International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 15, No. 3, September 2026, pp. 1154~1166
ISSN: 2252-8776, DOI: 10.11591/ijict.v15i3.pp1154-1166

Enhanced anomaly detection in IoT networks via feature fusion and learning-based echo state networks

P. Palpandi[1], B. Sakthivel[2], M. Ponnrajakumari[3], M. Indirani[4], S Govindaraju[5], S. Deivasigamani[6]
[1]Department of Computer Science and Engineering (IoT), Sethu Institute of Technology, Vidudhunagar, India
[2]Department of Electronics and Communication Engineering, Pandian Saraswathi Yadav Engg College, Sivagangai, India
[3]Department of Electronics and Communication Engineering, Velammal Engineering College, Chennai, India
[4]Department of Computer Science and Business Systems, Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Avadi, India
[5]Department of Computer Science, School of Science and Humanities, St. Joseph University, Chennai, India
[6]Faculty of Engineering, Technology & Built Environment, UCSI University, Kuala Lumpur, Malaysia

Article Info

Article history:

Received Aug 12, 2025
Revised Apr 14, 2026
Accepted May 17, 2026

Keywords:

Black eagle optimizer
Intrusion detection systems
Internet of things
Learning-based echo state
Network

ABSTRACT

The fast development of internet of things (IoT) networks has led to an increased probability of cyberattacks. Intrusion detection systems (IDS) are needed for identifying unauthorised access and malicious activities in such dynamic environments. However, existing machine learning (ML) models failed to handle the complexity and variability of modern cyber threats. In this work, a hybrid deep learning (DL)-based anomaly detection model is presented for IoT cybersecurity. The model combines three types of features: (i) supervised feature extraction using linear discriminant analysis (LDA) to extract the most discriminative features, (ii) unsupervised feature learning through autoencoders to capture latent representations of the input data, and (iii) statistical features such as mean, variance, skewness, and kurtosis to learn input characteristics. The fused feature matrix is fed into a learning-based echo state network (LBESN) for final detection. The parameters of the LBESN model are tuned using black eagle optimizer (BEO). Experimental results on standard intrusion detection datasets such as UNSW-NB15, KDD99, and InSDN show that the proposed model achieves superior performance in terms of accuracy, precision, recall, and F1-score compared to conventional DL techniques.

Corresponding Author:

P. Palpandi
Department of Computer Science and Engineering (IoT), Sethu Institute of Technology
Vidudhunagar, India
Email: palpandi@sethu.ac.in

1. INTRODUCTION

The internet of things (IoT) has emerged as a transformative paradigm. This innovation interconnects billions of physical devices, sensors, and intelligent systems through the internet [1]. IoT is used for continuous data exchange and decision-making across different domains such as smart cities, healthcare, agriculture, transportation, and industrial automation. However, the performance of IoT networks is degraded by various cybersecurity vulnerabilities [2], [3]. With the growth of IoT, cybersecurity has become a critical concern. It involves various attacks like distributed denial of service (DDoS) attacks, malware propagation, botnet recruitment, and data exfiltration. Compared to conventional computing environments, IoT networks involve massive, diverse, and dynamic data streams. The development of real-time and accurate intrusion detection remains a difficult challenge.

To overcome these drawbacks, anomaly-based intrusion detection systems (A-IDS) have gained attention in the cybersecurity community [4]. These systems use learning algorithms to identify deviations from normal behavior. This learning approach is applied for classifying both predefined and novel attacks. The machine learning (ML) and deep learning (DL) techniques have recently gained attention for their effectiveness in A-IDS for the classification of anomalous activities within network environments [5]. In DL approaches, the well-known models like long short-term memory (LSTM) networks, convolutional neural networks (CNN), and Autoencoders are used to handle complex and high-dimensional data. These models involve different learning paradigms, like supervised, unsupervised, and semi-supervised learning. This provides flexibility in adapting to various types of cybersecurity challenges. However, conventional models struggle to capture the nonlinear relationship of network data. Also, the temporal and high-dimensional patterns of IoT traffic data cannot be effectively processed by existing models. Furthermore, the models based on handcrafted features failed to represent the complexity of malicious behaviors across different IoT protocols and environments.

To fill this gap, a hybrid DL model is proposed for IoT cybersecurity applications. The proposed model integrates three different modules for feature extraction. It involves linear discriminant analysis (LDA) and Autoencoders for hidden feature learning. Also, it uses Statistical Descriptors for feature extraction. These features are combined to capture both the statistical behavior and semantic patterns in the network traffic. Then, the fused features are processed by a learning-based echo state network (LBESN) for final detection.

2. RELATED WORK

Deshmukh and Bhaladhare [6] propose an attacker classification model for the network using a deep belief network (DBN). To improve accuracy, the DBN model is trained using taylor-spider monkey optimization (Taylor-SMO). Experimental results on the NSL-KDD dataset show that the optimization process increases false positive rate (FPR) and false negative rate (FNR) by choosing the most relevant features. Manimurugan *et al.* [7] propose a DBN-based IDS and evaluate it using the CICIDS 2017 dataset. It attains 94.28% accuracy for the normal class, with other attack classes varying from 93.5% to 95.8%.

Aburasain [8] propose black widow optimization (BWO) based hybrid IDS for attacker prevention in IoT-based precision agriculture systems. The developed model uses the bald eagle search (BES) optimizer for feature selection and BWO for parameter tuning to achieve higher security and reliability. A generative DL model-based IDS is developed by Abdalgawad *et al.* [9]. The hybrid model of adversarial autoencoders (AAE) and bidirectional GANs (BiGAN) is used to detect network attackers. Results on the IoT-23 dataset with 1.8 million network flows show that the AAE with the BiGAN model attained an F1-rate of 0.89 with an accuracy of 0.93. Nayak and Bhattacharyya [10] propose an IDS for IoT networks using the enhanced shuffle bidirectional channel attention-based network (ESBCA-Net). It uses reformed histogram equalisation and feature optimization via an improved honey badger optimiser to achieve higher accuracy compared to existing methods. Ali and Yousaf [11] propose a three-tier IDS for intrusion detection based on DL models. The DL model, combined with type-II fuzzy filtering, is used to classify packets into normal, suspicious, and malicious. Fernando *et al.* [12] proposed an intruder prevention model using ensemble learning. It combines XGBoost, gradient boosting, and decision tree to classify malicious activities. Results on the Bot-IoT dataset show that the ensemble model achieves an accuracy of up to 93% for binary classification and 92% for multiclass categorisation. Aljuhani *et al.* [13] propose a private blockchain-based secure communication technique for IoT using session-based mutual verification and key contract. In addition, attention-based bidirectional LSTM (Abi-LSTM) networks are applied for cyber-attack detection in networks. Ullah and Mahmoud [14] proposed an IDS model using a multi-dimensional CNN. The CNN model is formed with different 1D, 2D, and 3D layers to learn features deeply. The proposed approach achieves an accuracy level of 93.85 % compared to existing DL models. Lahasan and Samma [15] proposed a two-layer optimizer model for intruder detection. It constructs a deep autoencoder model based on the detection accuracy of the K-nearest neighbor (KNN) classifier.

A new type of DL model-based attack classification model called device-based IDS (DIDS) is proposed by Madhu *et al.* [16]. To improve the classification accuracy, the parameters of the DIDS architecture are altered using particle swarm optimization. A recurrent neural network model of LSTM-based intruder detection is proposed by Keshk *et al.* [17]. The important features from the network are identified using SPIP (shapley additive explanations, permutation feature importance, individual conditional expectation, partial dependence plot). Compared to ML models, the LSTM model shows higher accuracy on NSL-KDD, UNSW-NB15, and ToN-IoT datasets.

Ashiku and Dagli [18] present a DL-based adaptive IDS designed to identify both known and zero-day network attacks. The proposed model uses CNNs with a regularised multi-layer perceptron (MLP) to increase detection flexibility compared to traditional feed-forward networks. In intruder detection models,

deep transfer learning is an advanced ML technique that uses knowledge from a pre-trained DL model to detect different attacker patterns. Ahmad *et al.* [19] propose an anomaly detection approach for IoT networks using deep transfer learning. The EdgeIIoT dataset is transformed into an image format for CNN-based processing, and hyperparameters are tuned using Random Search.

Hizal *et al.* [20] propose an effective IDS for IoT networks using gate recurrent unit (GRU)-based DL models. The preprocessing techniques of feature selection, duplication removal, and normalization are applied to increase detection accuracy. Similarly, Nandanwar and Katarya [21] recommend a hybrid attacker detection model using GRU. At first, the features are captured using a CNN. Next, the GRU model is applied for final classification. In the N_BaIoT dataset, the CNN+GRU model reaches a high testing accuracy of 94.62% with a loss of 0.0063. Deshmukh and Ravulakollu, [22] proposed a metaheuristic optimizer combined detection model called intelligent intrusion detection network (IIDNet). The optimization strategy is used for feature extraction, selection, and parameter tuning to achieve higher attacker detection accuracy. Results on the UNSW-NB15 dataset show that IIDNet achieves the highest accuracy of 94.89%. Alars and Kurnaz [23] improve IDS performance by integrating network and host traffic data with DL techniques. Using a military-context network intrusion detection dataset, they apply a CNN with feature selection and dimensionality reduction for improved accuracy. An autoencoder combined with a MLP-based classifier model is proposed by Liu *et al.* [24]. It uses traffic features for anomaly identification, clustering, and classification. Altunay and Albayrak [25] propose three DL models: CNN, LSTM, and a hybrid CNN+LSTM to find intrusions in IoT networks. The hybrid CNN+LSTM model achieved the highest accuracy of 93.21% and 92.9% on the UNSW-NB15 dataset and 99.84% and 99.80% on the KDD dataset for binary and multi-class classification, respectively. Likewise, the authors [26], [27] propose the graph neural networks (GNN) and transformer-based models for attacker detection in networks.

3. PROPOSED MODEL

To effectively solve the complexities of cyberattacks in IoT networks, a hybrid anomaly-based intrusion detection model is proposed. In the proposed detection approach, a three-stage feature extraction process is carried out. The features are extracted using LDA, Autoencoder and statistical feature extraction. The extracted features are processed using the LBESN model for final detection. Also, the LBESN model is optimised using BEO to fine-tune its parameters for optimal detection performance. The workflow of the proposed model is shown in Figure 1.

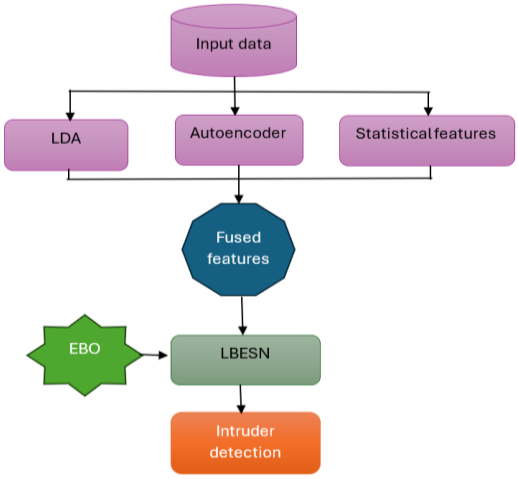

Figure 1. Workflow of proposed detection approach

3.1. Feature preprocessing

The input data collected from IoT-based networks contains missing values and inconsistent formats. Also, it includes categorical and numerical fields. In this stage, the data preprocessing is carried out to remove unwanted noise. Initially, missing values are handled using Simple Imputation. It calculates missing values using the mean value of numerical attributes. To ensure uniform scaling across features, Z-score normalisation is applied. This process centres the data around zero and scales it based on the standard deviation. This normalisation of input data can be computed as follows:

$$x' = \frac{x-\mu}{\sigma} \quad (1)$$

Where x denotes the original input, μ & σ denote the mean and standard deviation of the feature.

3.2. LDA

LDA is introduced as a supervised linear feature extraction approach. This technique projects high-dimensional input into a low-dimensional subspace and preserves the class separability. Unlike PCA, LDA focuses on maximizing variance without class information. This is needed for cybersecurity applications where distinct attack classes must be clearly separated. LDA operates by calculating the within-class scatter matrix S_W and the between-class scatter matrix S_B:

$$S_W = \sum_{i=1}^{C} \sum_{x \in D_i} (x - \mu_i)(x - \mu_i)^T, S_B = \sum_{i=1}^{C} \sum_{x \in D_i} N_i(\mu_i - \mu)(\mu_i - \mu)^T, \quad (2)$$

Then, LDA solves the generalized eigenvalue problem to find the optimal projection matrix W:

$$W = \arg max \frac{|W^T S_B W|}{|W^T S_W W|} \quad (3)$$

The projected features retain maximum class discrimination and increase the detection capability of rare attack types in IoT networks.

3.3. Autoencoder-based nonlinear feature learner

An Autoencoder is used to extract nonlinear feature representations from the preprocessed input data. The goal is to capture hidden patterns, redundancies, and correlations that traditional statistical or linear methods may overlook. It consists of two components: the encoder and decoder. The function of the encoder is to reduce the input feature vector into a lower-dimensional latent vector using a nonlinear transformation. The main function of a decoder is to recreate the original input from the compressed format. The encoder function is given by:

$$h = f(x) = \sigma(W_e x + b_e), \hat{x} = g(h) = \sigma(W_d h + b_d), L = ||x - \hat{x}||^2 \quad (4)$$

Overall, the network is trained to reduce the reconstruction error. Here, h is the latent features capturing meaningful non-linear structures, σ is the activation function (ReLU or sigmoid) and $\hat{x}$ is the reconstructed output. The latent vector h becomes a compact and informative feature set and preserves critical behavior characteristics such as protocol anomalies or abnormal traffic timing.

3.4. Statistical feature extraction module

These modules capture essential shape and distribution characteristics of input data for anomaly detection. The statistical features extracted from each input vector include mean (μ), standard deviation (σ), skewness (γ), and kurtosis (κ). The mean value is used to measure a central tendency. The standard deviation quantifies variability. The skewness indicates the asymmetry. Likewise, kurtosis captures tail heaviness or outliers. It can be framed as follows:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i, \sigma^2 = \frac{1}{n}(x_i - \mu)^2, \Upsilon = \frac{1}{n}\sum_{i=1}^{n}(\frac{x_i-\mu}{\sigma})^3, k = \frac{1}{n}\sum_{i=1}^{n}(\frac{x_i-\mu}{\sigma})^4 \quad (5)$$

These statistical metrics help identify outliers, bursts, and other anomalies common in cyberattack scenarios.

3.5. Feature fusion strategy

Instead of using a single feature processing method, the feature vectors from LDA, autoencoder, and statistical analysis are combined to increase detection accuracy:

$$X_{combined} = [X_{LDA}||X_{Autoencoder}||X_{Statistical}] \quad (6)$$

This hybrid feature vector uses the discriminative power of LDA, the nonlinear representation from the autoencoder, and the behavioral insights from statistics. The fused vector forms the final input to LIBESN.

3.6. LBESN classifier

The LBESN model is a powerful time-series classifier based on the echo state network (ESN) architecture. ESNs are a type of reservoir computing model that is highly efficient for sequential data due to their fast training and dynamic memory capability. The architecture of the LBESN model is shown in Figure 2. The LBESN contains an input layer that maps feature vector x(t) to the reservoir, and the reservoir transforms the input into high-dimensional dynamics. It can be stated as follows:

$$r(t+1) = (1-\alpha).r(t) + \alpha.\tanh\left(W_{in}x(t) + W_{res}r(t)\right) \quad (7)$$

Where, r(t) is the reservoir state at time t, W_{in} is the input weight matrix, W_{res} is the internal connection weights, α is the leak rate (controls temporal memory).

Unlike standard ESNs, LBESN incorporates bidirectional reservoirs and introduces trainable output layers via supervised learning. It is used for the network to better capture dependencies from both past and future contexts in a sequence. LBESN maintains two separate states: forward reservoir $r_f(t) +$ and reverse reservoir $r_b(t)$, both are updated using the same reservoir dynamics but potentially from opposite ends of the sequence.

$$\begin{aligned} r_f(t+1) &= (1-\alpha).r_f(t) + \alpha.\tanh\left(W_{in}x(t) + W_{res}r_f(t)\right), \\ r_b(t+1) &= (1-\alpha).r_b(t) + \alpha.\tanh\left(W_{in}x(t) + W_{res}r_b(t)\right) \end{aligned} \quad (8)$$

At the end of the sequence, both states are concatenated to form the final representation:

$$r_{final} = [r_f(T); r_b(t)] \quad (9)$$

Where T is the final time step. The output layer is a linear classifier trained using ridge regression on the reservoir states:

$$W_{out} = arg\min_{W} ||Y - WR||^2 + \lambda||W||^2 \quad (10)$$

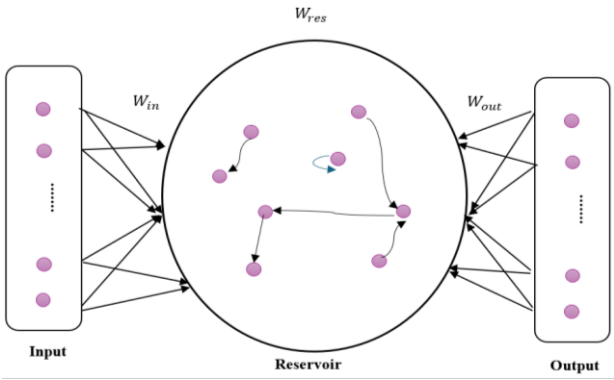

Figure 2. LBESN architecture

3.7. Black eagle optimizer (BEO)

The BEO is a bio-inspired metaheuristic algorithm developed by simulating the hunting behavior of black eagles [28]. It can be applied to complex optimization problems characterised by non-linearity, multi-modality, and large search spaces. The inspiration comes from the way black eagles cooperate, stalk, and capture their prey. This behavior is emulated in the algorithm through a multi-phase approach combining exploration (searching new regions) and exploitation (refining good solutions). These phases include Stalking, Attacking, Capturing and Returning/Homecoming.

Initialization phase:

We define an optimization problem as:

$$\text{Minimize or Maxime } f(X)=f\left(x_1, x_2 \ldots\ldots x_D\right) \quad (11)$$

Where, $X \in \mathbb{R}^D$ is the decision vector, D is the dimensionality, $x_j \in [Lb_j, Ub_j]$. Each variable lies within defined bounds. The initial population (set of black eagles) is generated randomly within the predefined search space:

$$P_{i,j} = Lb_j + rand(Ub_j - Lb_j) \quad (12)$$

Where, $rand$ ~U(0,1), $P_{i,j}$ is the position of eagle i in dimension j.

Stalking

In nature, black eagles fly at high altitudes and scan a wide area for potential prey. In BEO, this translates to a global search mechanism where each eagle updates its position based on: the best-known solution (global best), random influence to maintain diversity. It can be mathematically stated as follows:

$$P_{i,j}(t+1) = P_{i,j}(t) + c_1.r_1.\left(P_{best,j}(t) - P_{i,j}(t)\right) + c_2.r_2.\left(Ub_j - Lb_j\right).(2.rand - 1) \quad (13)$$

Where, $P_{best,j}(t)$: best eagle in dimension j, r_1 and r2 is a random variable which varies from zero to one. c_1 is the control's social attraction, c_2 is the control's random exploratory motion. This movement helps eagles converge to good areas while still searching elsewhere.

Attacking

Once the prey is spotted, the eagle begins to spiral downward in a focused but slightly unpredictable path. In optimization terms, this means the algorithm intensifies its local search around the best solutions using an oscillatory motion. It improves the chances of finding better optima in that region. It can be mathematically stated as follows:

$$P_{i,j}(t+1) = P_{best}(t) + A.\sin(w.t + \phi).(P_{mean,j}(t) - P_{i,j}(t) \quad (14)$$

Where, A is the spiral amplitude, ω is the frequency (controls oscillation), ϕ is the phase offset and $P_{mean,j}(t)$ is the mean location of the swarm. This represents non-linear and oscillatory motion. It increases local search around the best candidate.

Capturing

In nature, the eagle finally makes a sharp and precise movement to grab its prey. This phase models greedy search**,** which supports rapid convergence toward the optimal solution. It can be mathematically expressed as follows:

$$P_{i,j}(t+1) = P_{best,j}(t) + \epsilon.(P_{i,j}(t) - P_{best,j}(t) \quad (15)$$

Where, ϵ~U(−1,1 is a small perturbation). This helps fine-tune solutions near the optimum and ensures quick convergence.

Returning/Homecoming

After a successful hunt, eagles return to their nests. In BEO, this corresponds to checking and correcting any positions that have gone out of bounds. This ensures that all solutions remain feasible within the search space. It can be stated as follows:

If $P_{i,j}(t+1) \in [Lb_j, Ub_j]$, *apply,*

$$P_{i,j}(t+1) = \begin{cases} Lb_j \; if \; P_{i,j}(t+1) < Lb_j \\ Ub_j \; if \; P_{i,j}(t+1) > Ub_j \end{cases} \quad (16)$$

Alternatively, reflect or randomly reinitialize if desired:

$$P_{i,j}(t+1) = Lb_j + rand\,.(Ub_j - Lb_j) \quad (17)$$

The BEO effectively balances exploration and exploitation across its phases. By mimicking the biological intelligence of black eagles, the algorithm can navigate complex landscapes and find promising areas.

3.8. Parameter tuning of ESN model using BEO

The BEO is used to optimally tune the hyperparameters of the LBESN. These hyperparameters significantly impact the ESN's ability to model temporal dependencies and generalise to unseen data. The optimisation problem for ESN tuning is as follows:

$$\min_{\theta} L_{val}\left(ESN(\theta)\right) = 1 - Accuracy_{val} \quad (18)$$

Where, θ=[α,N_r,η] is the vector of ESN hyperparameters: α is the Leak Rate (temporal memory factor), Nr is the Reservoir Size (number of internal neurons), and η is the Learning Rate (output weight adjustment rate).

Each eagle in the BEO algorithm represents a candidate θ and is evaluated based on its ability to minimise the classification error (1 - accuracy) on the validation dataset.

The BEO is used to tune LBESN parameters by initializing a population of candidate solutions (eagles), each with random values for leak rate, reservoir size, and learning rate. Each eagle's fitness is evaluated by training an LBESN model. The error value is calculated on validation data. The best-performing eagle guides the search process through four phases: stalking (exploration), attacking (local search), capturing (greedy refinement), and homecoming (boundary correction). In each iteration, eagle positions are updated and fitness re-evaluated to identify better solutions. After several iterations, the best parameter set found is returned.

4. RESULTS AND DISCUSSION

The LBESN model is coded in Python and validated using the Google Colab environment. The performance of the model is assessed in three different datasets of KDDCup99, UNSW-NB15, and InSDN. The entire dataset is separated into a 70% training and 30% testing set. The KDDCup99 dataset consists of 22 distinct attack types. The UNSW-NB15 dataset consists of 10 different attacker categories. During preprocessing, the synthetic minority over-sampling technique (SMOTE) is applied to solve the class imbalance issues. The performance is analysed using four standard metrics: precision (P), recall (R), F1-score (F1), and accuracy (A). To optimize the performance of LBESN, the BEO is used with 100 iterations and a population size of 20 eagles. The leak rate (α) varies from 0.1 to 1.0. The reservoir size varies from 10 to 200 neurons. The learning rate is set to 0.01. The optimized values are given in Table 1.

The fitness curve of BEO is given in Figure 3. The fitness curve shows that the BEO is effective in reducing the loss of the LBESN model for attacker detection. It shows an initial exploratory phase, followed by a clear convergence towards a lower loss value. It is observed that the BEO successfully optimized the LBESN parameters to increase its performance in detecting attackers. Stabilisation at the end suggests that the algorithm has found a robust set of parameters or reached the limits of further significant improvement within the given iterations.

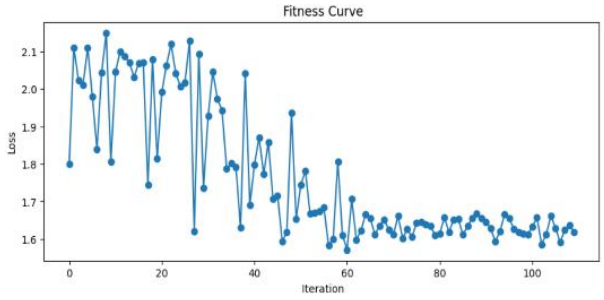

Figure 3. BEO fitness curve

Table 1. Adjusted values of the LBESN model

Parameter	Description	Min value	Max value	Optimized value
Leak rate (α)	Controls memory decay in the reservoir	0.1	1.0	0.52
Reservoir size	Number of neurons in the reservoir layer	10	200	133
Learning rate	Learning rate for training the readout layer	0.0001	0.01	0.0053

The training and testing curves of accuracy and loss for three datasets are given in Figure 4. The accuracy curves show a smooth and consistent rise in performance with minimal deviation between training and test sets. It denotes that strong generalization without overfitting. The models achieve high accuracy values within relatively few epochs. The loss curves, in parallel, show a steep decline in the early epochs and flatten out progressively. The confusion matrices prove the model's strong classification performance across all attack categories, as shown in Figure 5. For KDD99, the model attained near-perfect accuracy across multiple classes like neptune, smurf, and normal, with minimal misclassifications in minority classes such as Perl and phf and shows a robust generalization capability even with imbalanced data. In UNSW-NB15, the model achieves high precision in classifying major attack types such as Generic, DoS, and Exploits. Also, it achieves good detection performance for less frequent attacks like Shellcode and Worms. For InSDN, the model effectively identified large-scale DDoS, Probe, and Normal traffic patterns.

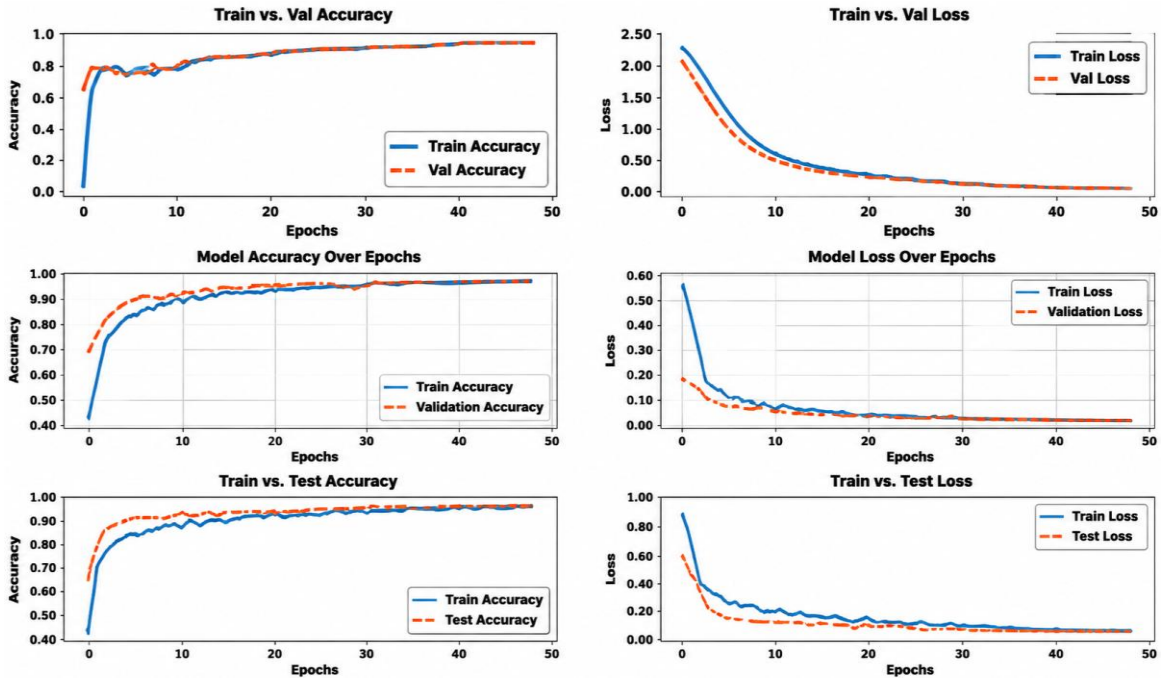

Figure 4. Model training and validation curve for three different datasets

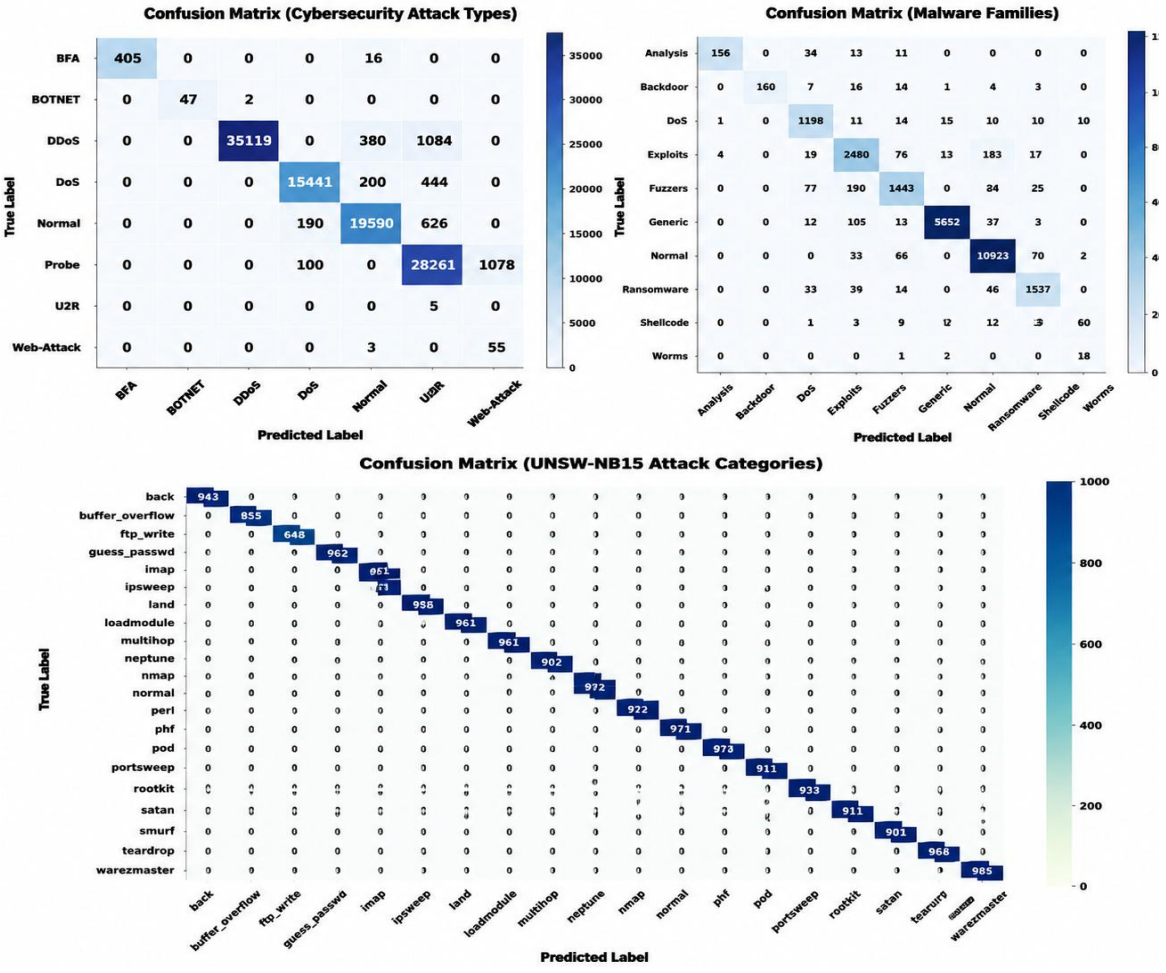

Figure 5. Confusion matrix visualisation of three different data sets

The performance metrics values of the model for three different datasets are given in Figures 6-8. For the UNSW-NB15 dataset, the model reaches an overall accuracy of 96.0% with a high precision and recall rate across most attack categories. It achieves F1-scores above 0.97 for the attack types such as BFA, BOTNET, DDoS, and DoS. It denotes the accurate detection with minimal false positives and negatives. In the KDD99 dataset, the model achieved an outstanding performance with an overall accuracy of 96.8%. It achieves F1-scores above 0.96 for the attack types such as smurf, neptune, portsweep, and nmap. It proves the model's ability to handle both frequent and infrequent intrusion types effectively. This strong performance across 21 classes indicates the model's robustness in high-dimensional multiclass environments. For the complex InSDN dataset, the model achieved an overall accuracy of 94.1%. It shows F1-scores above 0.86 for categories like backdoor and worms. Likewise, it achieves F1-scores around 0.83–0.87. Overall, the BEO-tuned LBESN constantly performed well across different datasets. The LBESN model accurately detects most known attack types with high reliability.

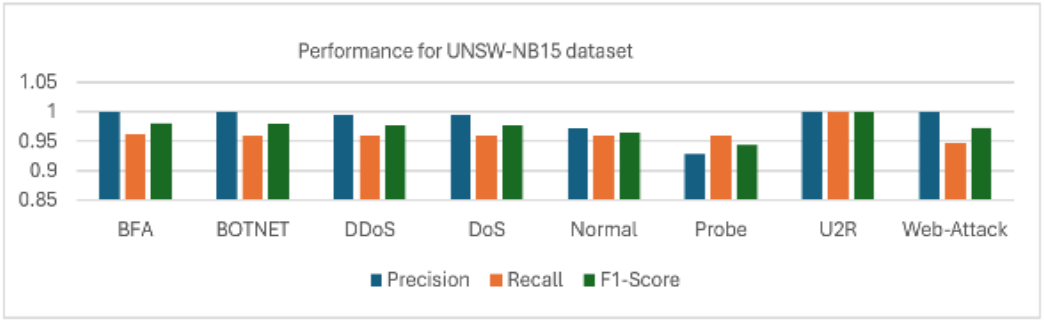

Figure 6. Performance for UNSW-NB15 dataset

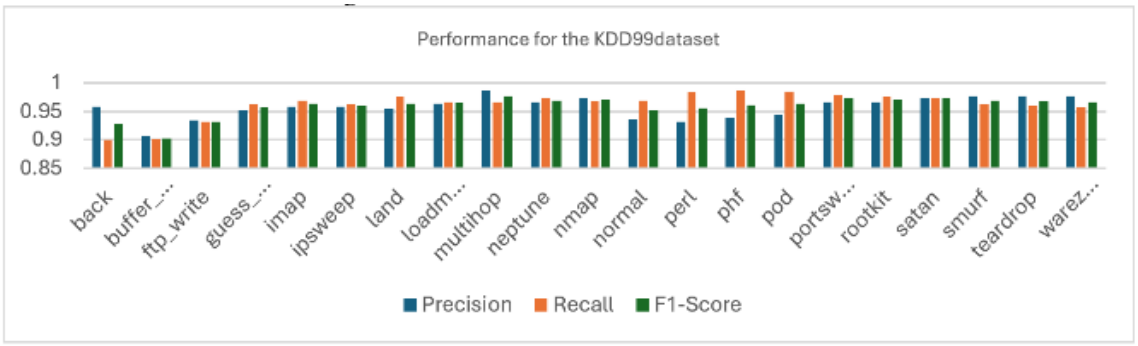

Figure 7. Performance for the KDD99dataset

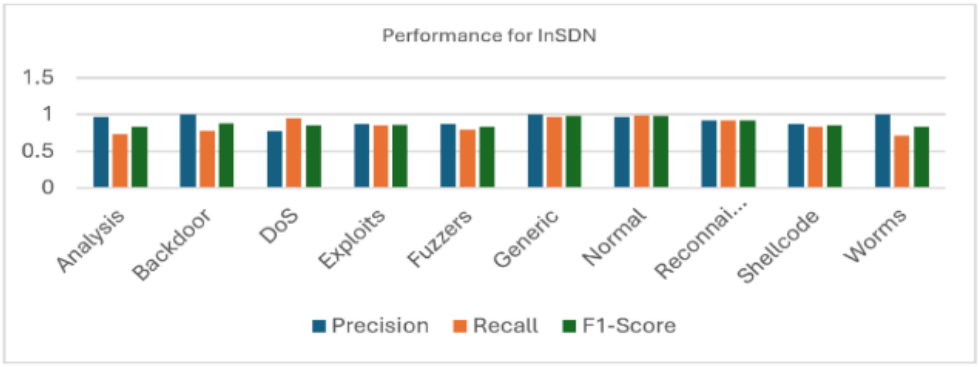

Figure 8. Performance for InSDN

The evaluation of the BEO-LBESN with existing architectures is given in Table 2. The *BEO-LBES* method achieves the highest accuracy of 95.6%. The existing models like AAE + BiGAN and ABiLSTM show accuracies of 94.1% and 94.2%, respectively. The CNN + GRU and LSTM show the accuracies of 94.5% and 92.8%, respectively. The models DBN and MLP show lower accuracies around 91.0% to 91.8%. Overall, the proposed BEO-LBESN model outperformed all existing methods with an accuracy of 95.6%. The powerful combination of ESN's dynamic reservoir computing with BEO increases learning accuracy across various network traffic classes.

To validate the performance of the model for real-world IoT traffic datasets, the model is applied to IoT-23 and Edge-IIoT datasets. This dataset consists of data collected from edge devices including attacks like Botnet, DDoS, DoS, and Worms. The results of BEO-LBESN on these real-world datasets are given in Table 3.

These results show that the BEO-LBESN model can effectively classify IoT-based attack traffic even in dynamic and evolving IoT environments. To evaluate the efficiency of the BEO-LBESN model in terms of runtime and memory usage, the tests are performed on both benchmark datasets and real-world IoT traffic datasets. The computational analysis results are given in Table 4.

It is observed that BEO-LBESN achieves strong detection performance with a moderate computational overhead for both training time and memory usage. These overheads are manageable in edge computing environments. Further, to analyze the efficiency of the proposed model for real-time adaptability, the performance is tested for zero-day attacks. To simulate zero-day attacks, a novel attack type is generated in both the KDDCup99 and IoT-23 datasets. The performance of the BEO-LBESN model on zero-day attacks is given in Table 5. The results show that the BEO-LBESN model can identify zero-day attacks effectively with the highest accuracy and F1-score rates.

Table 2. Comparison with other models

Author(s)	Technique	Accuracy (%)
M. S. Deshmukh *et al.*	DBN with Taylor-SMO	91.8
S. Manimurugan	DBN	92.3
R. Y. Aburasain *et al.*	BWO-based hybrid model	93.2
Abdalgawad *et al.*	AAE + BiGAN	94.1
N. K. S. Nayak *et al.*	ESBCA-Net	93.5
Marwa Keshk *et al.*	LSTM	92.8
Himanshu Nandanwar *et al.*	CNN + GRU	94.5
Qigang Liu *et al.*	MLP	91.0
U. C. Akuthota *et al.*	Transformer model	93.9
Sarika Saxena *et al.*	GNN	94.6
Proposed method (BEO- LBESN)	Optimized ESN using BEO	95.6

Table 3. Performance on real-world IoT traffic datasets

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Training time (seconds)	Inference time (seconds/sample)
IoT-23	95.2	94.5	96.0	95.3	245	0.42
Edge-IIoT	96.0	95.8	96.4	96.1	290	0.55

Table 4. Analysis of computational overhead

Metric	UNSW-NB15	KDDCup99	IoT-23	Edge-IIoT
Training time (seconds)	350	310	245	290
Inference time (seconds/sample)	0.50	0.45	0.42	0.55
Memory usage (MB)	1530	1450	1780	1820
Model size (MB)	12.5	11.2	14.0	15.6

Table 5. Performance analysis for zero-day attacks

Dataset	Zero-day attack type	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
KDDCup99	NewDoS	93.5	94.2	91.8	93.0
IoT-23	Botnet (variant)	94.8	94.6	95.4	95.0

5. CONCLUSION

In this work, a novel hybrid DL system is suggested to enhance anomaly detection in IoT networks. By integrating different feature extraction techniques, the model effectively captures both explicit and latent characteristics of network traffic. The fused features are classified using a LBESN. It uses dynamic memory for reservoir computing. In addition, the parameters are fine-tuned using the BEO to ensure optimal

performance. Experimental evaluation on benchmark datasets shows that the BEO-LBESN model achieves an average accuracy of 95.6% for validation sets.

FUNDING INFORMATION

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
P. Palpandi	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
B. Sakthivel		✓				✓			✓	✓	✓	✓		
M. Ponnrajakumari				✓			✓			✓	✓		✓	✓
M. Indirani								✓						
S. Govindaraju					✓									
S. Deivasigamani					✓		✓			✓		✓		✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available on Kaggle.

REFERENCES

[1] M. Abomhara and G. M. Køien, "Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks," *Journal of Cyber Security and Mobility*, vol. 4, no. 1, pp. 65–88, May 2015, doi: 10.13052/jcsm2245-1439.414.

[2] M. AL-Hawawreh, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial internet of things based on deep learning models," *Journal of Information Security and Applications*, vol. 41, pp. 1–11, Aug. 2018, doi: 10.1016/j.jisa.2018.05.002.

[3] M. Abdullah, A. S. Al-Shannaq, S. Almabdy, A. Balamash, and A. Alshannaq, "Enhanced intrusion detection system using feature selection method and ensemble learning algorithms," *International Journal of Computer Science and Information Security*, vol. 16, no. 2, pp. 48–55, 2018, [Online]. Available: https://sites.google.com/site/ijcsis/

[4] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for iot security based on learning techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671–2701, 2019, doi: 10.1109/COMST.2019.2896380.

[5] P. Toupas, D. Chamou, K. M. Giannoutakis, A. Drosou, and D. Tzovaras, "An intrusion detection system for multi-class classification based on deep neural networks," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, IEEE, Dec. 2019, pp. 1253–1258. doi: 10.1109/ICMLA.2019.00206.

[6] M. S. Deshmukh and P. R. Bhaladhare, "Intrusion detection system (DBN-IDS) for IoT using optimization enabled deep belief neural network," in *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*, IEEE, Oct. 2021, pp. 1–4. doi: 10.1109/ISCON52037.2021.9702505.

[7] S. Manimurugan, S. Al-Mutairi, M. M. Aborokbah, N. Chilamkurti, S. Ganesan, and R. Patan, "Effective attack detection in internet of medical things smart environment using a deep belief neural network," *IEEE Access*, vol. 8, pp. 77396–77404, 2020, doi: 10.1109/ACCESS.2020.2986013.

[8] R. Y. Aburasain, "Enhanced black widow optimization with hybrid deep learning enabled intrusion detection in internet of things-based smart farming," *IEEE Access*, vol. 12, pp. 16621–16631, 2024, doi: 10.1109/ACCESS.2024.3359043.

[9] N. Abdalgawad, A. Sajun, Y. Kaddoura, I. A. Zualkernan, and F. Aloul, "Generative deep learning to detect cyberattacks for the IoT-23 dataset," *IEEE Access*, vol. 10, pp. 6430–6441, 2022, doi: 10.1109/ACCESS.2021.3140015.

[10] N. K. S. Nayak and B. Bhattacharyya, "MAC protocol based iot network intrusion detection using improved efficient shuffle bidirectional COOT channel attention network," *IEEE Access*, vol. 11, pp. 77385–77402, 2023, doi: 10.1109/ACCESS.2023.3299031.

[11] A. Ali and M. M. Yousaf, "Novel three-tier intrusion detection and prevention system in software defined network," *IEEE Access*, vol. 8, pp. 109662–109676, 2020, doi: 10.1109/ACCESS.2020.3002333.
[12] G.-P. Fernando, A.-A. H. Brayan, A. M. Florina, C.-B. Liliana, A.-M. Héctor-Gabriel, and T.-S. Reinel, "Enhancing intrusion detection in IoT communications through ML model generalization with a new dataset (IDSAI)," *IEEE Access*, vol. 11, pp. 70542–70559, 2023, doi: 10.1109/ACCESS.2023.3292267.
[13] A. Aljuhani *et al.*, "A deep-learning-integrated blockchain framework for securing industrial IoT," *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 7817–7827, Mar. 2024, doi: 10.1109/JIOT.2023.3316669.
[14] I. Ullah and Q. H. Mahmoud, "Design and development of a deep learning-based model for anomaly detection in IoT networks," *IEEE Access*, vol. 9, pp. 103906–103926, 2021, doi: 10.1109/ACCESS.2021.3094024.
[15] B. Lahasan and H. Samma, "Optimized deep autoencoder model for internet of things intruder detection," *IEEE Access*, vol. 10, pp. 8434–8448, 2022, doi: 10.1109/ACCESS.2022.3144208.
[16] B. Madhu, M. Venu Gopala Chari, R. Vankdothu, A. K. Silivery, and V. Aerranagula, "Intrusion detection models for IoT networks via deep learning approaches," *Measurement: Sensors*, vol. 25, p. 100641, Feb. 2023, doi: 10.1016/j.measen.2022.100641.
[17] M. Keshk, N. Koroniotis, N. Pham, N. Moustafa, B. Turnbull, and A. Y. Zomaya, "An explainable deep learning-enabled intrusion detection framework in IoT networks," *Information Sciences*, vol. 639, p. 119000, Aug. 2023, doi: 10.1016/j.ins.2023.119000.
[18] L. Ashiku and C. Dagli, "Network intrusion detection system using deep learning," *Procedia Computer Science*, vol. 185, pp. 239–247, 2021, doi: 10.1016/j.procs.2021.05.025.
[19] B. Ahmad, Z. Wu, Y. Huang, and S. U. Rehman, "Enhancing the security in IoT and IIoT networks: an intrusion detection scheme leveraging deep transfer learning," *Knowledge-Based Systems*, vol. 305, p. 112614, Dec. 2024, doi: 10.1016/j.knosys.2024.112614.
[20] S. Hizal, U. Cavusoglu, and D. Akgun, "A novel deep learning-based intrusion detection system for IoT DDoS security," *Internet of Things*, vol. 28, p. 101336, Dec. 2024, doi: 10.1016/j.iot.2024.101336.
[21] H. Nandanwar and R. Katarya, "Deep learning enabled intrusion detection system for industrial IoT environment," *Expert Systems with Applications*, vol. 249, 2024, doi: 10.1016/j.eswa.2024.123808.
[22] A. Deshmukh and K. Ravulakollu, "An Efficient CNN-Based Intrusion Detection System for IoT: use case towards cybersecurity," *Technologies*, vol. 12, no. 10, p. 203, Oct. 2024, doi: 10.3390/technologies12100203.
[23] E. S. A. Alars and S. Kurnaz, "Enhancing network intrusion detection systems with combined network and host traffic features using deep learning: deep learning and IoT perspective," *Discover Computing*, vol. 27, no. 1, 2024, doi: 10.1007/s10791-024-09480-3.
[24] Q. Liu, D. Wang, Y. Jia, S. Luo, and C. Wang, "A multi-task based deep learning approach for intrusion detection," *Knowledge-Based Systems*, vol. 238, 2022, doi: 10.1016/j.knosys.2021.107852.
[25] H. C. Altunay and Z. Albayrak, "A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks," *Engineering Science and Technology, an International Journal*, vol. 38, p. 101322, Feb. 2023, doi: 10.1016/j.jestch.2022.101322.
[26] H. Zhang, H. San, J. Chen, H. Sun, L. Ding, and X. Wu, "Black eagle optimizer: a metaheuristic optimization method for solving engineering optimization problems," *Cluster Computing*, vol. 27, no. 9, pp. 12361–12393, 2024, doi: 10.1007/s10586-024-04586-1.
[27] U. C. Akuthota and L. Bhargava, "Transformer-based intrusion detection for IoT networks," *IEEE Internet of Things Journal*, vol. 12, no. 5, pp. 6062–6067, 2025, doi: 10.1109/JIOT.2025.3525494.
[28] S. Saxena, J. Grover, and S. Singhal, "Exploring graph neural networks for robust network intrusion detection," *Procedia Computer Science*, vol. 258, pp. 3630–3639, 2025, doi: 10.1016/j.procs.2025.04.618.

BIOGRAPHIES OF AUTHORS

Mr. P. Palpandi is an assistant professor in the Department of CSE (IoT) at Sethu Institute of Technology, with over 15 years of teaching experience. He holds a B.Sc. in Physics, an M.Sc. in Computer Science and Information Technology, and an M.Tech. in Computer and Information Technology. He can be contacted at email: palpandi@sethu.ac.in.

B. Sakthivel is an associate professor in the Department of Electronics and Communication Engineering at PSY Engineering College, Sivagangai, India. He holds a B.E. in Electronics and Communication from SACS MAVMMM Engineering College and an M.E. and Ph.D. in VLSI Design from Anna University. He can be contacted at email: sakthivelece@psyec.edu.in.

M. Ponnrajakumari is an assistant professor in the Department of Electronics and Communication Engineering at Velammal Engineering College, Chennai. Her research interests include wireless networks, cognitive radio networks, cybersecurity, and IoT. She can be contacted at email: ponnrajakumari23@gmail.com.

Dr. M. Indirani is an assistant professor at Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai. She earned her Ph.D. from Anna University in 2023. With 23 years of teaching experience, Dr. Indirani has published 16 journal papers, holds 7 patents, and authored 5 books. She can be contacted at email: mindirani2008@gmail.com.

Dr. S. Govindaraju is an associate professor in the Department of Computer Science at St. Joseph University, Chennai. He holds an MCA, M.Phil., and Ph.D. from Bharathiar University. His research primarily focuses on image retrieval using medical images. He can be contacted at email: govindindu2010@gmail.com.

S. Deivasigamani is an assistant professor at UCSI University, Malaysia. He holds a B.E. in Electrical and Electronics from Thiagarajar College of Engineering and an M.E. in Applied Electronics from Thanthai Periyar Government Institute of Technology. He earned his Ph.D. from Multimedia University, Malaysia. He can be contacted at email: deivasigamani@ucsiuniversity.edu.my.