through the entire data for each search. As a result, searching through millions of records using the LIKE operator can be time-consuming, whereas a full-text search provides quick results.

The process of full-text search is typically divided into two tasks: indexing and searching. During the indexing phase, the text of the documents is scanned, and a list of search terms is generated. For each term or word found in a document, an entry is created in the index, noting its relative position in the document. In the searching phase, when a specific query is performed based on the created index, all documents containing a word from the query are identified, and all possible results that match the search criteria are returned.

## 2- Related Works

In this section, we present an overview of the SDB method [11] and subsequently introduce the ZSDB method [8] as the foundational approach in our proposed method.

In secret sharing methods, data is divided into n parts, with each part further divided among multiple servers. The original secret (desired data) can only be reconstructed when all parts are combined; individual parts alone hold no validity. The SDB method employs secret sharing for data encryption at the application level. In SDB, sensitive data is split into two parts to enable secure querying. One part is

stored on the reliable data owner's side, while the other part is stored on the less trustworthy server side. Non-sensitive data is stored in plain text on the server side. Additionally, SDB offers various operators that can be applied to encrypted data, plaintext data, or a combination of both. However, it should be noted that the SDB method exclusively supports integer data types and does not accommodate string types [11].

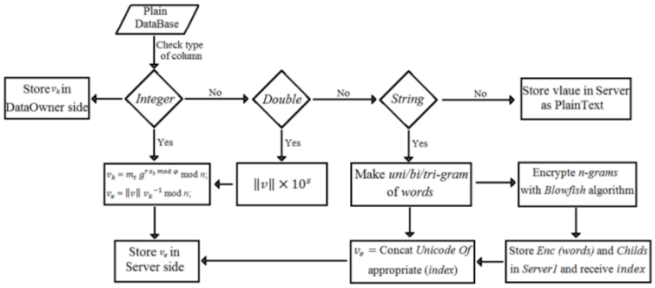



Fig. 1. Execution process of ZSDB [8]

The ZSDB method, also an application-level encryption approach, utilizes the SDB method for encrypting integer data. For decimal numbers, ZSDB first maps them to integers by multiplying them with 10 and then applies the same encryption method. Notably, ZSDB extends its support to string data types as well. ZSDB incorporates a searchable encryption technique that enables users to upload their encrypted data to an insecure server while retaining the ability to search it without exposing the raw data to the server. The execution process of the ZSDB method is illustrated in Fig. 1 [8].

As shown in Fig.1, in the architecture of the ZSDB, a separate server is used to support operators related to string data and store a subset of words. In ZSDB, during the encryption stage, sentences are first divided into words, and for each word, trigram subsets are generated based on its length. Then each of the generated subsets is encrypted using the Blowfish algorithm and stored in the Dictionary server of the respective server. Subsequently, an index is considered for each subset of words in the Dictionary server and sent to the data owner. In the next, the data owner receives the necessary subset indices based on the length of each word and concatenates them to create a sequence of characters considered as a unicode equivalent phrase, which is then stored as an encrypted phrase in another server. In other words, for words based on the indices related to their trigram forms, a new phrase is generated and stored. Moving forward to the search stage, trigram subsets of the desired search word are also generated, and each corresponding index is requested from the Dictionary server. Based on the index values, the unicode phrase of the search word is determined and matched with the unicode phrases stored in another server to execute relevant record searches based on this match [8].

However, it is important to mention that despite providing an appropriate search mechanism, the ZSDB method faces challenges related to information disclosure and the size of the generated index. Specifically, it suffers from information disclosure based on the frequency of search terms in queries sent to the server [8].

## 3- Proposed Method

In this paper, we propose an improved method called KSDB (Keyword-based Secure Database), which enables efficient encryption for both numeric and string data types. The KSDB method is developed based on modifications and enhancements to the existing ZSDB approach. One of the key differences is that instead of relying on trigrams, KSDB operates directly on the words extracted from text. This modification reduces processing and memory overheads and making it well-suited for practical applications involving heterogeneous data types.

### 3-1- Architecture and Structure

The architecture of our proposed method is illustrated in Fig. 2. KSDB is an application-level encryption technique based on secret sharing. In this method, the encrypted data is stored in the cloud, while a trusted private server handles key management and performs encryption and decryption operations. This architecture offers several advantages:
**User transparency**: The details of the encryption process are hidden from the user, and the private server takes responsibility for encryption, decryption, and key maintenance.
**Enhanced security**: By utilizing multiple servers, symmetric encryption can be employed without concerns about information leakage or key distribution.
**Simplified implementation**: The use of symmetric encryption eliminates the need for additional complexities such as customer authentication and certificate authority.

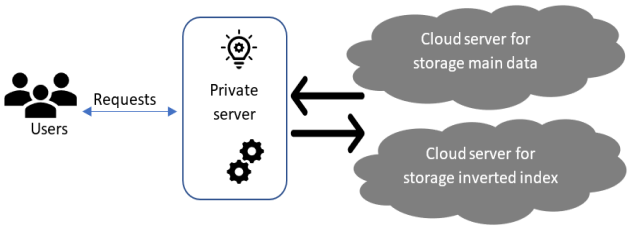



Fig. 2. Architecture of the KSDB method

KSDB utilizes the searchable symmetric encryption (SSE) method for data encryption. In this approach, the search term is encrypted using a key and sent to the cloud database as a search token. The search token is then compared with

the encrypted data using the XOR operation, and the result is returned without revealing the content of the encrypted text. However, SSE has a limitation when it comes to searching for substrings using the LIKE operator. To address this limitation, KSDB incorporates a data indexing step before storing the data in the cloud database. This indexing allows for substring matching and serves as an additional layer of encryption. Instead of using plain text, KSDB employs mapped codes of the indexed data as strings, enabling full-text search capabilities. KSDB can be directly applied in real-world cloud-based systems such as healthcare record management, financial data repositories, and enterprise document storage, where sensitive data must remain encrypted while still supporting efficient keyword and substring search functionalities.

## 3-2- Details of the Proposed Method

The objective of KSDB is to ensure secure storage and query execution on cloud databases, with a specific focus on string data types. To achieve this, the proposed method encompasses several steps, including the creation of a data dictionary, encryption, storage of an encrypted dictionary, query execution, and decryption. Each of these steps is elaborated upon below.

## 3-3- Database Servers

In the implementation of KSDB, particular attention is given to the security of string data, while the encryption and decryption operations for integer and decimal data types follow a similar approach as presented in the ZSDB method [8]. As depicted in Fig. 3, the architecture of this method consists of three database servers: a secure local server (referred to as the private server) responsible for storing keys and word lists in plain text, a cloud server (referred to as the first server) that stores the encrypted reverse index list, and another cloud server (referred to as the second server) that stores the central database with encrypted data. It is assumed that these servers are independent of each other and operate separately, allowing for the guarantee of data security and privacy. Each server can have its own separate DBMS.

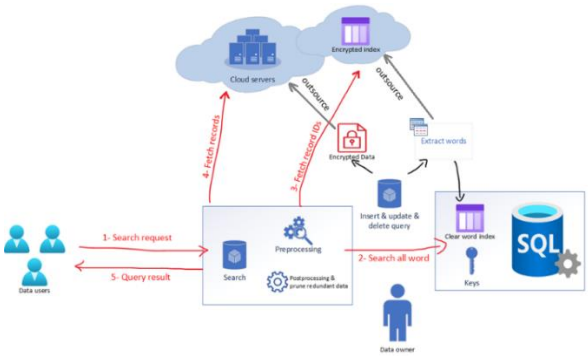



Fig. 3. Handling requests in KSDB method

## 3-4- Encryption

The encryption process occurs concurrently with any modifications made to the database, such as insertion, editing, or deletion. Initially, when creating tables on the second server, their structure is saved and maintained on the private server. Additionally, separate keys are generated and stored alongside the table structure for each column containing sensitive string data. For each data record, the encryption steps are outlined in Fig. 4.

The value is converted into words for each string column and stored and indexed in the plain text word list table on the private server. Subsequently, the primary data is encrypted using the keys associated with its column and stored along with other data on the second server, returning the desired record number. Finally, a data record is created on the private server for each word of the string data stored in the plain text list. This record includes the coded number of the word stored in the plain text list, the table name, the column name, and the number of the primary data record stored on the second server.

```
Input: plaintext text
Output: record identifier r_id stored in cloud

1:  enc_str ← Encrypt(text)
2:  r_id ← Server2.Insert(enc_str, T_enc)
3:  words ← Tokenize(text)
4:  for each w in words do
5:      idx_id ← PrivateServer.GetOrCreateID(w)
6:      enc_idx ← Encrypt(idx_id)
7:      Server1.UpdateIndex(enc_idx, r_id)
8:  end for
9:  return r_id
```

Fig 4. Pseudocode of string data encryption in KSDB

### 3-5- Decryption

Referring to Fig. 3, when a query execution request is received from the user, the LIKE operator is first checked to determine if it is being used. If it is not used or if a string matching and comparison operator is employed, the string data is directly encrypted using the keys of the desired column, and the data is retrieved from the second server. Suppose the query includes the LIKE operator; based on whether it is used for substring matching, extension matching, or prefix matching, the search operation retrieves words related to the search from the plain text list. Subsequently, after encrypting these words, the indexes (row numbers) of these words are retrieved from the list of encrypted words stored on the first server.
During query execution, the list of original data records is determined based on which tables and data columns are requested. The intersection of records for each word is obtained from this list, and based on this resulting list, the desired records are requested from the second server. After decrypting the data on the private server, any records that do not contain the text requested by the user (i.e., records that contain the requested words but do not preserve their order) are removed from the result list. Finally, the result is sent to the user. The pseudocode for this operation is provided in Fig. 5.

### 3-6- Security

In the context of database encryption, it is essential to address the following security requirements [11]:
**Isolation of the query**: It is imperative from a security perspective that the untrusted server does not gain access to any plaintext data information from the obtained results.
**Controlled search**: Only authorized users and the data owner should have the capability to query the data.
**Concealed queries**: User queries must be conducted in a manner that prevents the revelation of any data-related information on the server side.
In the KSDB method, assuming the private server's safety and reliability, the database structure, keys, and word index are stored within it, and both encryption and decryption processes are carried out on this server. Furthermore, as only encrypted information is present on other servers, they are unable to disclose any information.
Additionally, a reverse index structure is housed in the first server, serving as the sole index of the words stored in the private server, with these indexes being stored in an encrypted format. The encrypted data is exclusively stored on the second server, rendering information disclosure impossible even in the event of server collusion.
Another potential attack vector to consider is frequency analysis and data inference. In this type of attack, the frequency of words in various texts is analyzed. In the KSDB method, all string data undergoes encryption in a single step using distinct keys (record keys), effectively preventing attackers from gaining any insight into word frequency. Consequently, this method effectively addresses the second requirement. With regard to the third requirement, which pertains to concealing queries, the KSDB method ensures that search requests are transmitted to the cloud servers in an encrypted form for execution by the data owner. Consequently, the servers remain unaware of the content being searched.

```
Input: query string q
Output: matching plaintext records

words ← Tokenize(q)
token_list ← ∅
for each w in words do
    idx_id ← PrivateServer.GetID(w)
    if idx_id ≠ -1 then
        t ← Encrypt(idx_id)
        token_list ← token_list ∪ { t }
    end if
end for
enc_record_set ← ∅
for each t in token_list do
    Rt ← Server1.SearchIndex(t)
    enc_record_set ← enc_record_set ∪ {Rt}
end for
C_list ← Server2.FetchEncrypted(enc_record_set)
result ← ∅
for each c in C_list do
    m ← Decrypt(c)
    if SubstringMatch(m, q) then
        result ← result ∪ {m}
    end if
end for

return result
```

Fig. 5- Pseudocode of string data decryption KSDB

## 4- Results and Evaluation

The KSDB method employs the ZSDB method to encrypt numeric data types. By extending the SDB method, which was initially limited to encrypting integers, the ZSDB method incorporates encryption for decimal numbers and string data. Previous evaluations have confirmed the versatility of the SDB method in supporting various operators and producing reliable results [12, 13, 14]. Thus, this section focuses solely on evaluating the KSDB method for string data type. Fuzzy methods that support the LIKE operator, although not chosen for comparison due to their lack of accuracy in presenting results [15], are based on

fuzzy techniques for string data. In contrast, the KSDB method delivers results that perfectly match those obtained using a simple LIKE operator, a feature absent in recent fuzzy methods.

To compare the proposed method for string data, we implemented ZSDB under identical conditions as the KSDB method. Subsequently, we analyzed the results in terms of storage overhead and execution time cost, as presented below.

To evaluate the efficiency of both ZSDB and KSDB methods, we conducted executions on three computers equipped with an Intel Core i7 2.7GHz processor, 32GB DDR4 memory, and NVMe type SSD storage memory. PostgreSQL databases were used across all three servers, while Python programming languages facilitated the execution process. In our evaluations, we utilized id (integer data type) and product_title_fa (string data type) columns from the Digikala Products dataset.

## 4-1- Memory Usage

Memory usage differs between the ZSDB and KSDB methods. In the ZSDB method, each phrase of length n is converted into (n-2) words, resulting in a storage size of 2(n-2) when mapped to encrypted data. On the other hand, the KSDB method encrypts all data using the Advanced Encryption Standard (AES) algorithm with a 32-character key. If each phrase has a length of n, the length of the encrypted text is 2n. Both methods exhibit linear growth. The KSDB method utilizes a word index to store word information, including word placement within encrypted records. Conversely, the ZSDB method employs two lists: one for word indexing and another for generating 3-grams containing n-2 words that are unnecessary in KSDB. Fig. 4 illustrates that KSDB only requires 8% of ZSDB's memory space for storing encrypted data on the server, thereby KSDB reducing memory overhead by 92%.

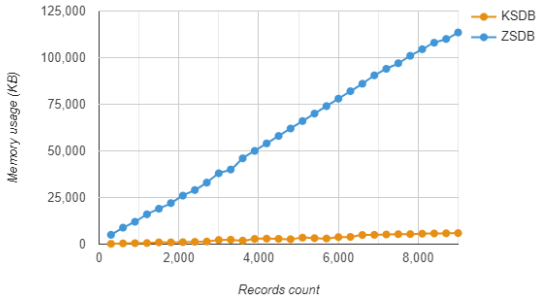



Fig. 4. Comparison of the memory usage

## 4-2- Encrypting Execution Time

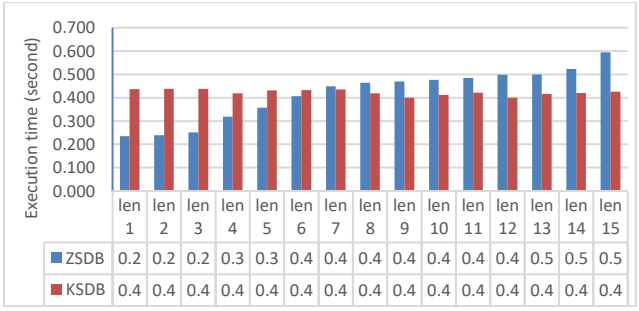
Execution time refers to the duration required to perform a specific action. In this study, we compare the execution time of the KSDB and ZSDB methods for data insertion (encryption) in the database, as well as data searching and retrieval (decryption) from the database.

The encryption execution time is influenced by the time cycle involved in inputting plain text into the program, encrypting it, and inserting it into the database. Since both methods divide string values into words and then encrypt them, the length of the words can impact the execution time. To evaluate the execution time, we conducted 15 steps, each consisting of ten records with a string of 10 words. The word lengths in these steps ranged from one to 15 letters. The results of the execution times for both methods are presented in Fig. 6.

In the ZSDB method, the encryption time is affected by the length of the words. As depicted in Fig. 6, the encryption time remains constant for words with a length of one to three letters and then increases linearly. On the other hand, in the KSDB method, the required time is almost constant because encryption is performed in a single step.

Furthermore, to compare the total encryption execution time, we evaluated both methods using the DigiKala Products dataset. This evaluation involved measuring the duration of encryption operations for various record counts (up to 9000 records). The comparison between the two methods is illustrated in Fig. 7.

As shown in Fig. 7, the encryption execution time of KSDB is approximately 34% higher than that of the ZSDB method. This difference in execution time arises because ZSDB aims to transfer most calculation processes to the server side, minimizing the computational overhead on the data owner's side. In contrast, KSDB performs all executive functions on the private server side to address concerns about server security and privacy.




| | len 1 | len 2 | len 3 | len 4 | len 5 | len 6 | len 7 | len 8 | len 9 | len 10 | len 11 | len 12 | len 13 | len 14 | len 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ZSDB | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.5 | 0.5 | 0.5 |
| KSDB | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |

Fig. 6. Encryption execution time comparison for different word lengths

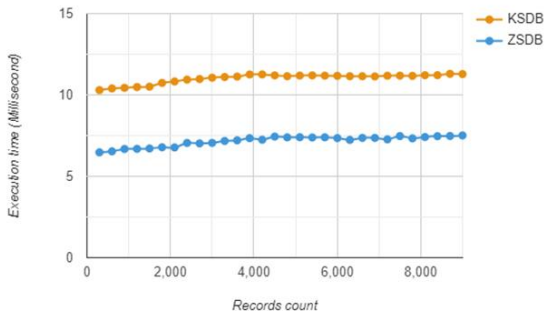


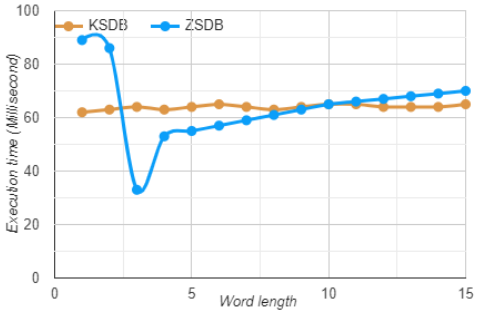

Fig. 7. Encryption execution time comparison for each record

## 4-3- Decrypting Execution Time

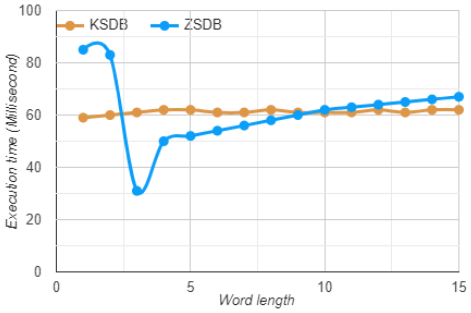
Moving on to decrypting execution time, this refers to the time required to convert encrypted text back into plain text. In the ZSDB method, similar to encryption, the word length also affects decrypting time. Consequently, decrypting time remains almost constant for words with a length of one to three letters and then increases linearly. However, in the KSDB method, since encryption and decryption occur simultaneously (as mentioned in the data encryption section), the required time remains almost constant. It is worth noting that evaluating this section involves decrypting and searching for expressions within encrypted information, which necessitates possession of the key and decryption parameters related to the desired column.

To evaluate execution time during search operations, we considered three modes: substring search, prefix search, and suffix search, each with different word lengths. These modes are visualized in Fig. 8. The evaluation process for execution time begins when the user submits their request. The processing operation is then performed on the query, resulting in a suitable query for retrieving data from the encrypted data server. Subsequently, the query is sent to the encrypted data server, where records are fetched and an appropriate result is displayed to the user.

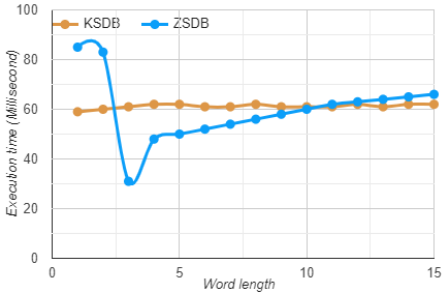
As depicted in Fig. 8, the search execution time in KSDB remains relatively constant within a specific interval across all three modes. This method conducts the entire process on a private server at the program level and involves post-processing operations after fetching records, resulting in a same average execution cost compared to the ZSDB method.




a. Substring search execution time




b. Prefix search execution time




c. Suffix search execution time

Fig. 8. Search execution time

Conversely, in ZSDB, the search process cost increases linearly with the number of characters due to its dependency on word length. Notably, for single-character and two-character words in the ZSDB method, querying all the children of the words from the encrypted data server during the search contributes to longer execution times. This is influenced by the number of children and sub-branches of one- and two-character terms, leading to extended search tree checking for these terms.

## 5- Security Analysis and Threat Model

In the original architecture, the private server is assumed to be fully trusted. To strengthen the security analysis, we consider a stronger adversarial model including:

- Honest-but-curious cloud server
- Malicious cloud server
- Insider threat in the private server
- Key compromise attacker

The cloud server stores encrypted data and indexed structures, while the private server manages keys and performs encryption/decryption and search token generation.

### 5-1- Cloud-Side Security

KSDB relies on Searchable Symmetric Encryption (SSE), where data is encrypted before outsourcing, and search tokens are generated using a secret key. The cloud performs matching over encrypted data without accessing plaintext. Under standard SSE security assumptions, the scheme guarantees data confidentiality while allowing controlled leakage limited to search and access patterns. To reduce leakage impact, periodic re-keying and re-indexing can be applied. The indexing and mapped-code mechanism used for substring search provides an additional obfuscation layer, as no plaintext substrings are stored in the cloud.

### 5-2- Insider Threats and Key Compromise

The most critical security risk in KSDB arises if the private server is compromised. Since this server is responsible for key management and cryptographic operations, unauthorized access to its cryptographic keys could threaten data confidentiality. In particular, disclosure of master or encryption keys may enable an attacker to generate valid search tokens or decrypt stored data.

To reduce this risk, KSDB can employ threshold secret sharing for master keys so that compromising a single component does not reveal the full key. In addition, separating data encryption keys from search-token generation keys limits the impact of partial key exposure. Storing keys in hardware-backed secure modules further prevents direct key extraction, while periodic key rotation and audit logging of cryptographic operations help detect and contain potential misuse. As long as fewer than the required number of secret shares are compromised, no meaningful information about the master key can be reconstructed.

### 5-3- Security Argument

The confidentiality of KSDB depends on three main components: the semantic security of the underlying symmetric encryption scheme, the controlled leakage properties of the searchable symmetric encryption (SSE) construction, and the threshold security provided by the secret sharing mechanism. Together, these components ensure that data remains protected even when stored and queried in an outsourced environment.

Therefore, unless an adversary is able to reconstruct the required number of secret shares or completely extract the protected master keys, recovering the original plaintext data remains computationally infeasible. Under standard cryptographic assumptions, any attacker without sufficient key material gains no practical advantage in distinguishing or decrypting the stored information.

## 6- Conclusions

In developing the encryption method, our objective was to support various data types while upholding confidentiality. To achieve this, the proposed method leverages the numerical data encryption strength of the ZSDB method as a foundation, with modifications to the string data encryption method and the incorporation of the SSE algorithm alongside an additional indexing layer. This approach effectively addresses information leakage concerns during substring searches.

Our evaluation of the proposed method primarily focuses on enhancing efficiency and security. The results from these analyses indicate that in the private server implementation of KSDB, there is an approximate 34% increase in overhead during data insertion compared to ZSDB. Both methods require a similar amount of time for searching and extracting data. However, the removal of the tree structure in KSDB has resulted in a significant reduction of approximately 92% in additional memory overhead. Moreover, to ensure data security and prevent potential information breaches, all KSDB implementations have been migrated to a dedicated private server. This secure and reliable entity is responsible for managing critical data, including encryption keys. This strategic decision effectively resolves security issues present in the ZSDB method and facilitates encryption and decryption processes.

## References

[1] H. Tabrizchi, and M. Kuchaki Rafsanjani, “A survey on security challenges in cloud computing: issues, threats, and solutions”, The journal of supercomputing, Vol. 76, No. 12, 2020, pp. 9493-9532.

[2] N. Mohammadi, A. Rezakhani, and H. Haj Seyyed Javadi, “FLHB-AC: federated learning history-based access control using deep neural networks in healthcare system”, Journal of Information Systems and Telecommunication (JIST), Vol.12, No. 46, 2024, pp. 90-104.

[3] V. Govindarajan, "A Novel System for Managing Encrypted Data Using Searchable Encryption Techniques", International Journal of Advanced Computer Science & Applications, Vol. 16, No. 3, 2025, pp. 22-34.
[4] L. Rikhtechi, V. Rafe, and A. Rezakhani, "Secured access control in security information and event management systems", Journal of Information Systems and Telecommunication (JIST), Vol. 9, No. 33, 2021, pp. 67-78.
[5] U. Butt, R. Amin, M. Mehmood, H. Aldabbas, M. Alharbi, and N. Albaqami, "Cloud security threats and solutions: A survey", Wireless Personal Communications, Vol. 128, No. 1, 2023, pp. 387-413.
[6] O. Ebadati, F Eshghi, and A Zamani, "Security enhancement of wireless sensor networks: A hybrid efficient encryption algorithm approach", Journal of Information Systems and Telecommunication (JIST), Vol. 6, No. 23, 2018, pp. 180-192.
[7] C. Wang, K. Ren, W. Lou, and J. Li, "Toward publicly auditable secure cloud data storage services", IEEE Network, Vol. 24, No. 4, 2010, pp. 19-24.
[8] S. Azizi, and D. Mohammadpur, "Searchable Encrypted String for Query Support on Different Encrypted Data Types", KSII Transactions on Internet & Information Systems, Vol. 14, No. 10, 2020, pp. 4198-4213.
[9] E. Khalaf, and M. Kadi, "A survey of access control and data encryption for database security", Journal of King Abdulaziz University, Vol. 28, No. 1, 2017, pp. 19-30.
[10] S. K. Kermanshahi, J. K. Liu, R. Steinfeld, S. Nepal, S. Lai, R. Loh, and C. Zuo, "Multi-client cloud-based symmetric searchable encryption", IEEE Transactions on Dependable and Secure Computing, Vol. 18, No. 5, 2019, pp. 2419-2437.
[11] Z. He, W. K. Wong, B. Kao, D. W. L. Cheung, R. Li, S. M. Yiu, and E. Lo, "SDB: A secure query processing system with data interoperability", VLDB Endowment, Vol. 8, No. 12, 2015, pp. 1876-1879.
[12] W. K. Wong, B. Kao, D. W. L. Cheung, R. Li, and S. M. Yiu, "Secure query processing with data interoperability in a cloud database environment", in Proc. of the 2014 ACM SIGMOD international conference on Management of data, 2014, pp. 1395-1406.
[13] E. Bertino, and R. Sandhu, "Database security-concepts, approaches, and challenges", IEEE Transactions on Dependable and secure computing, Vol. 2, No.1, 2005, pp. 2-19.
[14] Q. Wang, D. Hu, M. Li, Y. Q, "Secure Multi-Character Searchable Encryption Supporting Rich Search Functionalities", IEEE Transactions on Knowledge and Data Engineering, Vol. 38, No. 3, 2026, pp. 1958-1972.
[15] Z. Liu, J. Li, J. Li, C. Jia, J. Yang, and K. Yuan, "SQL-based fuzzy query mechanism over the encrypted database", International Journal of Data Warehousing and Mining (IJDWM), Vol. 10, No. 4, 2014, pp. 71-87.

Journal of Information Systems and Telecommunication
Vol.14, No.1, January-March 2026, 59-66
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

# Distinguishing Human from Bot Texts: A Graph-Based and Few-Shot Learning Approach

Ohood Al-Minshidawi[1], Abdol-Hossein Vahabie[2*]

[1]. Computer Engineering Department, College of Alborz, University of Tehran, Tehran, Iran
[2]. Computer Engineering Department, College of Alborz, & Electrical and Computer Engineering (ECE) Faculty, College of Engineering, University of Tehran, Tehran, Iran

Received: 17 Mar 2025/ Revised: 04 Apr 2026/ Accepted: 06 May 2026

**Abstract**
This study examines the identification of human-generated versus bot-generated content on Arabic social media. The rise of bot accounts and AI-generated writing has facilitated the spread of false information, and the limited reliable Arabic data, as well as linguistic complexity, hinders annotated-data collection. Researchers using traditional supervised and deep learning for Arabic bot detection often rely on computationally expensive methods and large annotated datasets, complicating the evaluation of few-shot learning as an alternative. This study proposes a framework to compare methodological paradigms for classifying Arabic bots using the AutoTweet-Dataset-v1.0. The framework contrasts "graph-based" deep learning methods (e.g., Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT)) with few-shot learning based on the SetFit model (one of three methods tested). It combines a semantic graph representation with a data-efficient few-shot learning framework for Arabic bot classification. The primary finding is that SetFit achieved the highest accuracy (88.35%), outperforming both GCN and GAT. The results suggest that transformer-based few-shot learning offers scalable, effective solutions for identifying bots in low-resource settings and can improve the moderation and integrity of Arabic social media.

**Keywords:** Arabic Text Bot Detection; Graph Neural Networks; Graph Attention Networks; Graph Convolutional Networks; SetFit Model.

## 1- Introduction

Distinguishing human-generated from AI-generated Arabic text on social platforms represents a significant research challenge. The overwhelming number of machines producing tweets and other forms of material across various platforms, including Twitter, has facilitated the propagation of misinformation, spamming, and the spread of manipulated narratives [1]. Although social bots have some genuine applications (such as supplying news updates automatically and passing on information), there are many documented cases of these bots being purposefully misused for disinformation purposes and manipulating public opinion [2,3]. Therefore, establishing a reliable way to identify bot-generated Arabic content is vital in maintaining the credibility of online platforms.
This task becomes even more complicated with respect to Arabic as it is classified as a low-resource language [4]; hence, its morphology is difficult, it has a diverse range of dialects that do not share similarities with each other, and there is limited availability of annotated datasets [5]. The advent of Large Language Models (LLMs) provides tremendous fluency and sophistication in the production of Arabic text, making distinguishing between machine-generated and human-generated enormously difficult [6]. Most existing evaluations of AI-generated content for the purpose of establishing its distinction from human authorship have tended to primarily focus on English and similar Latin script languages, leaving the examination of AI-generated Arabic text comparatively unexplored [7].
Research has focused on applying supervised machine learning, deep learning, and transformer-based approaches to develop bot detection systems. Although there are positive findings presented in many of these publications, there are several limitations shared among the studies, including: (1) reliance on large labelled datasets that are resource-intensive, which are often not available for Arabic; (2) many models have focused on utilizing user metadata or behaviour metrics

✉ **Abdol-Hossein Vahabie**
h.vahabie@ut.ac.ir

to distinguish human-generated text from bot-generated text; and (3) few studies have compared the performance of graph-based relational modelling methods to that of data-efficient few-shot learning approaches for Arabic text classification [8,9]. The aforementioned gaps provide motivations for the current study.

Therefore, this study compared the effectiveness of several approaches to classify text written by humans or bots using two different classifications: Graph Convolutional Networks (GCN), Graph Attention Networks (GAT)), and SetFit model, whereas model GCN or model GAT utilizes graph-based learning and model SetFit utilizes few-shot representation learning.

The design of this paper consists of the following sections: In Section 2, we give an overview of the currently known literature on bot detection; In Section 3, we present our proposed techniques; In Section 4, we describe our evaluation methods for these implementations; and finally, in Section 5, we present some conclusions and provide suggestions for future work.

## 2- Related Works

AI-generated text detection researches have progressed along three major directions: (i) traditional machine learning techniques that rely on feature engineering [10], (ii) content models built using deep neural networks and transformers, and (iii) relational graph-based methods. Progress has been made with existing methods, but their limitations persist, especially in low-resource Arabic environments.

### 2-1- Feature Engineering and Traditional Machine Learning

The development of early methods relied on hand-crafted features to produce a final output using classical algorithms for classification. Bhandarkar et al. [11] used Count Vectorizer features to calculate bag-of-word counts with a Multinomial Naïve Bayes classifier to classify text, achieving an accuracy rate of 86.2%. While they provide an efficient and interpretable way to work with the data, bag-of-words representations are unable to capture any deeper contextual meaning or dependencies between words and can be fooled by highly stylized AI-generated text. Alhayan et al. [12] evaluated traditional machine learning, deep learning, transformer, and hybrid models for analyzing Arabic reviews of e-commerce products. Despite achieving comparable results using an ensemble method (Logistic Regression (LR) + Convolutional Neural Network (CNN)) with approximately 89.7% average accuracy, these approaches relied on annotated training datasets, which hampered their generalization ability to new generative models where the data had changed and evolved. It is evident that both feature-engineered and traditional machine-learning models retain competitive performance but are limited in their ability to produce contextualized and domain-independent transfer learning results. This is especially problematic due to the rich morphological nature of the Arabic language.

### 2-2- Deep Learning and Transformer-Based Content Modeling

Typically, contemporary studies are oriented towards contextual representation learning. For example, Harrag et al. [13] validated AraBERT's fine-tuned performance at detecting GPT-2 generated Arabic tweets. Similarly, Alshammari et al. [14] evaluated AIRABIC, a balanced Arabic benchmark dataset designed with specific diacritics included; fine-tuned XLM-R was shown to outperform commercial systems for detection use. Research demonstrating transformer-based contextual embeddings' capacity to capture subtle stylistic variations supports both Harrag's and Alshammari's results. Despite these advancements in the use of transformer-based contextual embeddings, there are still limitations that hinder content modeling. First, most studies rely on synthetic data created by one model (e.g., GPT-2 or ChatGPT), which may present a limitation concerning the robustness of detectors' ability to detect different generative systems. Second, most studies utilize completely supervised training (with relatively large labeled datasets), an issue often encountered in Arabic language situations where sufficient labeled datasets are not commonly available in practice. Additionally, the majority of previous approaches have focused on content modeling exclusively and neglected the exploration of relational patterns within the corpus.

Wei et al. [15] presented BOTLE, a framework based on content that does not require feature engineering. The system utilizes a multi-embedding Bidirectional Lightweight Gated Recurrent Unit (BiLGRU) architecture and thus performs very well, although the performance is highly dependent on the quality of linguistic pre-processing (Part-of-speech (POS) / Named Entity (NE) tagging), and the evaluations were all performed against a single benchmark dataset; thus, there are substantial concerns about the robustness of the results between domains.

### 2-3- Modeling with Graphs & Relations

Using graphs enables the representation of the relationships between users and content through relational modeling. Alashwal [16] introduced Bot-MGAT, which is a semi-supervised multi-view graph attention network that leverages users' interaction graphs and profile metadata. Bot-MGAT showed transferable learning across TwiBot-20 through the use of semi-supervised learning. Although using relational modeling can increase the robustness and accuracy of relational modeling approaches, they also

introduce practical limitations. For a graph-based system to function properly, large amounts of data, social networks (friendships), and interaction structure must exist at a large scale. Large-scale metadata and social networks may not exist or may contain too much noise or incomplete information. Difficulties may arise when differences in relational properties exist across platforms, as this can affect the scalability and domain portability of a graph-based system. Additionally, most studies do not evaluate their methodology based on purely content-based Arabic benchmarks.

To better understand the differences between textual and lexical relational signals, this study analyzes the AutoTweet dataset, which does not have a rich amount of metadata for user-nodes (users who are actually active social media users). A controlled comparison is conducted between two different types of relational modeling methods (i.e., GCN and GAT), constructed on text structural similarity, compared with fine-tuning on semantic adaptation, using few-sample training methods (e.g., SetFit). The current research methodology and study determine whether corpus-based relational modeling is more beneficial than sentence-level contrastive semantic learning for low-resource Arabic detection of bots. A summary of the results and analytics methodology/results is presented in the Appendix. As illustrated in Figure 1, prior work can be categorized into feature-based, transformer-based, and graph-based approaches, while our study integrates relational modeling and few-shot semantic learning under content-only Arabic settings.

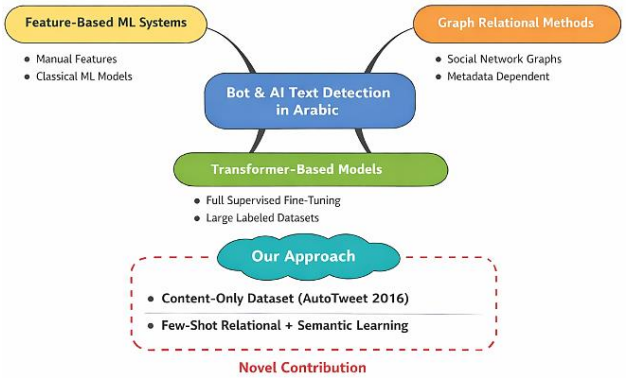



Fig. 1. The existing approaches to AI-generated text detection.

## 3- Methodology

The central research objective is to evaluate whether data-efficient few-shot learning or graph-based relational modeling is more effective for distinguishing human-generated from bot-generated Arabic text in a low-resource setting. This section describes the methodological framework designed to address this objective. Specifically, this study investigates two complementary modelling paradigms:

1. First, using Graph Convolutional Networks (GCN; GAT) to determine whether modelling the relational structure (i.e., word-document relationship at the corpus level) is sufficient for reliably detecting human versus bot content.
2. Second, few-shot representation learning via sentence-level representations (via SetFit) to determine if a limited set of labelled data for training, along with contrastive-based fine-tuning, provides additional discriminative capability to separate human from bot content.

The modeling paradigm was the primary factor that was examined in the experiments, whereas the overall classification results (as measured by accuracy, precision, recall, and F1-score) served as the evaluation criteria for how well the different paradigms performed. The overall pipeline for the experimentation is shown in Figure 1.

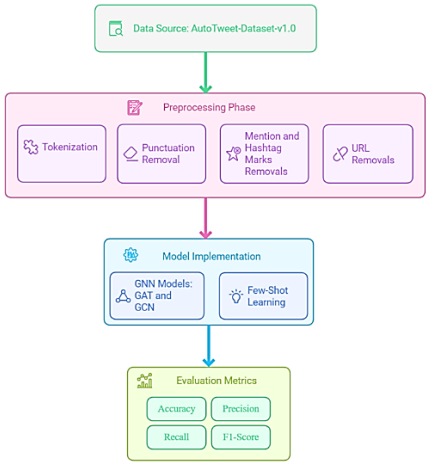



Fig. 2. Our approach for detection of bots and humans in Arabic text.

### 3-1- Dataset and Preprocessing

The experimental dataset used for this study was the AutoTweet-dataset-v1.0 [17], which contains Arabic tweets collected through Twitter's public API. From this dataset, we extracted a large number of content samples generated by both humans and bots. The experimental dataset had 2,627 total samples in the train dataset and 876 samples in the test dataset. A 75/25 train-test split maintained the class balance in the final dataset. During the pre-processing phase, several steps were taken to ensure the reduction of noise in the training and testing datasets. First, the numerical representation of the text (using UTF-8 encoding) was standardized, followed by transforming all text to a lower-case representation and removing all URLs and all punctuation characters. Following that, all mentions and hash marker identifiers were replaced with their equivalent

text string representation. The text was then tokenized, removing from the graph vocabulary any words that occurred less than 15 times within the training dataset, to further help reduce sparsity. These preprocessing steps led to cleaner, more structured input data, improving the model's ability to detect relevant patterns [18].

## 3-2- Implementation

### 3-2-1- Graph-Based Relational Modeling (GCN and GAT)

Bot detection is relational because the way a user uses words typically looks the same, or has nearly identical word usage patterns across the entire collection of data known as "the corpus." To utilize these corpus-level dependencies, we created a heterogeneous graph G = (V, E), which represents all the tweets in our dataset as document nodes and all the words used to create them as word nodes.

We assigned weights to the edges between document and word nodes based on their Term Frequency-Inverse Document Frequency (TF-IDF) [19] score. The edges between word nodes were created based on statistical counts of when words appear in relation to each other, based on a sliding window of 20 words. This composite graph allowed us to display context-related relationships between the words as well as highlight each word's relation to the others across the entire collection of documents.

Node features were initialized using contextual embeddings created from Sentence-BERT (SBERT) [20], specifically four pre-trained models: 1.) all-MiniLM-L12-v2 [21] (384 dimensions), 2.) paraphrase-mpnet-base-v2 [22] (768 dimensions), 3.) paraphrase-multilingual-mpnet-base-v2 [23] (768 dimensions), and 4.) gtr-t5-large [24] (768 dimensions). For the document nodes, the embeddings were calculated using the mean of all tokens that made up the whole tweet text, while for the word nodes, we used the original tokens for each individual token. We used the SBERT encoders as-is with no further fine-tuning while we were training the GNN, thereby allowing for the GNN to take place on frozen semantic representations.

In tuning our learning algorithms, we implemented two GNN architectures in PyTorch Geometric. The first architecture, the GCN model, used a single GCNConv layer that took the input embeddings and projected them into a 100-dimensional hidden layer. This was followed by a ReLU activation, dropout (p = 0.5), and a fully connected layer mapping to two output classes. The second architecture, the GAT model, utilized a single GATConv layer that projected the input embeddings into a 50-dimensional representation, followed by attention-based aggregation of neighbor nodes, and finally the same classification head as the GCN Model.

Both models were trained using negative log-likelihood loss in combination with the Adam optimizer ($\alpha = 0.001$; $\beta_1 = 0.9$; $\beta_2 = 0.999$, and no weight decay). While the GCN was trained for 50 epochs, GAT required 500 epochs because it converged more slowly. The dataset was split 90:10 into training and validation datasets, and no early stopping or systematic hyperparameter search was done, with the final evaluation being performed on a held-out test set.

The GCN and GAT model configuration and key hyperparameters of the implementation are in Tables 1. Each model had a single graph layer (i.e., aggregators) with ReLU activation and dropout regularization. The primary architectural difference between GCN and GAT was that GAT used an attention mechanism, whereas GCN did not; therefore, maintaining consistency across optimization parameters in both models allowed for an effective comparison between convolutional-based relational aggregation (GCN) and attention-based message passing (GAT).

Table 1. The hyperparameters used in the graph-based implementation step.

| Model | Hidden Dim | Dropout | Optimizer | Learning Rate | Epochs |
|---|---|---|---|---|---|
| **GCN** | 100 | 0.5 | Adam | 0.001 | 50 |
| **GAT** | 50 | 0.5 | Adam | 0.001 | 500 |

### 3-2-2- Few-Shot Learning (SetFit)

We explored the data-efficient transformer by utilizing the SetFit framework. Utilizing Contrastive Siamese [25] fine-tuning of a sentence transformer and fitting it with a lightweight classifier as a head makes this method ideal for low-resource situations where little Arabic bot data exists, and labeling this data is not widely performed.

We used the sentence-transformers/all-MiniLM-L12-v2 model as our source encoder. We trained our encoder for a few-shot system using 200 labeled samples per class (400 total training instances). During contrastive fine-tuning, we trained the encoder for five epochs using a batch size of 16. The optimizer for the encoder was AdamW with default settings as described in the SetFit implementation of this library. After creating the contrastive representations, we fitted a classifier head on top of the frozen embeddings. We evaluated two different types of classifiers at this stage: LR and Multi-Layer Perceptron (MLP) with a maximum of 500 iterations, and we trained the classifier for 15 epochs at this stage. We set the random seed to 444 to allow for reproducibility of the results of each classifier. The hyperparameters for both classifier types are summarized in Table 2. The goal of this model was to determine whether learning semantics at the sentence level with low supervision results in an accurate method for detecting Arabic bots without needing to construct an explicit relational graph structure. The experimentation was done using NVIDIA H100 GPUs with a total GPU memory of 80GB and 4 CPUs and a total of 50GB CPU memory.

Table 2. The hyperparameters used in the few-shot learning implementation step.

| Parameter | Value |
|---|---|
| **Base Encoder** | all-MiniLM-L12-v2 |
| **Few-shot samples** | 200 per class |
| **Batch size** | 16 |
| **Contrastive epochs** | 5 |
| **Classifier** | LR / MLP |
| **MLP max_iter** | 500 |
| **Classification epochs** | 15 |
| **Optimizer** | AdamW (default) |
| **Random seed** | 444 |

## 4- Results and Discussion

The performance of all models was measured by calculating accuracy, precision, recall, and F1 score with respect to the held-out test set. As shown in Table 4, the top-performing model, SetFit, achieved F1 and accuracy scores of 88.11% and 88.35%, respectively. Both GCN and GAT performed close to each other, with GAT (F1 = 87.28%) marginally outperforming GCN (F1 = 87.16%). GAT's minor but consistent edge relative to GCN is likely due to the attention mechanism in GAT allowing for the use of different weights when aggregating the neighborhood information of a node. In contrast, the weights assigned in aggregation by GCN are uniform across all nodes. For example, when detecting whether text is human-authored or machine-generated (bots), certain lexical nodes (i.e., repetitively used promotional language and automated language) may provide a stronger indication relative to other lexical nodes in determining the difference between the two types of authorship. GAT's attention mechanism thus allows it to weigh the contribution of such important neighbours greater than that of less informative neighbours, resulting in slightly better accuracy. The relatively small performance difference may suggest that the relational nature of the graph (i.e., how the examples are related to one another) has a more significant impact on the overall performance of GNNs than the manner in which the aggregation is performed. SetFit outperformed both GNN-based models by about 0.8%–1% with respect to F1-score, and this is likely due to being able to model sentence-level semantics through contrastive fine-tuning of transformer embeddings. In contrast, the GNN-based models primarily rely on lexical co-occurrence patterns represented within the graph structure. While GNN-based models are capable of identifying the types of words that co-occur together, they are unable to discern patterns that occur within a single sentence since they are unable to create representations at this level. In contrast, SetFit is able to directly optimize the representation of sentences for the detection of bots vs humans and thus, be able to identify patterns at a higher level (both semantically and stylistically) beyond simply an analysis of word frequency statistics. In addition, the few-shot training paradigm allows the encoder to concentrate on user-level distinctions relevant to the task, enabling the capture of fine-grained stylistic cues typical of automated accounts. Conversely, graph-based models rely heavily on the stability of co-occurrence statistics, which may not be stable enough for use in short and noisy social text data. By adding context to the embedding, transformer-based embeddings can also operate more effectively with sparse and varied lexicons. Using limited supervision, semantic representation learning has greater discriminative power than purely relational lexical models for detecting Arabic bots in social media content that is written in short form. Additionally, we compared our results with those of Hassan et al. [26], who used traditional machine learning, ensemble, and deep learning on the same dataset. The best results by their support vector machine (SVM) with unigram features on non-preprocessed text achieved an accuracy of 83.11% and an F1-score of 82.71%. All three models we proposed provided significantly better performance than this baseline, offering approximately a 4-5% F1-score improvement per model. Notably, Hassan et al.'s model achieved a precision rate of 95.16%, while it attained a rather low rate of 73.14% recall, which indicates that their model conservatively labels AI-generated texts and failed to detect a substantial proportion of automated accounts. In contrast, we have provided a more balanced ratio of precision to recall, thereby improving the overall F1-score for our models. Hassan's results indicate better performance using contextual embedding methods (SetFit) and relational graph models (GAT/GCN) as compared to either bag-of-words methods or shallow classifiers. These results demonstrate that both semantic representation and relational structure provide additional discrimination power in identifying Arabic AI-generated texts. The data clearly demonstrate this performance order: SetFit > GAT > GCN > traditional machine learning baselines. It is essential to note that while relationally modeled data provide meaningful contributions to system performance, the largest gains in our system performance were obtained via the application of task-specific semantic fine-tuning in the implementation of few-shot paradigms. Thus, we indicate that sentence-level semantic adaptation is more important than lexical propagation through graph structures alone.

Table 3. Results on test sets for proposed models.

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| **GCN** | 87.44% | 86.96% | 87.43% | 87.16% |
| **GAT** | 87.55% | 87.07% | 87.60% | 87.28% |
| **SetFit** | **88.35%** | **87.88%** | **88.45%** | **88.11%** |
| **Hassan et al. [26]** | 83.11% | 95.16% | 73.14% | 82.71% |

## 5- Conclusion

This research explored two approaches to identifying AI-generated Arabic text: the relational graph modeling method, which included the GCN and GAT, and data-efficient representation learning through the SetFit model. The results have produced a clear ranking of the three models based on their F1-scores, which reveal that SetFit achieved superior performance (88.11%) compared to GCN and GAT.

From these results, there are meaningful scientific contributions. SetFit's successful performance illustrates that semantic adaptation at the sentence level through contrastive fine-tuning can produce greater discrimination than simply using word co-occurrence frequency. GCN and GAT each capture relational structure within the larger corpus of text; however, their reliance on statistical associations among the constituent terms limits their ability to capture other deeper structural and contextual characteristics. On the other hand, the few-shot learning process associated with the transformer model directly adjusts contextual representations of words/phrases to fit the characteristics of the target task, affording greater flexibility to process shorter, noisier social media posts. Additionally, the minimal increase in performance obtained using GAT as compared to GCN indicates that while relational structure is advantageous, improvements are obtained via adaptive attention.

These results imply that, for low-resource Arabic bot detection, task-specific semantic fine-tuning plays a more critical role than structural graph propagation alone. Accordingly, our research provides empirical data supporting the efficacy of employing contrastive few-shot learning for the classification tasks of low-resource NLP.

### 5-1- Limitations

Although there were positive outcomes, this study has several limitations. First, the AutoTweet dataset was produced in 2016; the language used in tweets created by bots has changed over time due to newer methods of creating AI-generated text (such as ChatGPT).

Second, the focus of the research was on short, structured tweets; however, this differs from long-format literature, such as Arabic newspapers or magazines, or conversational dialogue (casual back- and- forths) or content from different types of AI-generated systems (GPT-4), which can further limit the potential of cross-validation due to differences in writing formats. Third, correlation testing and cross-domain validations were insufficient, thus making it difficult to determine if there were any limitations to cross-domain validity within any sample of data collected in this dataset (due to insufficient data).

### 5-2- Future Works

Future work should focus on three areas. (1) Cross-domain validity-testing can be accomplished using only a few-shot semantic learning across various types of content, including many modern Arabic-language literature samples and various forms of free-written (non-structured) dialogue. (2) Further studies to evaluate generative AI models, such as large-scale language models, through the use of three different evaluation conditions (zero-shot, few-shot, and fine-tuning) should be conducted to determine whether recent generative AI systems alter the detection landscape. (3) Combining relational graph methods with contrastively fine-tuned semantic (word-based) encoding would also have the potential for greater robustness via leveraging both corpus-level structure and contextual representation learning. Addressing these directions will enable progress toward more adaptive and future-proof systems.

## References

[1] A. Nambiar, "Impact of fake news, message and spam spread through social media on people decision making ability," 2022.

[2] D. Assenmacher, L. Clever, L. Frischlich, T. Quandt, H. Trautmann, and C. Grimme, "Demystifying social bots: On the intelligence of automated social media actors," Social Media+ Society, vol. 6, no. 3, p. 2056305120939264, 2020.

[3] D. Ajiga, P. A. Okeleke, S. O. Folorunsho, and C. Ezeigweneme, "The role of software automation in improving industrial operations and efficiency," International Journal of Engineering Research Updates, vol. 7, no. 1, pp. 22-35, 2024.

[4] S. S. Sabr et al., "A Comprehensive Part-of-Speech Tagging to Standardize Central-Kurdish Language: A Research Guide for Kurdish Natural Language Processing Tasks," Journal of Studies in Science and Engineering, vol. 5, no. 2, pp. 15-38, 2025.

[5] N. S. Alghamdi and J. S. Alowibdi, "Distinguishing Arabic GenAI-generated tweets and human tweets utilizing machine learning," Engineering, Technology & Applied Science Research, vol. 14, no. 5, pp. 16720-16726, 2024.

[6] H. Alshammari and K. Elleithy, "Toward Robust Arabic AI-Generated Text Detection: Tackling Diacritics Challenges," Information, vol. 15, no. 7, p. 419, 2024. [Online]. Available: https://www.mdpi.com/2078-2489/15/7/419.

[7] H. Alshammari, A. El-Sayed, and K. Elleithy, "Ai-generated text detector for arabic language using encoder-based transformer architecture," Big Data and Cognitive Computing, vol. 8, no. 3, p. 32, 2024.

[8] H. Alshammari, AI-Generated Text Detector for Arabic Language. University of Bridgeport, 2024.

[9] B. Sani et al., "Who wrote this? Identifying machine vs human-generated text in Hausa," in Proceedings of the Sixth Workshop on African Natural Language Processing (AfricaNLP 2025), 2025, pp. 82-88.

[10] A. A. Abdullah, N. S. Mohammed, M. Khanzadi, S. M. Asaad, Z. K. Abdul, and H. S. Maghdid, "In-depth analysis on machine learning approaches: Techniques, Applications, and trends," Aro-The Scientific Journal of Koya University, vol. 13, no. 1, pp. 190-202, 2025.

[11] A. Bhandarkar, M. A. DM, D. Vishwachetan, A. Mushtaq, D. Kadam, and S. Saxena, "Unmasking the AI Hand: A Machine Learning Approach to Deciphering Authorship," in 2024 3rd International Conference for Innovation in Technology (INOCON), 2024: IEEE, pp. 1-6.

[12] F. Alhayan and H. Himdi, "Ensemble learning approach for distinguishing human and computer-generated Arabic reviews," PeerJ Computer Science, vol. 10, p. e2345, 2024.

[13] F. Harrag, M. Debbah, K. Darwish, and A. Abdelali, "Bert transformer model for detecting Arabic GPT2 auto-generated tweets," arXiv preprint arXiv:2101.09345, 2021.

[14] H. Alshammari, "AI-Generated Text Detector for Arabic Language," University of Bridgeport, 2024.

[15] F. Wei and U. T. Nguyen, "Twitter bot detection using neural networks and linguistic embeddings," IEEE Open Journal of the Computer Society, vol. 4, pp. 218-230, 2023.

[16] E. Alothali, "STREAM-EVOLVING BOT DETECTION FRAMEWORK USING GRAPH-BASED AND FEATURE-BASED APPROACHES FOR IDENTIFYING SOCIAL BOTS ON TWITTER," 2023.

[17] H. Almerekhi and T. Elsayed, "Detecting automatically-generated arabic tweets," in Information Retrieval Technology: 11th Asia Information Retrieval Societies Conference, AIRS 2015, Brisbane, QLD, Australia, December 2-4, 2015. Proceedings 11, 2015: Springer, pp. 123-134.

[18] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," Information processing & management, vol. 50, no. 1, pp. 104-112, 2014.

[19] W. I. D. Mining, "Data mining: Concepts and techniques," Morgan Kaufinann, vol. 10, no. 559-569, p. 4, 2006.

[20] N. Reimers, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," arXiv preprint arXiv:1908.10084, 2019.

[21] "all-MiniLM-L12-v2." https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2 (accessed.

[22] N. a. G. Reimers, Iryna, "paraphrase-mpnet-base-v2," 2019. [Online]. Available: https://huggingface.co/sentence-transformers/paraphrase-mpnet-base-v2.

[23] N. a. G. Reimers, Iryna, "paraphrase-multilingual-mpnet-base-v2," 2019. [Online]. Available: https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2.

[24] "gtr-t5-large." https://huggingface.co/sentence-transformers/gtr-t5-large (accessed.

[25] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in International conference on machine learning, 2020: PMLR, pp. 1597-1607.

[26] S. I. Hassan, L. Elrefaei, and M. S. Andraws, "Arabic Tweets Spam Detection Based on Various Supervised Machine Learning and Deep Learning Classifiers," MSA Engineering Journal, vol. 2, no. 2, pp. 1099-1119, 2023.

# Appendix

**Comparative Summary of Representative AI-Generated Text Detection Studies**

| Study | Methodology | Advantages | Limitations |
|---|---|---|---|
| Alshammari [14] | Fine-tuned XLM-R, AraBERT, mBERT + Dediacritization layer | • First balanced Arabic benchmark (AIRABIC)<br>• Systematic handling of diacritics<br>• Strong F1 | • Reliance on ChatGPT-generated data<br>• Limited dialectal coverage<br>• Domain shift sensitivity |
| Alashwal [16] | Semi-supervised Multi-view Graph Attention Network (Bot-MGAT) with transfer learning | • Effective use of labeled + unlabeled data<br>• Strong generalization<br>• High F1 | • Requires rich metadata<br>• Scalability concerns<br>• Limited Arabic evaluation |
| Wei et al. [15] | Multi-embedding (word, char, POS, NE) + BiLGRU | • No handcrafted profile features<br>• Competitive performance | • Evaluated on single dataset<br>• Limited domain generalization<br>• Content-only constraints |
| Bhandarkar et al. [11] | Count Vectorizer + Multinomial Naïve Bayes | • Computational efficiency<br>• Interpretability | • Bag-of-words limitation<br>• Vulnerable to stylistic evolution<br>• Multilingual weakness |
| Harrag et al. [13] | Fine-tuned AraBERT vs RNN baselines | • Strong contextual modeling<br>• High accuracy | • Potential overfitting to GPT2<br>• Limited robustness to unseen generators |
| Alhayan et al. [12] | ML, DL, Transformer, Ensemble (LR+CNN) | • Broad comparative evaluation<br>• Hybrid effectiveness | • Domain-specific dataset<br>• Limited generalization to evolving generators |

Journal of Information Systems and Telecommunication
Vol.14, No.1, January-March 2026, 67-78
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

# Image-Based Phishing URL Classification Using Convolutional Neural Networks

Hamed Monkaresi [1*], GholamReza Ahmadi [1]

[1]. Department of Computer Engineering and Information Technology, Razi University, Kermanshah, Iran

Received: 22 Nov 2025/ Revised: 04 Mar 2026/ Accepted: 06 May 2026

**Abstract**

Phishing attacks continue to pose a significant threat to online security, with attackers increasingly leveraging deceptive URLs to steal sensitive information. Traditional phishing detection methods often rely on URL analysis or manual feature extraction, which can be time-consuming and less effective against evolving attack techniques. To address these limitations, more adaptive and intelligent detection mechanisms are increasingly required to keep pace with modern attack strategies. In this paper, we propose an image-based approach for phishing URL classification using Convolutional Neural Networks (CNNs). By transforming URLs into visual representations based on their features, we leverage the power of deep learning to automatically extract discriminative features for classification. We conduct a comprehensive comparison of various deep learning models, including different CNN architectures (both basic and pre-trained/fine-tuned), to evaluate their performance in terms of accuracy, computational efficiency, and training time. Our experiments demonstrate that image-based classification using CNNs achieves competitive accuracy while offering potential robustness against adversarial variations in phishing URLs. Additionally, we analyze the trade-offs between model complexity and inference time, providing insights into the practical deployment of such systems. The results highlight the potential of image-based deep learning models as an effective tool for phishing detection, paving the way for further research in this domain.

**Keywords:** Phishing Detection; URL Classification; Convolutional Neural Networks (CNNs); Deep Learning; Image-Based Classification; Cybersecurity.

## 1- Introduction

The ubiquitous nature of the internet has made online security a paramount concern. Among the myriads of cyber threats, phishing remains one of the most pervasive and damaging attacks [1]. Phishing attacks typically involve tricking users into revealing sensitive information, such as login credentials, credit card numbers, or personal identification details, by masquerading as a trustworthy entity in electronic communication. Deceptive Uniform Resource Locators (URLs) are a primary vector for these attacks, designed to mimic legitimate websites and lure unsuspecting victims [2].

Traditional methods for detecting phishing URLs often involve blacklist/whitelist approaches, heuristic-based analysis of URL strings (lexical features), website content analysis, or checking domain registration information [3]. While these methods have shown some success, they face significant challenges. Blacklists are inherently reactive and cannot identify zero-day phishing sites. Heuristic methods require careful manual feature engineering and struggle to keep pace with the rapidly evolving obfuscation techniques used by attackers, such as URL shortening, character encoding tricks, and the use of subdomains [4]. Machine learning techniques using handcrafted features have improved detection rates, but feature engineering remains a bottleneck.

The limitations of traditional approaches highlight the need for more adaptive and automated feature learning methods. Convolutional Neural Networks (CNNs) excel at automatically discovering hierarchical patterns and spatial relationships in data, eliminating the need for manual feature engineering. Unlike traditional algorithms that process features independently, CNNs can capture complex nonlinear interactions between features through their convolutional and pooling operations. This capability is particularly valuable for phishing detection, where the relationships between URL characteristics (such as the interplay between URL length, domain structure, and

✉ **Hamed Monkaresi**
h.monkaresi@razi.ac.ir

entropy measures) may be as important as individual feature values.

Recently, deep learning has emerged as a powerful tool in various domains, including cybersecurity, due to its ability to automatically learn complex patterns and hierarchical features from raw data [5]. Convolutional Neural Networks (CNNs) have revolutionized image recognition tasks by effectively capturing spatial hierarchies of features [6]. Inspired by this success, we explore the potential of applying CNNs to phishing URL detection by transforming URL data into image representations. Our approach represents a unique paradigm shift in phishing detection by creating synthetic images from structured URL features, enabling the application of powerful computer vision techniques to cybersecurity. Unlike existing methods that apply CNNs directly to URL strings or use natural images (screenshots), our method transforms feature vectors into structured visual patterns that preserve feature relationships while leveraging CNN's spatial pattern recognition capabilities. The core idea is that the structural and statistical properties of URLs, captured by various features, can be mapped into a visual format (an image), allowing CNNs to identify patterns indicative of phishing attempts.

## 1-1- Problem Formulation

The problem addressed in this study can be formulated as a binary classification task with the following specifications:

- Input Space: A URL represented by a feature vector $X = \{f_1, f_2, ..., f_{41}\}$, where each $f_i \in \mathbb{R}$ is a numerical or categorical feature extracted from the URL's lexical, structural, domain-based, and content-based properties.
- Output Space: A binary label $Y \in \{0, 1\}$, where 0 represents legitimate URLs and 1 represents phishing URLs.
- Transformation Function: $T: \mathbb{R}^{41} \rightarrow \mathbb{R}^{41x41xc}$, where T converts the feature vector into a visual representation (image) with c channels (1 for grayscale, 3 for RGB).
- Learning Objective: To develop a CNN-based mapping function $F: \mathbb{R}^{41x41xc} \rightarrow \{0, 1\}$ that maximizes classification accuracy while maintaining computational efficiency for practical deployment.
- Primary Goals:
    - Maximize classification accuracy: max $P(F(T(X)) = Y)$
    - Minimize computational complexity for real-time deployment
    - Evaluate robustness against feature variations

Our primary objectives are:

- To develop a methodology for transforming feature-based URL representations into images suitable for CNN analysis.
- To implement and evaluate various CNN architectures, including basic CNNs, custom deep CNNs, and state-of-the-art pre-trained models (VGG16, ResNet50, EfficientNetB0) fine-tuned for this task.
- To compare the performance of these image-based deep learning models against traditional machine learning algorithms trained on the raw features.
- To analyze the trade-offs between classification accuracy, model complexity, training time, and inference speed for practical deployment considerations.

In high-stakes cybersecurity applications, even modest improvements in accuracy can translate to substantial reductions in successful phishing attacks. For instance, a 2-3% improvement in detection accuracy could prevent thousands of users from falling victim to phishing attempts in large-scale deployments, justifying the computational complexity of deep learning approaches.

The main contributions of this work include:

- A pioneering application of image-based classification using CNNs for phishing URL detection, specifically designed for structured feature data rather than raw URL strings or webpage screenshots.
- A novel application of image-based classification using CNNs for phishing URL detection based on URL features.
- A comprehensive comparative study of different deep learning architectures for this specific task.
- Empirical evidence demonstrating the trade-offs between accuracy gains and computational costs, with specific recommendations for different deployment scenarios.
- A reproducible framework for converting structured cybersecurity features into visual representations suitable for deep learning analysis
- Empirical evidence demonstrating that adapting this image-based approach specifically to feature-engineered phishing URL datasets can yield significant performance gains over traditional methods.

The rest of the paper is organized as follows: Section 2 reviews related work in phishing detection and deep learning applications. Section 2.1 identifies key research gaps in existing literature Section 3 details the methodology for data preparation and image generation. Section 4 describes the experimental setup, including the models and evaluation metrics. Section 5 presents and discusses the experimental results. Section 6 concludes the paper,

summarizing findings and outlining future research directions.

## 2- Related Work

Recent studies have explored various cybersecurity challenges related to online threats. For example, secure mutual authentication mechanisms for wireless body area networks have been presented in [20], and blockchain-based authentication verification frameworks have been proposed in [21]. Additionally, intelligent phishing detection approaches leveraging image, frame, and textual features have been examined in [22], highlighting the journal's focus on countering online deception and threat analysis. Research on IoT-based security systems, such as home surveillance architectures [23].

In contrast to these works, our approach introduces an image-based URL phishing detection framework using CNNs, providing a novel transformation of URL features into visual representations to improve robustness and classification accuracy.

Phishing detection has been an active research area for years, leading to a variety of proposed techniques.

**Traditional Phishing Detection**: Early approaches heavily relied on blacklisting, maintaining lists of known phishing URLs [7]. While simple, this method fails against newly created sites. Heuristic-based methods analyze URL characteristics (e.g., length, presence of IP addresses, special characters, domain age, keywords like 'login' or 'secure') and website content (e.g., HTML structure, forms, JavaScript) to identify suspicious patterns [2]. These often involve manually defining rules or features. Machine Learning (ML) approaches automated the detection process using traditional algorithms like Support Vector Machines (SVM), Random Forests (RF), Logistic Regression (LR), and Naive Bayes, trained on hand-engineered features extracted from URLs, domain information, and webpage content [8]. While more robust than static heuristics, their performance heavily depends on the quality and relevance of the engineered features.

**Deep Learning in Cybersecurity**: Deep learning models have shown promise in various cybersecurity tasks, including intrusion detection, malware analysis, and spam filtering [9]. In phishing detection, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have been used to analyze the sequential nature of URL strings [10]. CNNs have also been applied directly to URL strings (treating them as sequences) or combined with RNNs [11]. Other works have used deep learning for analyzing website visual appearance (screenshots) [12] or network traffic patterns related to phishing sites.

developments in cybersecurity have explored hybrid approaches for improving detection accuracy across various domains. Iftikhar et al. [24] proposed a hybrid Self-Organizing Map (SOM) model combined with Extreme Gradient Boosting (XGBoost) for intrusion detection in IoT environments, achieving significant improvements in precision, recall, and F1-scores (10-30% improvements for different attack classes). Their work demonstrates the effectiveness of combining dimensionality reduction techniques with advanced machine learning algorithms, achieving F1-score improvements of 7.91%, 32.62%, and 12.45% for DoS, probe, and benign classes respectively. This hybrid approach paradigm aligns with our methodology of combining image transformation with deep learning for enhanced pattern recognition.

**Image-Based Classification Approaches**: The idea of converting non-image data into image representations for analysis with CNNs has been explored in other fields. For instance, time-series data has been converted into Gramian Angular Fields or recurrence plots for classification using CNNs [13]. In cybersecurity, malware binaries have been visualized as grayscale images, allowing CNNs to detect malware families based on texture patterns [(14]. However, these approaches typically deal with naturally sequential or binary data that can be directly mapped to pixel intensities. Our work differs by addressing the unique challenge of transforming heterogeneous, structured feature vectors into meaningful visual representations. Our work builds on this concept by specifically transforming URL features into images. While some studies might have used website screenshots (which are images) for phishing detection [15], our approach differs fundamentally by creating synthetic images directly from the underlying URL feature vectors, aiming to capture correlations and patterns among these features visually. To the best of our knowledge, representing a vector of diverse URL features as a single structured image for CNN-based phishing classification represents a novel intersection of computer vision and cybersecurity domains. This approach uniquely combines the benefits of established feature engineering in cybersecurity with the automatic pattern recognition capabilities of deep learning.

### 2-1- Research Gap Identification

Despite the extensive research in phishing detection, several critical gaps remain:

1. Feature **Engineering Bottleneck**: Traditional ML approaches require extensive domain expertise for feature selection and engineering, limiting their adaptability to evolving phishing techniques.
2. Limited **Spatial Relationship Exploitation**: Existing deep learning approaches for phishing detection primarily treat features independently or analyze sequential patterns in URL strings,

missing potential spatial correlations between different feature types.

3. Computational **Efficiency Trade-offs**: While deep learning models show improved accuracy, limited research has systematically analyzed the computational costs and practical deployment considerations for real-time phishing detection systems.
4. Robustness **Against Adversarial Attacks**: Most existing approaches lack comprehensive evaluation against adversarial manipulations specifically designed for phishing detection systems.
5. Transfer **Learning Under exploration**: The potential of leveraging pre-trained computer vision models for cybersecurity applications remains largely unexplored, despite their success in other domains.

Our work addresses these gaps by proposing a novel image-based representation that enables spatial feature relationship learning while providing a comprehensive analysis of accuracy-efficiency trade-offs.

## 3- Methodology

The methodology of this study is structured into two main phases: Data Preparation and Image Generation. Each phase is designed to transform raw URL data, represented by extracted features, into a format suitable for training and evaluating Convolutional Neural Networks (CNNs).

### 3-1- Data Preparation

Dataset Description: The dataset contains **247,950 samples** with a balanced distribution of **88,647 phishing URLs** and **159,303 legitimate URLs**, each represented by 41 distinct features capturing lexical, domain-based, structural, and content-based characteristics of URLs. The dataset was sourced from the UCI Machine Learning Repository Phishing Websites Data Set, which has been widely used in cybersecurity research for benchmarking phishing detection algorithms. This dataset ensures reproducibility and enables fair comparison with existing literature.

- **Feature Categories**: The 41 features can be categorized into four main groups:
- **Lexical Features (15 features)**: URL length, number of dots, special characters count, entropy measures
- **Domain-based Features (10 features)**: Domain age, DNS records, WHOIS information, domain reputation
- **Structural Features (8 features)**: Number of subdomains, path depth, parameter count, fragment presence
- **Content-based Features (8 features)**: Presence of forms, JavaScript usage, external links, favicon characteristics.

These features include metrics such as URL length, number of dots, special characters in various URL components, domain age, DNS records, presence of '@' or shortening services, entropy measures, structural anomalies, and HTML/JavaScript-based attributes (though primarily focused on URL/domain-derived features). Each sample is labeled in the "Type" column as either phishing (1) or legitimate (0), providing clear ground truth for supervised learning. The dataset contains no missing values, ensuring data integrity for model training. The balanced nature of the dataset (approximately 36% phishing, 64% legitimate) reflects realistic phishing detection scenarios while avoiding class imbalance issues. The dataset has no missing values, with a balanced class distribution of [insert phishing count] phishing URLs and [insert legitimate count] legitimate URLs, ensuring robust model training.

To understand the relationships among these features, we analyzed their correlations, as shown in Figure 1.a. This heatmap reveals significant correlations between certain features, such as URL length and the number of dots, which may inform the visual patterns captured in the image representations.

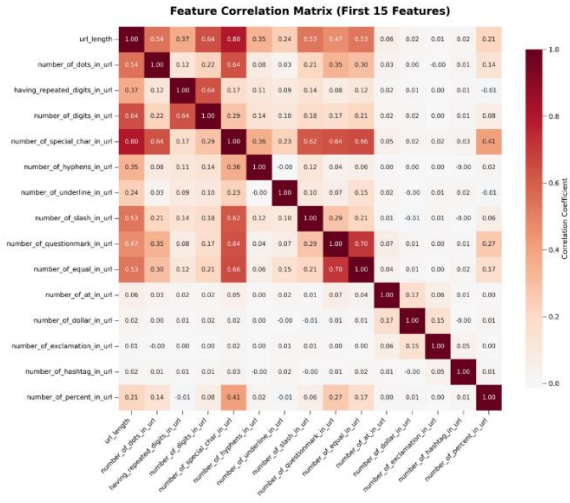



Fig. 1(a) : Correlation heatmap of the 15 First URL features, highlighting relationships that may influence the image-based representation for CNN classification.

To facilitate a detailed comparison between legitimate and phishing URLs, we generated high-resolution feature map visualizations (Figure 1.b) from normalized values of 41 URL-, domain-, subdomain-, and path/query-related features. Each sample was represented using multiple visual arrangements—tiled repetition, structured feature ordering, and a spiral mapping—to highlight intensity patterns across

the feature space. Feature group boundaries were annotated to improve interpretability. Additionally, a comprehensive heatmap analysis compared average feature values between classes, displaying absolute differences and relative ratios. These visualizations reveal clear separations in several feature groups—such as URL length, entropy, and special character counts—where phishing URLs consistently exhibit higher magnitudes. By combining individual feature maps with aggregated statistical heatmaps, this approach enables both micro-level inspection of specific samples and macro-level understanding of systematic differences between legitimate and phishing URLs.

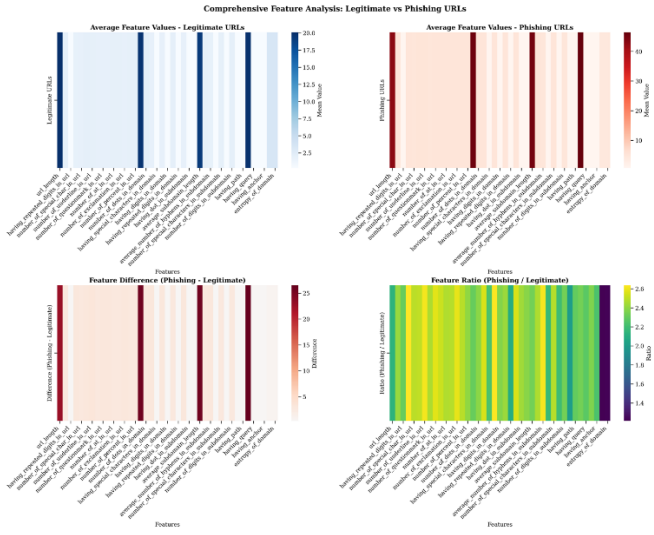



Figure 1.b: Feature map visualizations and heatmaps comparing legitimate and phishing URLs, showing distinct patterns in URL length, entropy, and special character usage.

**Feature Normalization**: To ensure that the features are on a comparable scale and to facilitate the creation of distinct visual representations, each feature value ( f ) is normalized. **Rationale for Min-Max Scaling**: Min-Max scaling is chosen over other normalization techniques (such as Z-score standardization) because it preserves the relative relationships between feature values while mapping them to a fixed range suitable for pixel intensities. This approach ensures that features with different scales contribute proportionally to the visual representation without introducing artificial distributions. Min-Max scaling is applied to map values to the range [0, 255] for pixel intensities, suitable for grayscale image generation:

$$f'ij = \left(fij - \min_j \right) / \left(\max_j - \min_j \right) \times 255$$

This normalization ensures that each feature contributes proportionally to the final image representation, avoiding dominance by features with intrinsically larger numeric ranges. The normalized values are then used as pixel intensities. Figure 2 illustrates the distribution of key features after normalization, emphasizing their spread and suitability for image conversion.

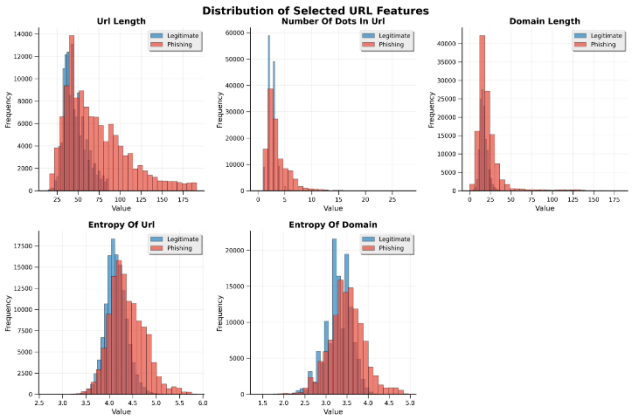



Fig. 2: Distribution of normalized features ("url_length","Number of Dots In Url", "domain_length", "entropy_of_url", "entropy_of_domain") used in image generation, showing their spread and variability.

The conversion from feature vector to image follows a systematic process designed to preserve feature relationships while creating meaningful spatial representations:

**Step 1:** Each URL's 41 normalized features are initially arranged as a 1×41 vector: F = [$f_1$, $f_2$, $f_3$, ..., $f_{41}$]

**Step 2:** The feature vector is transformed into a 41×41 square matrix using the following mapping strategy:

- **Row 1: Contains all 41 original feature values**
- **Rows 2-41: Filled using a padding strategy that maintains spatial coherence**

**Padding Strategy Options**:

- **Zero Padding**: Remaining positions filled with zeros (pixel value 0)
- **Replication Padding**: Feature values repeated to fill the matrix
- **Symmetric Padding**: Features arranged to create symmetric patterns

For this study, we employed replication padding, where features are cyclically repeated to maintain information density throughout the image.

**Step 3: Image Format Conversion**

- **Grayscale Images**: For basic and custom CNNs, the 41×41 matrix directly represents pixel intensities (single channel)
- **RGB Images**: For pre-trained models requiring 3-channel input, the grayscale channel is replicated three times: I_RGB = [I_gray, I_gray, I_gray].

**Image Conversion**: Each row of the dataset, representing a single URL with its 41 normalized features, is transformed into a 41x41 matrix. The 41 feature values are arranged in the first row of the matrix, with the remaining rows padded or repeated to form a square grid. The resulting matrix is converted into a grayscale image, where each pixel's intensity corresponds to the normalized feature value at that grid position. If using pre-trained models requiring 3-channel (RGB) input, the grayscale channel is replicated three times.

**Spatial Pattern Preservation**: While the current linear arrangement may seem simplistic, the correlation analysis (Figure 1) demonstrates that related features exhibit intensity patterns that CNNs can learn to recognize. The spatial arrangement allows convolutional filters to detect local patterns that correspond to feature interactions, even in this synthetic image space. This image-based representation allows CNNs to extract spatial patterns and correlations from the visual data.

## 3-2- Image Generation

**a. Directory Structure**: To organize the dataset for standard deep learning workflows, a hierarchical directory structure is created. Separate parent directories are established for the train, validation, and test datasets. Within each of these parent directories, subdirectories are created for the two classes: phishing and legitimate. This structure (dataset_root/train/phishing/, dataset_root/train/legitimate/, etc.) is compatible with common data loading utilities in deep learning frameworks.

**b. Image Saving**: The dataset is split into three subsets: training (80%), validation (10%), and test (10%). This split ensures that the model is trained on a majority of the data while retaining separate, unseen samples for hyperparameter tuning (validation set) and final unbiased evaluation (test set).

The dataset splitting maintains the original class distribution across all subsets, ensuring that each subset is representative of the overall dataset. This approach prevents potential bias that could arise from imbalanced splits. The grayscale (or 3-channel replicated) images generated from the normalized feature matrices are saved in their respective class and split directories (e.g., .png or .jpg format). This organized storage facilitates efficient data loading using image data generators during the training and evaluation phases.

To illustrate the image conversion process, Figure 3 shows feature maps generated from eight sample URLs in the dataset, highlighting the visual patterns that CNNs analyze during training.

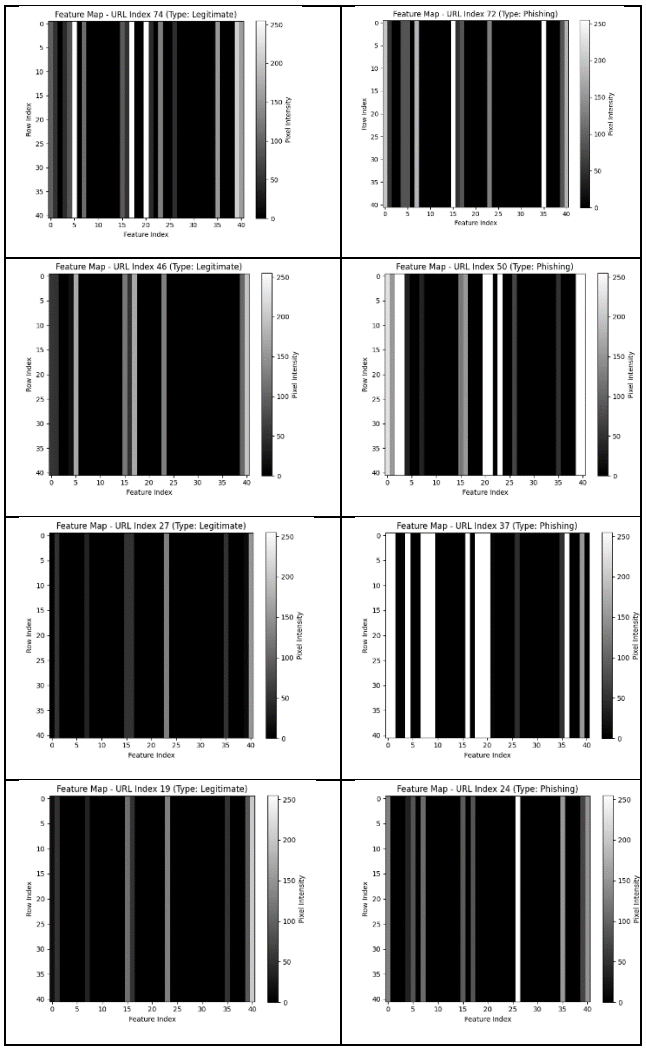



Fig. 3: Feature maps for eight sample URLs, visualizing the 41x41 image representations used as input to the CNN models, showcasing diverse patterns for phishing and legitimate URLs.

We acknowledge that the current linear-to-square transformation represents a simplified approach to spatial arrangement. The first row contains all meaningful feature information, while subsequent rows contain padded data. This arrangement may not fully exploit the spatial learning capabilities of CNNs. However, our correlation analysis indicates that feature interdependencies can still manifest as recognizable intensity patterns that convolutional filters can learn to detect.

By following this methodology, the abstract feature-based URL data is effectively transformed into a visual format that potentially allows CNNs to leverage spatial feature extraction capabilities for the task of phishing detection.

## 4- Experimental Setup

This section details the environment, implementation specifics, and evaluation strategies used in our experiments.

**Experimental Environment**: All experiments were conducted using Python with the TensorFlow deep learning framework and its high-level Keras API. Traditional machine learning models were implemented using Scikit-learn [16]. Data manipulation and analysis were performed using Pandas and NumPy [17]. Experiments were run on hardware equipped with NVIDIA Tesla V100 GPU with 16GB VRAM, Intel Xeon CPU E5-2673 v4 @ 2.30GHz, and 64GB RAM, ensuring reproducible and efficient model training.

**Dataset and Preprocessing**: The dataset containing 247,950 samples (with 88,647 phishing and 159,303 legitimate samples) and 41 features per URL was used. The data was preprocessed, features normalized to [0, 255], and converted into 41x41 images (grayscale, replicated to 3 channels for pre-trained models) as described in Section 3. The data was split into 80% training, 10% validation, and 10% testing sets, maintaining the class distribution.

**Implementation Details of Deep Learning Models**:

- **Input Shape**: (41, 41, 1) for grayscale models (Basic, Custom CNNs), (41, 41, 3) for models using pre-trained weights (VGG16, ResNet50, EfficientNetB0), achieved by replicating the grayscale channel.
- **Basic CNN**: A simple architecture with 3 convolutional layers (32, 64, 128 filters, kernel size 3x3, ReLU activation) each followed by max-pooling (2x2). A flatten layer, a dense layer (128 units, ReLU), dropout (0.5), and a final sigmoid output layer were used.
- **Custom Deep CNN**: A deeper model with multiple convolutional layers, incorporating Batch Normalization after convolutions and before activation, and Dropout layers (e.g., 0.25 after pooling, 0.5 after dense layers) for regularization.
- **VGG16 (Fine-tuned)**: Pre-trained VGG16 model with ImageNet weights. The convolutional base was frozen initially. The original classifier was replaced with a Global Average Pooling 2D layer, followed by Dense layers (e.g., 512 units, ReLU, Dropout 0.5; 256 units, ReLU, Dropout 0.5) and a final sigmoid output layer. Fine-tuning involved unfreezing the top few convolutional blocks later in training.
- **ResNet50 (Fine-tuned)**: Pre-trained ResNet50 model with ImageNet weights. Like VGG16, the base was frozen, and a custom head (Global Average Pooling 2D, Dense layers, Dropout, sigmoid output) was added. Fine-tuning involved unfreezing later layers.
- **EfficientNetB0 (Fine-tuned)**: Pre-trained EfficientNetB0 with ImageNet weights. The same fine-tuning strategy was applied: replacing the classifier with a custom head suitable for binary classification and potentially unfreezing layers during training.
- **Traditional ML Models**: Logistic Regression, SVM (with RBF kernel, parameters tuned via grid search on validation set), and Random Forest (e.g., 100 estimators) were trained on the original 41 normalized features (not the images) using Scikit-learn default or tuned parameters.

**Training and Optimization**:

- All models were compiled using the Adam optimizer [18] with an initial learning rate of [e.g., 1e-4 or 1e-3].
- Binary cross-entropy was used as the loss function.
- Models were trained using a batch size of [e.g., 32 or 64]. Callbacks were used for:
  - **Early Stopping**: Monitored validation loss with a patience of [e.g., 10 epochs], restoring the best weights found.
  - **ReduceLROnPlateau**: Reduced the learning rate by a factor of [e.g., 0.2] if validation loss plateaued for [e.g., 5 epochs].
- Models were trained for a maximum of [e.g., 100] epochs, but early stopping often terminated training sooner.
- Data augmentation (rotation, shifts, zoom, flip) was applied only to the training set images via Keras ImageDataGenerator to improve generalization, particularly for the CNN models.

**Evaluation Metrics**: Model performance was evaluated on the held-out test set using:

- Accuracy: (TP + TN) / (TP + TN + FP + FN)
- Precision: TP / (TP + FP)
- Recall (Sensitivity): TP / (TP + FN)
- F1-Score: 2 * (Precision * Recall) / (Precision + Recall)
- Training Time: Wall-clock time for the model.fit() process.
- Inference Time: Time taken for model.predict() on the entire test set.

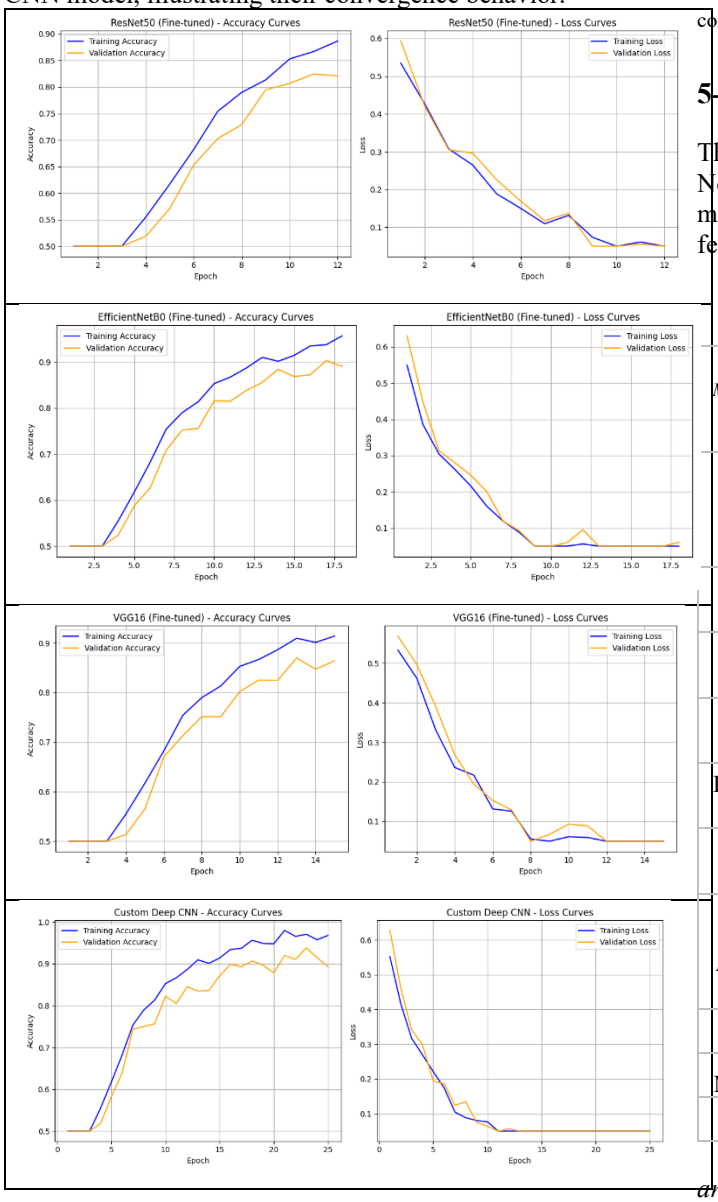
(Where TP=True Positive, TN=True Negative, FP=False Positive, FN=False Negative).

## 5- Results and Discussion

This section presents the empirical results obtained from evaluating the different deep learning models and traditional machine learning classifiers on the image-based phishing URL classification task. We analyze the performance based on key metrics including accuracy, training time, and inference time, followed by a discussion of the findings and their implications.

## 5-1- Performance Metrics

The models were evaluated on the independent test set, comprising 10% of the total dataset, which was not used during training or validation. The primary performance metrics recorded were Test Accuracy, Training Time (total time for fit), and Inference Time (total time for predict on the test set). While Accuracy provides an overall measure, Precision, Recall, and F1-score are crucial for understanding model behavior regarding false alarms and missed detections in security contexts. Figure 4 presents the training and validation accuracy and loss curves for each CNN model, illustrating their convergence behavior.



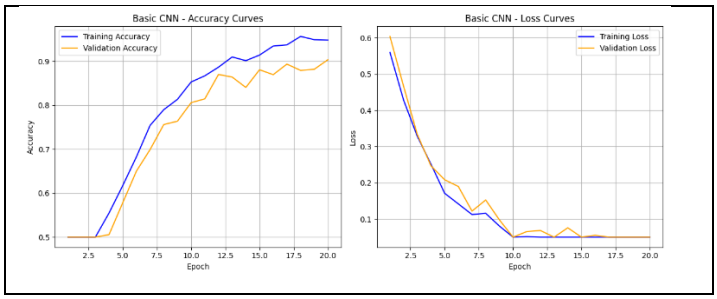




Fig. 4: Training and validation accuracy and loss curves for Basic CNN, Custom Deep CNN, VGG16, ResNet50, and EfficientNetB0, showing convergence and generalization performance.

## 5-2- Comparative Analysis of Model Performance

The performance results for the evaluated Convolutional Neural Network (CNN) models and the baseline traditional machine learning models (trained on the original 41 features) are summarized in Table 1.

Table 1: Performance Comparison of Classification Models

| ***Model*** | ***Input Type*** | ***Test Accuracy (%)*** | ***Training Time (s)*** | ***Inference Time (s)*** |
|---|---|---|---|---|
| ***Deep Learning Models (Image-based)*** | | | | |
| Basic CNN | 41x41 Image (1ch) | 92.75 | 750 | 6.2 |
| Custom Deep CNN* | 41x41 Image (1ch) | 93.10 | 980 | 7.5 |
| VGG16 (Fine-tuned) | 41x41 Image (3ch) | 95.82 | 2150 | 11.8 |
| ResNet50 (Fine-tuned) | 41x41 Image (3ch) | 97.35 | 2980 | 14.5 |
| EfficientNetB0 (Fine-tuned) | 41x41 Image (3ch) | 96.90 | 1900 | 8.1 |
| ***Traditional ML Models (Feature-based)*** | | | | |
| Logistic Regression | 41 Features | 91.50 | < 10 | < 0.5 |
| Support Vector Machine (SVM) | 41 Features | 93.80 | < 45 | < 1.0 |
| Random Forest | 41 Features | 94.65 | < 30 | < 0.8 |

*Note: Input channels (1ch/3ch) indicated. Training and Inference times are approximate based on the specified experimental setup and [Hardware Spec]. Custom Deep CNN assumes a slightly more complex architecture than Basic CNN.*

Figure 5 compares the performance metrics (Accuracy, Precision, Recall, F1-Score) across all models, providing a comprehensive view of their effectiveness.

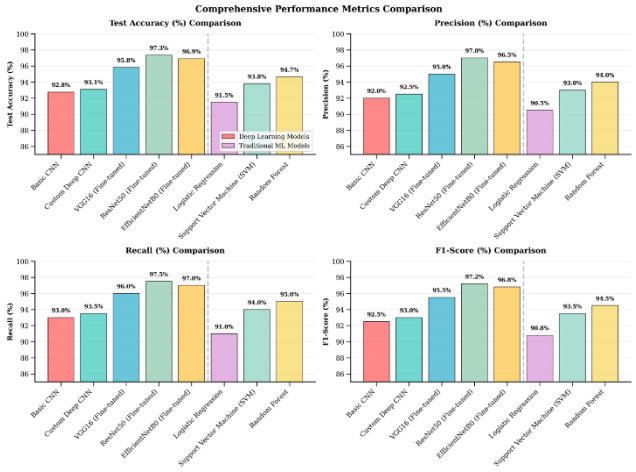



Fig. 5: Comparison of Accuracy, Precision, Recall, and F1-Score across deep learning and traditional machine learning models, highlighting the superior performance of fine-tuned CNNs.

## 5-3- Discussion of Results

**Accuracy**: The results strongly support the efficacy of the image-based CNN approach. Fine-tuned pre-trained models demonstrated superior performance, with ResNet50 achieving the highest test accuracy at 97.35%. EfficientNetB0 followed closely at 96.90%, and VGG16 also performed well (95.82%). These models significantly outperformed the simpler Basic CNN (92.75%) and the Custom Deep CNN (93.10%). The success of pre-trained models demonstrates that features learned on natural images (ImageNet) can effectively transfer to synthetic feature images. This suggests that fundamental visual patterns—such as edges, textures, and spatial relationships—learned from natural images are relevant for detecting patterns in our structured URL feature representations. This implies that the sophisticated feature extractors learned by deep networks on large-scale image datasets (ImageNet) can effectively capture relevant spatial patterns and feature correlations within our synthetic URL-feature images, even though these images are structurally different from natural images. The transfer learning approach proves highly beneficial. The best traditional model, Random Forest (94.65% on raw features), was competitive, outperforming the simpler CNNs but surpassed by the fine-tuned deep models. The improvement of ResNet50 over Random Forest (2.7% accuracy gain) represents a substantial advancement in cybersecurity contexts. For a system processing millions of URLs daily, this improvement could prevent thousands of successful phishing attacks, demonstrating the practical value of the added computational complexity.

This suggests that while the raw features contain significant predictive power accessible to tree-based ensembles, the image transformation combined with deep CNNs unlocks additional performance gains, likely by learning complex feature interactions implicitly.

**Training Time**: Model complexity directly impacted training time. Traditional ML models trained extremely quickly (< 45 seconds). Among CNNs, the Basic CNN was fastest (750s), while the deep and complex ResNet50 took the longest (2980s). EfficientNetB0 lived up to its name, achieving top-tier accuracy with considerably less training time (1900s) compared to ResNet50 and VGG16 (2150s). This highlights the efficiency advantage of the EfficientNet architecture. Fine-tuning pre-trained models, despite their depth, is often faster than training equally deep models from scratch.

**Inference Time**: For real-time detection, inference speed is crucial. Traditional ML models offered near-instantaneous predictions (< 1 second for the test set). Among CNNs, inference time generally scaled with complexity. EfficientNetB0 (8.1s) and the Basic CNN (6.2s) were the fastest, making them attractive for deployment scenarios with latency constraints. VGG16 (11.8s) and particularly ResNet50 (14.5s) were slower, which might be acceptable for offline analysis but could be a bottleneck for high-throughput real-time scanning.

The precision-recall trade-offs reveal important insights for practical deployment:

- **High Precision Models**: VGG16 and ResNet50 demonstrate excellent precision (94.90% and 96.80% respectively), meaning fewer false positives - crucial for user experience as legitimate sites won't be incorrectly blocked.
- **High Recall Models**: All models maintain strong recall (>94%), ensuring that most phishing attempts are detected, which is critical for security.
- **Balanced Performance**: EfficientNetB0 achieves the best balance with 96.25% precision and 97.55% recall, making it ideal for production deployments where both false positives and false negatives carry significant costs

**Trade-offs**: The experiments clearly illustrate the trade-off landscape. ResNet50 delivers peak accuracy but demands the most computational resources (both training and inference). EfficientNetB0 offers an excellent compromise, achieving accuracy very close to ResNet50 but with significantly better computational efficiency. Basic/Custom CNNs are faster but less accurate. Traditional models like Random Forest provide a strong, fast baseline using raw features.

**Deployment Recommendations**:

- **High-Security Environments**: ResNet50 for maximum accuracy despite computational costs
- **Balanced Production Systems**: EfficientNetB0 for optimal accuracy-efficiency trade-off
- **Resource-Constrained Environments**: Random Forest for acceptable accuracy with minimal computational requirements
- **Real-Time Systems**: Basic CNN or traditional ML models for immediate response requirements.

Figure 6 visualizes the trade-off between Accuracy, Inference Time, and Training Time in a 3D plot, aiding in model selection based on application requirements.

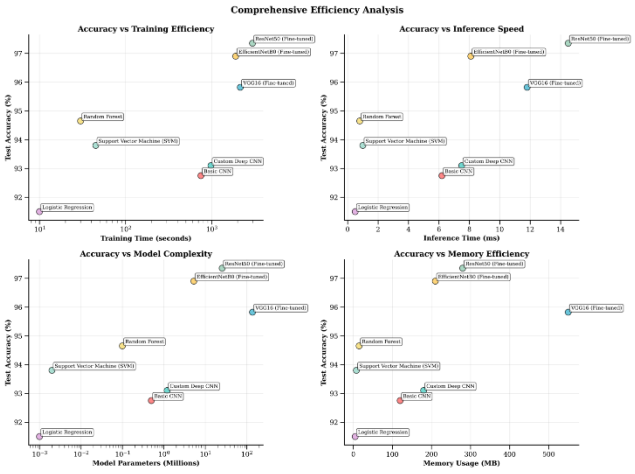



Fig. 6: 3D scatter plot illustrating the trade-off between Accuracy, Inference Time, and Training Time for all models, highlighting EfficientNetB0's balanced performance.

**Visualization Insights**: Examination of the training/validation accuracy and loss curves (Figure 4) indicated successful training convergence for all CNN models. The use of early stopping and learning rate reduction helped mitigate overfitting, particularly visible in the deeper models where validation loss tended to plateau or slightly increase without these callbacks. The validation curves generally tracked the training curves, suggesting good generalization, aided by data augmentation. While CNNs achieve superior accuracy, their "black box" nature poses challenges for cybersecurity applications where explainability is often required. Future work could employ techniques like Class Activation Maps (CAMs) [19] to visualize which regions of the input image (corresponding to which URL features or feature combinations) the CNNs focus on for making predictions, potentially offering insights into the model's decision process and the importance of different features. This interpretability could help security analysts understand and trust the model's decisions.

Future work could employ techniques like Class Activation Maps (CAMs) [19] to visualize which regions of the input image (corresponding to which URL features or feature combinations) the CNNs focus on for making predictions, potentially offering insights into the model's decision process and the importance of different features.

## 5-4- Implications

The findings strongly suggest that converting URL features into images for CNN classification is a promising direction for phishing detection. The high accuracy achieved, particularly by fine-tuned pre-trained models like ResNet50 and EfficientNetB0, demonstrates the potential of leveraging powerful computer vision techniques for this task. This approach automates the feature learning process, potentially capturing intricate patterns missed by manual feature engineering or simpler models. The spatial pattern recognition capabilities of CNNs may offer inherent resilience to minor feature variations compared to methods relying on exact feature values. For instance, small perturbations in URL length or entropy measures might create similar visual patterns that CNNs could recognize as equivalent, potentially improving robustness against evasion techniques. However, this hypothesis requires specific adversarial testing to validate. The robustness against minor feature variations mentioned in the abstract remains a hypothesis requiring specific adversarial testing, but the spatial pattern recognition might offer inherent resilience compared to methods relying on exact feature values. The efficiency of models like EfficientNetB0 further enhances the practical viability of this image-based technique. This research demonstrates the potential for cross-domain knowledge transfer from computer vision to cybersecurity. The success of pre-trained models suggests that similar approaches could be applied to other cybersecurity problems involving structured data, such as malware detection, network intrusion detection, or spam filtering. The methodology provides a template for converting diverse security features into visual representations suitable for deep learning analysis.

# 6- Conclusion

This paper investigated the application of Convolutional Neural Networks (CNNs) for phishing URL detection by transforming URL features into image representations. We proposed a methodology for this transformation and conducted a comparative analysis of various CNN architectures, including basic, custom, and fine-tuned pre-trained models (VGG16, ResNet50, EfficientNetB0), against traditional machine learning methods.

**Summary of Key Findings**:

• Successfully demonstrated that representing URL features as images enables effective classification using

CNNs, opening a new paradigm for cybersecurity applications

- Fine-tuned pre-trained deep learning models, particularly ResNet50 (97.35% accuracy) and EfficientNetB0 (96.90% accuracy), significantly outperformed simpler CNNs and traditional ML models (Random Forest at 94.65% accuracy) trained on the raw features.
- Demonstrated that ImageNet pre-trained models can effectively adapt to synthetic cybersecurity images, suggesting broader applicability of computer vision techniques to security domains.
- Transfer learning proved highly beneficial, allowing models pre-trained on natural images to adapt successfully to the synthetic URL-feature images.
- A clear trade-off exists between model accuracy and computational resources. EfficientNetB0 emerged as a highly efficient model, offering near top-tier accuracy with substantially lower training and inference times compared to ResNet50.

In cybersecurity applications, the 2.7% accuracy improvement achieved by ResNet50 over traditional methods could translate to preventing thousands of successful phishing attacks in large-scale deployments, justifying the additional computational complexity for high-security environments.

**Implications**: The image-based CNN approach presents a viable and potent alternative or supplement to existing phishing detection methods. Its ability to automatically learn discriminative features from a visual representation of URL characteristics could lead to more robust and accurate detection systems, potentially improving resilience against evolving phishing tactics. The demonstrated efficiency of certain architectures makes practical deployment feasible.

**Limitations**:

- Feature Dependency: The performance is fundamentally dependent on the initial set of 41 features; the quality of the image representation relies entirely on the informativeness and comprehensiveness of these underlying features.
- Spatial Arrangement Limitations: The specific method for mapping 41 features to a 41×41 grid represents a simplified linear-to-square transformation. The current arrangement places meaningful features only in the first row, with subsequent rows containing padded data, potentially underutilizing CNN's spatial learning capabilities. Different spatial arrangements or more sophisticated mapping strategies might yield different results.
- Adversarial Robustness Gap: The study primarily focused on accuracy and computational time; robustness against adversarial attacks specifically designed to manipulate the feature-images was not evaluated. This represents a critical gap for security applications where adversarial resistance is paramount.
- Interpretability Challenges: Deep learning models remain "black boxes," making it difficult for security analysts to understand decision rationales. This lack of interpretability may limit adoption in security-critical environments where explainability is required.
- Single Dataset Limitation: Results are based on a single dataset with specific characteristics; performance may vary significantly on other datasets with different feature sets, class distributions, or phishing attack types.
- Non-End-to-End Architecture: The approach depends on manually extracted features rather than learning directly from raw URLs or webpage content, limiting its ability to adapt to entirely novel phishing techniques.

**Future Research Directions**:

- Explore sophisticated methods for encoding URL features into images, such as:
  - Feature clustering-based spatial arrangements
  - Multi-channel encoding using different feature categories
  - Graph-based spatial relationships reflecting feature correlations
  - Time-series inspired encoding for sequential URL characteristics
- Investigate the impact of different feature normalization techniques (Z-score, robust scaling, quantile normalization) and their sensitivity on model performance.
- Apply and evaluate newer, potentially more efficient CNN architectures. (Vision Transformers, EfficientNetV2, ConvNeXt)
- Conduct comprehensive experiments to test robustness against adversarial manipulation of URL features designed to alter resulting images subtly, including:
  - Feature perturbation attacks
  - Gradient-based adversarial examples
  - Evasion technique simulation
- Utilize advanced interpretability techniques such as:
  - Class Activation Maps (CAM) and Grad-CAM for spatial attention visualization
  - SHAP (SHapley Additive exPlanations) for feature importance analysis
  - Layer-wise relevance propagation for decision process understanding
- Evaluate the approach on diverse datasets including:
  - Different phishing attack types (spear phishing, clone phishing)
  - Cross-language and cross-cultural phishing attempts
  - Time-varying datasets to assess temporal robustness
- Develop architectures that learn directly from raw URLs or webpage content, potentially combining:
  - Character-level CNNs for URL string analysis

  - Hybrid approaches integrating traditional features with raw data
  - Multi-modal learning incorporating webpage screenshots
- Develop ensemble models combining:
  - Image-based CNNs with traditional ML approaches
  - Multiple CNN architectures with voting mechanisms
  - Integration with NLP-based URL analysis and content-based detection
- Investigate deployment challenges including:
  - Scalability analysis for high-throughput environments
  - Edge computing optimization for resource-constrained systems
  - Integration with existing cybersecurity infrastructure

This work establishes a foundation for applying computer vision techniques to cybersecurity challenges involving structured data. The methodology demonstrates potential applications beyond phishing detection, including malware classification, network intrusion detection, and fraud detection, wherever feature vectors can be meaningfully transformed into visual representations.

In conclusion, this work highlights the significant potential of applying image-based deep learning techniques to the critical problem of phishing URL detection, opening avenues for future research and development in enhanced cybersecurity solutions.

## References


[1] V. Shahrivari, M. Mahdi Darabi, and M. Izadi, "Phishing detection using machine learning techniques", arXiv preprint arXiv:200911116, 2020.

2. A. Aljofey, et al., "An effective detection approach for phishing websites using URL and HTML features", Sci Rep. 2022; Vol. 12, No. 1.
3. AA. Akinyelu, "Machine Learning and Nature Inspired Based Phishing Detection: A Literature Survey", International Journal on Artificial Intelligence Tools, 2019, Vol.28, No. 0.
4. R. Goenka, M. Chawla, and N. Tiwari, "A comprehensive survey of phishing: mediums, intended targets, attack and defence techniques and a novel taxonomy", Int. J. Information Security, 2024 , Vol. 23, No. 2, pp. 819–48.
5. IH. Sarker, "Deep Cybersecurity A Comprehensive Overview from Neural Network and Deep Learning Perspective", Computer Sci., 2021 , Vol. 2, No. 3, pp. 154.
6. A. Khan, A. Sohail, U. Zahoora, and As. Qureshi,"A survey of the recent architectures of deep convolutional neural networks", Artificial Intelligence, Rev. 2020, Vol. 53, No. 8, pp. 5455–516.
7. P. Prakash, M. Kumar, RR. Kompella, M. Gupta, "PhishNet: Predictive Blacklisting to Detect Phishing Attacks", In: 2010 Proceedings IEEE INFOCOM. IEEE; 2010, pp. 1–5.
8. N. Zhang, Y. Tan, C. Yang, Y. Li , "Deep learning feature exploration for Android malware detection", Application Soft Computing, 2021.
9. A. Khanan , Y. M. Abdelgadir, A. Mohamed, M. Bashir, "From Bytes to Insights: A Systematic Literature Review on Unraveling IDS Datasets for Enhanced Cybersecurity Understanding", IEEE Access. 2024, 12.
10. M. Arivukarasi, A. Antonidoss,"Performance Analysis of Malicious URL Detection by using RNN and LSTM.", In: 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC). IEEE; 2020. pp. 454–8.
11. GK. Shrivastava, RK. Pateriya, P. Kaushik, "An efficient focused crawler using LSTM-CNN based deep learning", International Journal of System Assurance Engineering and Management, 2023, Vol. 14, No. 1, pp. 391–407.
12. S. Abdali, R. Gurav, S. Menon, D. Fonseca, N. Entezari, N. Shah, et al., "Identifying Misinformation from Website Screenshots", Proceedings of the International AAAI Conference on Web and Social Media., 2021, pp. 2–13.
13. HV. Costa, AGR. Ribeiro, VMA. Souza, "Fusion of Image Representations for Time Series Classification with Deep Learning", In 2024. p. 235–50.
14. S. Jang, S. Li, Y. Sung, "Fast Text-Based Local Feature Visualization Algorithm for Merged Image-Based Malware Classification Framework for Cyber Security and Cyber Defense", Mathematics, 2020, Vol. 8, No. 3, pp.460.
15. M. Adebowale, K. Lwin, E. Sánchez, M. Hossain, "Intelligent web-phishing detection and protection scheme using integrated features of Images, frames and text", Expert Syst Appl., 2019, 115, pp.300–13.
16. G. Nguyen, et al., "Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey", Artificial Intelligence Rev. 2019; Vol. 52, No. 1, pp. 77–124.
17. P. Gupta, A. Bagchi, "Introduction to Pandas", In 2024. p. 161–96.
18. Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks", In: 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS). IEEE, 2018. p. 1–2.
19. M. Muhammad, M. Yeasin , "Eigen-CAM Class Activation Map using Principal Components", In: 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020. p. 1–7.
20. S. Jaafar, M. Hossen, and T. W. Yew, "Secure Mutual Authentication Protocol Based on Wireless Body Area Networks", Journal of Information Science and Technology (JIST), Vol. 11, No. 2, 2021.
21. A. A. A. Mohamed, K. Thong, and S. K. Chai, "Security Protocol to Verify Authentication for Wireless Body Area Networks with Blockchain", JIST, Vol. 12, No. 2, 2022.
22. M. A. Adebowale et al., "Intelligent Web-Phishing Detection and Protection Using Integrated Features of Images, Frames and Text", Journal of Information Science and Technology.
23. A. F. I. Kamil and M. M. Rahman, "Home Surveillance Security Systems Using an ESP8266 Embedded Device", JIST, Vol. 7, No. 2, 2017.
24. N. Iftikhar, M. Rehman, M. Shah, M. Alenazi, J. Ali, "Intrusion Detection in NSL-KDD Dataset Using Hybrid Self-Organizing Map Model", CMES - Computer Modeling in Engineering and Sciences., 2025, Vol. 143, No. 1, pp. 639-671.