SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I2P119

Volume 13 Issue 2, 254-263, February 2026
© 2026 Seventh Sense Research Group®

Original Article

An Efficient Intrusion Detection System using Dimensionality Reduction with Horse Optimization Based Ensemble Classifier for Internet of Things Devices

M. Arun[1], R. Balamurugan[2], S. Kannan[3]

[1]Department of Information Technology, FEAT, Annamalai University, Chidambaram, Tamil Nadu, India.
[2]Department of Information Technology, FEAT, Annamalai University (Deputed to GCE, Bargur), Chidambaram, Tamil Nadu, India.
[3]Department of Computer Science and Engineering, Kings College of Engineering, Punalkulam, Pudukottai, Tamil Nadu, India.

[1]Corresponding Author : mailstoroyalarun@gmail.com

Received: 23 November 2025 Revised: 25 December 2025 Accepted: 26 January 2026 Published: 20 February 2026

Abstract - *Cybersecurity has evolved progressively due to the proliferation of the Internet of Things (IoT), since numerous compact, intelligent gadgets transmit vast quantities of data to the Internet. Nevertheless, these systems have numerous defence faults that result from the absence of security tools and support for hardware security, making them vulnerable to cyberattacks. Hence, the growth of IoT devices to deliver security over resistance to threats is a requirement to create an IoT that is secure and effective. Defending these things is very significant for the security of the system. Moreover, it is significant to incorporate the Intrusion Detection System (IDS) with an IoT device. IDS aims to perceive and analyse network traffic from dissimilar sources and detect malicious actions. It is an important fragment of cybersecurity technology. Presently, the Deep Learning (DL)-based system plays a fundamental part in the IDS scheme to detect and classify attacks. This paper develops an Efficient Intrusion Detection System Using Dimensionality Reduction and an Ensemble-based Classification Model (EIDS-DRECM) for Internet of Things Devices. The main intention of the EIDS-DRECM paper is to enhance cybersecurity in IoT networks by developing efficient threat detection and mitigation mechanisms to ensure data integrity, confidentiality, and system resilience. The min-max scaler methodology has been employed initially in the data preprocessing stage by changing and organizing raw data into a suitable format. Followed by, the process of Feature Selection (FS) is executed by the Horse Optimization Algorithm (HOA) to retain the most relevant feature from a dataset. At last, the ensemble models of Graph Convolutional Network (GCN), Temporal Convolutional Network (TCN), and Deep Recurrent Q-Network (DRQN) are used for the attack detection process. An extensive set of simulations was involved in exhibiting the promising results of the EIDS-DRECM method. The experimentation results inferred the proficient performance of the EIDS-DRECM system in the attack detection procedure.*

Keywords - *Cybersecurity, Feature Selection, Internet of Things, Ensemble models, Intrusion Detection System, Deep Learning.*

1. Introduction

Currently, the Internet of Things (IoT) has influenced several fields, namely healthcare, agriculture, transportation, and the automotive sector [1]. To integrate each physical entity into digital platforms, IoT networks comprise billions of interconnected devices, equipped with actuators, sensors, and similar technology to the Internet, producing vast amounts of data [2]. By thoroughly examining this data, IoT vendors deliver several smart services, like predictive maintenance in smart manufacturing and automated irrigation in smart farming.

Such services increase business gains and improve user experience. In general, IoT is viewed as a crucial component of the digital transformation shaping the modern world [3]. Cybersecurity systems have been crucial since the onset of the computer networking age. Nevertheless, security breaches occurred even earlier. Comparable events have continued into the current era.

The ten most common attack methods today include malware, credential theft, unknown threats, software flaws, code manipulation, unauthorized access, Denial of Service (DoS), content alteration, and data robbery. Therefore, both identified and unidentified attack methods remain major cyber risks. Figure 1 depicts the general framework of IDS using IoT devices.

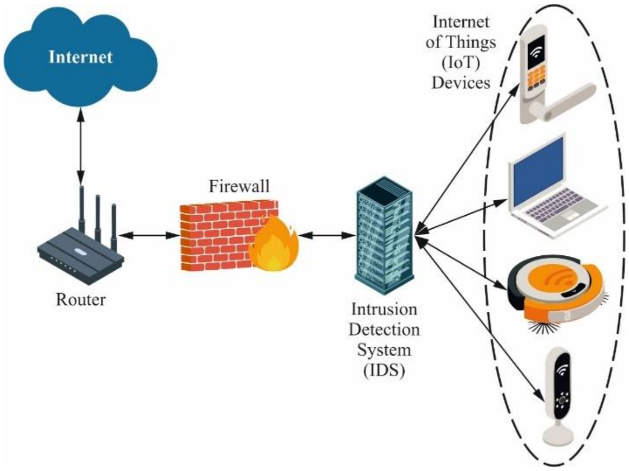
Internet
Internet of Things (IoT) Devices
Firewall
Router
Intrusion Detection System (IDS)

Fig. 1 Framework of IDS using IoT devices

Furthermore, due to the swift expansion of the IoT, its security becomes one of the most difficult issues in such an interconnected and shared environment [4]. Data and its accuracy can be threatened by malicious software, malware, and cyber attackers. Data vulnerability can directly impact the security of the whole IoT, leading to various harmful activities. As a result, several wireless communications enable unrestrained surveillance. The Intrusion Detection System (IDS) was designed to identify unauthorized use of computer systems. The IDS can notify the system operator about suspicious behavior or irregularities [5]. The IDS is considered a flexible network security tool that can provide useful insights to the network operator regarding emerging threat forms. It is also important to refresh the network with a sample of the threat form before it can recognize it. IDS is separated into 3 types: signature-based, anomaly-based, and specification-based [6]. Signature-based techniques recognize threats through existing attack signatures and known behaviour patterns. An anomaly-based detection method identifies irregularities from established typical actions. In the same way, a specification-based method applies the guidelines set by the operator. It is difficult to maintain IDS with the latest updates because of intricate and evolving conditions [7]. An IDS initially gathers and analyzes information, then uses a detection strategy to trigger alerts, which are passed to a human network specialist for further evaluation.

To reduce the security threats of IoT networks, a network IDS is often placed at an Internet access point to protect the systems [8]. Deep Learning (DL) has shown exponential performance in many regions, namely Computer Vision (CV), Natural Language Processing (NLP), and pattern recognition. Applying DL approaches in the field of network cybersecurity presents valuable chances to improve the precision and performance of IDS [9]. DL-driven IDS paradigms can train on massive amounts of network information, recognize unusual trends, and adjust to changing attack methods. This method offers the ability to strengthen overall security by lowering false positives and identifying previously unseen threats [10]. The structure of IDS using DL for network cybersecurity includes several essential parts. Moreover, the selection of DL methods plays an important part in the IDS structure.

2. Literature Survey

Aljabri [11] presented a Multi-Head Attention-based ID alongside an Improved White Shark Optimizer Algorithm (MHAID-IWSOA) technique within IoT systems. The motive of this technique is to optimize the cybersecurity recognition and migration approach utilizing sophisticated optimization models. Moreover, the SCSO method is employed for the FS procedure. This method utilizes the BiGRU-MHA methodology for recognizing and classifying attacks. Assiri [12] presented a maintainable BC-powered Edge Verification alongside a Consensus Approach-based Optimum DL (BCEVCA-ODL) technique to recognize faults in maintainable IoT infrastructures. The SSAE paradigm was utilized for detecting faults within IoT environments. Alkhonaini et al. [13] projected a new Sandpiper Optimizer alongside a Hybrid DL-driven ID (SPOHDL-ID) method from the BC-empowered IoT setting. This method aims to achieve security through detecting and classifying intrusions from the IoT setting. In this regard, BC technologies are employed for a safe process of data exchange. In this method, the FS from the network traffic transformation is carried out through the SPO paradigm. Moreover, the projected method leverages the HDL methodology for ID.

Hammadeh et al. [14] addressed these difficulties by presenting an advanced hybrid technique combined with an ONOS controller. This technique integrates entropy-based examination and an ML to improve the detection of low- and high-volume DDoS using a binary classification. It enhances ID to provide a deep understanding of network patterns and fortifies flexibility against emerging cyberattacks. The authors [15] presented a new system design and a model that merges Ensemble learning with a honeypot UAV for DDoS attack identification. Instead of depending on a single model, utilize dual methodologies merged by a bagging-based ensemble method to combine their outcomes. In this context, the recognized attacks are rapidly transmitted to second SDN controllers for the application of active security procedures. A honeypot UAV is integrated into the structure to improve the system's efficiency. Setitra et al. [16] introduced an approach for detecting DDoS attacks. This approach utilizes an amalgamation of CNN and Multilayer Perceptron (MLP) to boost the ML-driven DDoS-detection system's performance in SDN settings.

The authors [17] proposed an Adaptive DL (ADL) paradigm alongside a data preprocessing mechanism, a neural network pretraining mechanism, and a classifier mechanism. This ADL acquires the counts of layers and neurons for every layer by assessing the network traffic's characteristic dimension. In Transfer Learning (TL), ADL can capture the raw data dimensions and attain novel features. By integrating DL algorithms alongside conventional ML-driven classification approaches, the classification performance of network traffic data is greatly enhanced. Kavitha et al. [18] developed an Intelligent IDS through an Enhanced Arithmetic Optimizer Algorithm with the DL (IIDS-EAOADL) scheme. This scheme carried out the data standardization procedure for input data normalization. Furthermore, an Equilibrium Optimizer-based FS (EOFS) methodology is designed for selecting an optimum feature subset. For ID, a Deep Wavelet AE (DWAE) classifier is used. Because the appropriate DWNN's parameter tuning is considerably significant, the EAOA method is employed for tuning them.

3. Proposed Methodology

This paper develops an EIDS-DRECM approach for IoT Devices. The main intention of the EIDS-DRECM paper is to enhance cybersecurity in IoT networks by developing efficient threat detection and mitigation mechanisms to ensure data integrity, confidentiality, and system resilience. It contains distinct kinds of phases, such as min-max scaler, HOA-based feature reduction, and an ensemble of classification models. Figure 2 illustrates the complete workflow of the EIDS-DRECM approach.

3.1. Min-Max Scaler

Primarily, the min-max scaler method was utilized for data normalizing. The database undertakes a process of normalization to agree with the modeling standards usually necessitated in NN [19]. Normalization serves as a transformation method intended to decrease significant variations to map every data point into a normalized interval between zero and one. This procedure enables a more stable and efficient training level for DL techniques. Normalization is the MinMax Normalization model, which broadly identifies the MinMax Scaler. The mathematical model is specified in Equation (1).

$$\breve{z}_i = \frac{z_i - z_{\min}}{z_{\max} - z_{\min}} \tag{1}$$

Where, $\breve{z}_i$ means the normalized value of i_{th} observation that is a restrained range of zero and one. The symbol z_i represents an original, whereas $z_{\max}$ and $z_{\min}$ depict maximal and minimal values in the equivalent feature, correspondingly. The utilization of z emphasizes that the variable is modified from its original condition, assisting in differentiating the standardized values in an overall investigation.

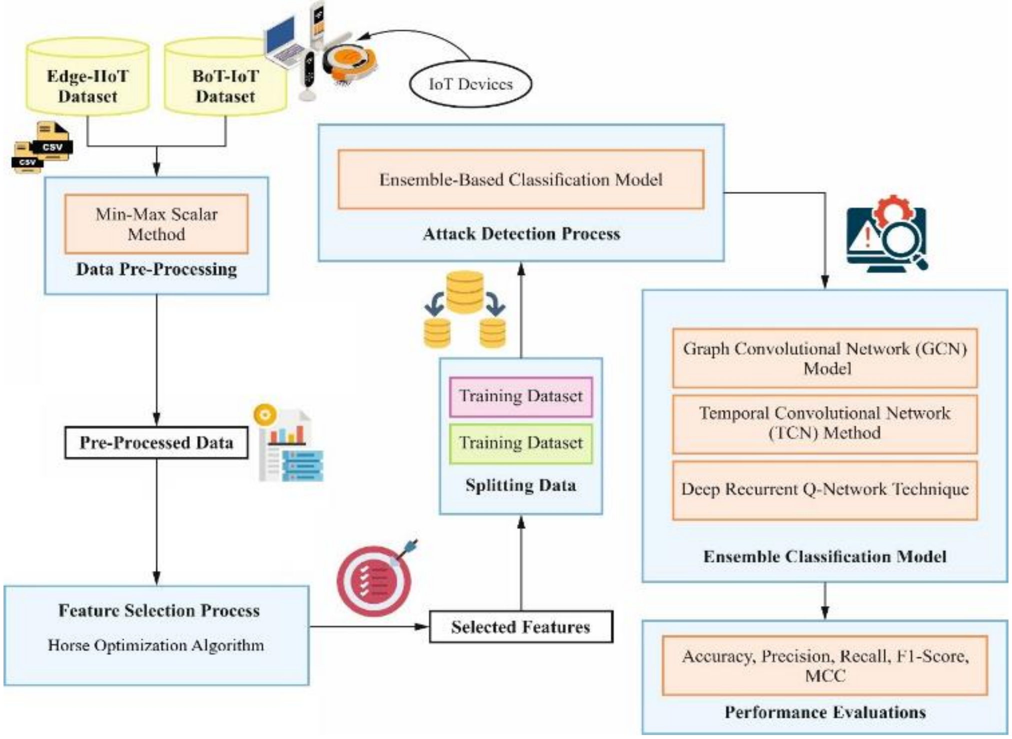
Edge-IIoT Dataset
BoT-IoT Dataset
IoT Devices
CSV
CSV
Min-Max Scalar Method
Data Pre-Processing
Ensemble-Based Classification Model
Attack Detection Process
Pre-Processed Data
Training Dataset
Training Dataset
Splitting Data
Graph Convolutional Network (GCN) Model
Temporal Convolutional Network (TCN) Method
Deep Recurrent Q-Network Technique
Ensemble Classification Model
Feature Selection Process
Horse Optimization Algorithm
Selected Features
Accuracy, Precision, Recall, F1-Score, MCC
Performance Evaluations

Fig. 2 Overall flow of EIDS-DRECM model

3.2. Feature Selection Process using HOA

The HOA is employed for electing an optimal set of features. The HOA is stimulated by social, behavioral, and environmental patterns of horses [24]. These models consist of social grazing, activity, hierarchy, defense mechanism, and imitation through different phases of a horse's life. The HOA utilizes these behaviors to mimic the optimizer process, using the movement of the horse through iterations as described by the succeeding Equation (2):

$$X_m^{Iter,AGE} = \vec{V}_m^{Iter,AGE} + X_m^{(iter-1),AGE} \; AGE = \alpha, \beta, \delta \qquad (2)$$

Here, $X_m^{Itmer,AGE}$ signifies the location of the mth horse at the provided iteration ($Iter$) and specific age range (AGE), whereas $V_m^{Iter,AGE}$ refers to a consistent vector of velocity of the mth horse. The population of the horse is separated into 4 age groups, all of which show dissimilar behavioral qualities: δ characterizes horses aged between 0-5 years, y symbolizes horses between the ages of 5 to 10 years, β means horses within the age bracket of 10 to 15 years, α exemplifies horses above 15 years old.

In all iterations, they are graded depending on resolving the optimizer issue, and the population was separated into 4 age groups based on the hierarchical matrix. The highest 10 percent of horses are categorized as α. The following 20 percent are assembled as β, accompanied by 30 % as γ, and the outstanding 40 % as δ, signifying the younger horses. This stratum guarantees a balance of youth and experience, improving the model's exploitative and exploratory abilities. The horse's movements are imitated through 6 main behavioral features: herd behavior, grazing, defense, hierarchy, arbitrary movement, and imitation.

$$\vec{V}_m^{Iter,\alpha} = \vec{C}_m^{Iter,\alpha} + \vec{D}_m^{Iter,\alpha}$$

$$\vec{V}_m^{Iter,\beta} = \vec{C}_m^{Iter,\beta} + \vec{H}_m^{Iter,\beta} + \vec{S}_m^{Iter,\beta} + \vec{D}_m^{Iter,\beta}$$

$$\vec{V}_m^{Iter,\gamma} = \vec{C}_m^{Iter,\gamma} + \vec{H}_m^{Iter,\gamma} + \vec{S}_m^{Iter,\gamma} + \vec{I}_m^{Iter,\gamma} + \vec{D}_m^{Iter,\gamma} + \vec{R}_m^{Iter,\gamma} \qquad (3)$$

$$\vec{V}_m^{Iter,\beta} = \vec{C}_m^{Iter,\beta} + \vec{H}_m^{Iter,\beta} + \vec{S}_m^{Iter,\beta} + \vec{D}_m^{Iter,\beta}$$

$$\vec{V}_m^{Iter,\delta} = \vec{C}_m^{Iter,\delta} + \vec{I}_m^{Iter,\delta} + \delta + \vec{D}_m^{Iter,\beta}$$

All equations combine numerous behavioral features, Social behavior (S), comprising Grazing (G), Random exploration (R), Imitation of other horses (I), Hierarchical movement (H), and Defense mechanism (D). The behavior diversities used to dissimilar age groups guarantee that the HOA model reaches a balance among exploitation (refining the present optimal solutions) and exploration (searching novel places within the solution area) stages of the optimizer procedure. Older horses (α) concentrate more on exploitative behaviors such as defense and grazing, whereas young horses (δ) rank exploration over random and simulated movement. This adaptive tactic lessens the computational efficiency while preserving the model's strength through dissimilar kinds of optimizer difficulties. By mimicking horse behavior in this age-graded and hierarchical way, the HOA model may effectively navigate composite solution areas, guaranteeing a wide-ranging exploration of likelihoods and efficient convergences to best solutions.

This paradigm integrates objectives into a singular formulation, wherein a specified weight categorizes the value of each aim. The fitness function that integrates both aims of feature selection is as follows.

$$Fitness(X) = \alpha \cdot E(X) + \beta * \left(1 - \frac{|R|}{|N|}\right) \qquad (4)$$

Here, Fitness(X) signifies the fitness value of a subset X, $E(X)$ denotes the classification error rate in the X subset, $|R|$ and $|N|$ are the no. of selected and original features, respectively, α and β are the weights of the error of the classifier and the reduction proportion.

3.3. Ensemble of Classification Models

At last, the ensemble models of the GCN approach, the TCN model, and the DRQN classifier are used for the attack detection process.

3.3.1. GCN Method

GCN depicts a kind of neural network intended to process data structured as a graph [21]. It involves generalizing the convolution operation from a regular grid structure to an irregular graph domain. This is accomplished by iteratively combining data from the local neighborhood node and then converting the aggregated data, thus learning the dominant representation of the node. GCN operates on graph data depicted as $G = (V, E, A)$. Now, $V \in R^{n \times d}$ creates a matrix for node features, where d indicates the dimensionality of features related to every node and n signifies the overall number of nodes. E depicts a set of edge-connected nodes, and A indicates an adjacency matrix. The input graph framework was originally employed to employ the KNN model.

$$\min(x_i) = KNN(k, x_i, \beta) \qquad (5)$$

Now β indicates a sample subset from the neighbors that are chosen, and k represents the nearest neighbor counts to retrieve. In the KNN graph, the weighted e_{ij} of edge-connected nodes x_i and x_j are calculated to employ a Gaussian kernel to reflect the similarity of the feature:

$$e_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\gamma^2}\right) for \; x_j \in \min(x_i) \qquad (6)$$

Now e_{ij} depicts an edge weight, and γ refers to a parameter that regulates the bandwidth of the Gaussian kernel.

The GCN method is widely classified into spectral models and spatial techniques that describe convolution directly on graph topology. Assume that $G = (V, E, A)$, the graph Laplacian L was specified as:

$$L = D - A \qquad (7)$$

Here A refers to the adjacency matrix, D represents the degree matrix, D_{ii} indicates the degree of node i.

$$D_{ii} = \sum_{j=1}^{n} A_{ij} \tag{8}$$

$$L_{sym} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{9}$$

Now A indicates an adjacency matrix, D represents a diagonal degree matrix, and I depicts an identity matrix. L_{ym} denotes a real symmetric matrix:

$$L_{sym} = U \Lambda U^{T} \tag{10}$$

Here U refers to a matrix, and Λ indicates the diagonal matrix. The matrix U contains an eigenvectors that act as the principle of Graph Fourier Transform (GFT). It converts a graph signal $y \in R^{n}$ into the domain y:

$$\hat{y} = U^{T} y \tag{11}$$

On the other hand, the Inverse GFT (IGFT) rebuilds the spatial signal from its spectral depiction:

$$\hat{y} = U \hat{y} \tag{12}$$

The GFT decays the graph signal into elements related to diverse graph frequencies, analogous to the conventional FT. This transformation is intended for sensor signals onto different patterns that are separable to examine their spectral coefficient $\hat{y}$. Spectral graph convolution is described as employing a filter g_{θ} promptly.

$$x * g_{\theta} = U g_{\theta}(\Lambda) U^{T} x \tag{13}$$

Now $g_{\theta}(\Lambda)$ depicts a filter function g_{θ}, parameterized by learnable weight θ, is utilized element-wise on the diagonal eigenvalue matrix Λ.

$$Z = \sigma\left(\hat{A} X W\right) \tag{14}$$

Here $X \in R^{n \times d_{in}}$ indicates the input node feature matrix with n nodes and d_{in} is an input feature; $A \in R^{d_{in} \times d_{out}}$ signifies the preprocessed adjacency matrix and $W \in R^{d_{in} \times d_{out}}$ refers to the layer-specific trainable weighted matrix, renovating the feature from the dimension d_{in} to d_{out}. σ indicates a non-linear activation function employed for element-wise and $Z \in R^{d_{in} \times d_{out}}$ refers to the resultant feature matrix of output, depicting node embedding once the graph activation and convolution are over.

3.3.2. TCN Classifier

A TCN is a kind of DL method applied to process sequence data [22]. This model removes features from the data sequence over a sequence of 1D convolutional layers. In comparison with RNNs, this model increases training efficacy over parallel calculation and deals with the problem of vanishing gradients in longer sequence dependencies. The causal convolution guarantees the temporal sequence of an input order, dilated convolution expands the receptive area, the residual block improves the prognostic precision of the method, and, in addition, it combines Dropout layers to avoid overfitting. This model contains dual basic features: dilated and causal convolutions. Causal convolution guarantees that an output only relies on present and preceding inputs, preserving time series sequence consistency. Dilated convolution presents a gap in the convolution calculation, permitting the receptive area of the kernel of the convolutional layer to develop without improving the parameter counts or computing cost, thus taking long sequence dependencies.

The fundamental equation of the TCN method is defined as demonstrated:

3.3.3. Causal Convolution

$$y[t] = \sum_{k=0}^{K-1} \omega\, [k] \cdot x[t-k] \tag{15}$$

Whereas y characterizes the output of the convolution, x refers to an input signal at $t - k$, ω refers to the parameter of the convolutional kernel, and K is the convolution kernel size.

3.3.4. Dilated Convolution

$$y[t] = \sum_{k=0}^{K-1} \omega\, [k] \cdot x[t - d \cdot k] \tag{16}$$

Now, d refers to the rate of dilation, which is capable of controlling the receptive area dimensions by fine-tuning d.

3.3.5. DRQN Model

Q-Learning is a traditional RL model, which approximates the long-term predictable implementation of reward [23]. Such projected longer-term rewards are referred to as Q-values. The highest Q-value $Q(s_{it}, a_{it})$ specifies an action a_{it} is considered that producing an improved, longer-term reward specified by the state s_{it}.

For the period comprising a well-organized course of state, action, and reward $\{s_{it}, a_{it}, r_{it}, s_{i(t+1)}\}$ gained by implementing the action a_{it} at tth interaction, Q-value $Q(s_{it}, a_{it})$ is described as a noticeable instant reward r_{it} adds the max Q-value through every action in the following state $s_{i(t+1)}$:

$$Q(s_{it}, a_{it}) = r_{it} + \gamma \max_{a_{i(t+1)}} Q(s_{i(t+1)}, a_{i(t+1)} \tag{17}$$

Whereas $\gamma \in [0,1]$ refers to the factor of discount, which balances present and upcoming rewards. Therefore, the aim of the agent of RL is for learning a policy $S \rightarrow A$ that selects the action a_t that results in the best Q value in all states. During various real-time situations, the area that defines the action and state is frequently very large to approximate Q values for each $|S| \times |A|$ pair. Therefore, like a flexible and general method, the Deep Neural Network (DNN) has been utilized for learning the composite state model. Accordingly, incorporating Q-Learning using the DNN (for example, Deep Q-Network or DQN) is applied to help compose input. The models were accepted by investigators to improve the sequential promotion model. Particularly, the DQN model, like an NN with parameters θ, Q values may be evaluated utilizing the function $(s_{it}, a_{it}|\theta)$. For training such an NN, the parameters θ are selected to reduce the temporal change through the succeeding function of loss:

$$\mathcal{L}(S, A, R|\theta) = \sum_{i,t} (Q(s_{it}, a_{it}|\theta) - \left(r_{it} + \gamma \max_{a_{i(t+1)}} Q(s_{i(t+1)}, a_{i(t+1)}|\theta)) \right)^2 \qquad (18)$$

The fundamental notion of DQN is the Markov Decision Process (MDP), whereas the upcoming condition depends only on the recent state and action: $\left(s_{i(t+1)} \middle| s_{it}, a_{it}, s_{i_1}, a_{i1}\right) = P\left(s_{i(t+1)} \middle| s_{it}, a_{it}\right)$. Such a property of Markov, nevertheless, occasionally has in real-time, particularly for the state of the prospect, occasionally be completely defined by simply the latest observations. To establish the feasibility and benefit of proposing DRQN, an incorporation of a Recurrent Neural Network (RNN) and Deep Q-Learning. Unlike DQN, which just applies the latest frames to encapsulate the environmental conditions, DRQN uses an RNN for processing sequential frames in order to encapsulate the state expansively. Following the previous research, might accept DRQN for learning the optimum targeting policies.

Historical Interaction data $\{hi_{i_1}, hi_{i_2}, \ldots, hi_{it}\}$ related to prospect i till the tth interaction should be handled by RNN to acquire the prospect of s_{it}. Formerly, forecast the Q values of explaining dissimilar activities according to the state s_{it}.

RNN is applied for handling historical interactions, while hi_{it} defines an agent's tth interaction with prospect i, which incorporates the context interaction attributes. With a sequence of interaction $\{hi_{i_1}, hi_{it}\}$ related to prospect i, GRU is used for processing the interaction sequence for learning the state s_{it}:

$$s_{it} = GRU\left(hi_{i_1}, hi_{i_2}, hi\right) \qquad (19)$$

Also, combine Dueling Network Architecture and Double Q-Learning into the DRQN method.

4. Experimental Results

The simulation validation of the EIDS-DRECM model is examined in Edge-IIoT [24] and BoT-IoT [25] databases. The Edge-IIoT database contains 24000 records with 12 types of events. There is a total of 63 attributes, and only 30 are chosen. The BoT-IoT database holds 2056 instances with five classes. There are 34 features in total, but only 22 are chosen.

Table 1 exhibits the attack detection of the EIDS-DRECM system at the Edge-IIoT database. Under 70% TRPHE, the proposed EIDS-DRECM model obtains average $accur_y$ of 99.41%, $preci_n$ of 96.44%, $recal_l$ of 96.44%, $F1_{Score}$ of 96.44%, and MCC of 96.12%. Likewise, at 30% TSPHE, the proposed EIDS-DRECM model gets average $accur_y$ of 99.47%, $preci_n$ of 96.83%, $recal_l$ of 96.84%, $F1_{Score}$ of 96.83%, and MCC of 96.55%.

Table 1. Attack detection of EIDS-DRECM on Edge-IIoT database

Classes	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**MCC**
TRPHE (70%)					
Normal	99.80	98.95	98.74	98.84	98.74
DDoS-UDP	99.81	98.52	99.22	98.87	98.77
DDoS-ICMP	99.90	99.56	99.20	99.38	99.33
SQL injection	99.87	99.36	99.08	99.22	99.15
DDoS-TCP	99.91	99.56	99.35	99.46	99.41
Password	99.83	99.13	98.78	98.95	98.86
DDoS-HTTP	99.82	98.99	98.78	98.89	98.79
Uploading	99.89	99.21	99.49	99.35	99.29
Backdoor	99.85	98.94	99.22	99.08	99.00
XSS	99.83	98.79	99.21	99.00	98.91
Ransomware	99.85	99.03	99.24	99.13	99.05
Fingerprinting	99.87	99.34	99.05	99.19	99.12
Average	**99.85**	**99.12**	**99.11**	**99.11**	**99.03**
TSPHE (30%)					
Normal	99.88	99.13	99.30	99.21	99.14
DDoS-UDP	99.83	98.81	99.15	98.98	98.89
DDoS-ICMP	99.92	99.19	99.84	99.52	99.47
SQL injection	99.93	99.83	99.32	99.57	99.53
DDoS-TCP	99.88	98.88	99.68	99.28	99.21
Password	99.86	99.02	99.34	99.18	99.11
DDoS-HTTP	99.81	99.17	98.52	98.84	98.74
Uploading	99.82	99.51	98.38	98.94	98.85
Backdoor	99.86	98.65	99.66	99.15	99.08
XSS	99.82	99.33	98.51	98.92	98.82
Ransomware	99.88	99.46	98.92	99.19	99.12
Fingerprinting	99.86	99.06	99.37	99.21	99.14
Average	**99.86**	**99.17**	**99.17**	**99.17**	**99.09**

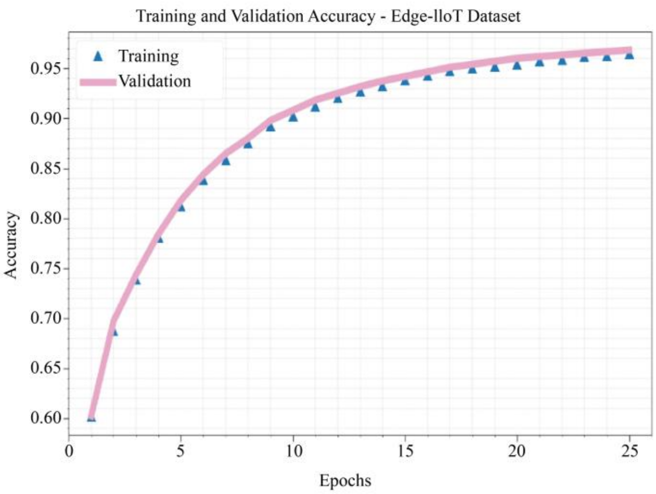

Fig. 3 $Accu_y$ curve of the EIDS-DRECM technique on the Edge-IIoT database

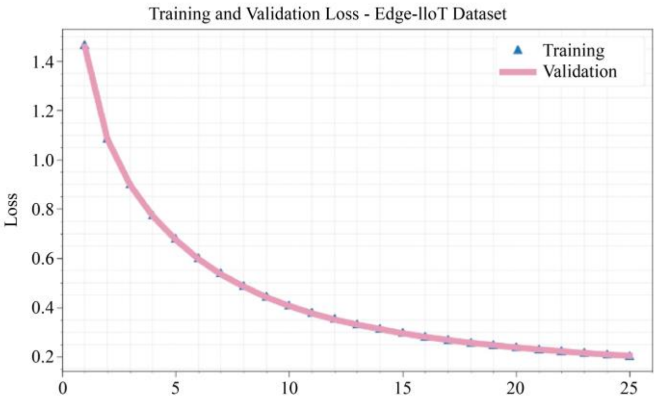

Fig. 4 Loss curve of EIDS-DRECM model on Edge-IIoT database

Figure 3 exemplifies the training (TRAIN) $accu_y$ and validation (VALID) $accu_y$ of an EIDS-DRECM method at the Edge-IIoT database over 25 epochs. Around the epoch**,** the VALID $accu_y$ minimally exceeds the training accuracy, indicating good generalization without over-fitting. As training advances, it reflects maximum performance and a minimum performance gap between TRAIN and VALID.

Figure 4 exemplifies the TRAIN and VALID losses of the EIDS-DRECM model at the Edge-IIoT database over 25 epochs. The close correlation between the TRAIN and VALID loss curves during training indicates that the model has not overfitted and maintains strong generalization to unseen data.

Table 2 portrays the comparative study of the EIDS-DRECM technique at the Edge-IIoT database with current techniques [26-29]. The outcomes underlined that the proposed EIDS-DRECM model got the highest $accur_y$, $preci_n$, $recal_l$, and $F1_{Score}$ of 99.47%, 96.83%, 96.84%, and 96.83%, respectively. The present methodologies, such as LR, Naïve Bayes, Decision Tree, LSTM, BiGRU, DNN, and RNN, have shown worse performance under various metrics.

Table 2. Comparative analysis of EIDS-DRECM methodology on Edge-IIoT database

Edge-IIoT Database				
Technique	**Accuracy**	**Precision**	**Recall**	**F1-Score**
LR Model	96.16	91.14	95.25	89.00
Naïve Bayes	92.98	94.52	91.11	91.73
Decision Tree	93.17	89.06	91.50	90.99
LSTM	91.94	92.94	91.90	95.90
BiGRU	97.69	89.56	94.11	89.80
DNN	92.62	93.26	89.82	93.57
RNN	93.04	90.11	96.50	95.04
EIDS-DRECM	99.86	99.17	99.17	99.09

Table 3 presents the attack detection of the EIDS-DRECM system at the BoT-IoT database. Under 70% TRPHE, the proposed EIDS-DRECM model gets average $accur_y$ of 99.25%, $preci_n$ of 98.11%, $recal_l$ of 96.61%, $F1_{Score}$ of 97.31%, and MCC of 96.86%. Similarly, at 30% TSPHE, the proposed EIDS-DRECM model obtains average $accur_y$ of 99.35%, $preci_n$ of 97.95%, $recal_l$ of 98.66%, $F1_{Score}$ of 98.29%, and MCC of 97.88%.

Table 3. Attack detection of the EIDS-DRECM technique on the BoT-IoT database

Class Labels	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**MCC**
TRPHE (70%)					
DDoS	99.17	98.54	97.97	98.26	97.71
DoS	98.89	97.45	98.01	97.73	96.99
Recon	99.24	98.55	98.26	98.40	97.90
Theft	99.51	98.00	89.09	93.33	93.20
Normal	99.44	97.99	99.71	98.84	98.48
Average	**99.25**	**98.11**	**96.61**	**97.31**	**96.86**
TSPHE (30%)					
DDoS	99.51	99.35	98.71	99.03	98.71
DoS	99.35	98.01	99.33	98.67	98.24
Recon	99.03	99.34	96.77	98.04	97.41
Theft	99.84	96.00	100.00	97.96	97.90
Normal	99.03	97.06	98.51	97.78	97.16
Average	**99.35**	**97.95**	**98.66**	**98.29**	**97.88**

Figure 5 establishes the TRAIN $accu_y$ and VALID $accu_y$ of an EIDS-DRECM method at the BoT-IoT database over 25 epochs. The proximity of both curves during training indicates that the approach is effectively regularized and generalized. Figure 6 establishes the TRAIN and VALID losses of the EIDS-DRECM model at the BoT-IoT database over 25 epochs. As training progresses, both losses persistently decline, displaying that the model is efficaciously learning and enhancing its parameters.

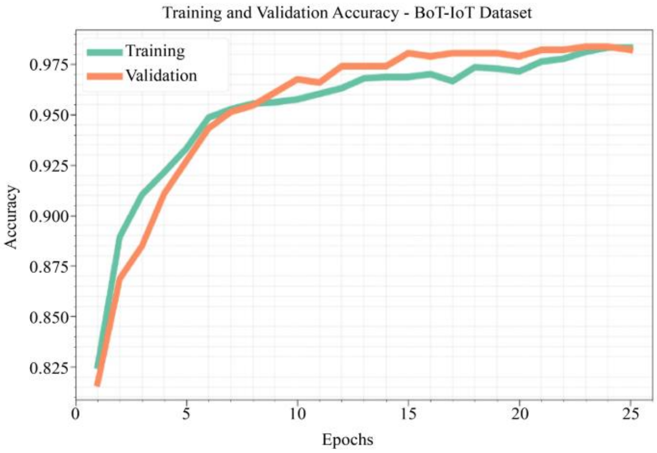

Fig. 5 $Accu_y$ curve of the EIDS-DRECM model on the BoT-IoT database

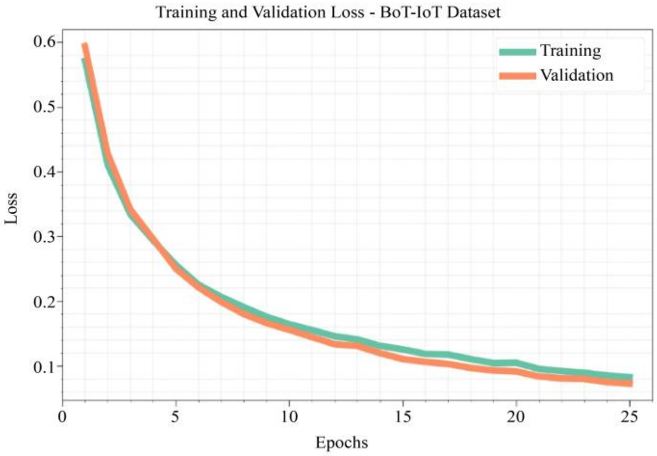

Fig. 6 Loss curve of the EIDS-DRECM model on the BoT-IoT database

Table 4 represents the comparative analysis of the EIDS-DRECM methodology at the BoT-IoT database with current approaches. The outcomes underscored that the proposed EIDS-DRECM model got the highest $accur_y$, $preci_n$, $recal_l$, and $F1_{Score}$ of 99.35%, 97.95%, 98.66%, and 98.29%, respectively.

Table 4. Comparative analysis of the EIDS-DRECM model on the BoT-IoT database

BoT-IoT Database				
Techniques	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Random Forest	96.81	97.55	89.33	89.07
MLP	93.52	89.21	92.04	93.73
ARNN	93.90	90.35	92.67	97.64
1D CNN	92.68	91.74	96.70	91.59
Attention-based GRU	98.36	94.09	89.00	92.21
VGG16+PSO-PCA	93.35	96.53	97.76	91.80
DNN-MIFS	93.63	97.72	92.19	90.83
EIDS-DRECM	99.35	97.95	98.66	98.29

However, the compared methodologies, such as Random Forest, MLP, ARNN, 1D CNN, Attention-based GRU, VGG16+PSO-PCA, and DNN-MIFS, have shown worse performance under various metrics. Therefore, the EIDS-DRECM approach gains superior outcomes.

5. Conclusion

In this manuscript, an EIDS-DRECM methodology is developed for IoT Devices. The main intention of the EIDS-DRECM approach is to enhance cybersecurity in IoT networks by developing efficient threat detection and mitigation mechanisms. It contains distinct kinds of phases, such as min-max scaler, filter-based feature reduction, and an ensemble of classification models. The min-max scaler approach has been employed in data preprocessing to transform and structure raw data into an appropriate format. The HOA is employed to preserve the most pertinent features from the data. At last, the ensemble models of GCN, TCN, and DRQN are used for the attack detection process. An extensive set of simulations was involved in exhibiting the promising results of the EIDS-DRECM method. The experimentation results inferred the proficient performance of the EIDS-DRECM system in the attack detection process.

Data Availability

The data that support the findings of this study are openly available in the Kaggle repository at https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot and https://www.kaggle.com/datasets/vigneshvenkateswaran/bot-iot, reference numbers [24, 25].

References

[1] Gangadhar Sadaram et al., "Internet of Things (IoT) Cybersecurity Enhancement through Artificial Intelligence: A Study on Intrusion Detection Systems," *Universal Library of Engineering Technology*, pp. 1-9, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Xavier Larriva-Novo et al., "An IoT-Focused Intrusion Detection System Approach Based on Preprocessing Characterization for Cybersecurity Datasets," *Sensors*, vol. 21, no. 2, pp. 1-15, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] Xiao Chun Yin et al., “Toward an Applied Cyber Security Solution in IoT-Based Smart Grids: An Intrusion Detection System Approach,” *Sensors*, vol. 19, no. 22, pp. 1-22, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[4] Syeda Nazia Ashraf et al., “IoT Empowered Smart Cybersecurity Framework for Intrusion Detection in Internet of Drones,” *Scientific Reports*, vol. 13, pp. 1-20, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Albara Awajan, “A Novel Deep Learning-Based Intrusion Detection System for IoT Networks,” *Computers*, vol. 12, no. 2, pp. 1-17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Arash Heidari, and Mohammad Ali Jabraeil Jamali, “Internet of Things Intrusion Detection Systems: A Comprehensive Review and Future Directions,” *Cluster Computing*, vol. 26, pp. 3753-3780, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Irfan Ali Kandhro et al., “Detection of Real-Time Malicious Intrusions and Attacks in IoT Empowered Cybersecurity Infrastructures,” *IEEE Access*, vol. 11, pp. 9136-9148, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Xuan-Ha Nguyen et al., “Realguard: A Lightweight Network Intrusion Detection System for IoT Gateways,” *Sensors*, vol. 22, no. 2, pp. 1-18, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Eirini Anthi et al., “A Supervised Intrusion Detection System for Smart Home IoT Devices,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9042-9053, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] Mohammed I. Alghamdi, “An Investigation into the Effect of Cybersecurity on Attack Prevention Strategies,” *Journal of Cybersecurity and Information Management*, vol. 3, no. 2, pp. 53-60, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] Jawhara Aljabri, “Attack Resilient IoT Security Framework using Multi Head Attention based Representation Learning with Improved White Shark Optimization Algorithm,” *Scientific Reports*, vol. 15, pp. 1-21, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[12] Haitham Assiri, “Piranha Foraging Optimization Algorithm with Deep Learning Enabled Fault Detection in Blockchain-Assisted Sustainable IoT Environment,” *Sustainability*, vol. 17, no. 4, pp. 1-124, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[13] Mimouna Abdullah Alkhonaini et al., “Sandpiper Optimization with Hybrid Deep Learning Model for Blockchain-Assisted Intrusion Detection in Iot Environment,” *Alexandria Engineering Journal*, vol. 112, pp. 49-62, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[14] Karam Hammadeh, M. Kavitha, and Nadim Ibrahim, “Enhancing Cybersecurity in Software-Defined Networking: A Hybrid Approach for Advanced DDoS Detection and Mitigation,” *The Social and Ethical Implications of Nanotechnology and Engineering*, vol. 20, no. S4, pp. 514-529, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Mohamed Amine Ould Rabah et al., “Detection and Mitigation of Distributed Denial of Service Attacks Using Ensemble Learning and Honeypots in a Novel SDN-UAV Network Architecture,” *IEEE Access*, vol. 12, pp. 128929-128940, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Mohamed Ali Setitra et al., “Optimized MLP-CNN Model to Enhance Detecting DDoS Attacks in SDN Environment,” *Network*, vol. 3, no. 4, pp. 538-562, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Xue Jun Li, Maode Ma, and Yihan Sun, “An Adaptive Deep Learning Neural Network Model to Enhance Machine-Learning-Based Classifiers for Intrusion Detection in Smart Grids,” *Algorithms*, vol. 16, no. 6, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] S. Kavitha, N. Uma Maheswari, and R. Venkatesh, “Intelligent Intrusion Detection System using Enhanced Arithmetic Optimization Algorithm with Deep Learning Model,” *Technical Gazette*, vol. 30, no. 4, pp. 1217-1224, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[19] Madilyn Louisa, Gumgum Darmawan, and Bertho Tantular, “Enhancing Stock Price Forecasting with CNN-BiGRU-Attention: A Case Study on INDY,” *Mathematics*, vol. 13, no. 13, pp. 1-16, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[20] Hamid Kardan Moghaddam et al., “Sustainable Water Allocation Under Climate Change: Deep Learning Approaches to Predict Drinking Water Shortages,” *Journal of Environmental Management*, vol. 385, pp. 1-20, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[21] Fan Li et al., “Rolling Bearing Fault Diagnosis via Temporal-Graph Convolutional Fusion,” *Sensors*, vol. 25, no. 13, pp. 1-19, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[22] Yongjie Wan et al., “Prediction of Typical Power Plant Circulating Cooling Tower Blowdown Water Quality Based on Explicable Integrated Machine Learning,” *Processes*, vol. 13, no. 6, pp. 1-14, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[23] Yicheng Song, Wenbo Wang, and Song Yao, “Customer Acquisition via Explainable Deep Reinforcement Learning,” *Information Systems Research*, vol. 36, no. 1, pp. 1-50, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[24] Mohamed Amine Ferrag et al., “Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning,” *IEEE Access*, vol. 10, pp. 40281-40306, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] Nickolaos Koroniotis et al., “Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset,” *Future Generation Computer Systems*, vol. 100, pp. 779-796, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[26] Urikhimbam Boby Clinton, and Nazrul Hoque, “MU-IoT: A New IoT Intrusion Dataset for Network and Application Layer Attacks Analysis,” *IEEE Access*, vol. 12, pp. 166068-166092, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[27] Erol Gelenbe, and Mert Nakip, “IoT Network Cybersecurity Assessment with the Associated Random Neural Network,” *IEEE Access*, vol. 11, pp. 85501-85512, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[28] Prabu Kaliyaperumal et al., “Enhancing Cybersecurity in Agriculture 4.0: A High-performance Hybrid Deep Learning-based Framework for DDoS Attack Detection,” *Computers and Electrical Engineering*, vol. 126, pp. 1-30, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[29] Tasneem Qasem Al-Ghadi et al., “Leveraging Federated Learning for DoS Attack Detection in IoT Networks based on Ensemble Feature Selection and Deep Learning Models,” *Cyber Security and Applications*, vol. 3, pp. 1-19, 2025. [CrossRef] [Google Scholar] [Publisher Link]