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.

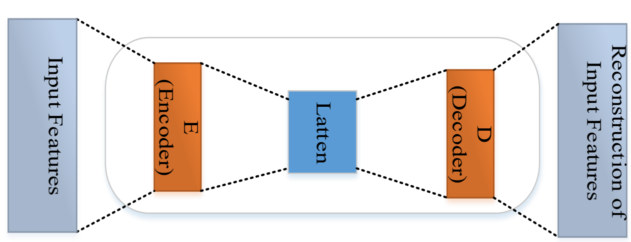

Fig 1. The basic structure of autoencoder

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.

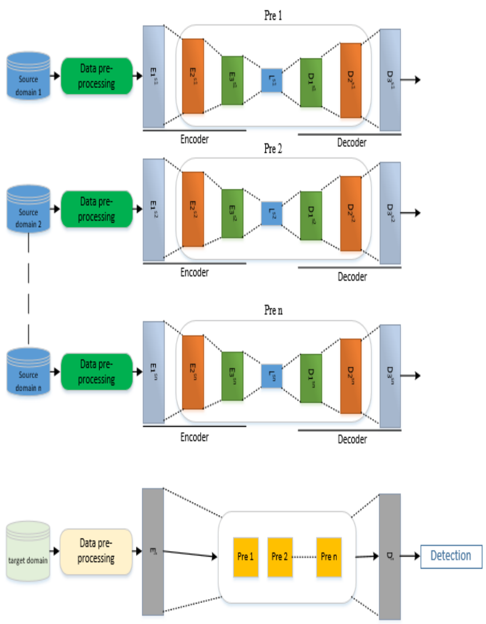

Fig 1. The architecture of CaFFM

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.

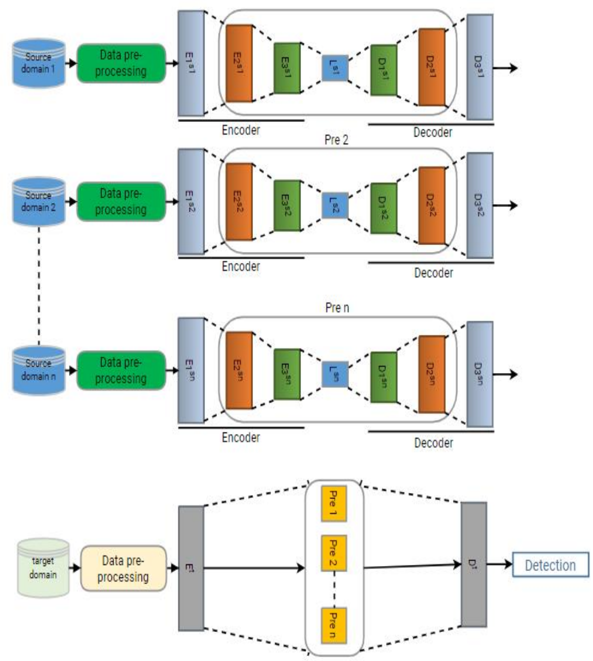

Fig 2. The architecture of the CoFFM

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.

.

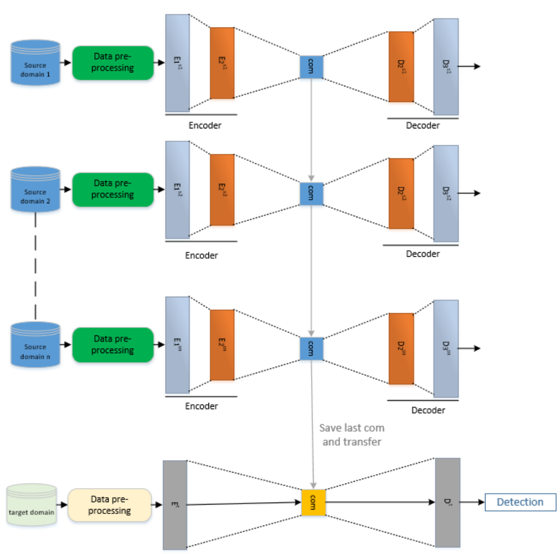

Fig 3. The architecture of UFFM

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

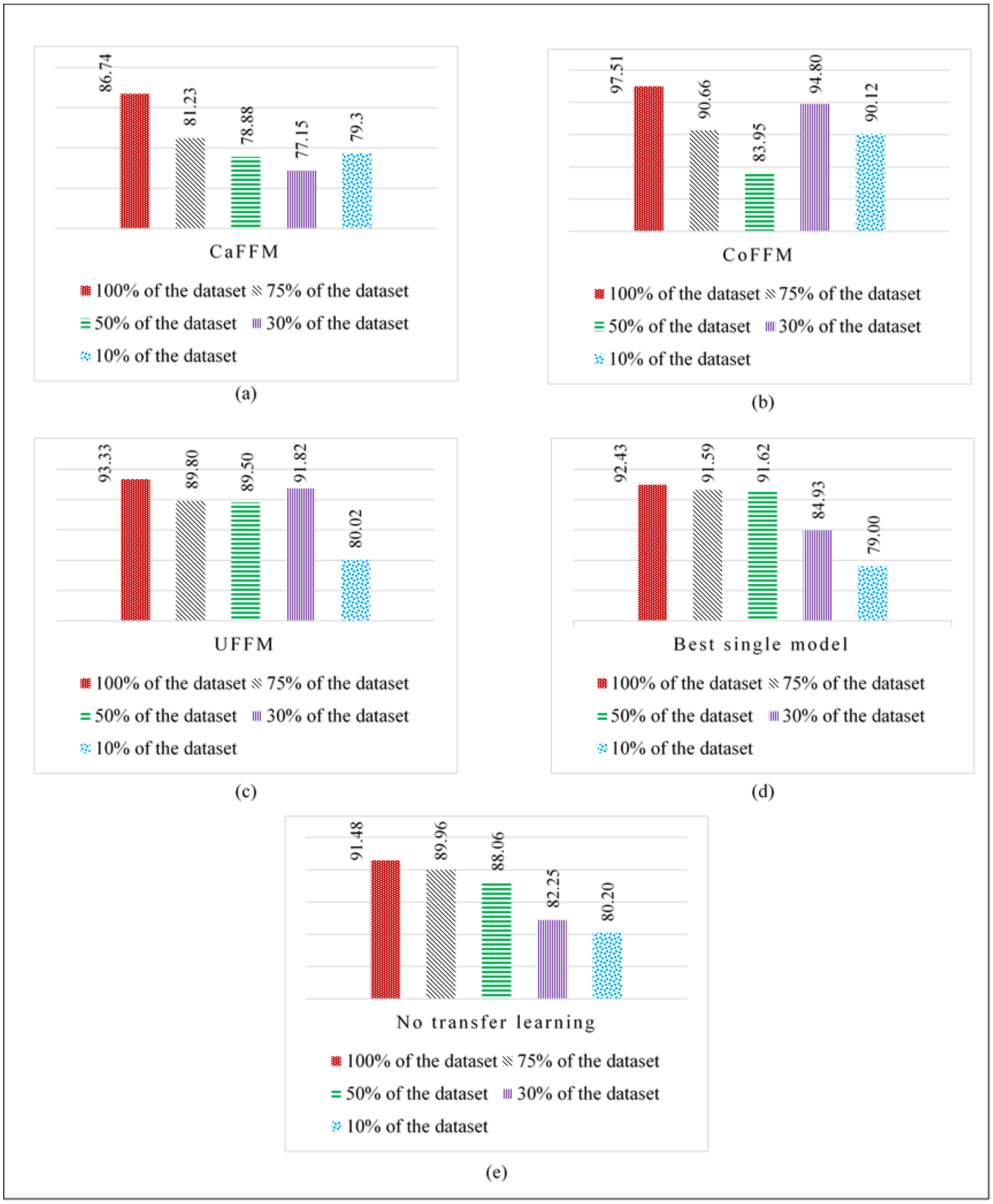

Fig 6. Precision rates of the proposed methods on the IDSAI dataset for detecting unknown attack

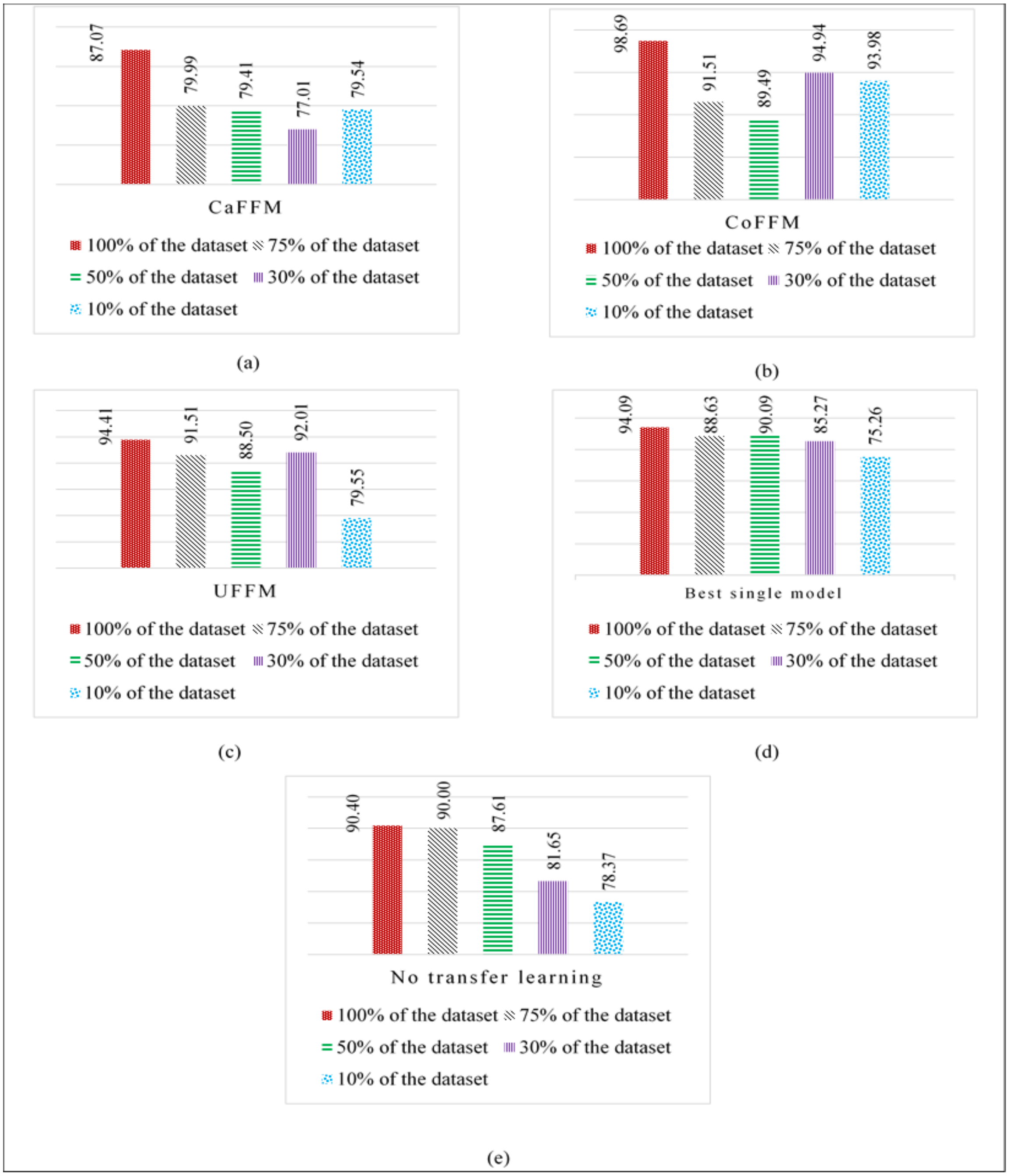

Fig 7. Recall rates of the proposed methods on the IDSAI dataset for detecting unknown attacks

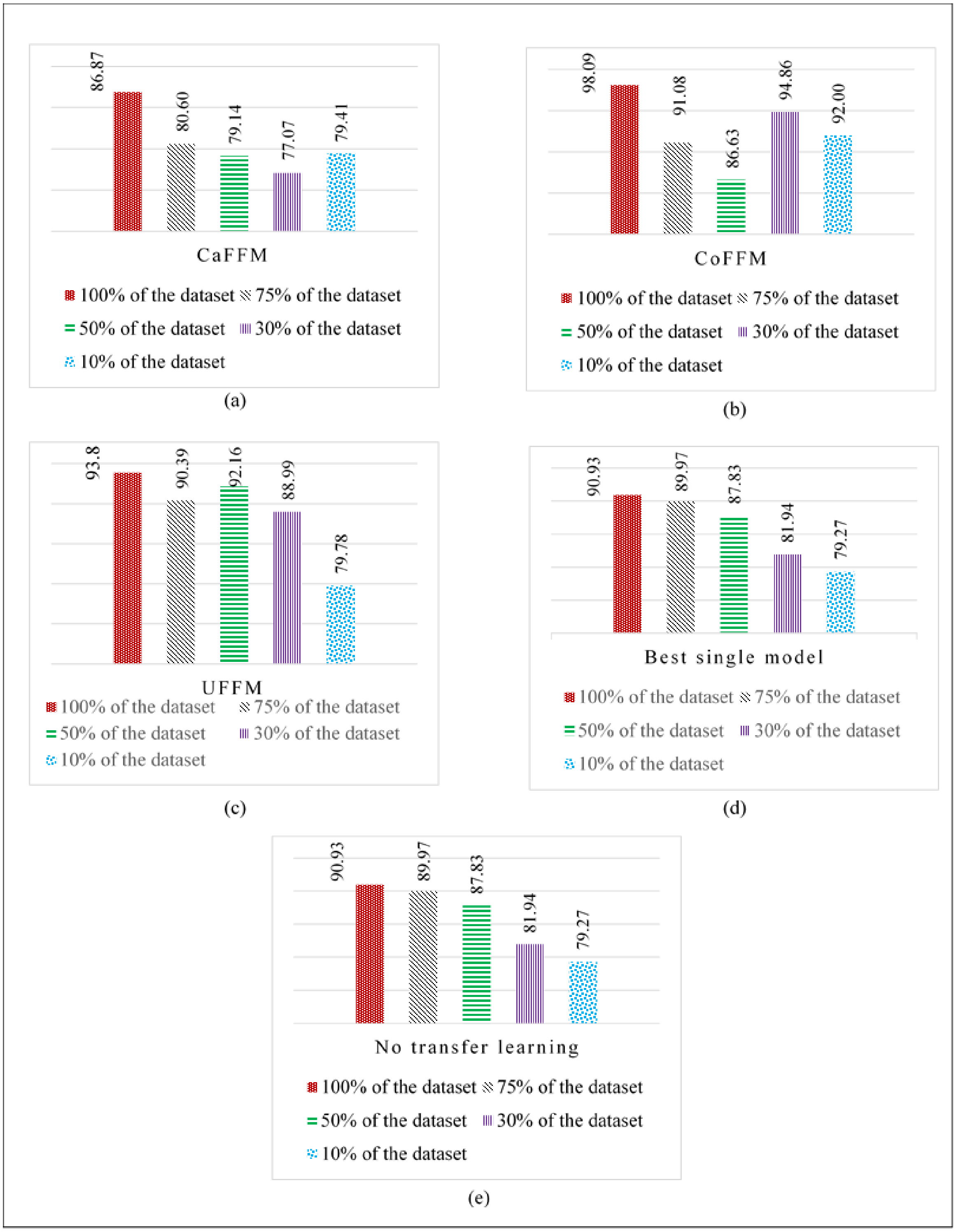

Fig 8. F-score rates of the proposed methods on the IDSAI dataset for detecting unknown attacks

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.