International Journal of Innovation Studies

ISSN:2096-2487 | E-ISSN:2589-2975

SWARM BASED FEATURE SELECTION AND ENSEMBLE DEEP LEARNING MODEL (EDLM) FOR BOTNET ATTACK DETECTION IN IoT HEALTHCARE SYSTEMS

Mrs.B.Praveena
a)Research Scholar, Department of Computer Science, Dr.SNS Rajalakshmi College of Arts and Science, b)Assistant Professor, Department of Computer Science with Data Analytics Kongunadu Arts and Science College, Coimbatore, E-mail: praveenamtp@gmail.com

Dr.A.Devi
Research Supervisor & Associate Professor, Department of Computer Applications
Dr.SNS Rajalakshmi College of Arts and Science, Coimbatore
Email: devialagarsamy111@gmail.com

ABSTRACT: A variety of devices make up the Internet of Things (IoT). Because of the large number of attack routes and the constant growth of viruses, botnet detection (BND) is getting more and more difficult. Because of the complexity and diversity of attacks, conventional detection methods that rely solely on a single Machine Learning (ML) technique could not be completely effective. The Ensemble (DL) Deep Learning Model (EDLM) is presented in this study. Divergence Weight (LSTM) Long Short-Term Memory (DWLSTM), Levy Weight (LW) Bi-Directional Gated Recurrent Unit (LWBi-GRU), and CDBN are some of the models that are combined to produce the findings of this EDL. To choose the most relevant features from the dataset, the Inertia Weight Mother Optimisation Algorithm (IWMOA) is presented. The data sequences are processed in both forward (F) and backward (B) directions by the DWLSTM classifier. An update gate (UG) and a reset gate (RG) process LWBi-GRU in both directions (F and B). This classifier's capacity to analyse information from both sides allows it to interpret some inputs effectively. The Conditional Gaussian-Bernoulli Restricted Boltzmann Machine (CGBRBM) for botnet attack detection (BN AD) is a component of the Conditional Deep Belief Network (CDBN). Here, the ensemble averaging (EA) method (EAM) is utilized for the purpose of integrating benefit of many classifiers. The diversity of samples from multiple sources are detected by the potential of the ensemble model (EM) via averaging the predictions of various models. From the Kaggle and the University of California, Irvine (UCI), the Bot-IoT and N-BaIoT datasets are used, and it may simulate attack scenarios. In the IoT datasets, the testing data was used to assess the DL models, and the metrics like precision (P), recall (R), F1-score, and accuracy (ACC) were used for this evaluation.

INDEX TERMS: Cybersecurity, botnet detection (BND), deep learning (DL), Ensemble DL Model (EDLM), Levy Weight Bi-Directional Gated Recurrent Unit (LWBi-GRU), Conditional Deep Belief Network (CDBN), and Internet of Things (IoT).

1. INTRODUCTION

The introduction of the IoT offers novel advancements to many sectors. The unprecedented levels of simplicity and connectivity was also offered by this IoT. The global connectivity of the Botnet attacks (BNA) made this IoT backgrounds suspectible to many risks in security. The Botnet is defined as a system of infected gadgets that are in control of the suspicious users. This Botnets thus impacts the integrity and privacy of IoT deployments. So, IoT background has major risks because of this BNA [1]–[2]. The cyberattacks against vital structures are executed by the botnet [3]. For securing sensitive data in IoT backgrounds, Real-time (RT) detection and mitigation of botnet activity is crucial. The entire network is continuously monitored for detecting any anomalies, and it was facilitated by the Network (IDS) Intrusion Detection Systems (NIDS) via aggregating network data from every host [4]. The IoT environment has heterogeneous devices having many types, producers, and structures [5]. In the network of several host features, it is complicated to detect anomalies in a heterogeneous IoT device, because these heterogeneous IoT device contains many traffic patterns.

The function of the IoT devices are then limited, when installing conventional security software. For the purpose of identifying and classifying harmful traffic, the study that integrates "intelligence" into security systems and using Artificial Intelligence (AI) methods are wide-spreading. Then, the AI-driven IDS have the potential in detecting malicious patterns, and it also adapts to various attack methods. Large amounts of network traffic (NT) may be automatically analysed by AI-driven IDS**.**

The ML method is a kind of the AI. This ML has become an effective method for IoT intrusion detection (ID) [6]. Models that classify the normal and malicious network behavior has been created by the potential of ML method, and it can be done by ML in learning past data, and obtaining relevant data. Regardless of its benefits, IoT ID also have several disadvantages that needs to resolve. Among the main obstacles for resolving are the absence of labelled training data, the requirement for RT processing and low latency, and the interpretability of intricate AI models. AI models must also be continuously adjusted to new attack vectors and changing network topologies because of the dynamic nature of IoT backgrounds.

However, there are still significant issues that need to be solved, such as prompt detection, RT monitoring, and attack adaptation. In training and post-deployment, classical machine learning algorithms use signatures of known malware, which is the primary reason for this. Preprocessing data before feeding it could cause the detection model's quality to deteriorate. Finding the ideal subset of features, often known as feature selection (FS), is actuality a NP-complete problem. Consequently, for classification procedure, the choice of pertinent feature subset is obtained by the application of optimisation techniques. By identifying the features that are most pertinent to the classification procedure, FS is used to decrease the dataset's dimensionality.

The computational resources and time needed for BND are reduced when the feature set size is reduced. Additionally, the impact of superfluous features on the detection process may be avoided. Because of its ease of use and adaptability, meta heuristics (MH) algorithms (MHA) are widely used for a variety of optimisation problems. As a result, the BND domain has employed MH optimisation techniques for various objectives. The global search (GS) capabilities and applicability for the FS process of nature-inspired optimisation algorithms (NIOA) are particularly well-known.To optimise FS for BND, some studies employed algorithms based on swarm intelligence (SI) [8]. To address the complex issues, DL techniques are presented for their generic deep layer architecture [9]. To gain inspiration for using DL on (AD) attack detection, a number of literature evaluations have been carried out [10–12].

By merging the diversity of the training models, Subset training is produced by ensemble learning (EL). To improve result prediction, this EL will generate a subset classifier. By analysing data with varied behaviour in EL, NIDS can identify more general patterns of assault in IoT networks with diverse gadget [13–15]. Training and testing the created DL models and their ensemble model is the primary goal of this paper. With a lower (ER) error rate and a higher detection ACC, this study seeks to achieve optimal performance. EDLM that aggregates the outcomes by merging multiple models, such as CDBN, LWBi-GRU, and DWLSTM. The accurate outcomes are attained by EA, via averaging its predictions. In IoT datasets, the following metrics can be used for the purpose of determining the classifier efficiency, those metrics are P, R, F1-score, and ACC.

2. LITERATURE REVIEW

To identify the subsets of the most pertinent features for the detection procedure, Baker and Samarneh [16] suggested an optimisation technique. In order to detect IoT botnets using the N-BaIoT, this study used the efficacy of Equilibrium Optimisation (EO), Battle Royale Optimisation (BRO), and Adaptive Equilibrium Optimisation (AEO) for FS. 3 classifiers: K Nearest Neighbour (KNN), Random Forest (RF), and Gaussian Naive Bayes (GNB) are used to assess the efficiency of the chosen features. True Positive (TP) Rate (TPR), False Positive (FP) Rate (FPR), sensitivity (S), specificity (SP), feature count, ACC, and time are among the metrics taken into account. By runtime, count of FS, and ACC, the outcomes demonstrate that EO and AEO outperform existing work on the same dataset.

For selecting the most pertinent attributes, a new FS method based on a hybrid filter and wrapper selection called Fisher Grasshopper Optimisation algorithm (FGOA) is introduced by Taher et al. [17]. After ranking the attributes using the new approach combined with clustering, the top-ranked features are minimised by using the GOA. Botnet detection is achieved using the Improved Harris Hawks Optimisation Algorithm (IHHO), which chooses and modifies the hyper parameters (HP) of the neural networks (NN). The GS procedure is improved for optimal solutions by adding three improvements to HHO. A chaotic map function (CMF) is used for initialisation for resolving the problem of population diversity. A novel nonlinear (NL) is added to the hawks' escape energy in order to avoid local minima and enhance the exploration-exploitation (E-E) balance. Additionally, Opposite-Based Learning (ROBL) and a new elite operator, Refraction principle, are used to improve the exploitation phase of HHO. The KNN and NN classifiers are validated using the N-BaIoT dataset.

To generate a training model (TM) from every diverse IoT device, a Deep NN (DNN) was suggested by Wardana et al. [18]. The traffic is then estimated using every TM from every different IoT gadgets. Using the EAM, the prediction results from all TM are averaged to determine the ultimate outcome. The suggested model is assessed by using the N-BaIoT dataset. By using a mix of DNN and EA for anomaly detection, NIDS is best able to detect BNA patterns in diverse IoT devices. EA DNN can BN AD in heterogeneous IoT gadgets, according to experimental findings.

In order to detect IoT botnet attacks, the Deep (AE) Autoencoder (DAE) model was suggested by Meidan et al. [19]. From benign traffic data, this DAE derives statistical features. The four main processes in this procedure are anomaly detector training, data collection (DC), feature extraction (FE), and continuous monitoring. Anomalies found when applied to fresh (potentially hacked) data from an IoT device could be a sign that botnet attacks have compromised the device. After being compressed, an AE is trained to reconstruct its inputs. and the AE is a NN. The network has the ability to learn the important ideas and the relationships between its input features due to the compression.

Using long short-term memory AE (LAE) encoding, using large-scale IoT NT data, Reducing the dimensionality of features was recommended by Popoola et al. [20]. Deep Bidirectional LSTM (Bi-LSTM) is used to assess the long-term correlated changes in the low-dimensional feature set produced by LAE in order to accurately categorise NT samples.The BoT-IoT dataset is used to confirm the efficacy of the hybrid DL technique. LAE significantly reduced the amount of memory required for large-scale NT data storage and outperformed state-of-the-art (SOTA) feature (DR) dimensionality reduction approaches. The deep Bi-LSTM model exhibits resilience against model underfitting and overfitting in spite of the notable decrease in feature size. In situations involving binary class (BC) and multiclass (MC) classification, it also demonstrates strong generalisation abilities.

A novel IDS for IoT systems was introduced by Ge et al. [21] utilising a DL method. IoT traces and real attack traffic, such as Denial of Service (DoS), Distributed DoS (DDoS), reconnaissance, and information theft attacks, are included in this SOTA IoT dataset. Feed-Forward NN (FFNN) algorithms with embedding layers are utilised for MC classification, whereas header field information in each packets is employed as generic features to capture general network features. To construct a BC, high-dimensional (HD) categorical features are encoded using the transfer learning (TL) concept.The results of the examination of the suggested method show that both the BC and MC classifiers have the highest classification ACC.

Considering a DNN for an IoT network, Ahmad et al. [22] suggested an effective anomaly detection method utilising Mutual Information (MI). When performing various DL models, such as DNN, Convolutional NN (CNN), Recurrent NN (RNN), Gated Recurrent Unit (GRU), and LSTM, the IoT-Botnet 2020 is taken into account. Metrics like P, R, f1-score, ACC, False Acceptance Rate (FAR), True Negative Rate (TNR), and False Negative Rate (FNR) are used to compare experimental outcomes to the widely used DL models.

An EL model for BN AD in IoT networks (ELBA-IoT) has been suggested by Abu Al-Haija et al. [23]. It uses EL for AD NT from hacked IoT devices and keeps an eye on the behavioural features of IoT networks. Furthermore, the evaluation of 3 distinct ML techniques that are part of the decision tree (DT) methodology family (AdaBoosted, RUSBoosted, and bagged) is characterised by the IoT-based botnet detection strategy. The N-BaIoT-2021 dataset, that includes records of both regular IoT NT and botnet attack (BNA) traffic of compromised IoT devices, is used to assess the ELBA-IoT. For BNA launched from affected IoT devices, the ELBA-IoT model has a high (DR) detection rate and a low inference overhead (40 μ-seconds).

A hybrid DL (CNN-LSTM) approach has been suggested by Alkahtani and Aldhyani [24] to detect BNA on 9 commercial IoT devices. An IoT environment is used to apply the DL model to BN AD. By expediting the procedure of disconnecting the majority of IoT devices from the Internet, early detection (ED) of DDoS attacks can aid network security (NS). This will assist in preventing and stopping the acceleration of botnet attacks. An actual N-BaIoT database extracted from a real-world structure. It is employed for in-depth research.With the best ACC, the CNN-LSTM model has the ability for detecting BNA from a variety of IoT devices.

To stop and identify IoT BNA, a two-fold ML strategy was suggested by Hussain et al., [25]. For scanning the AD (ResNetScan-1) model, a ResNet-18 is created in the first fold. The BNA was no effectively prevented by the previous scanning detection model. Then, for detecting the DDoS attacks, the second ResNet-18 model (ResNetDDoS-1 model) is used. Then, the effectiveness of the ResNetScan-1 and ResNetDDoS-1 models were determined via some simulations, and this simulation was executed with the support of the scan and DDoS traffic samples from 3 widely accessible databases. After training the ResNet-18 model on these datasets, the ResNetScan and ResNetDDoS models that were produced were stored. When compared to other trained models, the experimental findings demonstrate how effectively the recommended technique can stop and identify BNA.

3. PROPOSED METHODOLOGY

In this study, EDLM is introduced which combine the results by combining several models like DWLSTM, LWBi-GRU, and CDBN. DWLSTM data sequences are processed in two directions (F and B). LWBi-GRU is performed with an UG and the RG in both directions. The CDBN is composed of the CGBRBM for BN AD. Ensemble averaging is introduced to merge the strengths of individual classifiers. Bot-IoT and N-BaIoT datasets are used for performance validation of the suggested method and current methods. ACC, P, R, and F1-score were employed for determining the efficacy of the suggested method. The recommended scheme's overall process is depicted in Figure 1.

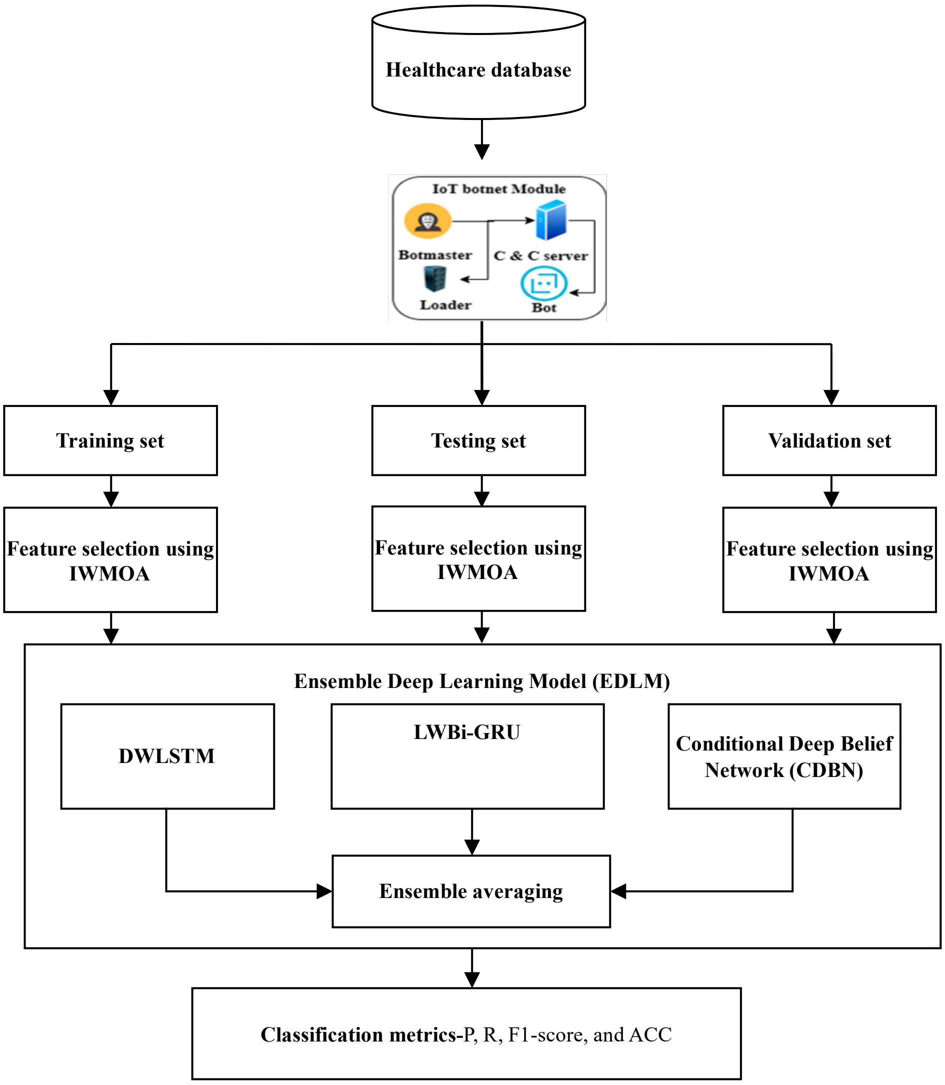
Healthcare database
IoT botnet Module
Botmaster
C & C server
Loader
Bot
Training set
Testing set
Validation set
Feature selection using IWMOA
Feature selection using IWMOA
Feature selection using IWMOA
Ensemble Deep Learning Model (EDLM)
DWLSTM
LWBi-GRU
Conditional Deep Belief Network (CDBN)
Ensemble averaging
Classification metrics-P, R, F1-score, and ACC

FIGURE 1. FLOWCHART OF SUGGESTED IWHO AND DWLSTM STRUCTURE

3.1. IoT Botnet

A packet tracer (https://www.netacad.com/courses/packet-tracer accessed on 29 April 2024), an open source tool for IoT simulation, is used to construct an IoT botnet. To control the IoT nodes, a Python script was used to construct the C&C server. To configure the setup, a Python script module called Scapy was used. An IoT network is subjected to 4 attack types as a result of the attack simulation.

3.2. BoT-IoT AND N-BAIoT DATABASE

For cyber security study, the Bot-IoT dataset is openly and freely accessible [3]. It includes four BNA scenarios: DoS, DDoS, reconnaissance, and information theft as well as benign IoT NT. The N-BaIoT dataset is also publicly available and free for use in cyber security research [19]. Two doorbells, a thermostat, a baby monitor, four security cameras, and a webcam were all part of the IoT testbed that produced this dataset. Six redundant features were found and eliminated from the Bot-IoT dataset in this investigation. An NT sample is represented by a total of 37 features. Min-max normalisation (MMN) was used to re-scale the values of these features among 0 and 1 in order to effectively train a NN model.

3.3. IWMOA BASED FS

IWMOA is a population-based MHA that uses an iterative technique to solve FS. The algorithm population is made up of vectors in the AD space that represent candidate feature-selected solutions. Three periods of relation among a mother and her children: education, guidance, and upbringing will be simulated by IWMOA. Unquestionably, the home is the primary academic institution in society, and mothers are crucial to a child's upbringing. Children inherit their mother's skills and life lessons, and they grow in their own abilities by following her guidance. The three stages of (i) education, (ii) advice, and (iii) Upbringing are among the most important forms of relationships among a mother and her children.

Stage 1: Education (exploration stage): Children's education serves as the model for population update education in the suggested IWMOA strategy. It aims to enhance GS and exploration skills by drastically changing the feature positions of the population members (PM). The MOA design models the mother's training of her children to resemble the educational stage, as she is considered the best feature of the population.

Stage 2: advice (exploration stage): By significantly altering the PM' locations, the advice phase increases the IWMOA capability in GS and exploration. According to the IWMOA design, each PM's feature position relative to other PM whose ACC values are higher than theirs is seen as abnormal behaviour that is to be avoided. The mother's GS and Local Search (LS) capacity are adjusted using the inertia weight, which also modifies the influence of previous outcomes.

Stage 3: upbringing (exploitation stage): In numerous ways, the children's skill are enhanced by mother's contribution throughout the educational experience. By making little adjustments to the PM' feature positions, the parenting increases LS's capacity and facilitates exploitation throughout the IWMOA phase.

The population's search control is utilised to identify the best feature solution, and every population member establishes the values of ACC according to its location in the FS search space (SS) [26]. The classification accuracies best and worst values can be used to determine which PM are the best and worst. The optimal feature solution is updated and saved continuously during the process. Once the method is fully developed, IWMOA gives the problem's optimal feature solution [26].

3.4.ENSEMBLE DEEP LEARNING MODEL (EDLM)

The main advantages of an EL system and a number of DL techniques are intended to be efficiently combined by EDLM. Performance gains over individual techniques have been demonstrated by DL architectures such as DWLSTM, LWBi-GRU, and CDBN. Even though using separate techniques can improve prediction performance, EDLM has a significant impact. In contrast, base learners (BL) such as DWLSTM, LWBi-GRU, and CDBN are created concurrently in the parallel ensemble technique [27, 28]. As, Figure 2 illustrates, every piece of data in the BL is created separately. The fundamental benefit of this method is that it takes advantage of the independence between BL. The process of combining the outputs of the baseline classifiers into a single output is known as output fusion. The fusion techniques can be applied to parallel or sequential baseline classifiers, as well as independent or dependent data samples.

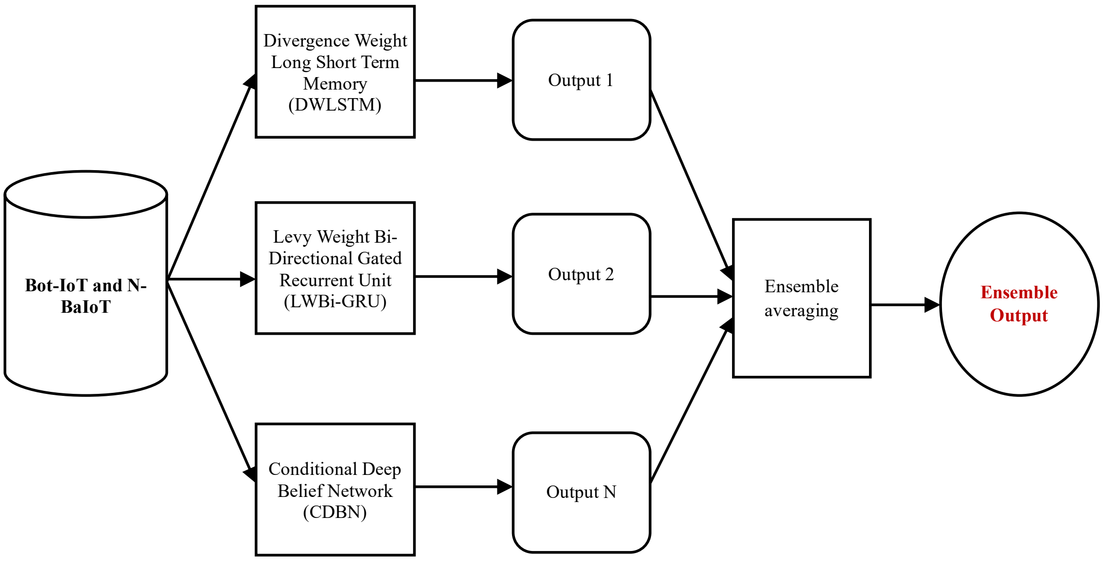
Divergence Weight Long Short Term Memory (DWLSTM)
Levy Weight Bi-Directional Gated Recurrent Unit (LWBi-GRU)
Conditional Deep Belief Network (CDBN)
Bot-IoT and N-BaIoT
Output 1
Output 2
Output N
Ensemble averaging
Ensemble Output

FIGURE 2. EDLM

3.4.1. DWLSTM CLASSIFIER

The DWLSTM network is one specific kind of RNN structure.When it comes to learning long-term dependencies, DWLSTM outperforms the conventional RNN [29–30]. Through the use of specially designed gates and memory cells, over time, LSTM networks can choose preserve or discard information due to their deep structure.

(Figure 3).

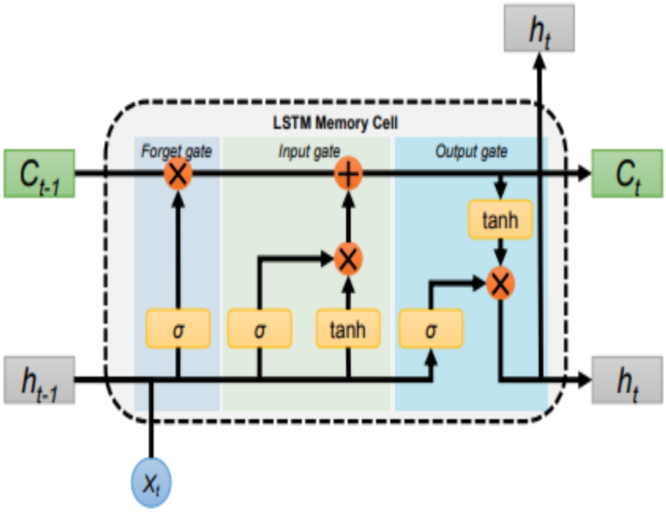

FIGURE 3. STRUCTURE OF DWLSTM

The cell state (C_t), or LSTM, allows data to move along it unaltered. Three gates control the C_t, which allows information to pass through optionally. The first gate, sometimes referred to as the forget gate (FG). FG determines which features of the cell state vector C_{t-1} will be forgotten.

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \quad (1)$$

Here, the forgetting degree is indicated by the sigmoid layer (SL)'s output vector, f_t, which has values among 0 and 1. The set of trainable parameters for the FG is defined by $W_f$and b_f. The input gate then selects the value that has to be changed.

$$i_t = \sigma(W_i . [h_{t-1}, x_t] + b_i) \tag{2}$$

Here, the value of the output variable that ranges from 0 to 1 is denoted as i_t. Trainable parameters are $W_i$and b_i. The current input (x_t) and the last hidden state, h_{t-1}, are then used to calculate a possible vector of cell state.

$$\tilde{C} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

The hyperbolic tangent is called $\tanh$. The values of the vector $\tilde{C}$ range from 0 to 1.The trainable parameters are $W_C$and b_C. The previous cell state C_{t-1} can then be updated by element-wise multiplication to the new C_t.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

In the end, the output gate determines which an SL should output.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5}$$

Here, a vector that has values between 0 and 1 is denoted as o_t. The output gate's trainable parameters are $W_o$and b_o. After that, equations (5–6) are combined to determine the new hidden state h_t.

$$h_t = o_t * \tanh(C_t) \tag{6}$$

Kullback Leibler (KL) divergence has been used to calculate the classifier's weight values. A statistical distance that quantifies the degree to which a probability distribution class Q differs from a true probability distribution P is called $D_{KL}(P \parallel Q)$.

$$D_{KL}(P||Q) = \sum_{w \in W} P(w) \log\left(\frac{P(w)}{Q(w)}\right) \tag{7}$$

It frequently has high HP, whose value is predetermined before the learning process starts. The loss function (LF) for HP optimisation in this study is Mean Square Error (MSE). It is the procedure that produces a model that minimises the LF by identifying a tuple of HP.

3.4.2. LWBi-GRU

For BN AD, the Bi-GRU model has UG and RG. Because Bi-GRU lacks a forget gate, it has fewer constraints, making it computationally economical, less likely to overfit, and an effective option for datasets of a smaller size. The update gate (d_{TS}) is used in the Bi-GRU model in place of the input gate and FG of the LSTM network. In order to determine which historical data should be passed along with the future data, the (d_{TS}) aids the model. The Bi-GRU model's vanishing-gradient problem (VGP) is lessened by this procedure. Equation (8) provides a mathematical specification for the [31, 32].

$$d_{TS} = \sigma(LW_d \times [h_{TS-1}, f_{TS}] + b_d) \tag{8}$$

Here, $\mathrm{LW_d}$ is the representation of the levy weight matrix. The symbol $\mathrm{b_d}$ represents the bias matrix. The input matrix (FS) at time step (TS) is denoted by $\mathrm{f_{TS}}$. The sigmoid (AF) activation function is represented by the symbol σ. $\mathrm{h_{TS-1}}$ represents the hidden state at the preceding time step (TS−1). The historical (TS) time-series data is controlled in the Bi-GRU model by means of the reset gate ($\mathrm{p_{TS}}$). In the hidden state, $\mathrm{p_{TS}}$ is in charge of the network's short-term memory. Equation (9), which provides a numerical expression for $\mathrm{p_{TS}}$,

$$\mathrm{p_{TS}} = \sigma\left(\mathrm{LW_p} \times [\mathrm{h_{TS-1}}, \mathrm{f_{TS}}] + \mathrm{b_p}\right) \tag{9}$$

Here, $\mathrm{b_p}$and $\mathrm{LW_p}$ stand for the bias matrix and the LW matrix of the $\mathrm{p_{TS}}$. Next, equation (10), which specifies the hidden state candidate ($\tilde{\mathrm{h}}_{\mathrm{TS}}$),

$$\tilde{\mathrm{h}}_{\mathrm{TS}} = \tanh(\mathrm{LW_h} \times [\mathrm{h_{TS-1}} \odot \mathrm{p_{TS}}, \mathrm{f_{TS}}] + \mathrm{b_h}) \tag{10}$$

Here, tanh stands for the tangent AF. ⊙ is used to represent the dot multiplication operation. The symbols as $\mathrm{b_h}$and $\mathrm{LW_h}$ stand for the bias matrix and LW matrix of the memory cell state, respectively. The influence of the prior weight on the present weight is managed by the levy weight. Using equation (11) and the Mantegna method with step size s, Levy weight is produced,

$$\mathrm{s} = \frac{\mathrm{u}}{|\mathrm{v}|^{\frac{1}{\beta}}} \tag{11}$$

The equations (12–13) are used to extract u and v from normal distributions.

$$\mathrm{u} \sim \mathrm{N}(0, \sigma^2), \mathrm{v} = \mathrm{N}(0, \sigma_\mathrm{v}^2) \tag{12}$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}}\right)^{\frac{1}{\beta}}, \sigma_\mathrm{v} = 1 \tag{13}$$

Therefore, equation (14) can be used to illustrate a basic strategy.

$$\mathrm{Levy(w)} = 0.01 \times \frac{\mathrm{u} \times \sigma}{|\mathrm{v}|^{\frac{1}{\beta}}} \tag{14}$$

Equation (15) illustrates the linear interpolation of $\tilde{\mathrm{h}}_{\mathrm{TS}}$, and $\mathrm{h_{TS-1}}$ to get the output ($\mathrm{h_{TS}}$).

$$\mathrm{h_{TS}} = (1 - \mathrm{d_{TS}}) \odot \mathrm{h_{TS-1}} + \mathrm{d_{TS}} \odot \tilde{\mathrm{h}}_{\mathrm{TS}} \tag{15}$$

Figure 3 mentions the LWBi-GRU model architecture. Feature information is extracted forward in the Bi-GRU model. The backward historical data is then automatically rejected by the Bi-GRU model. Because LWBi-GRU can simultaneously investigate information from both directions, it can process multiple inputs more effectively than conventional frameworks. As shown in Figure 4, the suggested framework takes the information among the features from the F and B directions.

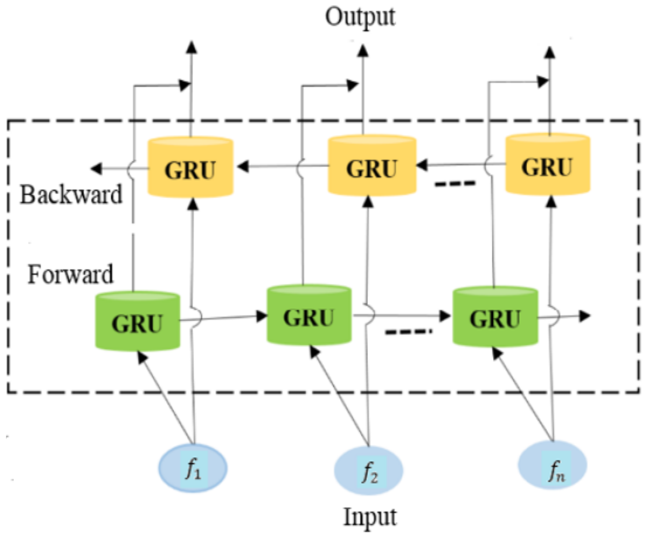

FIGURE 4. STRUCTURE OF THE LWBi-GRU FRAMEWORK

The forward GRU in the LWBi-GRU model takes information regarding the past from the historical data, while the backward GRU does the same for the future. Equation (16) specifies the LWBi-GRU model's numerical expression.

$$o_{TS} = A(\overrightarrow{h}_{TS}, \overleftarrow{h}_{TS}) \tag{15}$$

Here, A stands for the output of the B and F directions. Furthermore, $\overleftarrow{h}_{TS}$ and $\overrightarrow{h}_{TS}$ are the symbols representing the hidden states of the B and F GRUs. A particular set of parameters is used by the Bi-GRU model to maximise its performance. These consist of the Adam optimiser, a learning rate, a batch size of 50, a dropout rate of 0.5, 80 neurons, a look-back of 8 time steps, and the Mean Square Error (MSE) LF.

3.4.3. CDBN

In order to identify the attack activity, a CDBN classifier made of the CGBRBM is suggested. The suggested CDBN classifier, which is based on CGBRBM, is capable of efficiently learning the dataset's features and RT-detecting botnet attacks. CGBRBM uses an attack dataset as its input. The correlations between the current and historical input datasets can be captured by the CGBRBM. This is an RT method to identify the newly input dataset because, once the CDBN framework has been trained, instead of being inputted into the model all at once, the testing data will be added progressively. As shown in Figure 5, the CGBRBM unit serves as the initial layer in a CDBN-based detector, with N-1 conventional RBMs on top of it. This means that the entire CDBN design has N hidden layers. To be clear, the CDBN design is topped with a multiple classifier output unit that can determine if the input dataset is an attack sample and output the classification label.

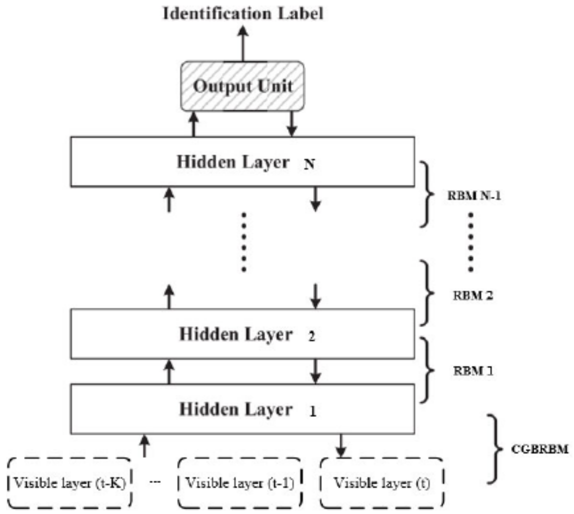

FIGURE 5. FRAMEWROK OF CDBN

PRE-TRAINING PROCEDURE: To initialise the system attributes, which include the offset values of each layer neuron and the link weights among layers, the CDBN classifier's pre-training procedure is utilised. Consider an RBM, which has a hidden layer (HL) with n hidden units (HU) and a visual layer (VL) with m visible units (VU). The following is a definition of a traditional RBM's energy function:

$$E(v,h) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}h_iv_j - \sum_{j=1}^{m} c_jv_j - \sum_{i=1}^{n} d_ih_i \tag{16}$$

Here, the j^{th} element of the VL vector is denoted by v_j. The i^{th} sample of the HL vector is denoted by h_i. In the weight matrix between the VU and HU, w_{ij} is the ij^{th} element. For the HL and VL, define $d_i$and c_j as the i^{th} sample and j^{th} member of the bias vector. Equation (16) uses the values of the neighbouring layer units to calculate the activation conditional probability distributions (CPD) of HU and VU.

$$\begin{cases} p(h_i = 1|v) = \mathrm{sigm}\left(d_i + \sum_{j=1}^{m} w_{ij}v_j\right) \\ p(v_j = 1|h) = \mathrm{sigm}\left(c_i + \sum_{i=1}^{n} w_{ij}h_i\right) \end{cases} \tag{17}$$

The sigmoid function is denoted by sigm(.). The following updates are made to the weights and biases of the traditional RBMs using the Contrastive Divergence (CD) approach [33]:

$$\begin{cases} w_{ij} = w_{ij} - \alpha(\langle v_jh_i\rangle_m - \langle v_jh_i\rangle_l) \\ d_i = d_i - \alpha(\langle h_i\rangle_m - \langle h_i\rangle_l) \\ c_j = c_j - \alpha(\langle v_j\rangle_m - \langle v_j\rangle_l) \end{cases} \tag{18}$$

Here, the learning rate is denoted by α. The expectations calculated across the dataset and model distributions are $\langle . \rangle_m$and $\langle . \rangle_l$.

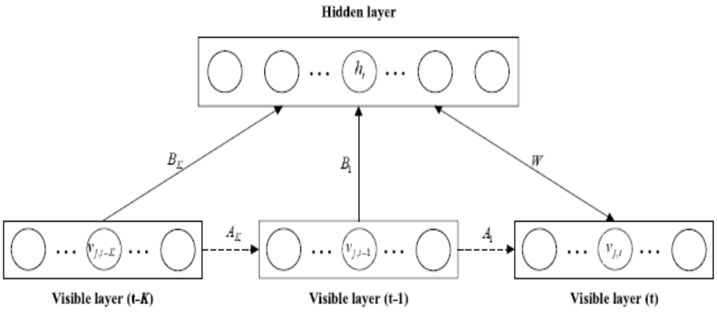

FIGURE 6. CONFIGURATION OF CGBRBM FOR CDBN

The framework of the CGBRBM using one HL and KC1 VL is shown in Figure 6. Observation window size is denoted as K. Equation (16) defines the CGBRBM energy function as follows:

$$E(v_t, \dots, v_{t-K}, h) = -\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{v_j}{\widehat{\sigma}_j^2}h_i w_{ij} - \sum_{i=1}^{n} d_{i,t}h_i + \sum_{j=1}^{m}\frac{\left(v_{j,t}-c_{j,t}\right)^2}{2\widehat{\sigma}_j^2} \qquad (19)$$

Here, the i^{th} sample of the HL is denoted by h_i and the j^{th} member of the VL vector by v_j. The ijth component of the weight matrix (WM) among the HL and VL units is denoted by w_{ij}. The standard deviation of the visible vector's j^{th} component is denoted by $\widehat{\sigma}_j$. The count of VU is m, and the count of HU is n. The HL and VL bias vectors are denoted by d and c. Then, compute d_t and c_t in the following way:

$$\begin{cases} d_t = d + \sum_{k=1}^{K} v_{t-k}B_k \\ c_t = d + \sum_{k=1}^{K} v_{t-k}A_k \end{cases} \qquad (20)$$

Here, the k^{th} prior VL vector is denoted by v_{t-k}. The CPD of the HL and VL units can be computed using equation (20) below:

$$\begin{cases} p(h_i = 1|v_t, \dots., v_{t-N}) = \mathrm{sigm}\left(d_{i,t} + \sum_{j=1}^{m}\frac{w_{ij}v_{j,t}}{\widehat{\sigma}_j^2}\right) \\ p\left(v_{j,t} = v\middle|h\right) = N\left(c_{j,t} + \sum_{i=1}^{n} w_{ij}h_i\,, \widehat{\sigma}_j^2\right) \end{cases} \qquad (21)$$

The CGBRBM structure can be modified by the gradient-based CD technique, and it is given below

$$\begin{cases} w_{ij} = w_{ij} - \alpha\left(\left\langle \frac{v_{j,t}}{\hat{\sigma}_j^2} h_i \right\rangle_m - \left\langle \frac{v_{j,t}}{\hat{\sigma}_j^2} h_i \right\rangle_l\right) \\ a_{ijk} = a_{ijk} - \alpha\left(\left\langle \frac{v_{j,t-k}}{\hat{\sigma}_j^2} v_{i,t} \right\rangle_m - \left\langle \frac{v_{j,t-k}}{\hat{\sigma}_j^2} v_{i,t} \right\rangle_l\right) \\ b_{ijk} = b_{ijk} - \alpha\left(\left\langle \frac{v_{j,t-k}}{\hat{\sigma}_j^2} h_i \right\rangle_m - \left\langle \frac{v_{j,t-k}}{\hat{\sigma}_j^2} h_i \right\rangle_l\right) \\ d_i = d_i - \alpha(\langle h_i \rangle_m - \langle h_i \rangle_l) \\ c_{j,t} = c_{j,t} - \alpha\left(\left\langle \frac{v_{j,t}}{\hat{\sigma}_j^2} \right\rangle_m - \left\langle \frac{v_{j,t}}{\hat{\sigma}_j^2} \right\rangle_l\right) \end{cases} \tag{22}$$

The WM defined as W, A_k and B_k, with w_{ij}, a_{ijk} and b_{ijk}being the corresponding elements. Define the expectations derived from the data and model distributions as $\langle . \rangle_l$and $\langle . \rangle_m$. After pre-training, add a fully connected (FC) output node to the top of the model.The output node is constructed as a multiple node with sigmoid AF specified in equation (17) to display the 2 values denoting the attack and the normal samples. For accomplishing the learnt structure of the NN, the given labelled data and the previously indicated procedures are utilized, and with the application, the model may be fine-tuned (FT) by back-propagation (BP) supervised training [34].

FT PROCEDURE OF CDBN: The FT approach is employed to modify the features, including the weights and biases, following the pre-training phase. The WM and bias vector of the hth HL can be updated as follows if the learning rate is defined as η.

$$\begin{cases} \Delta W_{h,i,j} = -\eta \delta_{h,j} p_{h-1,j} \\ \Delta d_{h,j} = -\eta \delta_{h,j} \end{cases} \tag{23}$$

The updated values for the ijth component of the WM and the jth component of the bias vector are denoted by $\Delta W_{h,i,j}$ and $\Delta d_{h,j}$. The activation probability of the jth component of the (h-1)th HL is denoted by $p_{h-1,j}$.

$$\delta_{h,j} = p_{h,j}(1 - p_{h,j}) \sum_{k=1}^{M} \delta_{h+1,k} W_{h+1,j,k} \tag{24}$$

In the (h+1)th HL, M is the number of elements. $W_{h+1,j,k}$ and $p_{h,j}$represent the activation probability of the jth element of the hth HL and the jkth component of the WM of the (h+1)th HL, respectively. Equation (25), the output layer's (OL) single-unit weight vector and bias value are modified in the following way:

$$\begin{cases} \Delta W_{o,j} = -\eta \delta_o p_{H,j} \\ \Delta d_o = -\eta \delta_o \end{cases} \tag{25}$$

Here, the updated value for the weight vector's jth component is denoted by $\Delta W_{o,j}$. The bias's updated value is Δd_o. The bias's updated value is Δd_o. The activation probability of the jth component of the final HL, whose index is h=H, is denoted by $p_{H,j}$.

$$\delta_o = p_o(1 - p_o)(l_o - L) \tag{26}$$

The output label's predicted value and its actual value are represented as $l_o$and L. The single output unit's activation probability is denoted by p_o.

3.4.4. ENSEMBLE AVERAGING

By merging the detections of several different models, the EA method can assist in addressing the diversity of classifiers. To capture its unique features, each model is trained on a dataset from a particular source.By averaging various models' predictions, the EM is able to represent the diversity of samples.With this strategy, the EM can decide using several techniques depending on dataset variances and trends from various sources.

$$y_{AVG} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{27}$$

For numerous TM i, where i $\in \{1, 2, \dots, n\}$, EA is just the average of the detection result y_i [18].

4. EXPERIMENTS AND DISCUSSION

This part uses the Bot-IoT and N-BaIoT to replicate the DL approaches. The classifiers were executed using the MALABR2020a programming language on a laptop running 64-bit Windows 10 OS with an Intel Core i7 2.2 GHz CPU and 32 GB of RAM in order to measure the experimental findings. The efficacy of these approaches for BN AD was assessed by experiments.

4.1. DATASETS

Bot-IoT Dataset: Research on cyber security can use the publicly available Bot-IoT dataset. It includes four botnet attack scenarios, as well as harmless IoT NT. Additionally, a sliding window of 100 was used to produce new features based on network connection transaction flows. There are 3,668,045 mbotnet attack samples and 477 benign IoT NT samples in this dataset. To characterise the behaviour of an NT sample, 43 features were taken from a network packet [35].

N-BaIoT Dataset: For cyber security research, a freely accessible N-BaIoT dataset is used. Two doorbells, a thermostat, a baby monitor, four security cameras, and a webcam were all part of the IoT testbed that produced this dataset. The 115 attributes are obtained from FE for each of these 23 features, taking into account five time windows (100, 500, 1.5, 10, and 1 minute). 115 statistical features that depict the behaviour snapshots of the NT were taken from the network packets over a number of temporal windows after these commercial IoT devices were attacked with the Mirai and BASHLITE botnets [36]. Benign IoT NT and IoT botnet scenarios, including as ACK, Scan, SYN, and UDPP flooding attacks, are included in this dataset. The outcomes of this study are evaluated using 1,483,658 IoT BN attack samples and 363,979 benign IoT NT samples.

4.2. SIMULATION OUTCOMES

The classification outcomes were assessed using the confusion matrix (CM) and a number of performance metrics, such as P, R, F1-score, and ACC.The number of TP, True Negative (TN), FP, and False Negative (FN) cases were among the metrics used. These metrics are formulated below, (in Eqns (28–31)),

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{28}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{29}$$

$$\mathrm{F1 - Score} = \frac{2 * \mathrm{P} * \mathrm{R}}{\mathrm{P} + \mathrm{R}} \tag{30}$$

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{31}$$

4.3. PERFORMANCE OF DEEP LEARNING METHODS ON Bot-IoT DATASET

The class-wise performance of the DL approaches in the four FS methods (Gaussian Weight Black Widow Optimisation (GWBWO), Local Search Algorithm-Pigeon-Inspired Optimisation (LS-PIO) [38], Multi-Objective Particle Swarm Optimisation (MOPSO) [37], and IWMOA) on the Bot-IoT dataset is assessed in this subsection. Table 1, EDLM classifier achieved a highest precision of 99.00%, other classifiers such as CNN-LSTM [24], Long Short-Deep Recurrent Neural Network (LS-DRNN) [39], DNN, DWLSTM, LWBi-GRU, and CDBN gives results of 91.08%, 94.69%, 96.57%, 97.59%, 97.80%, and 98.38% for IWMOA. EDLM classifier achieved a highest recall of 98.84%, other classifiers such as CNN-LSTM, LS-DRNN, DNN, DWLSTM, LWBi-GRU, and CDBN gives results of 88.71%, 92.62%, 95.43%, 96.65%, 97.27%, and 98.03% for IWMOA. EDLM classifier achieved a highest F1-score of 98.92%, other classifiers such as CNN-LSTM, LS-DRNN, DNN, DWLSTM, LWBi-GRU, and CDBN gives results of 89.88%, 93.64%, 96.02%, 97.10%, 97.63%, and 98.21% for IWMOA. EDLM classifier achieved a highest accuracy of 99.05%, other classifiers such as CNN-LSTM, LS-DRNN, DNN, DWLSTM, LWBi-GRU, and CDBN gives results of 89.59%, 92.56%, 95.32%, 97.25%, 97.74%, and 98.38% for IWMOA.

TABLE 1. METRICS COMPARISON OF DL METHODS ON BoT-IoT DATASET

Methods	**Precision (%)**						
	CNN-LSTM	**LS-DRNN**	**DNN**	**DWLSTM**	**LWBi-GRU**	**CDBN**	**EDLM**
MOPSO	83.62	88.29	90.83	92.71	93.61	94.48	95.63
LS-PIO	85.86	90.01	93.03	94.47	95.02	95.85	96.70
GWBWO	89.12	93.08	94.41	96.52	97.13	97.69	98.33
IWMOA	91.08	94.69	96.57	97.59	97.80	98.38	99.00
Methods	**Recall (%)**						
	CNN-LSTM	**LS-DRNN**	**DNN**	**DWLSTM**	**LWBi-GRU**	**CDBN**	**EDLM**
MOPSO	81.56	85.72	89.18	90.76	91.93	93.12	94.34
LS-PIO	83.73	87.91	90.92	92.40	93.32	94.35	95.76
GWBWO	86.72	90.96	93.97	95.16	95.72	96.32	97.37

IWMOA	88.71	92.62	95.43	96.65	97.27	98.03	98.84
Methods	**F1-Score (%)**						
	CNN-LSTM	**LS-DRNN**	**DNN**	**DWLSTM**	**LWBi-GRU**	**CDBN**	**EDLM**
MOPSO	82.57	86.98	90.00	91.72	92.79	93.80	94.98
LS-PIO	84.78	88.94	91.96	93.41	94.16	95.09	96.23
GWBWO	87.90	92.01	94.18	95.84	96.42	97.00	97.85
IWMOA	89.88	93.64	96.02	97.10	97.63	98.21	98.92
Methods	**Accuracy (%)**						
	CNN-LSTM	**LS-DRNN**	**DNN**	**DWLSTM**	**LWBi-GRU**	**CDBN**	**EDLM**
MOPSO	83.22	85.98	89.30	91.04	92.10	93.08	94.04
LS-PIO	85.08	87.87	91.33	92.81	93.47	94.54	95.72
GWBWO	87.62	90.94	93.24	95.66	96.43	97.16	98.09
IWMOA	89.59	92.56	95.32	97.25	97.74	98.38	99.05

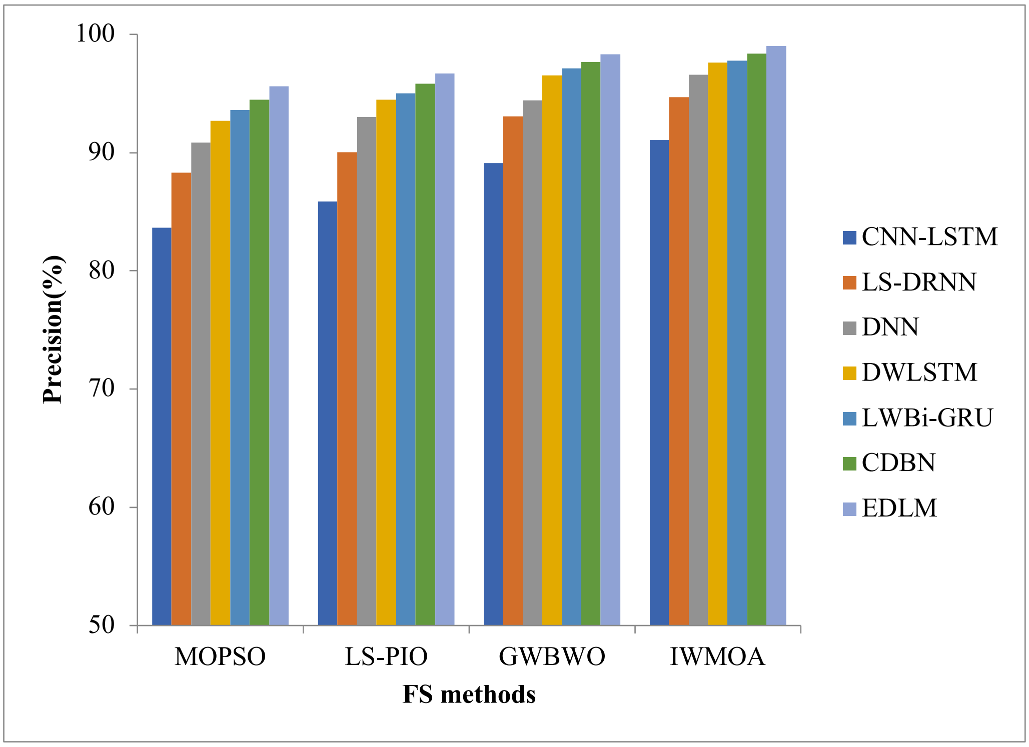

FIGURE 7. PRECISION ANALYSIS OF DL METHODS (BoT-IoT)

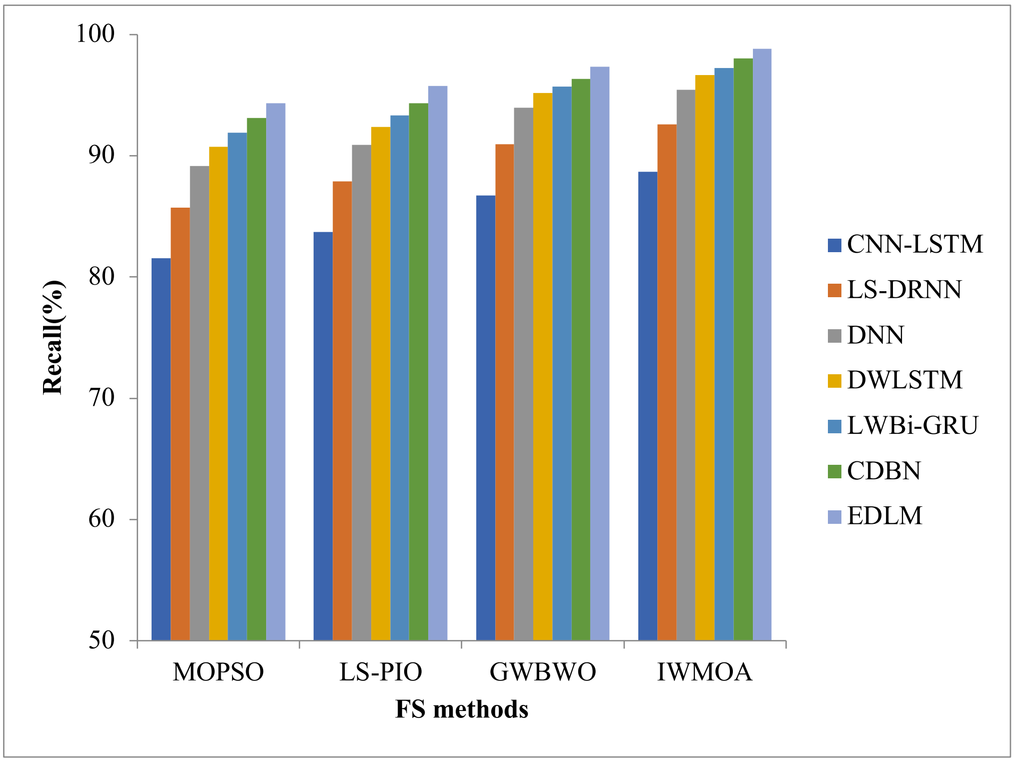

FIGURE 8. RECALL ANALYSIS OF DL METHODS (BoT-IoT)

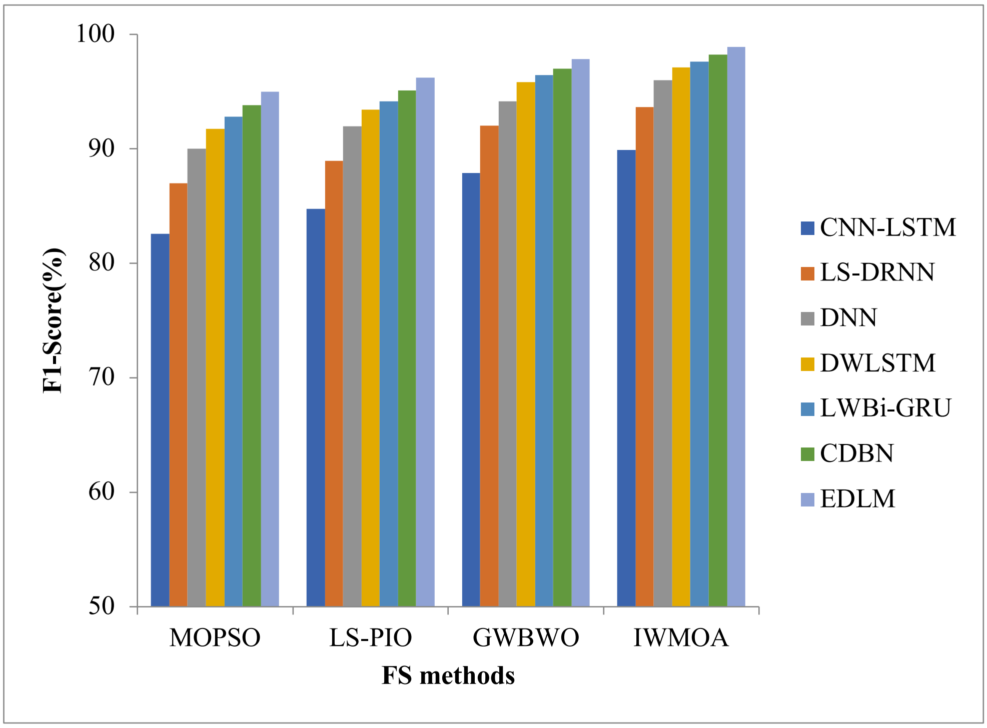

FIGURE 9. F1-SCORE ANALYSIS OF DL METHODS (BoT-IoT)

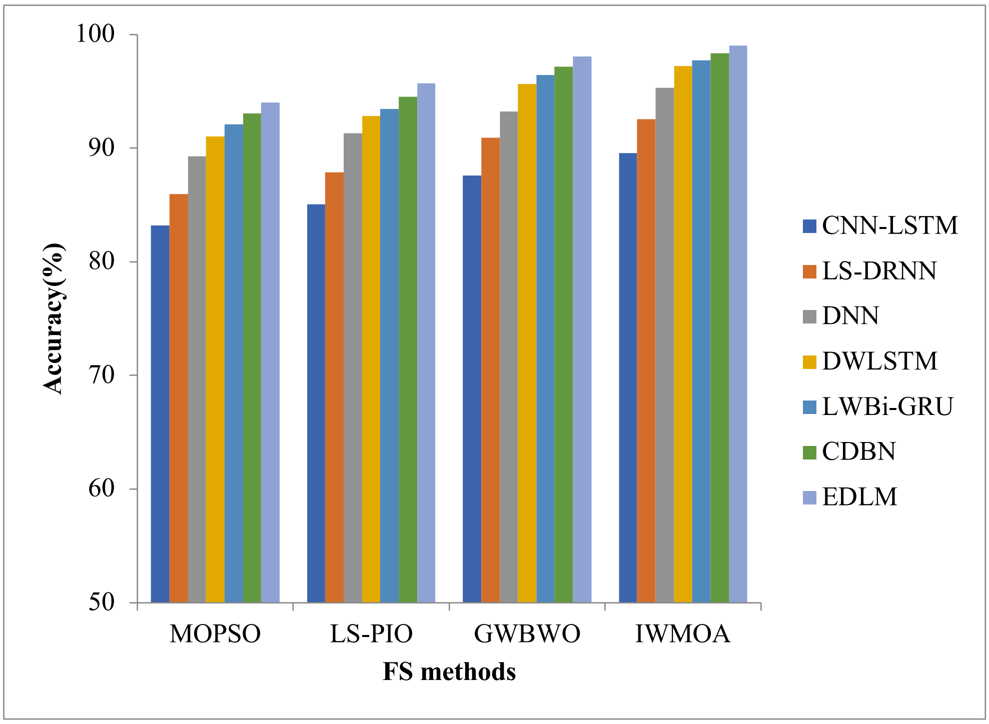

FIGURE 10. ACCURACY ANALYSIS OF DL METHODS (BoT-IoT)

Performance metrics of the DL methods in the four FS methods on Bot-IoT dataset are illustrated in figures 7-10. The EDLM method has highest results of 99.00%, 98.84%, 98.92%, and 99.05% for P, R, f1-score, and ACC. Figure 7, proposed classifier gives highest precision of 95.63%, 96.70%, 98.33%, and 99.00% for MOPSO, LS-PIO, GWBWO, and IWMOA. IWMOA gives lowest precision of 91.08%, 94.69%, 96.57%, 97.59%, 97.80% and 98.38% for CNN-LSTM, LS-DRNN, DNN, DWLSTM, LWBi-GRU, and CDBN. Figure 8, proposed classifier gives highest recall of 94.34%, 95.76%, 97.37%, and 98.84% for MOPSO, LS-PIO, GWBWO, and IWMOA. IWMOA gives lowest recall of 88.71%, 92.62%, 95.43%, 96.65%, 97.27%, and 98.03% for CNN-LSTM, LS-DRNN, DNN, DWLSTM, LWBi-GRU, and CDBN. Figure 9, proposed classifier gives highest f1-score of 94.98%, 96.23%, 97.85%, and 98.92% for MOPSO, LS-PIO, GWBWO, and IWMOA. IWMOA gives lowest f1-score of 89.88%, 93.64%, 96.02%, 97.10%, 97.63%, and 98.21% for CNN-LSTM, LS-DRNN, DNN, DWLSTM, LWBi-GRU, and CDBN. Figure 10, proposed classifier gives highest accuracy of 94.04%, 95.72%, 98.09%, and 99.05% for MOPSO, LS-PIO, GWBWO, and IWMOA. IWMOA gives lowest accuracy of 89.59%, 92.56%, 95.32%, 97.25%, 97.74%, and 98.38% for CNN-LSTM, LS-DRNN, DNN, DWLSTM, LWBi-GRU, and CDBN.

4.4. PERFORMANCE OF DEEP LEARNING METHODS ON N-BaIoT DATASET

The DL methods' class-wise performance in the four FS techniques on N-BaIoT was assessed in this subsection. According to Table 2, the EDLM model had the maximum precision of 99.28%. For IWMOA, the results from other classifiers, including CNN-LSTM, LS-DRNN, DNN, DWLSTM, LWBi-GRU, and CDBN, were 90.21%, 91.80%, 95.97%, 97.22%, 97.96%, and 98.49%. EDLM model achieved a highest recall of 98.51%, other classifiers such as CNN-LSTM, LS-DRNN, DNN, DWLSTM, LWBi-GRU, and CDBN gives results of 89.06%, 91.04%, 94.08%, 95.49%, 96.48%, and 97.55% for IWMOA. EDLM model achieved a highest f1-score of 98.89%, other classifiers such as CNN-LSTM, LS-DRNN, DNN, DWLSTM, LWBi-GRU, and CDBN gives results of 89.62%, 91.41%, 94.98%, 96.34%, 97.21%, and

98.02% for IWMOA. EDLM framewrok attained a highest ACC of 98.84%, other classifiers such as CNN-LSTM, LS-DRNN, DNN, DWLSTM, LWBi-GRU, and CDBN gives results of 90.76%, 92.93%, 95.34%, 96.50%, 97.14%, and 97.96% for IWMOA.

TABLE 2. METRICS COMPARISON OF DL METHODS ON N-BaIoT DATASET

Methods	**Precision (%)**						
	CNN-LSTM	**LS-DRNN**	**DNN**	**DWLSTM**	**LWBi-GRU**	**CDBN**	**EDLM**
MOPSO	83.83	86.21	87.99	89.56	91.07	92.50	94.08
LS-PIO	85.87	88.05	90.25	91.86	93.01	94.16	95.47
GWBWO	88.49	90.31	94.81	95.93	96.90	97.44	98.10
IWHO	90.21	91.80	95.97	97.22	97.96	98.49	99.28
Methods	**Recall (%)**						
	CNN-LSTM	**LS-DRNN**	**DNN**	**DWLSTM**	**LWBi-GRU**	**CDBN**	**EDLM**
MOPSO	83.36	85.57	87.52	89.22	90.34	91.57	92.64
LS-PIO	85.32	87.55	89.89	90.85	92.03	93.28	94.56
GWBWO	87.37	89.51	92.65	93.66	94.59	95.62	96.57
IWHO	89.06	91.04	94.08	95.49	96.48	97.55	98.51
Methods	**F1-Score (%)**						
	CNN-LSTM	**LS-DRNN**	**DNN**	**DWLSTM**	**LWBi-GRU**	**CDBN**	**EDLM**
MOPSO	83.58	85.87	87.74	89.38	90.70	92.03	93.35
LS-PIO	85.59	87.78	90.06	91.34	92.52	93.71	95.02
GWBWO	87.91	89.89	93.72	94.77	95.73	96.52	97.33
IWHO	89.62	91.41	94.98	96.34	97.21	98.02	98.89
Methods	**Accuracy (%)**						
	CNN-LSTM	**LS-DRNN**	**DNN**	**DWLSTM**	**LWBi-GRU**	**CDBN**	**EDLM**
MOPSO	84.26	87.30	89.37	91.16	92.32	93.30	94.47
LS-PIO	86.48	89.08	91.14	92.48	93.24	94.21	95.56
GWBWO	88.85	91.15	93.72	94.96	95.77	96.72	97.60
IWHO	90.76	92.93	95.34	96.50	97.14	97.96	98.84

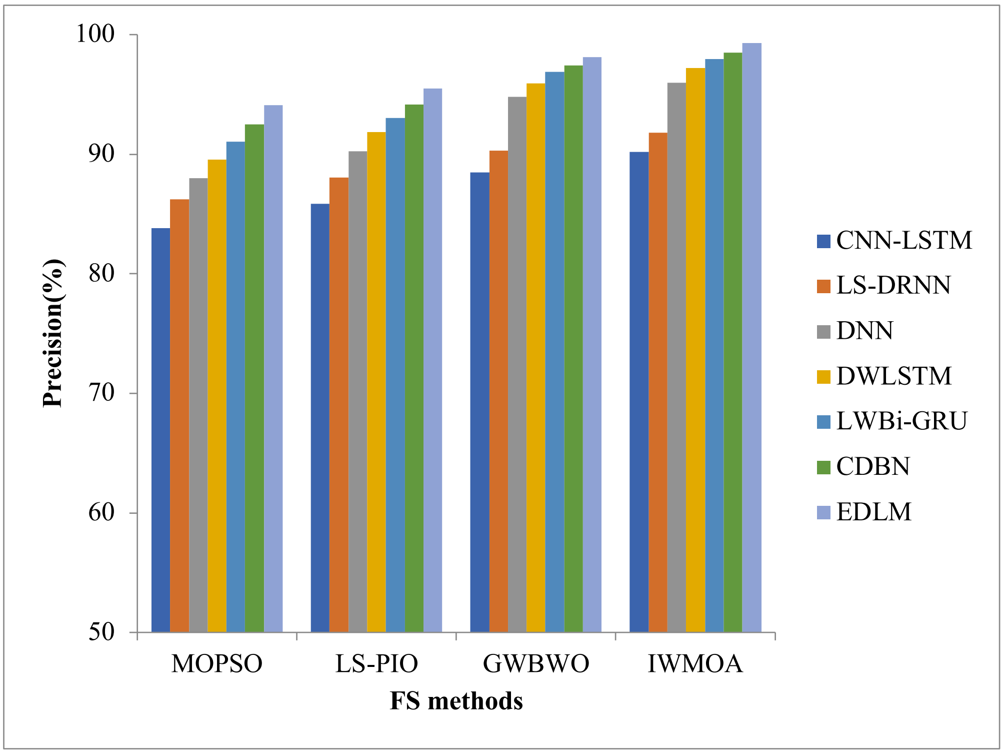

FIGURE 11. PRECISION ANALYSIS OF DL METHODS (N-BaIoT)

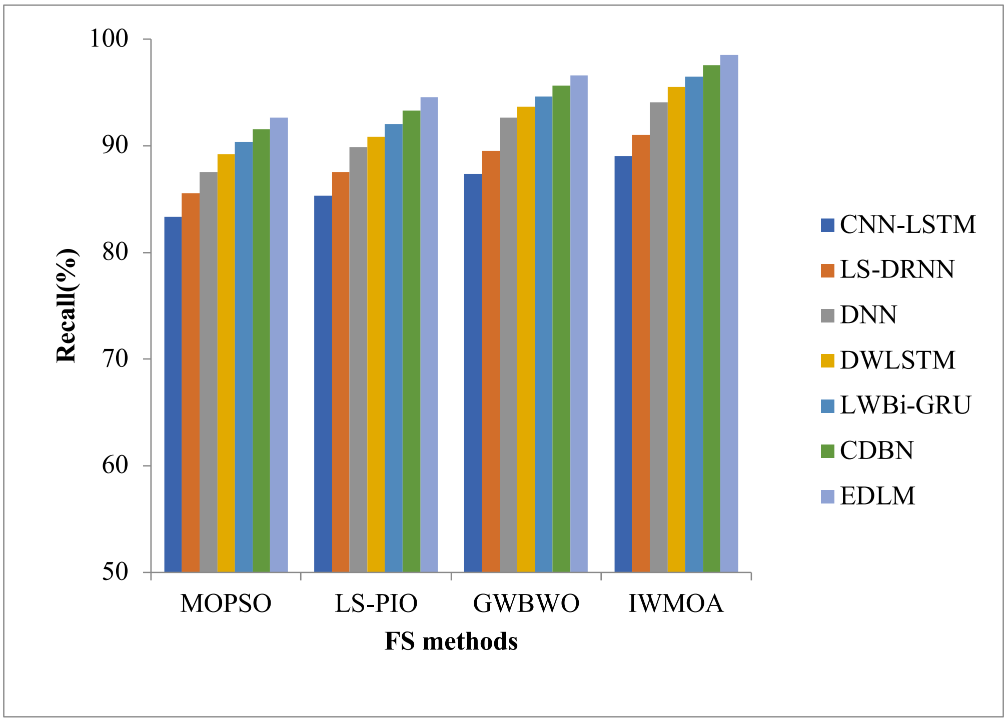

FIGURE 12. RECALL ANALYSIS OF DL METHODS (N-BaIoT)

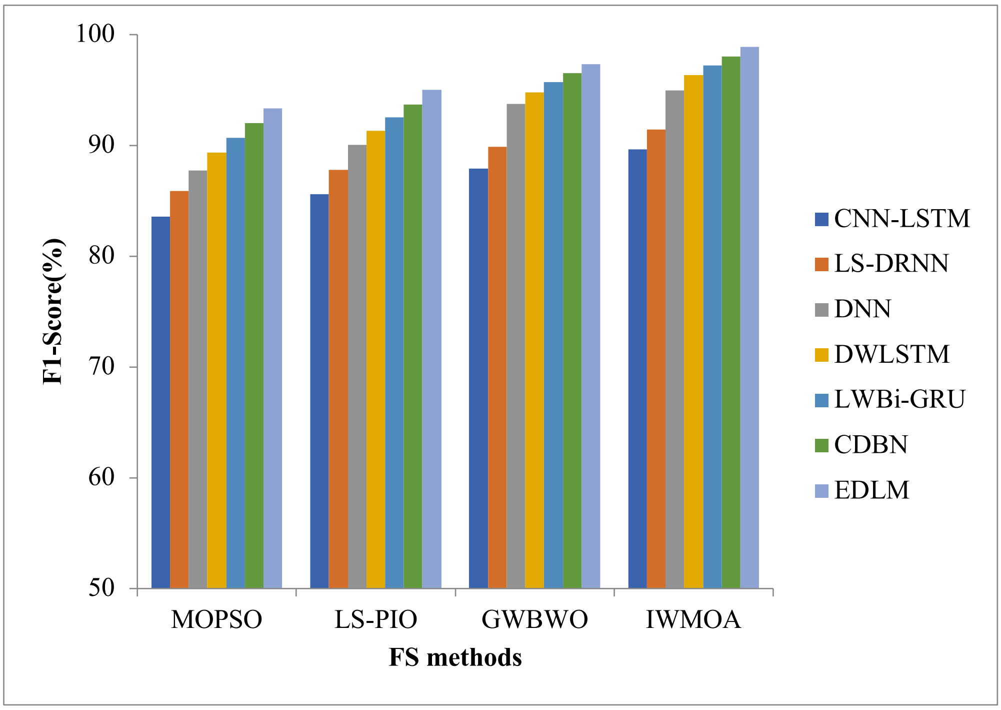

FIGURE 13. F1-SCORE ANALYSIS OF DL METHODS (N-BaIoT)

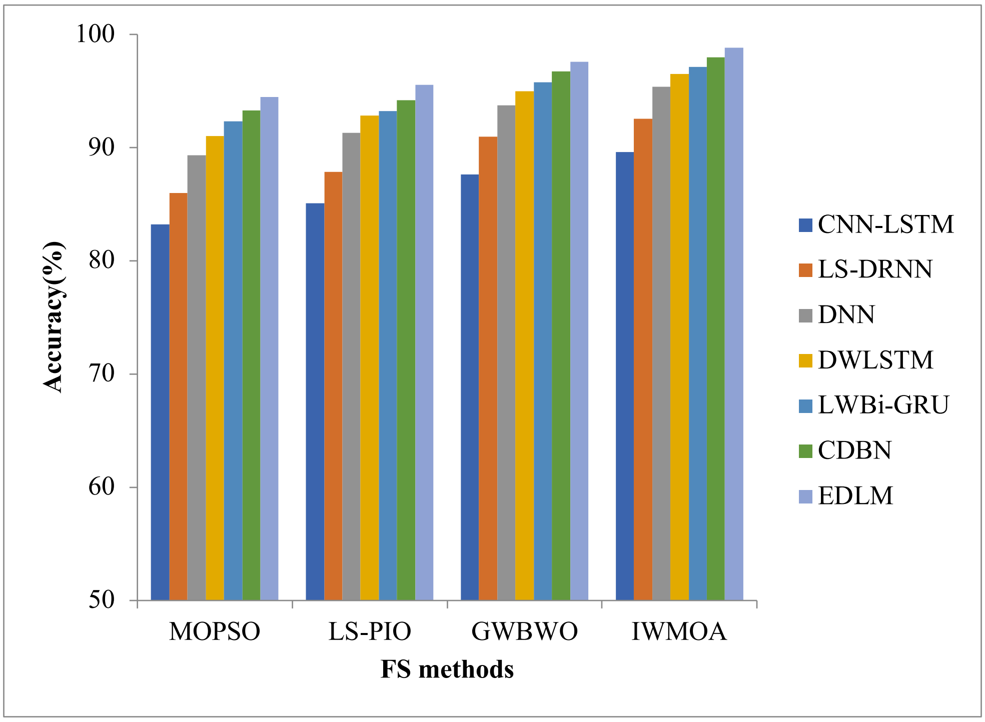

FIGURE 14. ACCURACY ANALYSIS OF DL METHODS (N-BaIoT)

The performance metrics of the DL methods in the four FS approaches on the N-BaIoT database are assessed in Figures 11–14. The EDLM method has highest results of 99.28%, 98.51%, 98.89%, and 98.84% for P, R, f1-score, and ACC. Figure 11, proposed classifier gives highest precision of 94.08%, 95.47%, 98.10%, and 99.28% for MOPSO, LS-PIO, GWBWO, and IWMOA. IWMOA gives lowest precision of 90.21%, 91.80%, and 93.66% for CNN-LSTM, LS-DRNN, DNN, DWLSTM, LWBi-GRU, and CDBN. Figure 12, proposed classifier gives highest recall of 92.64%, 94.56%, 96.57%, and 98.51% for MOPSO, LS-PIO, GWBWO, and IWMOA. IWMOA gives lowest recall of 89.06%, 91.04%, 94.08%, 95.49%, 96.48%, and

97.55% for CNN-LSTM, LS-DRNN, DNN, DWLSTM, LWBi-GRU, and CDBN. Figure 13, proposed classifier gives highest f1-score of 93.35%, 95.02%, 97.33%, and 98.89% for MOPSO, LS-PIO, GWBWO, and IWMOA. IWMOA gives lowest f1-score of 89.62%, 91.41%, 94.98%, 96.34%, 97.21%, and 98.02% for CNN-LSTM, LS-DRNN, DNN, DWLSTM, LWBi-GRU, and CDBN. Figure 14, proposed classifier gives highest accuracy of 94.47%, 95.56%, 97.60%, and 98.84% for MOPSO, LS-PIO, GWBWO, and IWMOA. IWMOA gives lowest accuracy of 90.76%, 92.93%, 95.34, 96.50%, 97.14%, and 97.96% for CNN-LSTM, LS-DRNN, DNN, DWLSTM, LWBi-GRU, and CDBN.

5. CONCLUSION AND FUTURE WORK

This study introduces an EDLM-based classifier for AD and an IWMOA-based FS. For FS, IWMOA simulates the human contact between a mother and her children, has been introduced. Compared to individual models, EDLM has demonstrated superior predictors and the ability to automate BND. The ensemble of DWLSTM, LWBi-GRU, and CDBN serves as the foundation for the suggested BND. Through the use of specially designed gates and memory cells, it can train using the duration to recall the state data and when to forget via DWLSTM.Using the Levy distribution, the update gate in the LWBi-GRU model assists in identifying the historical data that must be transmitted with the future data and weight updates. The LWBi-GRU model automatically excludes historical data that goes backwards and extracts feature information in a forward direction. With the help of the CDBN classifier, CGBRBM can efficiently identify botnet attacks and learn the dataset's features. By averaging the predictions of various models, the EM can capture the diversity of samples. The Bot-IoT and N-BaIoT database, that have been downloaded from UCI and Kaggle, are used to simulate botnet attack scenarios. A thorough set of evaluation indicators, including P, R, F1-score, and ACC, are specifically applied for evaluating the efficacy of the recommended method. This illustrates how important it is to properly balance model efficacy and complexity across a variety of purposes. Future study should use reinforcement learning for greater training because it will be fully automated.

REFERENCES

1. Koroniotis N., N. Moustafa, and E. Sitnikova, “Forensics and deep learning mechanisms for botnets in internet of things: A survey of challenges and solutions,” IEEE Access, vol. 7, pp. 61 764–61 785, 2019.
2. Bertino E. and N. Islam, “Botnets and internet of things security,” Computer, vol. 50, no. 2, pp. 76–79, 2017.
3. Koroniotis N., N. Moustafa, E. Sitnikova, and B. Turnbull, “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.
4. Abdulganiyu, O.H., Ait Tchakoucht, T. and Saheed, Y.K., 2023. A systematic literature review for network intrusion detection system (IDS). International journal of information security, 22(5), pp.1125-1162.
5. Mishra, N. & Pandya, S. Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review. IEEE Access **9**, 59353–59377 (2021).

6. Thakkar A, Lohiya R (2019) Review on machine learning and deep learning perspectives of ids for IoT: recent updates, security issues, and challenges. Arch Computat Methods Eng 28:3211–3243.
7. Nguyen, B.H.; Xue, B.; Zhang, M. A survey on swarm intelligence approaches to feature selection in data mining. Swarm Evol. Comput. **2020**, 54, pp.1-27.
8. Jiang, F., Fu, Y., Gupta, B.B., Liang, Y., Rho, S., Lou, F., Meng, F. and Tian, Z., 2018. Deep learning based multi-channel intelligent attack detection for data security. IEEE transactions on Sustainable Computing, 5(2), pp.204-212.
9. Wu, Y., Wei, D. and Feng, J., 2020. Network attacks detection methods based on deep learning techniques: A survey. Security and Communication Networks, 2020(1), pp.1-17.
10. Al-Garadi, M.A., Mohamed, A., Al-Ali, A.K., Du, X., Ali, I. and Guizani, M., 2020. A survey of machine and deep learning methods for internet of things (IoT) security. IEEE communications surveys & tutorials, 22(3), pp.1646-1685.
11. Berman, D.S., Buczak, A.L., Chavis, J.S. and Corbett, C.L., 2019. A survey of deep learning methods for cyber security. Information, 10(4), pp.1-35.
12. Ferrag, M.A., Maglaras, L., Moschoyiannis, S. and Janicke, H., 2020. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, pp.1-20.
13. Dong, X., Yu, Z., Cao, W., Shi, Y. and Ma, Q., 2020. A survey on ensemble learning. Frontiers of Computer Science, 14, pp.241-258.
14. Mienye, I.D. and Sun, Y., 2022. A survey of ensemble learning: Concepts, algorithms, applications, and prospects. IEEE Access, 10, pp.99129-99149.
15. Albulayhi, K.; Sheldon, F.T. An Adaptive Deep-Ensemble Anomaly-Based Intrusion Detection System for the Internet of Things. In Proceedings of the 2021 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 10–13 May 2021; pp. 187–196.
16. Baker, Q.B. and Samarneh, A., 2024. Feature selection for IoT botnet detection using equilibrium and Battle Royale Optimization. Computers & Security, 147, p.104060.
17. Taher, F., Abdel-Salam, M., Elhoseny, M. and El-Hasnony, I.M., 2023. Reliable machine learning model for IIoT botnet detection. IEEE Access, 11, pp.49319-49336.
18. Wardana, A.A., Kołaczek, G., Warzyński, A. and Sukarno, P., 2024. Ensemble averaging deep neural network for botnet detection in heterogeneous Internet of Things devices. Scientific Reports, 14(1), pp.1-18.
19. Meidan Y., M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-BaIoT: Network-based detection of IoT botnet attacks using deep autoencoders," IEEE Pervasive Computing, vol. 17, no. 3, pp. 12–22, 2018.
20. Popoola S. I., B. Adebisi, M. Hammoudeh, G. Gui, and H. Gacanin, "Hybrid deep learning for botnet attack detection in the internet of things networks," IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4944– 4956, 2021.
21. Ge, M., Syed, N.F., Fu, X., Baig, Z. and Robles-Kelly, A., 2021. Towards a deep learning-driven intrusion detection approach for Internet of Things. Computer Networks, 186, pp.1-20.
22. Ahmad, Z., Shahid Khan, A., Nisar, K., Haider, I., Hassan, R., Haque, M.R., Tarmizi, S. and Rodrigues, J.J., 2021. Anomaly detection using deep neural network for IoT architecture. Applied Sciences, 11(15), pp.1-19.

23. Abu Al-Haija, Q. and Al-Dala'ien, M.A., 2022. ELBA-IoT: An ensemble learning model for botnet attack detection in IoT networks. Journal of Sensor and Actuator Networks, 11(1), pp.1-15.
24. Alkahtani, H. and Aldhyani, T.H., 2021. Botnet attack detection by using CNN-LSTM model for Internet of things applications. Security and Communication Networks, 2021(1), pp.1-23.
25. Hussain, F., Abbas, S.G., Pires, I.M., Tanveer, S., Fayyaz, U.U., Garcia, N.M., Shah, G.A. and Shahzad, F., 2021. A two-fold machine learning approach to prevent and detect IoT botnet attacks. IEEE Access, 9, pp.163412-163430.
26. Matoušová, I., Trojovský, P., Dehghani, M., Trojovská, E. and Kostra, J., 2023. Mother optimization algorithm: A new human-based metaheuristic approach for solving engineering optimization. Scientific Reports, 13(1), pp.1-26.
27. Al-Andoli, M.N., Tan, S.C., Sim, K.S., Seera, M. and Lim, C.P., 2023. A parallel ensemble learning model for fault detection and diagnosis of industrial machinery. IEEE Access, 11, pp.39866-39878.
28. Huang, Y., Feng, X., Li, B., Xiang, Y., Wang, H., Liu, T. and Qin, B., 2024. Ensemble learning for heterogeneous large language models with deep parallel collaboration. Advances in Neural Information Processing Systems, 37, pp.119838-119860.
29. Fan H., M.Jiang, L.Xu, H.Zhu, J. Cheng, and J.Jiang, "Comparison of long short term memory networks and the hydrological model in runoff simulation," Water, vol. 12, no. 1, pp.1-15.
30. Alosaimi S., and S.M.Almutairi, "An intrusion detection system using BoT-IoT," Applied Sciences, vol. 13, no. 9, pp.1-15, Apr 2023.
31. Li, X., Ma, X., Xiao, F., Xiao, C., Wang, F. and Zhang, S., 2022. Time-series production forecasting method based on the integration of Bidirectional Gated Recurrent Unit (Bi-GRU) network and Sparrow Search Algorithm (SSA). Journal of Petroleum Science and Engineering, 208, pp.109309.
32. Mogarala Guruvaya, A., Kollu, A., Divakarachari, P.B., Falkowski-Gilski, P. and Praveena, H.D., 2024. Bi-GRU-APSO: Bi-Directional Gated Recurrent Unit with Adaptive Particle Swarm Optimization Algorithm for Sales Forecasting in Multi-Channel Retail. In Telecom, Vol. 5, No. 3, pp. 537-555.
33. Wu, E.Q., Zhou, G.R., Zhu, L.M., Wei, C.F., Ren, H. and Sheng, R.S., 2019. Rotated sphere haar wavelet and deep contractive auto-encoder network with fuzzy Gaussian SVM for pilot's pupil center detection. IEEE transactions on cybernetics, 51(1), pp.332-345.
34. Tran S. N. and A. S. d'Avila Garcez, ``Deep logic networks: Inserting and extracting knowledge from deep belief networks," IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 2, pp. 246-258, Feb. 2018.
35. Gümüşbaş, D., Yıldırım, T., Genovese, A. and Scotti, F., 2020. A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems. IEEE Systems Journal, 15(2), pp.1717-1731.
36. Alosaimi, S. and Almutairi, S.M., 2023. An intrusion detection system using BoT-IoT. Applied Sciences, 13(9), pp.1-15.

37. Habib, M., Aljarah, I., Faris, H. and Mirjalili, S., 2020. Multi-objective particle swarm optimization for botnet detection in internet of things. Evolutionary Machine Learning Techniques: Algorithms and Applications, pp.203-229.
38. Alghanam, O.A., Almobaideen, W., Saadeh, M. and Adwan, O., 2023. An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning. Expert Systems with Applications, 213, pp.1-16.
39. Popoola, S.I., Adebisi, B., Ande, R., Hammoudeh, M. and Atayero, A.A., 2021. Memory-efficient deep learning for botnet attack detection in IoT networks. Electronics, 10(9), pp.1-18.