HSL-DCRNet: Hybrid Sequential-Local Deep CNN-RNN Feature Extractor Network for Efficient Intrusion Detection in IoT Network

Abdullah Makki Jebur, Murtadha Talib Abbas, and Bashaer Makki Jebur

Original scientific article

***Abstract*—Protecting Internet of Things (IoT) networks necessitates intrusion detection systems (IDS) capable of accurately identifying both temporal behaviors and structural characteristics of malicious traffic. This paper proposes HSL-DCRNet, a Hybrid Sequential Local CNN RNN Feature Extractor Network, to address this challenge. The model employs a Gated Recurrent Unit (GRU) to learn sequential dependencies in traffic flows and a Convolutional Neural Network (CNN) with Inception blocks to extract multi-scale structural features. Their outputs are fused into a unified latent space, and Maximum Relevance Minimum Redundancy (MRMR) feature selection is applied to enhance discriminative power while reducing redundancy. Classification is performed using Enhanced Dense Layers with Adaptive Learning Rate Optimizer (EDL-ALRO), enabling parameter-specific learning rates and faster convergence. Experiments on the UNSW-NB15 dataset show that HSL-DCRNet achieves 99.82% accuracy, surpassing existing IDS approaches. The results confirm its robustness and scalability for securing IoT environments.**

***Index terms*—Intrusion Detection, Convolutional Neural Network, Recurrent Neural Network, Adaptive Learning Rate, Internet of Things.**

I. Introduction

THE Internet of Things (IoT) has been spreading and has transformed the contemporary digital ecosystems by providing pervasive connectivity to various disciplines [1]. IoT offers innovation and efficiency opportunities never seen before by combining billions of heterogeneous devices. Nonetheless, large-scale interconnectivity is also a source of increased attack space, which makes IoT infrastructures extremely susceptible to malicious use [2]. Denial-of-service (DoS), data exfiltration, reconnaissance, and advanced persistent threats are some of the attacks that are increasingly affecting the IoT networks, which can lead to serious disruption, loss of finances, and sensitive information [3, 4].

The nature of the IoT environment poses a specific challenge to securing them, with limited resources available to devices, protocols that are heterogeneous, and dynamic traffic patterns. The effectiveness of traditional intrusion detection systems (IDS), especially signature-based systems is still useful in entering familiar threats but not zero-day and polymorphic attacks, which are mostly ineffective [5]. Anomaly-based IDS solutions offer greater coverage but are often characterised by a large rate of false-positive results, which invalidates their use in practical applications [6]. The challenges highlight why scalable, intelligent, and adaptive intrusion detection systems tailored to the unique properties of the IoT environments need to be created [7].

Machine learning has proved to be an effective intrusion detection tool that can produce data-driven models with the capability of learning sophisticated attack patterns [8]. Ensemble-based techniques like the Random Forests and Support Vector Machines have proven to be reasonably accurate at detecting metrics in benchmark datasets, but fail to scale in high-dimensional space and do not tend to capture time or context-related dependencies in traffic flows. In addition, they cannot be flexible to changes in attack vectors because of their dependence on handcrafted features [9].

Recent developments in deep learning have allowed improving the performance of intrusion detection significantly [10]. Recurrent Neural Networks (RNNs) are effective in temporal sequence modeling such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), which are useful in the analysis of traffic patterns through time [11]. In the meantime, Convolutional Neural Networks (CNNs) have been shown to be extremely efficient in the process of extracting spatial and structural attributes out of raw traffic data. Hybrid CNN-RNNs have been of interest due to their complementary abilities, where they have better accuracy than single models. Still, there are problems: overfitting issues such as redundant and noisy features impair the ability to generalize, the imbalance in classes within intrusion datasets impairs the ability

Manuscript received November 17, 2025; revised December 14, 2025. Date of publication April 8, 2026. Date of current version April 8, 2026. The associate editor prof. Hrvoje Karna has been coordinating the review of this manuscript and approved it for publication.

A. M. Jebur and M. T. Abbas are with the Department of Electronic and Communications Engineering, University of Kufa, Najaf, Iraq (e-mails: abdullahm.algburi@uokufa.edu.iq, Murtadhat.mullayousif@uokufa.edu.iq).

B. M. Jebur is with the Department of Electronic and Communications Engineering, Al-Muthanna University, Samawah, Iraq (e-mail: bashaer.makki@mu.edu.iq).

Digital Object Identifier (DOI): 10.24138/jcomss-2025-0228

to learn, and fixed learning rate optimizers tend to slow down learning in deep architectures [12, 13]. This paper introduces a new architecture, called Hybrid Sequential Local Deep CNN-RNN Feature Extractor Network (HSL-DCRNet), that will help in the effective identification of intrusion in the Internet of Things (IoT) networks. A Gated Recurrent Unit (GRU) will be used in the proposed approach to learn the short and long-term temporal dependencies, thus successfully leveraging the dynamism of network traffic flows. At the same time, a Convolutional Neural Network (CNN) with an Inception block partly uses multi-scale filters in finding local and structural patterns related to malicious behaviors. The results of the two parts are combined into a single feature vector, thus, enhancing the representation space by means of localized structural patterns as well as time sequences. In order to extend a further use of discriminative power, an algorithm known as Maximum Relevance Minimum Redundancy (MRMR) algorithm is employed to select the features, which is applied to eliminate redundant features and adopt only the most informative features, and this is an important boost to the power of the model to learn. Lastly, an Enhanced Dense Layer with Adaptive Learning Rate Optimizer (EDL-ALRO) is included to enhance the classification process, in which the adaptive optimization of the learning rate allows the process to converge much faster and more precisely predicts. Altogether, the proposed model is comprehensive, with the combination of both time- and location-specific feature extraction, optimal feature selection, and smart classification, it offers a powerful and very precise framework of intrusion detection in IoT networks.

The major contributions of this work include:

- Introduction of a novel hybrid framework (HSL-DCRNet):

A hybrid architecture is designed that integrates an Inception-enhanced CNN with a GRU, enabling simultaneous extraction of multi-scale local features and sequential temporal patterns from network traffic data, thereby offering a comprehensive representation of attack behaviors.

- Enhanced feature representation via multi-level fusion:

A feature fusion mechanism is developed to integrate the representations obtained from the CNN and GRU into a shared latent space. This integration strengthens inter-class separability and improves the model's ability to capture complex intrusion patterns.

- Optimal feature selection using the MRMR algorithm:

To enhance classification accuracy while reducing computational overhead, feature selection is performed using the Maximum Relevance Minimum Redundancy (MRMR) approach. By leveraging mutual information as a criterion for both relevance to class labels and redundancy among features, the method identifies an optimal subset of features. This approach helps prevent overfitting while enhancing the model's robustness and ability to generalize when differentiating between normal and malicious IoT traffic.

- Utilization of Enhanced Dense Layers with Adaptive Learning Rate Optimizer (EDL-ALRO) for effective classification:

A novel mechanism, EDL-ALRO, is introduced to optimize the classification process. By adaptively tuning parameter updates based on the mean and variance of historical gradients, EDL-ALRO enables parameter-specific learning rates. This refinement results in faster convergence and superior detection accuracy in network attack classification tasks.

The remainder of this paper is organized as follows. Section II presents a comprehensive review of the relevant literature, highlighting key theoretical frameworks and prior empirical findings that inform the current study. Section III details the methodology employed, including the research design. Section IV reports the results and provides a critical discussion of the findings in relation to the existing body of knowledge.

II. Literature Review

According to [14], the IoT devices are considered prone to cyber threats due to their limited computational and storage and a lack of standardization. Conventional security mechanisms that include use of cryptography algorithms and firewalls cannot be used to confront these challenges. In order to enhance the security of IoT, scholars have considered deep learning-based intrusion detection systems, but the weaknesses of most of the current methods include poor generalization and weak features extraction. To overcome these limitations, the authors propose a temporal convolutional residual model with an attention mechanism, achieving superior accuracy on ToN_IoT and UNSW-NB15 datasets compared to current state-of-the-art methods.

In [15], cyberattacks, particularly DDoS attacks, are identified as major threats to IoT networks, often disrupting communication traffic and draining sensor node energy. Detecting such attacks is crucial, but traditional methods struggle with analyzing the vast network traffic generated by numerous IoT devices. To solve this, a hybrid deep learning architecture that uses 1D-CNN and LSTM are constructed and trained on benchmark data. The approach attains excellent performance.

The development of IoT is pointed out in [16] as a key factor of connectivity and automation but also a cause of critical security concerns. The conventional intrusion detection systems are usually not able to handle the dynamic and multi-faceted nature of the IoT threats. Possibly in response, the authors perform a systematic review on Deep Reinforcement Learning (DRL) as an IoT security approach. They conclude that DRL significantly increases the adaptability of IDS and boosts the detection performance and minimizes the number of false positives. Others that are found to pose open challenges in the review are diversity of the datasets, their reproducibility, and the integration with the emerging IoT technologies.

The IoT devices are demonstrated in [17] to be extremely prone to advanced cyberattacks, and effective security measures, including intrusion detection systems (IDS), are required. To manage those issues, the authors offer a hybridization of deep learning methods, which can be seen as a combination of convolutional neural networks (CNN) and long short-term memory (LSTM) models. The combination will use the capabilities of CNN as a spatial feature extractor and LSTM as a temporal dependencies capturer to effectively classify IoT traffic as benign or malicious activity. The model was tested on the CICIoT2023 and CICIDS2017 datasets and presented an

accuracy of 98.42, and a high F1 score, indicating its capability to ensure the safety of an IoT environment.

In [18], the rapid expansion of IoT and changing vectors of attacks are pointed out as the significant security issues, and the centralized AI-based intrusion detection systems (IDSs) have constraints in scaling and privacy. In order to meet these challenges, the authors study a decentralised approach based on Federated Learning (FL), which allows to train jointly without violating data confidentiality. Their experiments take place on the realistic ToN-IoT data set, and associate each IP address with an individual FL client and look at pre-training and different aggregation strategies. Results show that data heterogeneity has a negative impact on performance but by using a pre-trained global model for initialisation, we can improve the detection by more than 20% in F1-score.

In [19], the growth of IoT in smart cities is mentioned, which allows remote monitoring, control and real-time data analysis on one hand, but also makes them more vulnerable to cyberattacks. To improve security, the authors suggest an intrusion detection system (IDS) using ensemble learning consisting of AdaBoost, Boruta feature selection, and XGBoost methods. The model is tested on the NSL-KDD and BoT-IoT datasets, showing a better performance than available IDS methods. The findings demonstrate that there are high accuracy, recall, and F1-scores, and that it has efficient computing time, which means that it can be used to ensure the security of IoT-based smart environments.

In [20], the conventional approaches to network intrusion detection are pointed out as incapable of managing intricate attack schemes. To counter this, the authors suggest a deep learning-based detection model based on automatic feature extraction and nonlinear modeling. The hybrid CNN-LSTM model is optimized based on KDD Cup 1999 and UNSW-NB15 datasets to optimize data preprocessing and minimal design. It has been experimentally found that our model performs better than the conventional machine learning, as well as the SVM models, which have proven to be highly effective in identifying hidden attacks.

According to [21], the influx of IoT, cloud computing, and smart devices is observed to produce more than 400 zettabytes of network traffic every year, which shows that it is necessary to use a robust cybersecurity approach. The study highlights the significance of using machine learning to improve the effectiveness of IDS, and it is recommended to reduce the number of false alarms and increase the accuracy. The authors test the logistic regression, SVM, decision tree, and random forest models in UNSW-NB15 data using the feature selection and exploratory data analysis. Findings suggest that the Random Forest model is the best to use, as it has the highest accuracy of 98.63% and F1 of 97.8% with the minimal false alarm of 1.36 hence the model is very effective in improving IDS.

Satellite-terrestrial integrated networks (STINs) are known in [22] to be susceptible to distinct security issues despite such benefits as high throughput, low latency and large coverage. The authors suggest four hybrid intrusion detectors (SAT-IDSs) based on sequential forward feature selection (SFS), machine learning, and deep learning models (Random Forest (RF), LSTM, ANN, and GRU) to increase protection. Accuracy and efficiency were highly enhanced on the STIN dataset and UNSW-NB15 dataset with feature selection. SFS-RF and RF-SFS-GRU models had the best results in terms of detection and they have shown that considering key features is important in improving the effectiveness of an IDS in a satellite and a terrestrial network setup.

The weaknesses of the IoT networks to cyber attacks are overcome in [23] with the proposal of a dual feature optimization deep learning system known as FOUND. It uses Bald Eagle Search (BES) and Butterfly Optimization Algorithm (BOA) to extract both flow and packet-level features to enhance intrusion detection accuracy. To classify the attack and non-attack traffic, multi-head attention-based bidirectional GRU (MHA-BiGRU) is used. On BoT-IoT and UNSW-NB15 datasets, the FOUND method is shown to be more effective than current models, with improvements of 1.5, 1.1 and 2.5 per cent in accuracy over GRU, RNN and GCN methods, justifying its use as an effective method to enhance the security of IoT.

III. Methodology

This study primarily seeks to present a robust framework, HSL-DCRNet (Hybrid Sequential–Local Deep CNN–RNN Feature Extractor Network), designed to detect multiple categories of attacks within IoT networks. The motivation for adopting a hybrid GRU–CNN architecture stems from the intrinsic characteristics of IoT traffic, which exhibits both temporal dependencies and localized structural patterns. The CNN network is employed to extract local and spatial features from network traffic, while the GRU recurrent network is capable of capturing temporal dependencies and sequential behavioral patterns that unfold over time within the same traffic flow. In particular, the Gated Recurrent Unit (GRU) was selected instead of Long Short-Term Memory (LSTM) due to its ability to efficiently capture both short-term and long-term dependencies in sequential data, while avoiding some of the vanishing gradient issues that LSTM can encounter. GRU's simplified gating mechanism helps in reducing overfitting, stabilizing training, and improving generalization, especially when dealing with high-dimensional and complex network traffic data. Moreover, several studies have demonstrated that GRU can achieve comparable or even superior performance to LSTM in tasks such as network traffic modeling and intrusion detection, particularly in capturing temporal patterns critical for differentiating between attack types. The first step is to obtain raw data out of the dataset and then goes through a preprocessing phase involving cleaning the data, normalization, and preliminary preparation. Given the imbalance of attack types, a data balancing process is used. This aids the increase of diversity in the data set, it avoids domination of majority classes and thus leads to a better model performance which lessens overfitting and improves the accuracy of intrusion detection. After the preprocessing, features extraction commences and it draws on the hybrid combination of two strong neural network structures. A Gated Recurrent Unit (GRU) is applied to extract temporal and sequential patterns and it has proven to have a high ability in learning short-term and long-term dependency on network traffic samples, and thus successfully model the temporal dynamics of data flows. Simultaneously, a Convolutional Neural Network (CNN) with an Inception block obtains local and structural attack-related features. This block

makes use of multi-scale filters to create the richer representation and the fine-grained local patterns are identified accurately. GRU and CNN outputs are then converted to feature vectors; they are then combined into one representation vegetable. This combination allows the system to learn about temporal dependencies and local spatial structures, on a common space, which provide a complete description of intrusion behaviors. In order to minimize redundancy further, and project features into a lower dimensional and more discriminating space, Maximum Relevance Minimum Redundancy (MRMR) feature selection is used. This approach is an optimal choice since the concept of mutual information implies that the method chooses the most significant features and eliminates those that are less informative or correlated from the features, thus improving the efficiency and accuracy of the model. The feature vectors thus chosen are submitted to fully connected layers to undertake the classification process. We use Enhanced Dense Layers with Adaptive Learning Rate Optimizer (EDL-ALRO) to enhance the performance of the classifier to make a distinction between different types of attacks. The optimizer is a dynamic control over the learning rate of every parameter based on the average and standard deviation of the already calculated gradients. Consequently, every parameter receives its best learning rate, which allows obtaining a quicker convergence and a more precise attack classification. Overall, the proposed model combines the advantages of deep learning in the simultaneous extraction of both temporal and local features, optimal feature selection, and adaptive optimization; hence, providing a lightweight, precise, and efficient architecture to detect intrusion in the IoT. The overall architecture of the proposed framework is illustrated in Figure 1.

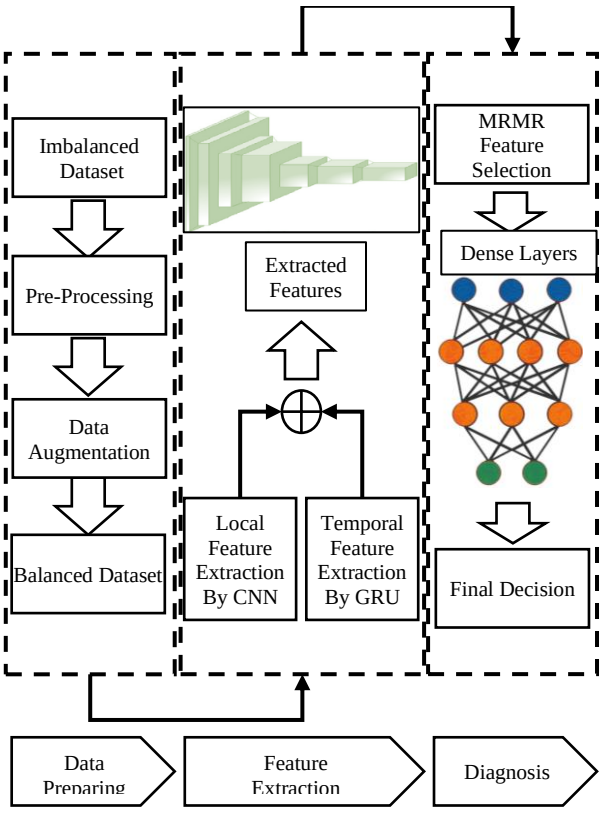


Fig. 1. Schematic diagram of the proposed method

A. Dataset and Data Preprocessing

In this work, the publicly available UNSW-NB15 data set, which can be accessed through the official website [24], was used in the intrusion detection in IoT networks. This data set is particularly aimed at testing and making intrusion detection models. It has 45 different features, which are the aspects of network traffic behavior. Each record is tagged as either normal traffic or a specific attack type and as such, the researchers can effectively train their models and evaluate performance in diverse attack scenarios. The dataset uses instances of several attack classes such as DoS, Fuzzers, Reconnaissance, Backdoor, Exploits, Shellcode, Worms, Generic and Analysis and one Normal traffic class. Table I gives the distribution of records in these different categories. As shown, the dataset contains a total of 257,673 records from 10 classes (one Normal and nine types of attack), which is the basis for multi-class intrusion detection.

TABLE I
DISTRIBUTION OF THE UNSW-NB15 DATASET USED IN SIMULATIONS

Class No.	Category	No. of records
1	Analysis	2677
2	Backdoor	2329
3	DoS	16353
4	Exploits	44525
5	Fuzzers	24246
6	Generic	58871
7	Normal	93000
8	Reconnaissance	13987
9	Shellcode	1511
10	Worms	174
-	Total Records	257673

A.1 Data Balancing

As shown in Table I, the number of samples in each class is highly imbalanced. Therefore, in the proposed method, to achieve the data balancing, the samples of majority classes (those with more than 20,000 samples) were first reduced to 20,000 using the undersampling techniques. Then, the number of samples in the minority classes was augmented to 20,000 with over-sampling techniques. In other words, new samples were created for minority classes to balance the number of samples in all classes. Since, after balancing, the real and synthetic samples are randomly mixed and only then split into training and test sets, the model encounters a combination of real and augmented data during both training and evaluation. This prevents the classifier from becoming overly dependent on synthetic patterns. It is important to note that for classes with very limited real data (such as class 10, which contains only 174 real samples), increasing the number of samples to a level comparable with other classes can significantly improve the learning performance of the model. Otherwise, the minimal

contribution of such a class during training may lead to unstable decision boundaries, bias toward majority classes, and weakened generalization capability. In this study, in order to increase the diversity and the amount of data, two data augmentation techniques have been used; namely, rescaling and noise injection. Initially, the data had to be transformed by multiplying them with a variety of scaling factors, so that the model would be adaptable to possible scale variations in the real world. Then random noise based on a normal distribution with zero mean and low standard deviation was added to the data. The introduction of noise helps the model to be more robust in the face of noise in the environment and natural fluctuations in the data.

These two techniques were used to increase both the diversity and number of training samples, in order to improve the performance of the model for various realistic conditions. The use of such augmentation methods is used to enhance the performance of models, reduce overfitting and improve detection accuracy. Moreover, it is a part of the generalizability of the classifier. Table II presents the class-wise distribution of data after balancing operations. Figure 2 illustrates the distribution of data before and after augmentation, with 70% of the data allocated for training and 30% reserved for evaluation.

TABLE II
DATA DISTRIBUTION AFTER BALANCING OPERATIONS

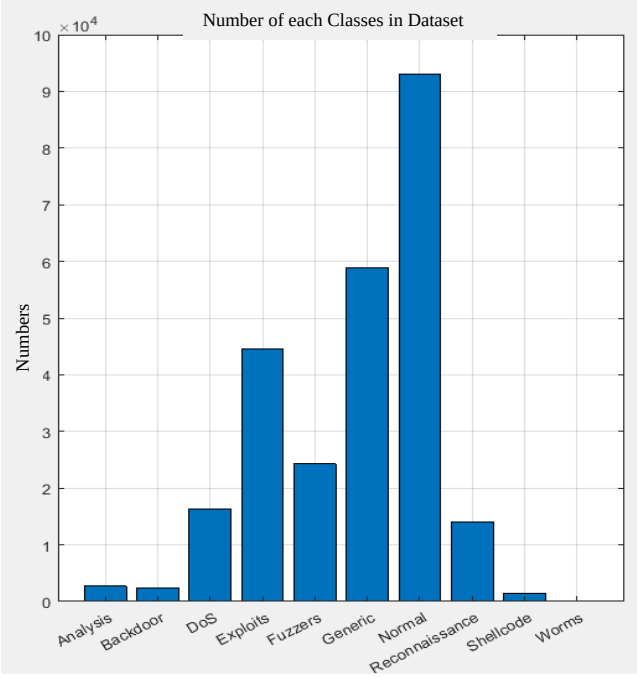
Class No.	Category	No. of records	No. of Train Data	No. of Test Data
1	Analysis	20000	14000	6000
2	Backdoor	20000	14000	6000
3	DoS	20000	14000	6000
4	Exploits	20000	14000	6000
5	Fuzzers	20000	14000	6000
6	Generic	20000	14000	6000
7	Normal	20000	14000	6000
8	Reconnaissance	20000	14000	6000
9	Shellcode	20000	14000	6000
10	Worms	20000	14000	6000
-	Total Records	200000	140000	60000

B. Feature Extraction

In the proposed method, two neural networks—GRU and CNN—are employed to extract sequential and local features, respectively. Sequential and temporal patterns are captured using a Gated Recurrent Unit (GRU), which effectively models short- and long-term dependencies in network traffic flows. Simultaneously, local and structural attack behavior-related features are learned with the help of a Convolutional Neural Network (CNN) with Inception block additions. Both networks produce feature vectors, which are later combined into a single feature vector, and this allows the system to co-exist in terms of representing temporal dependencies and local spatial substructures with a common latent space. This combined representation gives a holistic picture of the behavior of intrusion related data in IoT networks. The description of feature extraction by each of these networks can be found in the following subsections.

B.1 Temporal Feature Extraction using GRU

a)



b)

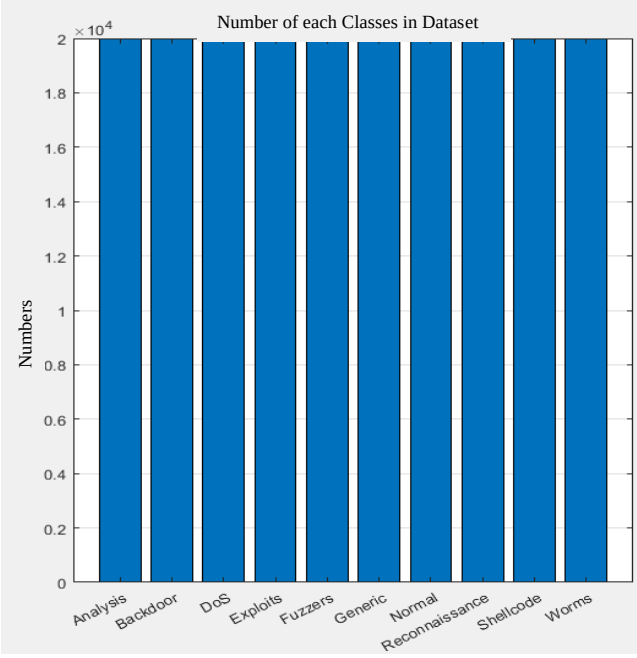


Fig. 2. Data distribution before and after balancing operations

A Gated Recurrent Unit (GRU) is a type of Recurrent Neural Networks (RNNs), which is used to extract both temporal and sequence-dependent features in the proposed structure. GRU is a special-purpose model to identify short- and long-term relationships in a time-series data including network traffic flows. GRUs unlike the traditional RNN, counter the issue of the vanishing gradient hence enhancing their learning capabilities of long-range temporal relationship. GRU has two

gating mechanisms; update gate and reset gate that control the flow of information, where information about the past time steps is either preserved or disregarded:

- Update Gate (z_t): Controls how much of the previous hidden state h_{t-1} is carried forward to the new hidden state h_t, thereby preventing loss of important information across long sequences:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \quad (1)$$

- Reset Gate (r_t): Determines how much of the previous hidden state $h_{t-1}$is considered when computing the candidate hidden state $\tilde{h}_t$, effectively filtering out irrelevant past information:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \quad (2)$$

Using these gates, the hidden states are updated as follows:

- Candidate hidden state ($\tilde{h}_t$): This candidate is calculated by combining the current input (x_t) with the reset-gate version of the previous hidden state.

$$\tilde{h}_t = \tanh(W \cdot [r_t \odot h_{t-1}, x_t] + b) \quad (3)$$

- Final hidden state (h_t): The new hidden state (representing the temporal feature vector) is found using a linear combination of the previous hidden state (h_{t-1}) and the candidate hidden state ($\tilde{h}_t$) and its weight is determined by the update gate.

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (4)$$

Here, x_t denotes the input vector at time step t, $h_{t-1}$is the previous hidden state, W and b represent trainable weights and biases, σ is the sigmoid activation function, and ⊙ indicates the Hadamard (element-wise) product. Through its gating mechanisms, the GRU effectively models complex temporal dependencies and dynamic behaviors within network traffic, enabling the detection of anomalous or malicious activities with higher accuracy than traditional sequence-learning approaches. Figure 3 demonstrates the GRU architecture employed in this study.

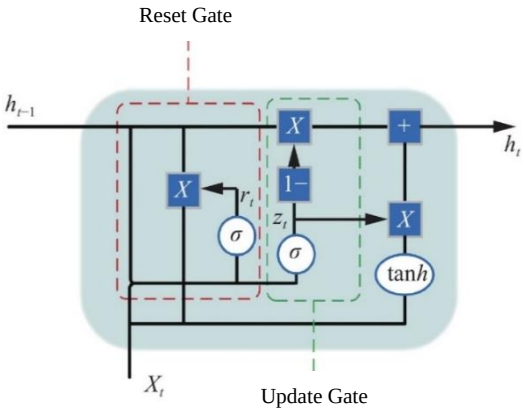


Fig. 3. The GRU architecture

B.2 Local Feature Extraction using CNN with Inception Blocks

To extract local and structural features, the proposed model employs a Convolutional Neural Network (CNN) augmented with Inception blocks, which are designed to capture multi-scale features effectively. Unlike the conventional CNNs which make use of a single fixed filter size, Inception blocks use parallel convolutions with different size of kernels to enable the extraction of features at different scales at the same time. Inception blocks have multiple parallel convolutional paths which have varying receptive fields as well as a pooling path. The results of these paths are then combined, and they produce a more meaningful feature representation:

$$O_{inc} = \text{Cat}\big(C_{1\times k}(X).C_{3\times k}(X).C_{5\times k}(X).P_{max}(X)\big) \quad (5)$$

where C represents convolutional operations with different sized kernel, and P_{max} represents max pooling operations. This architecture enables the network to both capture fine-grained local structures and capture contextual patterns, which is more robust to capture behaviors related to attacks in IoT settings. Using these multi-scale feature maps, the CNN–Inception architecture improves the detection of complex and dynamic patterns of intrusion by the system beyond localized features. Figure 4 illustrates the CNN architecture that is improved with Inception blocks, which is applied in the proposed structure.

C. Optimal Feature Selection Using the MRMR Algorithm

A dimension reduction and selection of features step is carried out in the proposed framework to maximize the computational efficiency as well as overall model performance.

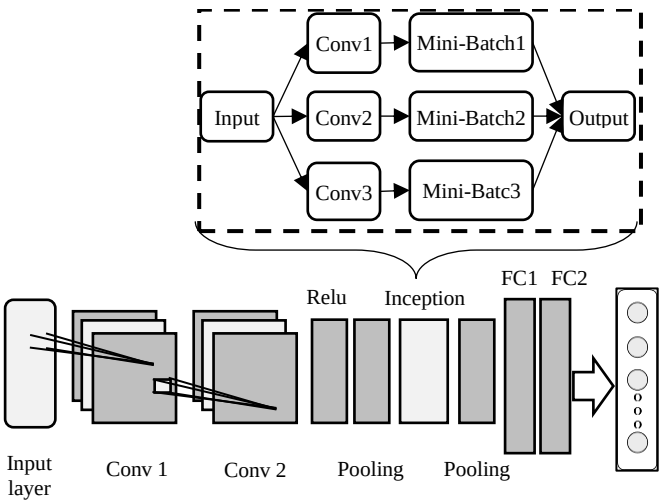


Fig. 4. CNN architecture enhanced with Inception blocks

The primary objective of this stage is to eliminate redundant and irrelevant features while retaining the most informative ones. To this end, the Maximum Relevance Minimum Redundancy (MRMR) algorithm, grounded in the concept of Mutual Information (MI), is adopted. MRMR simultaneously optimizes two criteria: maximizing the relevance of the selected

features with respect to the target class, and minimizing redundancy among the selected features. Using MI, MRMR formulates these two conditions as follows [25]. The mutual information between two discrete random variables, X and Y, is expressed as follows:

$$I(X;Y) = \sum_{y\in y} \sum_{x\in X} p\,(x.y)\log\left(\frac{p(x.y)}{p(x)p(y)}\right) \quad (6)$$

where $p(x.y)$ represents the joint probability distribution and $p(x)$, $p(y$ are the marginal distributions.

1. Maximum Relevance:

This measure is a measure of importance of each feature f_i with respect to the target class c. The way to do this is to select features that have the maximum discriminative information about the class label:

$$\max\left(\frac{1}{|\mathcal{S}|}\sum_{f_i\in\mathcal{S}} I\,(f_i;c)\right) \quad (7)$$

where $\mathcal{S}$ is the subset of selected features.

2. Minimum Redundancy:

This criterion measures the degree of redundancy or overlap among the selected features. The objective is to select features that contribute unique information and minimize redundancy:

$$\min\left(\frac{1}{|\mathcal{S}|^2}\sum_{f_i,f_j\in\mathcal{S}} I\,(f_i;f_j)\right) \quad (8)$$

The MRMR algorithm integrates these two conditions into a single objective function:

$$\max_{f_i\in\mathcal{F}-\mathcal{S}_{k-1}}\left[I(f_i;c) - \frac{1}{k-1}\sum_{f_j\in\mathcal{S}_{k-1}} I\,(f_i;f_j)\right] \quad (9)$$

where $\mathcal{F}$ is the full feature set, and $\mathcal{S}_{k-1}$ is the set of features selected in the previous iteration. At every iteration, the feature which has maximum relevance to the target class and minimum redundancy with previous selected features is selected until the number of features k is reached.

This technique not only makes the feature space less dimensional, but it also can help the model to be more accurate and efficient by getting rid of less informative or redundant features. In addition, it alleviates the curse of dimensionality, which is of paramount importance to high-dimensional intrusion detection tasks.

D. Data Classification Using Enhanced Dense Layers with Adaptive Learning Rate Optimizer (EDL-ALRO)

After extracting comprehensive temporal–local features and performing dimensionality reduction, the final feature vector $\mathrm{F}_{\mathrm{combined}} \in \mathbb{R}^d$ is passed to the classification block composed of dense (fully connected) layers. The primary role of these layers is to map the extracted representations into the decision space, where explicit class boundaries can be defined. If a denotes the initial input to the first dense layer, then:

$$\mathrm{a}^{(0)} = \mathrm{F}_{\mathrm{combined}} \quad (10)$$

For a network with L dense layers:

- $\mathrm{a}^{(0)}$: input to the first dense layer
- $\mathrm{a}^{(1)}$: output of the first dense layer
- $\mathrm{a}^{(2)}$: output of the second dense layer
- …
- $\mathrm{a}^{(L-1)}$: output of the last hidden dense layer

Moreover, z denotes the output scores (logits) of the final Dense layer, over which the softmax function is applied. That is, the output of every Dense layer is determined based on the following equation:

$$\mathrm{a}^{(\ell)} = f\left(\mathrm{W}^{(\ell)}\mathrm{a}^{(\ell-1)} + \mathrm{b}^{(\ell)}\right). \qquad \ell = 1.\ldots, L-1 \quad (11)$$

The parameters of the layer are the weight $\mathrm{W}^{(\ell)}$ and bias $\mathrm{b}^{(\ell)}$ and the output is converted by a nonlinear activation function, say ReLU, simply $f(\cdot)$. The scores z, also known as logits are generated in the last layer (output layer), without the use of an activation function, as the following equation:

$$\mathrm{z} = \mathrm{W}^{(L)}\mathrm{a}^{(L-1)} + \mathrm{b}^{(L)} \quad (12)$$

The resultant scores z are then passed through the softmax function, which converts the scores into the likelihood of each of the K classes:

$$p(\,y = k \mid \mathbf{z}\,) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}. \qquad k = 1.\ldots.K \quad (13)$$

This generates a probability distribution vector $\mathrm{p} = [p(y = 1).\ldots.p(y = K)]$ that sums to one, whose maximum value in a prediction is the predicted label $\mathrm{argmax}_k p(y = k)$. The categorical cross-entropy loss is applied during model training as follows:

$$\mathcal{J}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}\ \log p_{ik} \quad (14)$$

where y_{ik} is the one-hot representation of the ground truth of sample i of class k, and p_{ik} is the predicted probability. Assuming that it is additional to increase the discriminative power of the dense block, this paper uses Enhanced Dense Layers with Adaptive Learning Rate Optimizer (EDL-ALRO), which is based on the principles of adaptive moment estimation. The main idea is to adapt the learning rate of each parameter dynamically according to the statistic of historical gradients, so as to speed up the speed of convergence and at the same time ensure the stability of training. At iteration t, the gradient of the loss with respect to all the parameters is calculated as:

$$\mathrm{g}_t = \nabla_\theta \mathcal{J}(\theta_t) \quad (15)$$

Here, θ represents any trainable parameter of the system, which can represent the weight vector W, the bias vector b, or any other parameter of the network. The first-order moment estimate of the gradient (m_t) and the second order moment estimate (v_t) of each parameter th are then updated as moving averages using the following equations:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad (16)$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t \odot g_t \quad (17)$$

where $\beta_1, \beta_2 \in (0,1)$ are smoothing coefficients, and $\odot$ denotes element-wise multiplication. To correct the bias at the start, unbiased gradient statistics for the gradient mean m_t and gradient variance v_t are calculated as follows:

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \quad (18)$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \quad (19)$$

Finally the parameters are updated using an adaptive learning rate:

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\varepsilon} \quad (20)$$

where α is the base rate of learning and ϵ is a very small constant so that we do not divide by zero.

From the interpretational perspective, mt gives the most prominent search direction in the parameter space, whilst v_t takes a step in the update that is sensitive to the variance in the gradient. Therefore, EDL-ALRO adjusts multiple learning rates to each parameter, which can achieve faster convergence and higher classification accuracy.

The fundamental innovation of EDL-ALRO, compared to conventional adaptive optimizers, lies in the employment of a gradient-statistics-driven weight adjustment mechanism designed to optimize the mapping process within the decision space. Unlike classical optimizers that apply a learning rate globally or with limited sensitivity, EDL-ALRO utilizes Equations (18) and (19) to manage the bias correction process, ensuring that unbiased estimates of the gradient mean and variance are achieved during the early stages of training. This effectively prevents severe oscillations in the search direction (m_t) and guarantees update stability when dealing with high-dimensional feature vectors.

The operational distinction of this approach from existing methods is its variance-sensitivity; specifically, according to Equation (20), the base learning rate (α) is modulated by the square root of the second moment ($\hat{v}_t$). This feature ensures that parameters with small and sparse gradients undergo larger steps, whereas parameters with rapid variations receive shorter, more precise updates. By establishing a dynamic equilibrium between momentum—to maintain the overall search trajectory—and variance normalization—for fine-grained accuracy in the final stages—EDL-ALRO minimizes the risk of divergent oscillations around global minima. This property renders the model robust against varying probability distributions in the input data and, in contrast to existing methods that necessitate manual learning rate tuning, allows it to automatically adapt to the complex geometry of the loss function in every iteration. Figure 5 shows the structure of the proposed dense-layer classifier with adaptively updated weights that consists of an input layer, a hidden dense layer, and an output layer using EDL-ALRO.

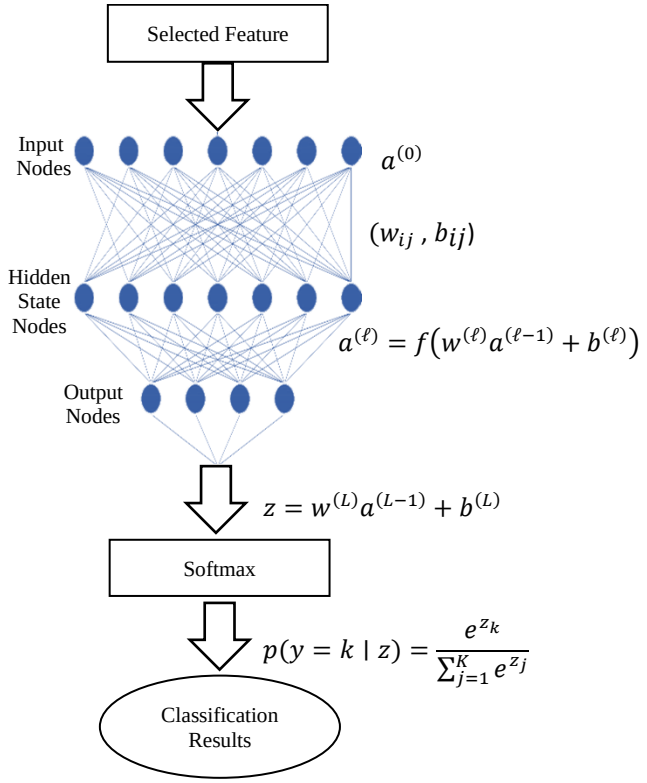


Fig. 5. The architecture of the proposed dense-layer classifier

IV. Results and Discussion

In this section, the results of the proposed method for intrusion detection of computer networks are presented and were compared with the previous studies. The proposed algorithm was implemented on MATLAB R2024a and tested on a system having Intel Core i5 CPU, 16 GB RAM, and NVIDIA Quadro K2000 GPU. The experiments were carried out on the UNSW-NB15 dataset, out of which after Data Augmentation process, 70% of the dataset was used for training and the remaining 30% were used for testing. In order to avoid overfitting, 10-fold cross-validation (K=10) was used. Also, The model parameters were tuned through a systematic trial and error procedure. This manual search strategy involved evaluating multiple configurations and iteratively refining them. The process included numerous experiments with different values for the number of GRU units, dropout rates, the number of CNN filters, and kernel sizes. To validate the robustness of the selected parameters and to mitigate the inherent risks of overfitting associated with manual tuning, a 10 fold cross validation scheme was employed to assess the obtained results. Ultimately, the combination of parameters that provided the best balance among the Accuracy, Precision, Recall, and F score metrics—while ensuring model stability and generalization capability—was selected. In order to evaluate the performance of the proposed framework in comparison to other existing approaches, the standard classification accuracy, precision, recall, and F-Measure were used and defined as follows:

- Precision: the ratio of correctly predicted instances of attacks to all instances predicted as attacks.
- Recall (Detection Rate): the ratio of properly classified attack instances in all actual attack instances.
- Accuracy (Detection Accuracy): the ratio of correctly-classified instances to the total number of instances; This metric is only reliable if the data set is balanced.
- F-Measure: the harmonic mean of Precision and Recall, providing a balanced evaluation of detection performance.

$$Precision = \frac{TP}{TP+FP} \quad (21)$$

$$Recall = Detection\ Rate = \frac{TP}{TP+FN} \quad (22)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (23)$$

$$F\ Measure = 2\left(\frac{Precision \times Recall}{Precision+Recall}\right) \quad (24)$$

Figure 6 shows the learning curve of the proposed model on the Loss metric. This curve measures the changes in loss function during training for the iterations and epochs. The x-axis is the number of training steps (Iterations) and y-axis is the error (Loss) of the model with each step. As seen, the loss value gets decreasing with a higher number of iterations, which is a sign of an effective learning and improving model performance with time. A strong decrease in the loss function is visible in the first training phase (Epoch 1), in which the model quickly learns from the input data. In the following epochs (Epochs 2-4) although the overall tendency is still towards lower values, there are some noticeable fluctuations. These oscillations, in the form of small peaks and valleys in the curve, can be explained due to the use of an Adaptive Learning Rate mechanism in the ALRDS structure, which dynamically changes the optimization rate for each parameter. This behavior is representative of the fine-grained search behavior of the model in the parameter space to obtain a stable convergence point (local minimum). Ultimately, the curve is a demonstration of convergence towards some value loss close to zero, which signifies the success of the training and the model's preparedness to accurately classify the unseen samples.

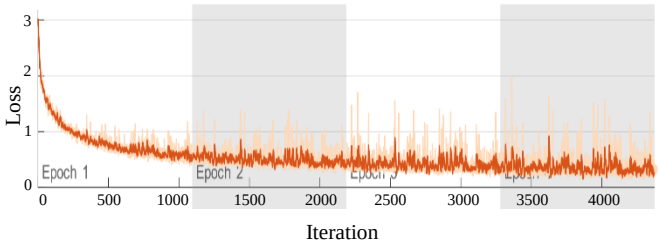


Fig. 6. The learning curve of the presented approach

The ROC curve of the proposed method is shown in Figure 7. The ROC curve is a plot of the True Positive Rate (TPR) versus False Positive Rate (FPR) of the detection system as a function of the threshold value, and hence provides a visual representation of the trade off between the detection sensitivity of the system and the false alarms. An effective classifier has a curve inclined towards the top-left corner with high TPR and low FPR, while a weak classifier has a curve inclined towards the bottom-right corner with low TPR and high FPR. A random classifier is along the diagonal line, where TPR=FPR. As shown in Figure 7, the proposed model is able to obtain a ROC curve with high TPR and low FPR, with the inflection point of this curve near around the top-left corner of the plot. This proves that the model is highly able to detect accurately the intrusion attempts with fewer false alarms.

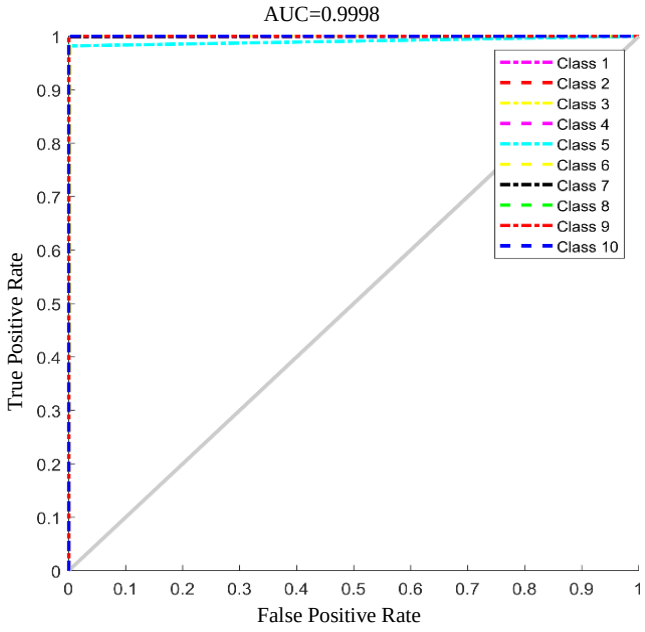


Fig. 7. The ROC curve

The confusion matrix resulting from the evaluation of the suggested approach has been demonstrated in figure 8. This matrix is a typical analytical tool to evaluate the performance of a classifier when dealing with supervised learning tasks. Each row in the matrix corresponds to the actual class labels and each column represents the predicted class labels of the model. The diagonal elements represent the number of correct classified samples (True Positives) while the elements off diagonal represent the misclassified samples (False Positives and False Negatives). Based on the confusion matrix, the model proposed provides a perfect classification accuracy (100%) for seven out of the ten classes (Classes 1, 3, 4, 5, 8, 9 and 10). This means that all samples that belong to these classes were correctly recognised by the model, there was no misclassification into other classes. The only misclassifications that have been observed are between Classes 2 and 7. In particular, out of 5,913 actual samples in Class 2, 1 sample was incorrectly predicted as Class 3. Similarly, class 7, which has 6043 actual samples, 1 sample was misclassified as class 5, and 108 samples were misclassified as class 6. These minor errors caused a small decrease in accuracy in Classes 2 and 7. Nevertheless, given the extremely small number of misclassified examples in comparison to the total number of data points, the proposed model has an extremely high accuracy of prediction. These results highlight that hybrid GRU-CNN architecture in

conjunction with MRMR feature selection process is highly effective in terms of extracting discriminative features and reliable classification of network traffic samples. The near-perfect performance of the model in 8 out of 10 classes highlights this model's robustness and efficiency in solving the challenges of detecting complex patterns of attack within IoT environments. It is worth noting that in this study, after performing the oversampling/undersampling procedures and balancing the dataset, the entire data were divided into 70% for training and 30% for testing. However, prior to splitting the data into training and test sets, the complete dataset was randomly permuted (shuffled) to eliminate any potential bias arising from the original ordering of the samples and to ensure an appropriate distribution of instances across both subsets. Due to this random reordering and the application of a global 70/30 split, the number of test samples per class is not necessarily exactly 6,000, although it remains very close to this value, as reflected in the confusion matrix presented in Figure 8. It is important to emphasize that the overall proportion of training and test data strictly adheres to the 70% and 30% ratio, respectively, and the final evaluation of the model has been conducted using the entire test set resulting from this partitioning.

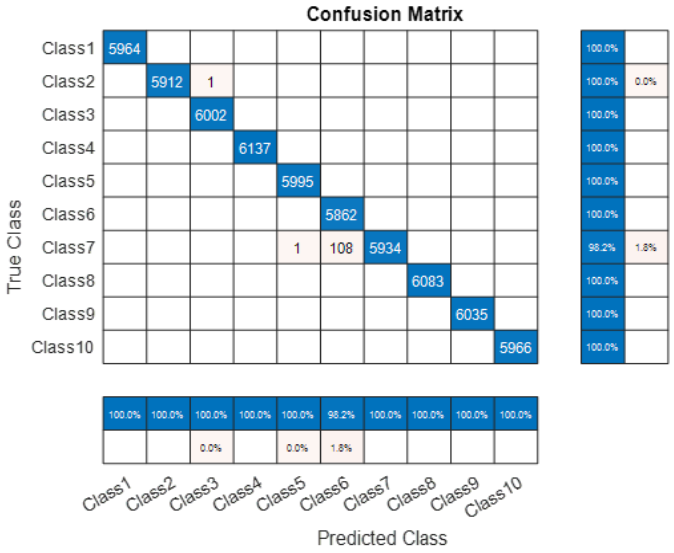


Fig. 8. The confusion matrix

Table III shows a comparative evaluation of the proposed method with some of the recently published intrusion detection methods for IoT networks. The efficacy of this model is evaluated in terms of accuracy and compared with the methods of CNN-LSTM [20], Random Forest [21], SFS-RF [22], RF-SFS-GRU [22], GRU [23], RNN [23], and MHA-BiGRU [23]. As seen in the table, the accuracy of the proposed HSL-DCRNet is 99.82%, which is more than the accuracy of all other techniques considered together. This superior performance goes to show how it is capable of being strong in identifying malicious activities with minimum classification errors. Among the compared methods Random Forest also showed competitive results with accuracy 99.45% ranking 2 but still a little bit lower compared to the proposed approach. Deep learning-based architectures, such as MHA-BiGRU and GRU, have achieved accuracies of 98.73% and 97.12%, respectively - very good results, but still at a good distance from the proposed method. On the other hand, traditional or hybrid feature selection based methods (i.e., CNN-LSTM, RNN, SFS-RF, and RF-SFS-GRU) had a significant decrease in the accuracy, demonstrating their shortcomings when considering the complex and dynamic intrusion patterns in IoT scenarios.

TABLE III
THE ASSESSMENT OF THE SUGGESTED APPROACH RELATIVE TO ALTERNATIVE METHODS

Author	Year	Method	Accuracy
Chen et al.	2025	CNN-LSTM [20]	87.00
Shweta, et al.	2024	Random Forest [21]	99.45
Azar et al.	2023	SFS-RF [22]	88.52
Azar et al.	2023	RF-SFS-GRU [22]	89.00
Biju, A. et al.	2025	GRU [23]	97.12
Biju, A. et al.	2025	RNN [23]	95.5
Biju, A. et al.	2025	MHA-BiGRU [23]	98.73
Proposed Model	-	HSL-DCRNet	99.82

In conclusion, the comparative analysis also indicates that the proposed HSL-DCRNet, due to the highest accuracy, is the most effective method of intrusion detection for IoT and outperforms the conventional machine learning and the advanced deep learning methods to ensure IoT network security.

In addition, Figure 9 shows a comparison of the proposed model with other techniques in terms of Precision, Recall and F-score. The results show that the proposed method has superior performance to all the competitors in those three important evaluation metrics. Specifically, the proposed approach boasted 99.82% in Precision, Recall and F-score, and reflects not only the high ability of correctly identifying the true positive samples (Precision) but also the high capacity of retrieving the majority of the relevant instances (Recall). The well-balanced improvement in both dimensions leads to a much higher F-score which confirms the robustness and stability of the proposed framework. Although Random Forest achieved close values i.e. 99.72%, 99.65% and 99.65% for Precision, Recall and F-score respectively, but it still performed slightly worse than the proposed model. Other methods like CNN-LSTM and GRU with F-scores of 86.00% and 89.57%, respectively, had reasonable but significantly lower results. Moreover, hybrid methods such as RF-SFS-GRU achieved weak performance results, proving that they are not very competitive in real-world IoT intrusion detection tasks. Meanwhile, MHA-BiGRU with an F-score of 93.98%, achieved significantly better results than the baseline models, but still failed to achieve the same precision and balance as the proposed HSL-DCRNet.

Overall, the comparative results clearly indicate that the proposed approach, by achieving the highest scores across all performance metrics, delivers superior classification capability and offers a reliable and efficient approach for securing IoT environments.

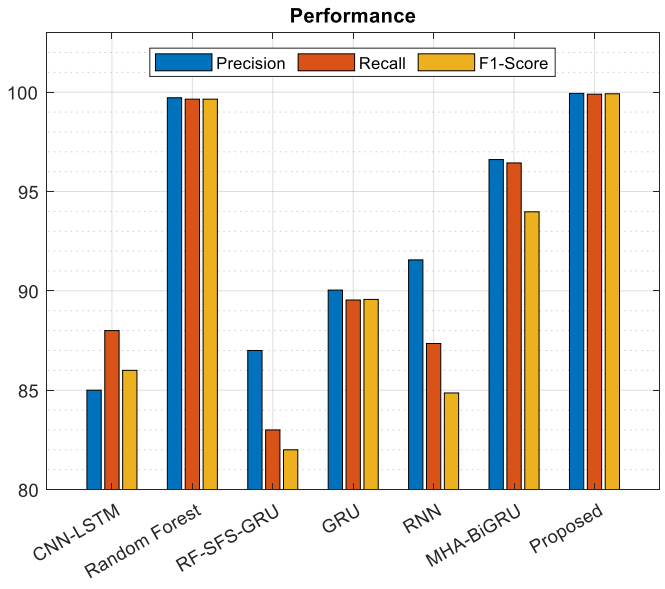


Fig. 9. The comparison of proposed model with other approaches in terms of precision, recall, F-score

Table IV presents the results of the ablation study, conducted to evaluate the impact of the MRMR feature-selection mechanism and the adaptive learning-rate adjustment using the EDL-ALRO technique on the performance of the proposed method. As the results indicate, when the model is executed without MRMR and with a fixed learning rate (Experiment 1), an accuracy of 99.01% is achieved. Adding the MRMR feature-selection stage under the same fixed learning-rate setting (Experiment 2) improves the accuracy to 99.33%, demonstrating the effective role of MRMR in extracting discriminative features and reducing unnecessary dimensions. Furthermore, employing the adaptive EDL-ALRO optimizer without MRMR (Experiment 3) increases the accuracy to 99.61%, highlighting the significant contribution of this optimizer in enhancing model stability and convergence. Finally, the simultaneous use of both MRMR and EDL-ALRO (Experiment 4) yields the highest accuracy of 99.82%. This result shows that these two components complement each other, and their combined use leads to the best model performance. Overall, the ablation study demonstrates that both the feature-selection mechanism and the adaptive optimization strategy are essential and effective for achieving optimal performance in the HSL-DCRNet architecture.

TABLE IV
ABLATION STUDY FOR EVALUATE PERFORMANCE OF PROPOSED METHOD IN DIFFERENT SITUATION

Exp. No.	description	ACC
1	HSL-DCRNet With Fixed Learning Rate Without MRMR	99.01
2	HSL-DCRNet With Fixed Learning Rate With MRMR	99.33
3	HSL-DCRNet With EDL-ALRO Without MRMR	99.61
4	HSL-DCRNet With EDL-ALRO With MRMR	99.82

V. CONCLUSION

In this paper, the authors proposed a new Hybrid Sequential-Local Deep CNN-RNN Feature Extractor Network (HSL-DCRNet) to detect intrusion in an Internet of Things (IoT) system. The suggested architecture uses a Gated Recurrent Unit (GRU) to learn temporal correlations with a Convolutional Neural Network (CNN) further developed with Inception blocks to extract local features. Combining these two complementary representations and applying the Maximum Relevance Minimum Redundancy (MRMR) to select the best features, the model has become an efficient way to strike a balance between expressiveness and computer efficiency. Moreover, the use of the Enhanced Dense Layers with Adaptive Learning Rate Optimizer (EDL-ALRO) can be considered as one of the most effective ways to accelerate the convergence speed and classification. Experimental testing on the UNSW-NB15 dataset showed that HSL-DCRNet can accurately predict 99.82, which is higher in the accuracy of the traditional machine learning models and the state of the art deep learning methods. The findings affirm the capacity of the framework to identify different classes of intrusions with great precision, recall, and F-score, and is also strong in crowdsouring class imbalance and noise. The results highlight the possibilities of hybrid deep learning models in the context of further development of intrusion detection in IoT networks. Along with the future work, this research will be extended by assessing the model on real-time streaming data, incorporating lightweight deployment strategies to deploy resource-constrained IoT devices, and investigating adaptation mechanisms to counter new and zero-day attackers. The objectives of these directions are to make even more complex cyber-physical systems scalable, generalizable and resilient with regard to IDS solutions.

REFERENCES

[1] R. Mishra, A. Mishra: “Current research on Internet of Things (IoT) security protocols: A survey”, Computers & Security, 2025, p.p. 104310.

[2] T. Mazhar, D. B. Talpur, T. A. Shloul, Y. Y. Ghadi, I. Haq, I. Ullah, H. Hamam: “Analysis of IoT security challenges and its solutions using artificial intelligence”, Brain Sciences, Vol.13, No.4, p.p. 683, 2023.

[3] A. Heidari, M. A. Jabraeil Jamali: “Internet of Things intrusion detection systems: a comprehensive review and future directions”, Cluster Computing, Vol.26, No.6, p.p. 3753-3780, 2023.

[4] O. H. Abdulganiyu, T. Ait Tchakoucht, Y. K. Saheed: “A systematic literature review for network intrusion detection system (IDS)”, International Journal of Information Security, Vol.22, No.5, p.p. 1125-1162, 2023.

[5] B. Nawaal, U. Haider, I. U. Khan, M. Fayaz: “Signature-based intrusion detection system for IoT”, in Cyber Security for Next-Generation Computing Technologies, CRC Press, p.p. 141-158, 2024.

[6] M. Bhavsar, K. Roy, J. Kelly, O. Olusola: “Anomaly-based intrusion detection system for IoT application”, Discover Internet of Things, Vol.3, No.1, p.5, 2023.

[7] F. S. Alsubaei: “Smart deep learning model for enhanced IoT intrusion detection”, Scientific Reports, Vol.15, No.1, p.20577, 2025.

[8] D. Manivannan: “Recent endeavors in machine learning-powered intrusion detection systems for the internet of things”, Journal of Network and Computer Applications, Vol.229, p.103925, 2024.

[9] M. Almohaimeed, F. Albalwy: "Enhancing IoT Network Security Using Feature Selection for Intrusion Detection Systems", Applied Sciences, Vol.14, No.24, 2024.
[10] R. Kimanzi, P. Kimanga, D. Cherori, P. K. Gikunda: "Deep Learning Algorithms Used in Intrusion Detection Systems – A Review", arXiv preprint, arXiv:2402.17020, 2024.
[11] P. Sinha, D. Sahu, S. Prakash, T. Yang, R. S. Rathore, V. K. Pandey: "A high performance hybrid LSTM CNN secure architecture for IoT environments using deep learning", Scientific Reports, Vol.15, No.1, p.9684, 2025.
[12] H. Dong, I. Kotenko, D. Levshun: "Next-generation IIoT security: Comprehensive comparative analysis of CNN-based approaches", Knowledge-Based Systems, Vol.316, p.113337, 2025.
[13] A. Sagu, N. S. Gill, P. Gulia, N. Alduaiji, P. K. Shukla, M. A. Shah: "Advances to IoT security using a GRU-CNN deep learning model trained on SUCMO algorithm", Scientific Reports, Vol.15, No.1, p.16485, 2025.
[14] B. Cui, Y. Chai, Z. Yang, K. Li: "Intrusion detection in IoT using deep residual networks with attention mechanisms", Future Internet, Vol.16, No.7, p.255, 2024.
[15] S. Yaras, M. Dener: "IoT-based intrusion detection system using new hybrid deep learning algorithm", Electronics, Vol.13, No.6, p.1053, 2024.
[16] S. Jamshidi, A. Nikanjam, K. W. Nafi, F. Khomh, R. Rasta: "Application of deep reinforcement learning for intrusion detection in Internet of Things: A systematic review", Internet of Things, p.101531, 2025.
[17] A. Gueriani, H. Kheddar, A. C. Mazari: "Enhancing IoT security with CNN and LSTM-based intrusion detection systems", in Proc. of the 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS), p.p. 1-7, IEEE, April 2024.
[18] O. Belarbi, T. Spyridopoulos, E. Anthi, I. Mavromatis, P. Carnelli, A. Khan: "Federated deep learning for intrusion detection in IoT networks", in Proc. of IEEE Global Communications Conference (GLOBECOM 2023), p.p. 237-242, Dec. 2023.
[19] C. Hazman, A. Guezzaz, S. Benkirane, M. Azrour: "Toward an intrusion detection model for IoT-based smart environments", Multimedia Tools and Applications, Vol.83, No.22, p.p. 62159-62180, 2024.
[20] Y. Chen: "Framework design of Network intrusion detection based on convolutional neural networks", Procedia Computer Science, Vol.261, p.p. 1356-1362, 2025.
[21] S. More, et al.: "Enhanced intrusion detection systems performance with UNSW-NB15 data analysis", Algorithms, Vol.17, No.2, p.64, 2024.
[22] A. T. Azar, et al.: "Deep learning based hybrid intrusion detection systems to protect satellite networks", Journal of Network and Systems Management, Vol.31, No.4, p.82, 2023.
[23] A. Biju, S. Wilfred Franklin: "Dual Feature-Based Intrusion Detection System for IoT Network Security", International Journal of Computational Intelligence Systems, Vol.18, No.1, p.p. 1-19, 2025.
[24] UNSW: "UNSW-NB15 Dataset", available at: https://research.unsw.edu.au/projects/unsw-nb15-dataset
[25] Z. H. Obaid, B. Mirzaei, A. Darroudi: "An efficient automatic modulation recognition using time–frequency information based on hybrid deep learning and bagging approach", Knowledge and Information Systems, 2024, p.p. 1-18.

Abdullah Makki Jebur was born in Najaf, Iraq, in 1997. He received his B.Sc. degree in Telecommunications Engineering from the Engineering Technical College of Najaf, Al-Furat Al-Awsat Technical University, Iraq, in 2019, and his M.Sc. degree in Telecommunications Engineering from Imam Reza International University, Mashhad, Iran, in 2021. He is currently an Assistant Lecturer at the University of Kufa, Iraq. His current research interests include artificial intelligence and its applications. He can be reached via email: abdullahm.algburi@uokufa.edu.iq.

Murtadha Talib Abbas is an assistant lecturer at university of kufa. He got his master degree from Obuda University, Budapest, Hungary in 2022. The BSc degree gotten Engineering Technical College of Najaf, Al- Furat Al-Awsat Technical University, Najaf, Iraq in 2009. Now, he is a member of faculty staff in university of kufa. Since 2022, he is occupied a responsible of lecturers affairs, administrative and finance affairs department, university of kufa. His works focused in the security issues in IoT networks. He can be reached via email: murtadhat.mullayousif@uokufa.edu.iq.

Bashaer Makki Jebur was born in Najaf, Iraq, in 1991. She received her B.Sc. degree in Telecommunications Engineering from the Engineering Technical College of Najaf, Al-Furat Al-Awsat Technical University, Iraq, in 2013, and her M.Sc. degree in Telecommunications Engineering from Imam Reza International University, Mashhad, Iran, in 2021. She is currently an Assistant Lecturer at Al-Muthanna University, Iraq. Her research interests include artificial intelligence and its applications. She can be contacted at bashaer.makki@mu.edu.iq.