AIMS Mathematics, 9(6): 15796–15818.
DOI: 10.3934/math.2024763
Received: 16 January 2024
Revised: 08 April 2024
Accepted: 10 April 2024
Published: 06 May 2024

https://www.aimspress.com/journal/Math

Research article

Enhancing cybersecurity in cloud-assisted Internet of Things environments: A unified approach using evolutionary algorithms and ensemble learning

Mohammed Aljebreen[1], Hanan Abdullah Mengash[2], Khalid Mahmood[3], Asma A. Alhashmi[4,*] and Ahmed S. Salama[5]

[1] Department of Computer Science, Community College, King Saud University, P.O. Box 28095, Riyadh 11437, Saudi Arabia
[2] Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[3] Department of Information Systems, College of Science & Art at Mahayil, King Khalid University, Saudi Arabia
[4] Department of Computer Science at College of Science, Northern Border University, Arar, Saudi Arabia
[5] Department of Electrical Engineering, Faculty of Engineering & Technology, Future University in Egypt, New Cairo 11845, Egypt

* **Correspondence:** Email: Asma.alhashmi@nbu.edu.sa.

Abstract: Internet of Things (IoT) security is an umbrella term for the strategies and tools that protect devices connected to the cloud, and the network they use to connect. The IoT connects different objects and devices through the internet to communicate with similarly connected machines or devices. An IoT botnet is a network of infected or cooperated IoT devices that can be remotely organized by cyber attackers for malicious purposes such as spreading malware, stealing data, distributed denial of service (DDoS) attacks, and engaging in other types of cybercrimes. The compromised devices can be included in any device connected to the internet and communicate data with, e.g., cameras, smart home appliances, routers, etc. Millions of devices can include an IoT botnet, making it an attractive tool for cyber attackers to launch attacks. Lately, cyberattack detection using deep learning (DL) includes training neural networks on different datasets to automatically detect patterns indicative of cyber threats, which provides an adaptive and proactive approach to cybersecurity. This study presents an

evolutionary algorithm with an ensemble DL-based botnet detection and classification (EAEDL-BDC) approach. The goal of the study is to enhance cybersecurity in the cloud-assisted IoT environment via a botnet detection process. In the EAEDL-BDC technique, the primary stage of data normalization using Z-score normalization is performed. For the feature selection process, the EAEDL-BDC technique uses a binary pendulum search algorithm (BPSA). Moreover, a weighted average ensemble of three models, such as the modified Elman recurrent neural network (MERNN), gated recurrent unit (GRU), and long short-term memory (LSTM), are used. Additionally, the hyperparameter choice of the DL approaches occurs utilizing the reptile search algorithm (RSA). The experimental outcome of the EAEDL-BDC approach can be examined on the N-BaIoT database. The extensive comparison study implied that the EAEDL-BDC technique reaches a superior accuracy value of 99.53% compared to other approaches concerning distinct evaluation metrics.

Keywords: evolutionary algorithm; IoT; ensemble learning; botnet detection; reptile search algorithm
Mathematics Subject Classification: 11Y40

1. Introduction

An IoT-based cloud infrastructure is a wide network that contains many IoT-assisted devices and applications [1]. An IoT-based cloud structure also contains services and standards vital for safeguarding, handling, and linking dissimilar IoT devices and uses. Cloud computing (CC) delivers scalability and steady upgrades on hardware and software for huge amounts of industrial uses [2]. Furthermore, the cloud allows the consumer to make effective use of network resources and offers a variety of safety performances. Of these benefits, it can be obvious that the viewpoint of CC is an effective perspective [3]. CC and basic technologies offer numerous possible chances for businesses, and it has a huge array of uses, platforms, services, and solutions, with more likely in the future. The achievement of some cloud-based performance is greatly dependent on delivering the best experience to software developers, cloud managers, and users [4]. There are exact features to the assumption of clouds like compliance, complexity, privacy, reliance, control, security, and price. Safety in CC is measured as a vital obstacle, and so data and uses can exist at many layers reliant on the preferred cloud service method [5]. Furthermore, IoT devices can effortlessly be affected by DDoS and Mirai botnet attacks, and both of these attacks are dangerous when compared to other attacks. Besides, the occurrence of DDoS attacks can affect the data link layer because it can close all the web pages, which is in the present procedure [6]. Attackers who launch the bots to corrupt or damage the method are termed a Mirai botnet, performing like a robot and taking control of the entire system.

Intrusion detection systems (IDS) have resulted in increased attention from researchers toward safe IoT devices, along with commenced attacks from challengers [7]. Most of the researchers often chose machine learning (ML) models to identify network traffic anomalies produced by recognized and recently presented assaults and to caution the suitable system control nodes to block such traffic [8]. ML has been considered by computing resources during all its stages. For IDS, extracting features from connection packets is an essential action for running, testing, and building the network. Composed data models need scaling and cleaning. Constructing a technique needs feature classification, and validation. All those actions must be implemented in time order or else slipping risky packets invisible is predictable [9]. Combining ML within an embedded system process must

regard the computing resources range like CPU design, the graphical processing unit (GPU), network connectivity, and the physical memory size [10]. Those kinds of features simulate the operational possibility of ML-IDS on IoT devices concerning packet miss rate, forecast output, and computing resource application.

Present methods for DL-based botnet recognition in the IoT-cloud face important challenges, including the collection of related features from varied and dynamic IoT data sources, the combination of numerous DL methods over ensemble learning to improve recognition accuracy and flexibility, and the optimization of hyperparameters to strike a balance between recognition efficacy and computational efficiency. These represent the difficulties related to managing varied IoT data streams, and are essential for strong feature extractor models personalized to the unique features of the IoT devices. Also, the efficiency and scalability of the DL method in handling massive quantities of streaming data presents important hurdles, compounded by the dynamic nature of botnet behaviors and the developing threat landscape. Hyperparameter tuning requires careful optimization to strike a balance between model performance and computational efficiency, a challenge impaired by the dynamic nature of IoT atmospheres. Solving these challenges efficiently is vital to understanding the complete potential of DL-based botnet recognition methods in the maintenance of IoT-cloud organizations besides sophisticated cyber dangers.

There is a persistent need to develop effective DL-based solutions personalized to the exclusive tasks of identifying botnets within IoT-cloud settings. This involves developing models proficient in precisely categorizing malicious actions while minimalizing false positives, and familiarizing them with the dynamic nature of IoT systems. Furthermore, safeguarding scalability, real-time observing, and compatibility with resource-constrained IoT strategies pose additional challenges. Thus, the problem consists of inventing strong DL-based botnet recognition devices that determine the details of IoT-cloud organizations, eventually improving cybersecurity and safeguarding crucial methods and data.

Therefore, this study presents an evolutionary algorithm with an ensemble DL-based botnet detection and classification (EAEDL-BDC) approach. In the EAEDL-BDC algorithm, the primary stage of data normalization using Z-score normalization is performed. For the feature selection (FS) process, the EAEDL-BDC technique uses a binary pendulum search algorithm (BPSA). Moreover, a weighted average ensemble of three models, including the modified Elman recurrent neural network (MERNN), gated recurrent unit (GRU), and long short-term memory (LSTM), are used. Furthermore, the hyperparameter selection of the DL models takes place using the reptile search algorithm (RSA). The experimental value of the EAEDL-BDC approach can be examined on the N-BaIoT dataset.

The remaining sections of the article are arranged as follows: Section 2 offers a literature review, and Section 3 presents the proposed method. Then, Section 4 evaluates the results, and Section 5 concludes the work.

2. Related works

The authors of [11] established the IoT with Cloud-Aided Botnet Detection and Classification employing Rat Swarm Optimizer with DL (BDC-RSODL) technique. Mainly, the system data was pre-processed to generate it well-suited for advanced processes. Also, the RSO technique was developed for effectual FS. In addition, the LSTM technique was employed for the detection and identification of botnets. Lastly, SCA was implemented for perfecting the parameters connected to the LSTM method.

In [12], an intelligent and safe edge-enabled computing (ISEC) technique was developed for maintainable towns utilizing Green IoT. The developed technique creates optimum features utilizing DL for data routes, which aids in training the sensors to forecast the best routes near edge servers (ES). Additionally, the combination of dispersed hashing with a chaining plan benefits safety and results in an effective computing method. Prabhu et al. [13] proposed a new DL plan named Modified Learning-based CAD (MLCAD) which adjusts the features from the conventional safety handle system termed IAIS. The projected MLCAD technique classifies the DDoS assaults over the cloud atmosphere by analyzing the authentication and authorization reasons of the particular consumer.

Alrowais et al. [14] presented a Botnet Recognition employing the Chaotic Binary Pelican Optimizer Algorithm with DL (BNT-CBPOADL) model. In this technique, the Z-score normalized was functional for pre-processed. The convolutional VAE (CVAE) technique has been useful for the recognition of botnets. Finally, the arithmetical optimizer algorithm (AOA) has been used for optimum hyperparameter tuning. In [15], a united structure for Leveraging the Safety of IoT Application (LSITA) with a Remote Patient Monitoring System (RPMS) was developed. It permits cloud-aided authentication, safe communications between gatherings involved in IoT use, and an enhanced main distribution technique for multiple user data analytics atmosphere. Dissimilar safety systems work composed with a unified combination. Aljebreen et al. [16] developed a Political Optimizer Algorithm by an HDL Aided Malicious URL Detection and Classification for Cybersecurity (POAHDL-MDC) method. This method executes a pre-processing step to convert the information to a well-matched setup, and a Fast Text word embedded procedure is involved. For mischievous URL recognition, an HDL method incorporates the features of stacked AE (SAE) and BiLSTM. Lastly, POA can be demoralized for boosting parameter tuning.

Wang et al. [17] developed a privacy-enhanced retrieval technology (PERT) for cloud-aided IoT. This structure has been intended over a hidden index sustained by ES and a graded retrieval method that conserves data confidentiality by hiding the info of data communication among the cloud and ES. For the categorized retrieval method, the technique aimed for a data partition plan. The ES stocks partial data. In [18], a novel Lightweight Hybrid Encryption (LHE) technique was developed. Primarily, the input medical images are encoded over effective substitution box (S-box) elliptic curves and a block cipher. An optimum block has been nominated by employing an adaptive COOT optimizer approach. Lastly, the encoded medical image information is safely kept in the cloud storage platform.

Despite the potential of DL in boosting the cybersecurity of cloud-assisted IoT, there is a prominent research gap about the addition of vital optimizer models. FS is vital for classifying the most related data features that contribute to precise threat recognition, yet its application in DL-based cybersecurity for cloud-assisted IoT remains underexplored. Likewise, effectual hyperparameter tuning is vital for enhancing the performance of DL techniques in managing various cyber threats professionally. Also, the application of EL models, which fuse manifold methods for improved predictive robustness and accuracy, remains unexplored in the situation of cloud-assisted IoT cybersecurity. Connecting these research gaps is vital for developing more effective and strong defense devices against cyberattacks directing cloud-aided IoT infrastructure. The summery of the existing work is illustrated in Table 1.

Table 1. Summary of existing works.

Reference No.	Objective	Method	Dataset	Measures
Alshahrani et al. [11]	To detect and classify IoT-assisted botnet activities	RSO feature selection, LSTM classification, and SCA-based hyperparameter tuning	Bot-IoT database	Accuracy, Precision, Recall, F-Score, and AUC-Score
Haseeb et al. [12]	To establish intellectual and safe edge-enabled computing for sustainable cities employing Green IoT	ISEC model	Real-time data	Energy consumption, throughput, delay, and route interruption
Prabhu, Prema, and Perumal [13]	To construct a DL model to detect DDoS attacks in the cloud atmosphere	MLCAD	Standard dataset	Accuracy, Precision, Recall, F-Score, and AUC-Score
Alrowais et al. [14]	The objective of the BNT-CBPOADL technique is in the precise recognition and classification of botnet attacks in the IoT atmosphere	CBPOA, CVAE, and AOA	Bot-IoT database	Accuracy, Precision, Recall, F-Score, and AUC-Score
Ahmed, Kannan, and Polamuri [15]	To improve the security and privacy of IoT applications, specifically in the context of a Remote Patient Monitoring System (RPMS)	LSITA	Standard dataset	Accuracy, Precision, Recall, F-Score, and AUC-Score
Aljebreen et al. [16]	To develop a Hybrid Deep Learning Assisted Malicious URL Detection and Classification for Cybersecurity	SAE-BiLSTM, POA memory (Bi-LSTM) POA	Malicious URL dataset	Accuracy, Sensitivity, Specificity, and F-Score
Wang et al. [17]	To improve privacy in cloud-assisted IoT retrieval	PERT	-	Accuracy, Precision, Recall, F-Score, and AUC-Score
Padma Vijetha Dev, and Venkata Prasad [18]	To improve the security of medical images in IoT utilizing a Lightweight Hybrid Encryption (LHE) approach	FEC, adaptive COOT optimization model	-	Accuracy, Precision, Recall, F-Score, and AUC-Score

3. The proposed model

In this manuscript, the EAEDL-BDC technique is introduced. The goal of the study is to enhance cybersecurity in cloud-assisted IoT platforms via a botnet detection process. The EAEDL-BDC technique comprises data normalization using Z-score normalization, BPSA-based FS, ensemble learning, and RSA-based parameter tuning. Figure 1 demonstrates the workflow of the EAEDL-BDC approach.

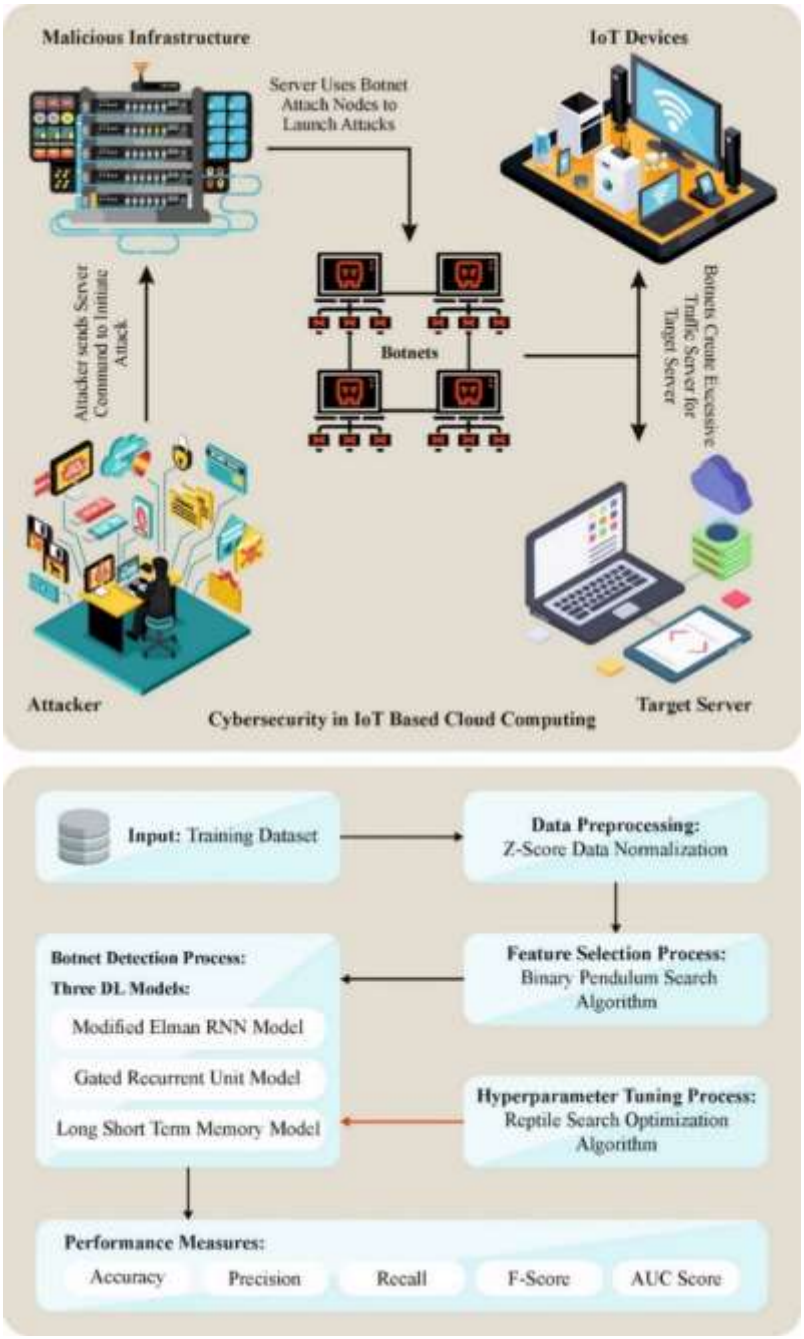

Figure 1. Workflow of EAEDL-BDC technique.

3.1. Data normalization

The primary stage of data normalization using Z-score normalization can be performed in this step. Z-score normalization, a.k.a. standardization, is a statistical approach used to center and rescale datasets dividing by standard deviation (SD) and subtracting by mean [19]. Z-score normalization transforms the data distribution into a uniform distribution with a mean of zero and an SD of one. This process is especially suitable in statistical analyses and ML as it ensures that variables with varying scales can equally contribute to the analysis, facilitating better model performance and preventing

dominance by variables with larger magnitudes.

3.2. Feature selection

For the feature selection process, the EAEDL-BDC technique uses the BPSA algorithm. In 2022, Ab. Aziz, N.A. and Ab. Aziz, K. introduced PSA, a new population-based metaheuristic algorithm based on the harmonic motion of a simple pendulum to resolve continuous optimization problems [20]. The equation of motion is related to the one suggested in SCA. This study incorporates an exponential function that enhances the balance between exploitation and exploration.

The search agent is initialized at random and their location is upgraded by the next expression.

$$X_{i,j}^{t} = X_{i,j}^{t} + pend_{i,j}^{t} \cdot \left(Best_j - X_{i,j}^{t}\right) \tag{1}$$

Where the location of i^{th} solution in the j^{th} dimension at the t^{th} iteration is expressed as $X_{i,j}^{t}$, $pend_{i,j}^{t}$ shows the parameter that can be evaluated by Eq (2), and $Best_j$ refers to the location of the optimum solution in j^{th} dimension at the t^{th} iteration:

$$pend_{i,j}^{t} = 2 \cdot e^{(-t/t\max)} \cdot cos\ (2 \cdot \pi \cdot rand) \tag{2}$$

In Eq (2), t stands for the current iteration, $tmax$ denotes the maximum iteration counts, and *rand* denotes a uniformly distributed random integer between zero and one. The pseudocode of PSA is given in Algorithm 1.

Algorithm 1: Pseudocode of PSA

Input: The population $X = \{X_1, X_2, \ldots, X_i\}$
Output: The updated population $X' = \{X_1', X_2', \ldots, X_i'\}$ and *Best*
Initialize X random population
Assess the objective function of all the individuals from the X population
Recognize the fittest individual from the population ($Best$)
For iteration (t) do
For performance (i) do
For dimensional (j) do
Upgrade $pend_{i.j}^{t}$ using Eq (2)
Upgrade the location of $X_{i,j}^{t}$ by Eq (1)
End for
End for
Calculate the objective function of all the individuals within the X population
Upgrade $Best$
End for
Return X' the upgraded population whereas $Best$ is an optimum outcome

PSA is a recent meta-heuristic algorithm intended to resolve continuous optimizer problems. It can be essential to convert the solution into the binary domain for resolving the FS. In addition, the classical Two-Step algorithm is used to binarize the continuous metaheuristic. This study presents five binarization rules, and eight different transfer functions are used. Equation (4) represents the

binarization rule, and Eq (3) represents the transfer function.

$$T(d_w^j) = \left|\frac{2}{\pi} arctan\left(\frac{\pi}{2} d_w^j\right)\right| \tag{3}$$

$$X_{new}^j = \begin{cases} 1 \; if \; rand \leq T(d_w^j) \\ 0 \; otherwise \end{cases} \tag{4}$$

Thus, the Binary PSA (B-PSA) is constructed. First, the solution in the binary domain is initialized, and the following steps are performed in all the iterations: (1) perturbing the binary solution with Eqs (1) and (2), the equation of movement of PSA. (2) After perturbing each solution, it can leave the binary domain and use Eqs (3) and (4), and binarization will be applied. This procedure is reiterated until the end of the iterations. Then, perform a feasibility test and solution repair after the binarization and solution generation steps. Here, we verify that all the solutions have a minimum of one activated feature. A new binary random solution is produced if this condition is not met, and then the feasibility test is repeated. This procedure is reiterated until each feasible solution is attained. The pseudocode of B-PSA is demonstrated in Algorithm 3.

Algorithm 2: Feasibility test and repair solution

Input: The population $X = \{X_1, X_2, \ldots, X_i\}$
Output: The feasible population $X = \{X_1, X_2, \ldots, X_i\}$
Repeat
for $solution$ (i) do
If $solution_i$ has only 0 then
Produce a new random binary solution 5:
Else
Possible result
End if
End for
Until all the results are possible
Return the X possible population

Algorithm 3: Pseudocode of BPSA

Input: The population $X = \{X_1, X_2, \ldots, X_i\}$
Output: The updated population $X' = \{X_1', X_2', \ldots, X_i'\}$ and *Best*
Initialize X binary random population
Implement a possibility test based on Algorithm 2
Estimate the objective function of all the individuals in X population
Detect the fittest individual from the population $(Best)$
For iteration (t) do
For performance (i) do
For dimensional (j) do
Upgrade $pend_{i.j}^t$ by using Eq (2)
Upgrade the place of $X_{i,j}^t$ by utilizing Eq (1)
End for
End for

Binarization of population X Execute possibility test based on Algorithm 2 Estimate the objective function of all the individuals within X population Upgrade *Best* End for Return the X' upgraded population whereas $Best$ is an optimum solution

3.3. Weighted average ensemble

A weighted AE is an extension of the average ensemble model, which describes the weight of every member's impact to the last estimate [21]. When compared to the lowest performing technique, the highest performing method will get large weights. The formula to unite the prediction of the base-learners can be given as:

$$P(t) = w_i p_j(t) \tag{5}$$

In Eq (5), N represents the overall count of the models, p_i indicates the probability for i, and w_i represents each model's weight.

3.3.1. MERNN model

MERNN has a unique learning strategy and is derived from the backpropagation neural network (BPNN) model [22]. This technique has successfully classified a long distance of crucial data. The architecture of MERNN multiple different layers to accomplish classification. The layers presented in the MERNN model are a recurrent or context, input, output, and hidden layer (HL). Each neuron has an activation function, a biased input, and one output. The input layer fetches the information and allows the next HL to transfer data to the output layer. This HL is given at the last moment in the Elman neural network (ENN). Later, the output of the HL is kept in the recurrent layer. Figure 2 depicts the framework of the MERNN technique.

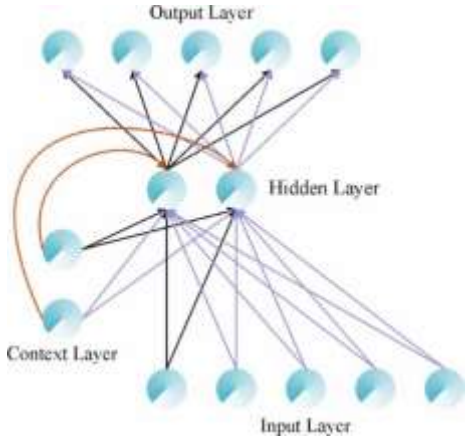

Figure 2. MERNN architecture.

Consider the hidden neuron counts as $j = 1,2..m$, the input counts as $i = 1,2 \ldots . n$, the network's weights as W_{ij}, W_{rj}, and W_{jo}, and the recurrent neuron count s as $r = 1,2..m$.

The output of the HL at t is formulated as

$$O_j(t) = \sum_{i=1}^{n}\sum_{j=1}^{m}\Big(W_{ij} \times i(t)\Big) + \sum_{r=1}^{m}\sum_{j=1}^{m}\Big(W_{rj} \times O_j(t-1)\Big) + b_j, \quad (6)$$

In Eq (6), b denotes the bias term.

$$Y_j(t) = g\Big(O_j(t)\Big), \quad (7)$$

In Eq (7), g represents the tangent hyperbolic function.

3.3.2. GRU model

The GRU is a NN that is adapted depending on the LSTM [23]. The GRU combines forget and input gates in the LSTM and exploits them into an update gate. Moreover, it establishes a reset gate. The appropriate computational formula of GRU is

$$r_t = \sigma(X_t W_{xr} + h_{t-1} W_{hr} + B_r) \quad (8)$$

$$z_t = \sigma(X_t W_{xz} + h_{t-1} W_{hz} + B_z) \quad (9)$$

$$\tilde{h}_t = tanh(X_t W_{xh} + r_t \odot h_{t-1} W_{hh} + B_h) \quad (10)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \quad (11)$$

where X_t signifies the input numbers, r_t stands for the outcomes of the update gate, W_{hr} represents the weights among the HL at the preceding moment h_{t-1} and the reset gate, B_r is the bias of the reset gate, W_{xr} defines the weights among X_t and the reset gate, z_t stands for the result of the update gate, W_{xh} stands for the weights among X_t and h_t, W_{hh} signifies the weights among h_{t-1} and h_t, W_{xz} signifies the weights among X_t and the update gate, W_{hz} defines the weights among h_{t-1} and the update gate, h_t indicates the existing HL, B_h refers to the bias of h_t, B_z illustrates the bias of the update gate, h_t defines the candidate layer attained by the compound function of X_t and h_{t-1}, $\sigma(\cdot)$ signifies the Sigmoid function, and $\odot$ stands for the point multiplication operation.

3.3.3. LSTM model

By comparison with conventional RNNs, the advance of LSTM models has included 3 control parts ("cells"), namely the output gate, input gate, and forget gate [24]. The functions of the gates will be explained as given below:

Forget gate: This gate resolves anything from prior data that can be disregarded. The existing stage's HL h_{t-1} and input x_t at the previous component are incorporated into a new vector. Increasing the weight parameter W_f of the gate, every component value of the resultant vector f_t is scaled from 0 to 1 over the unit-wise sigmoidal function $\mathfrak{a}$. A $0'$ module permits the related data in C_{t-1} to be removed, where a $1'$ represents consistent data that can be allowed to be accepted. The output f_t of the gate is generated according to Eq (12).

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big) \quad (12)$$

Input gate: It evaluates how many of the input x_t of the network will be kept in the cell state C_t. The achievement of the input gate's operation needs support among 2 parallel layers. The tangent state outputs candidate data C_t for collection, but the sigmoidal layer works as f_t as well as chooses which candidate data could be preferred by the decision vector i_t outputs. Next, the unit-wise increase of candidate data by the decision vector $C_t \times i_t$ can be carried out along with the last upgrade data, which will be comprised of the unit state to be calculated. The function of 2 layers can be represented as Eqs (13) and (14).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{13}$$

$$\overline{C}_t = \tan(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{14}$$

Consequently, the cell layer C_t of the present chain can be incorporated into the previously saved data $of\ C_{t-1}$, and upgrading data is preferred in C_t (Eq (15)).

$$C_t = C_{t-1} \times f_t + \overline{C}_t \times i_t \tag{15}$$

Output gate: It selects HL h_t from the present chain to outcome by multiplication of the decision vector 0_t via the candidate data elected at C_t, as denoted in Eqs (16) and (17).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{16}$$

$$h_t = \tan(C_t) \times o_t \tag{17}$$

3.4. Hyperparameter tuning using RSA

Eventually, the hyperparameter selection of the DL models takes place using RSA. Similar to other metaheuristic optimization algorithms, RSA exploits local and global search to effectively locate potential areas in the search range [25]. The original RSA is nature-inspired, mathematically modeling the hunting strategy of crocodiles. It can efficiently address complex challenges since it can be a gradient-free population-based technique. Based on the stochastic technique, a population of agents can be created in the initialization process. Then, the population is estimated and the optimum performance is considered near-optimum:

$$P = \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,n} \end{bmatrix} \tag{18}$$

In Eq (18), N denotes P size, n is the dimension problem, and χ is a promising solution. The set population is generated according to Eq (19):

$$x_{i,j} = rand \cdot \left(B_{lower} - B_{upper}\right) + B_{lower}, j = 1,2, \ldots, n \tag{19}$$

In Eq (19), B_{lower} and B_{upper} are the lower and upper boundaries of the search range, and $rand$ denotes a random number.

The algorithm can make progress with optimization once the population is established. The strategy used is largely based on the number of residual iterations. Two behaviors are distinctly

simulated for the exploration mechanism, namely the crocodile high walking and belly walking:

$$x_{i,j} = \begin{cases} B_j(t) \cdot -\left(\eta_{i,j}(t)\right) \cdot \beta - R_{i,j}(t).rand, & t \le \frac{T}{4} \\ B_j(t) \cdot x_{r_1,j} \cdot EX(t).rand, & t \le 2\frac{T}{4} \; and \; t > \frac{T}{4} \end{cases} \tag{20}$$

In Eq (20), $B_j(t)$ stands for the j^{th} component of the better candidate, $rand$ denotes a random number from [0,1], t and T are the existing and maximum iterations, and β denotes the sensitivity. R and ES are specialized values described as follows:

$$\eta_{i,j} = B_j(t) - PD_{i,j} \tag{21}$$

$$R_{i,j} = \frac{B_j(t) - x_{r_2,j}}{B_j(t) + \epsilon} \tag{22}$$

$$ES(t) = 2 \cdot r_3 \cdot \left(1 - \frac{1}{T}\right) \tag{23}$$

Here, the η parameter denotes the hunter operator. The role of R is to decrease the searching range, ES represents the evolutionary sense, r_2 and r_3 are random integers, and PD defines the percentage deviation among the existing and optimum solutions. Also, a smaller value can be added by ϵ to avoid a mathematical error.

Similarly, exploitation exploits 2 different hunting approaches: hunting coordination and cooperation.

$$x_{i,j} = \begin{cases} B_j(t) \cdot PD_{i,j}(t).rand, & t \le 3\frac{T}{4} \; and \; t > 2\frac{T}{4} \\ B_j(t) \cdot \eta_{i,j} \cdot \epsilon - R_{i,j}.rand & t \le T \; and \; t > 3\frac{T}{4} \end{cases} \tag{24}$$

The RSA method produces a fitness function (FF) to acquire a better solution of the classifier. This specifies a positive integer to label the best outcomes of the candidate outcome. In this study, the decreasing classifier error rate is defined as FF:

$$fitness(x_i) = ClassifierErrorRate(x_i) = \frac{No.of\ misclassified\ instances}{Total\ no.of\ instances} * 100 \tag{25}$$

4. Performance validation

This section observes the performance of the EAEDL-BDC algorithm on the N-BaIoT Database [26]. The database encompasses 17,001 instances with 3 class labels, as defined in Table 2.

Table 2. Details on database.

Classes	**No. of Instances**
Benign	5000
Mirai	7001
Gafgyt	5000
Total Instances	**17001**

Figure 3 illustrates the confusion matrices produced by the EAEDL-BDC system on 80:20 and 70:30 of TRPH/TSPH. The experimental outcome specifies the efficient recognition of the benign, Mirai, and Gafavt classes.

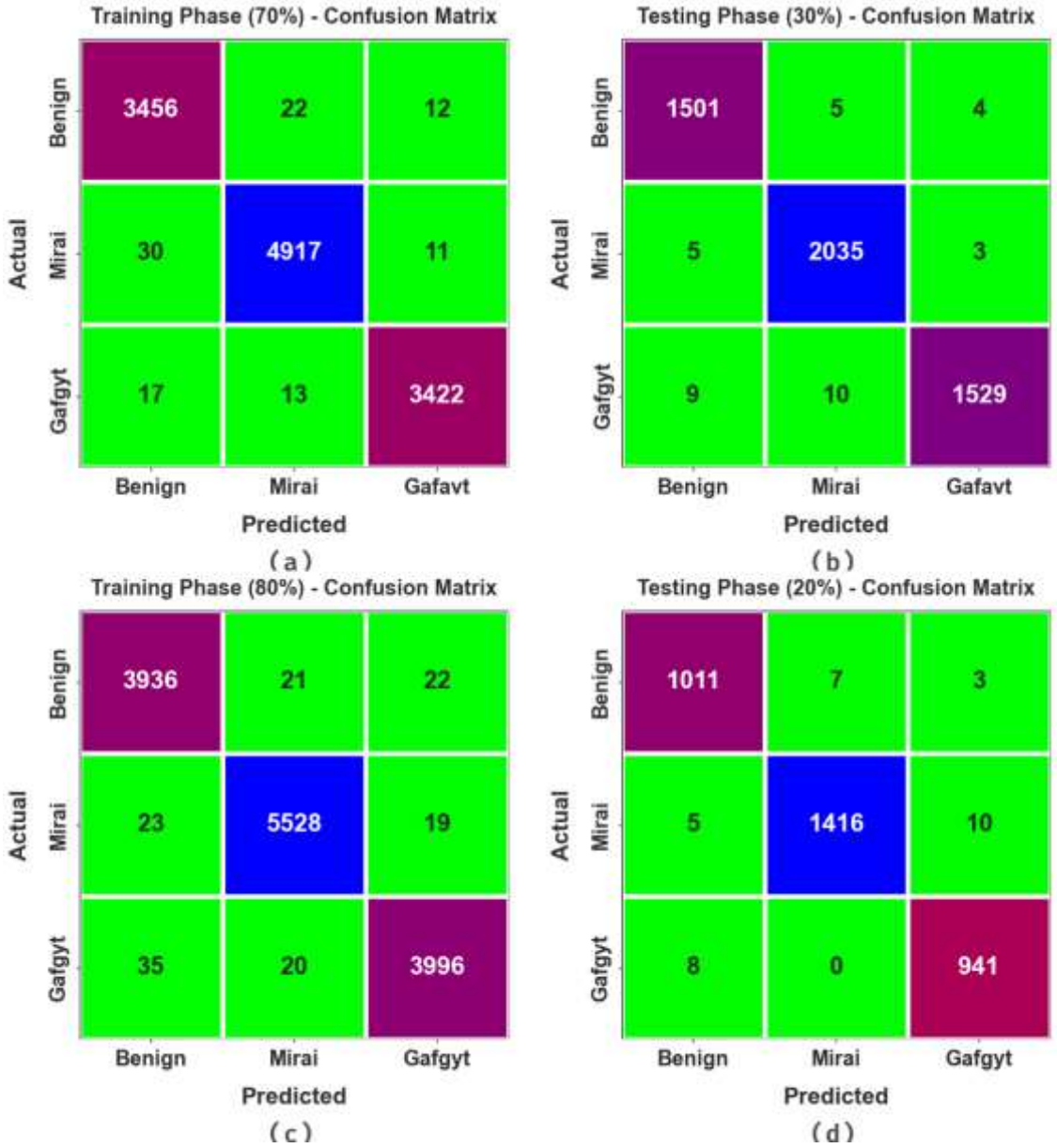

Figure 3. Confusion matrices of (a-b) 70:30 of TRPH/TSPH and (c-d) 80:20 of TRPH/TSPH.

An overall detection result of the EAEDL-BDC technique is 80% of TRPH and 20% of TSPH, as shown in Table 3. Figure 4 demonstrates an overall detection result of the EAEDL-BDC technique with 80% of TRPH. These obtained outcomes specify that the EAEDL-BDC system properly identifies benign, Mirai, and Gafavt classes. The EAEDL-BDC technique recognizes the benign class with $accu_y$ of 99.26%, $prec_n$ of 98.55%, $reca_l$ of 98.92%, F_{score} of 98.73%, and AUC_{score} of 99.16%. Additionally, the EAEDL-BDC system identifies the Mirai class with $accu_y$ of 99.39%, $prec_n$ of 99.26%, $reca_l$ of 99.25%, F_{score} of 99.25%, and AUC_{score} of 99.37%. The EAEDL-BDC algorithm

recognizes the Gafavt class with $accu_y$ of 99.29%, $prec_n$ of 98.98%, $reca_l$ of 98.64%, F_{score} of 98.81%, and AUC_{score} of 99.11%.

Table 3. Detection outcome of EAEDL-BDC system at 80:20 of TRPH/TSPH.

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	AUC_{Score}
TRPH (80%)					
Benign	99.26	98.55	98.92	98.73	99.16
Mirai	99.39	99.26	99.25	99.25	99.37
Gafavt	99.29	98.98	98.64	98.81	99.11
Average	**99.31**	**98.93**	**98.94**	**98.93**	**99.21**
TSPH (20%)					
Benign	99.32	98.73	99.02	98.88	99.24
Mirai	99.35	99.51	98.95	99.23	99.30
Gafavt	99.38	98.64	99.16	98.90	99.31
Average	**99.35**	**98.96**	**99.04**	**99.00**	**99.28**

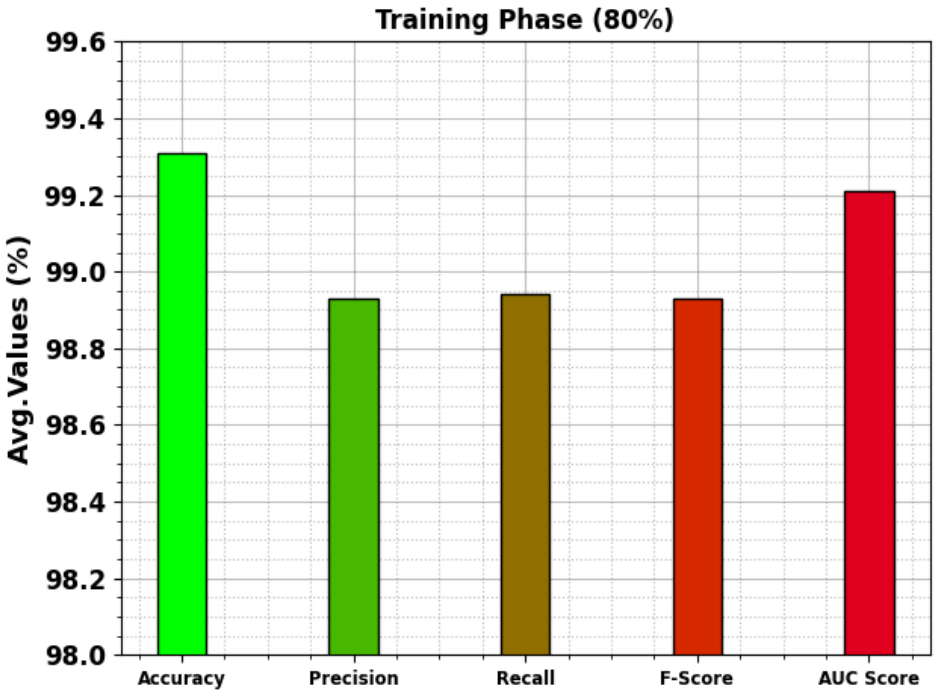

Figure 4. Average analysis of EAEDL-BDC system at 80% of TRPH.

Figure 5 displays an overall detection analysis of the EAEDL-BDC system with 20% of TSPH. These outcomes inferred that the EAEDL-BDC methodology properly recognizes benign, mirai, and Gafavt classes. The EAEDL-BDC method identifies the benign class with $accu_y$ of 99.32%, $prec_n$ of 98.73%, $reca_l$ of 99.02%, F_{score} of 98.88%, and AUC_{score} of 99.24%. In addition, the EAEDL-BDC algorithm identifies the Mirai class with $accu_y$ of 99.35%, $prec_n$ of 99.51%, $reca_l$ of 98.95%, F_{score} of 99.23%, and AUC_{score} of 99.30%. Last, the EAEDL-BDC algorithm detects the Gafavt class with $an\ accu_y$ of 99.38%, $prec_n$ of 98.64%, $reca_l$ of 99.16%, F_{score} of 98.90%, and AUC_{score} of 99.31%.

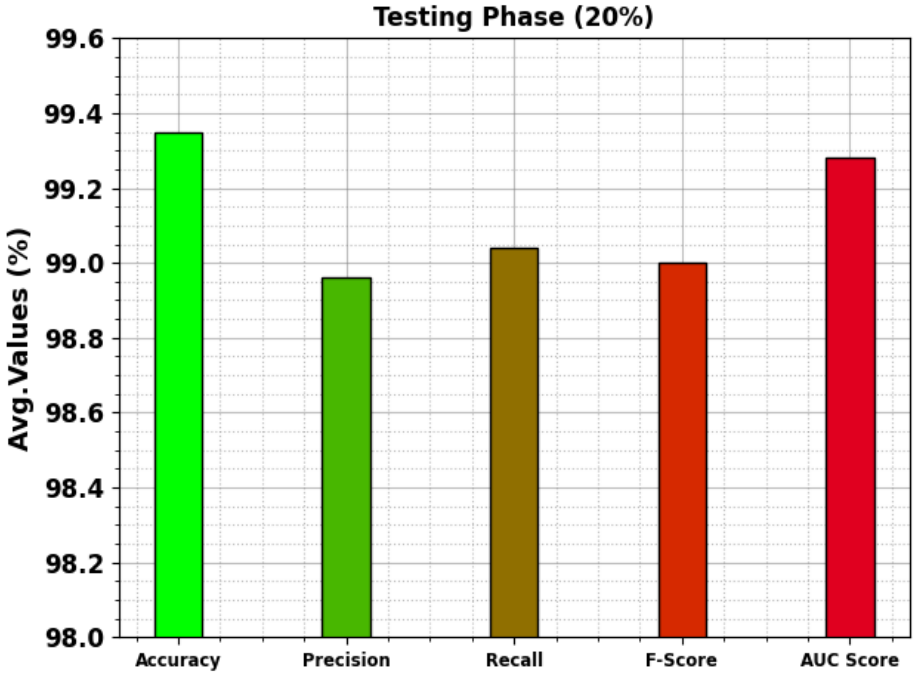

Figure 5. Average analysis of EAEDL-BDC method at 20% of TSPH.

In Table 4, a detailed detection investigation of the EAEDL-BDC algorithm can be provided with 70% of TRPH and 30% of TSPH.

Figure 6 exhibits an overall detection outcome of the EAEDL-BDC system with 70% of TRPH. These results specify that the EAEDL-BDC technique suitably recognizes benign, Mirai, and Gafavt classes. The EAEDL-BDC method recognizes the benign class with $accu_y$ of 99.32%, $prec_n$ of 98.66%, $reca_l$ of 99.03%, F_{score} of 98.84%, and AUC_{score} of 99.23%. Moreover, the EAEDL-BDC system finds the Mirai class with $accu_y$ of 99.36%, $prec_n$ of 99.29%, $reca_l$ of 99.17%, F_{score} of 99.23%, and AUC_{score} of 99.33%. Also, the EAEDL-BDC algorithm recognizes the Gafavt class with $accu_y$ of 99.55%, $prec_n$ of 99.33%, $reca_l$ of 99.13%, F_{score} of 99.23%, and AUC_{score} of 99.43% respectively.

Table 4. Detection analysis of the EAEDL-BDC model under 70:30 of TRPH/TSPH.

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	AUC_{Score}
70% of TRPH					
Benign	99.32	98.66	99.03	98.84	99.23
Mirai	99.36	99.29	99.17	99.23	99.33
Gafavt	99.55	99.33	99.13	99.23	99.43
Average	**99.41**	**99.09**	**99.11**	**99.10**	**99.33**
30% of TSPH					
Benign	99.55	99.08	99.40	99.24	99.51
Mirai	99.55	99.27	99.61	99.44	99.56
Gafavt	99.49	99.54	98.77	99.16	99.29
Average	**99.53**	**99.30**	**99.26**	**99.28**	**99.45**

Figure 6. Average of EAEDL-BDC technique at 70% of TRPH.

Figure 7 shows an overall detection outcome of the EAEDL-BDC method with 30% of TSPH. The achieved outcome represents that the EAEDL-BDC technique accurately recognizes benign, Mirai, and Gafavt classes. The EAEDL-BDC system recognizes the benign class with $an\ accu_y$ of 99.55%, $prec_n$ of 99.08%, $reca_l$ of 99.40%, F_{score} of 99.24%, and AUC_{score} of 99.51%. Next, the EAEDL-BDC algorithm finds the Mirai class with $accu_y$ of 99.55%, $prec_n$ of 99.27%, $reca_l$ of 99.61%, F_{score} of 99.44%, and AUC_{score} of 99.56%. Lastly, the EAEDL-BDC methodology recognizes the Gafavt class with $accu_y$ of 99.49%, $prec_n$ of 99.54%, $reca_l$ of 98.77%, F_{score} of 99.160%, and AUC_{score} of 99.29% respectively.

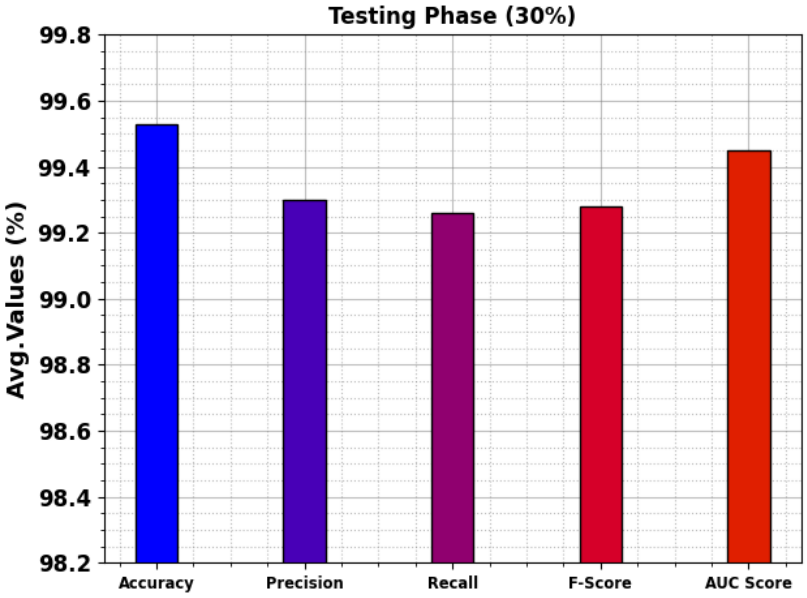

Figure 7. Average of EAEDL-BDC system at 30% of TSPH.

The $accu_y$ curves for training (TR) and validation (VL) illustrated in Figure 8 for the EAEDL-BDC system under 70:30 of TRPH/TSPH offer valued insights into its effectiveness in several epochs. Mainly, it can be a consistent upgrade in both TR and TS $accu_y$ with increased epochs, specifying the proficiency of the model in learning and recognizing patterns with both data of TR and TS. The increasing trend in TS $accu_y$ underscores the adaptability model to the TR dataset and the ability to produce exact predictions on unnoticed data, emphasizing the capabilities of robust generalization.

Figure 8. $Accu_y$ curve of EAEDL-BDC technique at 70:30 of TRPH/TSPH.

Figure 9 displays a wide-ranging overview of the TR and TS loss values for the EAEDL-BDC technique under 70:30 of TRPH/TSPH through several epochs. The TR loss consistently minimizes as the model refines weights for decreasing classification errors under both datasets. The loss curves show the alignment of the model with the TR data, underscoring its ability to capture patterns efficiently. Significant is the continuing refinement of parameters in the EAEDL-BDC algorithm, aimed at reducing discrepancies among forecasts and actual TR labels.

Figure 9. Loss curve of EAEDL-BDC algorithm at 70:30 of TRPH/TSPH.

In regard to the PR curve exhibited in Figure 10, the findings affirm that the EAEDL-BDC technique with 70:30 of TRPH/TSPH reliably accomplishes increased PR values in each class. These performances underline the methodologies' active capability to discriminate amongst distinct class labels, highlighting its ability in accurately detecting classes.

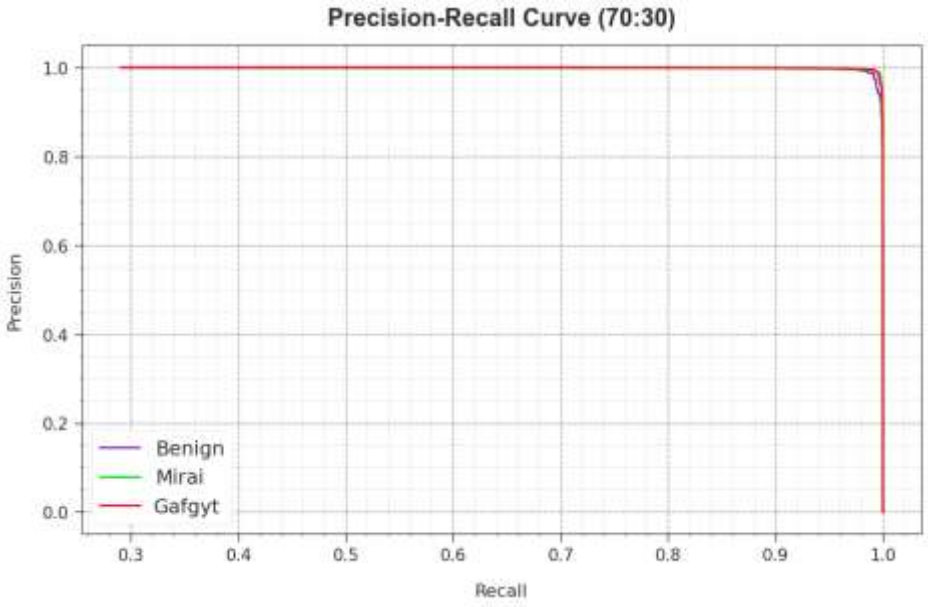

Figure 10. PR curve of EAEDL-BDC algorithm at 70:30 of TRPH/TSPH.

Likewise, in Figure 11, we illustrate ROC outcomes made by the EAEDL-BDC system at 70:30 of TRPH/TSPH, suggesting its abilities in unique amongst class labels. These curves provide appreciated perceptions of how the trade-off between FPR and TPR varied by diverse classification thresholds and epochs. These outcomes underscore the model's exact classification effectiveness on diverse class labels, underscoring its efficiency in overcoming several classification challenges.

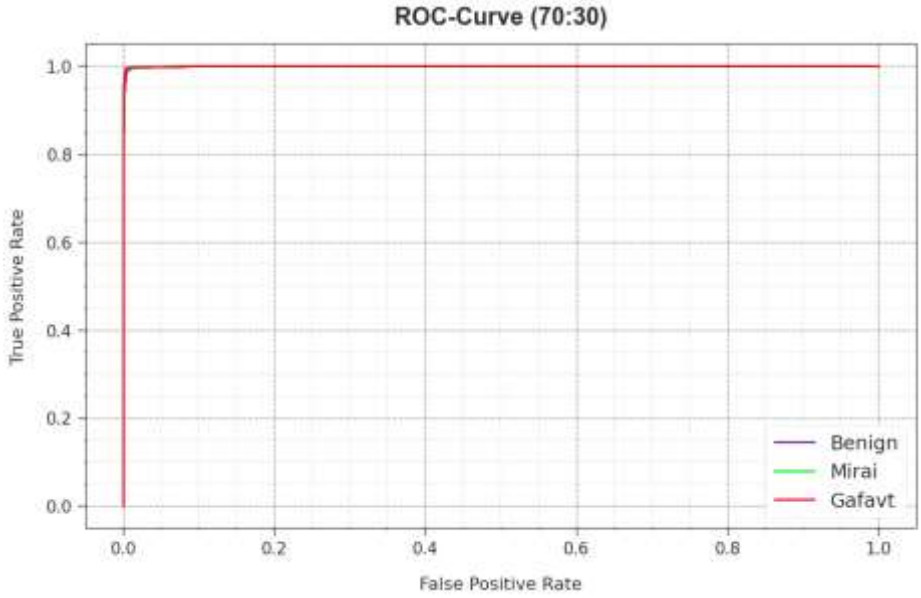

Figure 11. ROC curve of EAEDL-BDC technique at 70:30 of TRPH/TSPH.

In Table 5, a comparison analysis of the EAEDL-BDC method is provided in terms of distinct measures [27]. In Figure 12, a comparative $accu_y$ investigation of the EAEDL-BDC approach is provided. The outcome demonstrates that the EAEDL-BDC methodology has better results. Based on $accu_y$, the EAEDL-BDC technique exhibits an increased $accu_y$ of 99.53%, whereas the HMMLB-BND, BND-BMOML, DNN-LSTM, LSTM, CNN-RNN, LSTM-CNN, and DNN techniques obtain decreased $accu_y$ values of 99.44%, 99.05%, 98.85%, 96.90%, 96.20%, 98.61%, and 98.53%, respectively.

An extensive comparative $prec_n$, $reca_l$, and F_{score} analysis of the EAEDL-BDC system can be provided in Figure 13. These achieved outcomes indicate that the EAEDL-BDC technique acquires enhanced performance. According to $prec_n$, the EAEDL-BDC method exhibits boosted $prec_n$ of 99.3% while the HMMLB-BND, BND-BMOML, DNN-LSTM, LSTM, CNN-RNN, LSTM-CNN, and DNN algorithms get reduced $prec_n$ values of 99.14%, 98.68%, 98.11%, 95.71%, 93.74%, 96.75%, and 96.75%. Additionally, with $reca_l$, the EAEDL-BDC method exhibits raised $reca_l$ of 99.26%, but the HMMLB-BND, BND-BMOML, DNN-LSTM, LSTM, CNN-RNN, LSTM-CNN, and DNN methods get reduced $reca_l$ values of 99.13%, 98.67%, 98%, 94.36%, 97.36%, 97.43%, and 96.17%, correspondingly. Lastly, based on $the\ F_{score}$, the EAEDL-BDC system offers an improved F_{score} of 99.28%, while the HMMLB-BND, BND-BMOML, DNN-LSTM, LSTM, CNN-RNN, LSTM-CNN, and DNN techniques get diminished F_{score} values of 99.14%, 98.71%, 97.87%, 94.96%, 93.82%, 95.91%, and 94.52%, correspondingly.

Table 5. Comparison analysis of the EAEDL-BDC model with other algorithms.

Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
EAEDL-BDC	99.53	99.3	99.26	99.28
HMMLB-BND	99.44	99.14	99.13	99.14
BND-BMOML	99.05	98.68	98.67	98.71
DNN-LSTM	98.85	98.11	98.00	97.87
LSTM	96.90	95.71	94.36	94.96
CNN-RNN	96.20	93.74	97.36	93.82
LSTM-CNN	98.61	96.75	97.43	95.91
DNN Algorithm	98.53	96.75	96.17	94.52

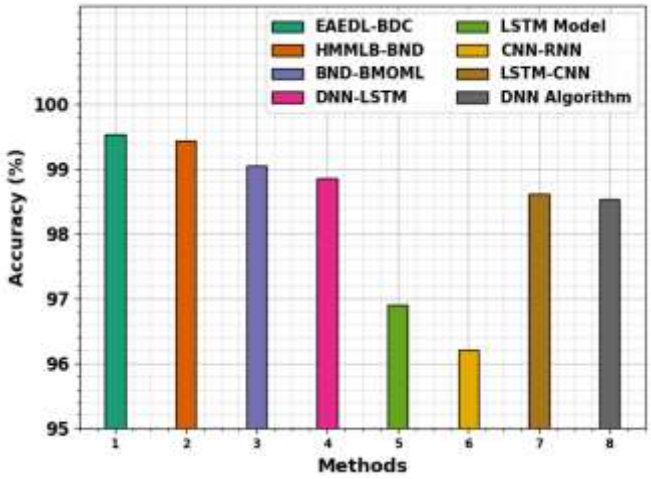

Figure 12. $Accu_y$ outcome of EAEDL-BDC technique with other approaches.

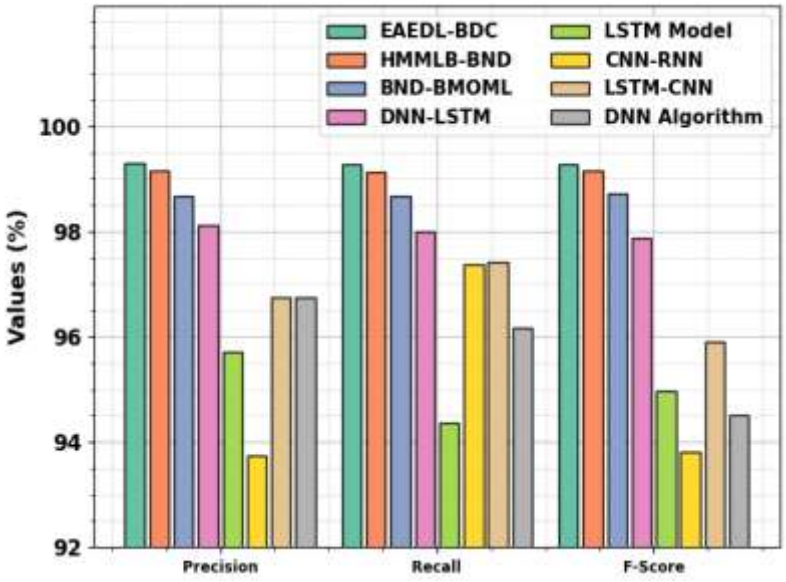

Figure 13. Comparative outcome of EAEDL-BDC technique with other approaches.

These experimental outcomes show that the EAEDL-BDC system gains excellent performance compared to other systems.

5. Conclusions

In this study, the EAEDL-BDC technique is presented. The goal of the study is to enhance cybersecurity in the cloud-assisted IoT environment via a botnet detection process. The EAEDL-BDC technique comprises data normalization using Z-score normalization, BPSA-based FS, ensemble learning, and RSA-based parameter tuning. For the FS process, the EAEDL-BDC technique uses BPSA. Moreover, a weighted average ensemble of three models, such as MERNN, GRU, and LSTM, can be employed. Furthermore, the hyperparameter choice of the DL approaches takes place using RSA. The simulation value of the EAEDL-BDC algorithm can be examined on the N-BaIoT database. The extensive comparison study demonstrated that the EAEDL-BDC technique researched a superior accuracy value of 99.53%, along with other approaches concerning distinct evaluation metrics. The EAEDL-BDC model may face threats in real-time scalability due to the computational complexity of ensemble DL methods. Future research may concentrate on optimizing the model for edge computing atmospheres and exploring dynamic adaptation mechanisms for evolving cyber threats in the cloud-based IoT context.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through large group Research Project under grant number (RGP2/112/44). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R114), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Research Supporting Project number (RSP2024R459), King Saud University, Riyadh, Saudi Arabia.

The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number "NBU-FPEJ-2024-2913-01". This study is partially funded by the Future University in Egypt (FUE).

Conflict of interest

The authors declare that they have no conflict of interest. The manuscript was written through the contributions of all authors. All authors have approved the final version of the manuscript.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

References

1. J. K. Samriya, C. Chakraborty, A. Sharma, M. Kumar, Adversarial ML-Based Secured Cloud Architecture for Consumer Internet of Things of Smart Healthcare, *IEEE Transactions on Consumer Electronics*, 2023. https://doi.org/10.1109/TCE.2023.3341696
2. Y. Djenouri, D. Djenouri, A. Belhadi, G. Srivastava, J. C. W. Lin, Emergent deep learning for anomaly detection in the Internet of everything, *IEEE Internet of Things Journal*, 2021.
3. M. A. Rahman, M. S. Hossain, A deep learning assisted software defined security architecture for 6G wireless networks: IIoT perspective, *IEEE Wirel. Commun.*, **29** (2022), 52–59. https://doi.org/10.1109/MWC.006.2100438
4. S. Bhingarkar, S. T. Revathi, C. S. Kolli, H. K. Mewada, An effective optimization enabled deep learning based Malicious behaviour detection in cloud computing, *Int. J. Intell. Robot.*, **7** (2023), 575–588. https://doi.org/10.1007/s41315-022-00239-x
5. N. Kansal, B. Bhushan, S. Sharma, Architecture, security vulnerabilities, and the proposed countermeasures in Agriculture-Internet-of-Things (AIoT) Systems, *Int. Things Anal. Agricul.*, **3** (2022), 329–353. https://doi.org/10.1007/978-981-16-6210-2_16
6. Q. Aslan, M. Ozkan-Okay, D. Gupta, Intelligent behavior-based malware detection system on cloud computing environment, *IEEE Access*, **9** (2021), 83252–83271. https://doi.org/10.1109/ACCESS.2021.3087316
7. B. B. Gupta, A. Gaurav, V. Arya, P. Kim, A Deep CNN-based Framework for Distributed Denial of Services (DDoS) Attack Detection in Internet of Things (IoT), *In Proceedings of the 2023 International Conference on Research in Adaptive and Convergent Systems*, 2023, 1–6. https://doi.org/10.1145/3599957.3606239
8. A. Lakhan, M. A. Mohammed, A. N. Rashid, S. Kadry, K. H. Abdulkareem, J. Nedoma, et al., Restricted Boltzmann machine assisted secure serverless edge system for internet of medical things, *IEEE J. Biomed. Health*, **27** (2022), 673–683. https://doi.org/10.1109/JBHI.2022.3178660
9. R. Zhou, X. Zhang, X. Wang, G. Yang, H. N. Dai, M. Liu, Device-Oriented Keyword-Searchable Encryption Scheme for Cloud-Assisted Industrial IoT, *IEEE Internet Things*, **9** (2021), 17098–17109. https://doi.org/10.1109/JIOT.2021.3124807

10. N. Kumar, V. Goel, R. Ranjan, M. Altuwairiqi, H. Alyami, S. A. Asakipaam, A Blockchain-Oriented Framework for Cloud-Assisted System to Countermeasure Phishing for Establishing Secure Smart City, *Secur. Commun. Netw.*, 2023. https://doi.org/10.1155/2023/8168075
11. S. M. Alshahrani, F. S. Alrayes, H. Alqahtani, J. S. Alzahrani, M. Maray, S. Alazwari, et al., IoT-Cloud assisted botnet detection using Rat Swarm Optimizer with deep learning, *Comput., Mater. Con.*, **74** (2023). https://doi.org/10.32604/cmc.2023.032972
12. K. Haseeb, I. U. Din, A. Almogren, I. Ahmed, M. Guizani, Intelligent and secure edge-enabled computing model for sustainable cities using green internet of things, *Sustain. Cities Soc.*, **68** (2021), 102779. https://doi.org/10.1016/j.scs.2021.102779
13. R. S. Prabhu, A. Prema, E. Perumal, A novel cloud security enhancement scheme to defend against DDoS attacks by using deep learning strategy, *In 2022 6th international conference on electronics, communication and aerospace technology*, IEEE, 2022, 698–704. https://doi.org/10.1109/ICECA55336.2022.10009177
14. F. Alrowais, M. M. Eltahir, S. S. Aljameel, R. Marzouk, G. P. Mohammed, A. S. Salama, Modeling of botnet detection using chaotic binary Pelican Optimization Algorithm with deep learning on Internet of Things Environment, *IEEE Access*, **11** (2023), 130618–130626. https://doi.org/10.1109/ACCESS.2023.3332690
15. M. I. Ahmed, G. Kannan, S. R. Polamuri, LSITA: An Integrated Framework for Leveraging Security of Internet of Things Application with Remote Patient Monitoring System, 2022. https://doi.org/10.21203/rs.3.rs-1948226/v1
16. M. Aljebreen, F. S. Alrayes, S. S. Aljameel, M. K. Saeed, Political Optimization Algorithm with a hybrid deep learning assisted Malicious URL detection model, *Sustainability*, **15** (2023), 16811. https://doi.org/10.3390/su152416811
17. T. Wang, Q. Yang, X. Shen, T. R. Gadekallu, W. Wang, K. Dev, A privacy-enhanced retrieval technology for the cloud-assisted internet of things, *IEEE T. Ind. Inform.*, **18** (2021), 4981–4989. https://doi.org/10.1109/TII.2021.3103547
18. V. D. Padma, K. Venkata, An adaptive lightweight hybrid encryption scheme for securing the healthcare data in cloud-assisted Internet of Things, *Wireless Pers. Commun.*, **130** (2023), 2959–2980. https://doi.org/10.1007/s11277-023-10411-6
19. L. Friedman, O. V. Komogortsev, Assessment of the effectiveness of seven biometric feature normalization techniques, *IEEE T. Inf. Foren. Sec.*, **14** (2019), 2528–2536. https://doi.org/10.1109/TIFS.2019.2904844
20. B. Crawford, F. Cisternas-Caneo, K. Sepúlveda, R. Soto, A. Paz, A. Peña, et al., B-PSA: A binary Pendulum Search Algorithm for the feature selection problem, *Computers*, **12** (2023), 249. https://doi.org/10.3390/computers12120249
21. H. Gunasekaran, K. Ramalakshmi, D. K. Swaminathan, M. Mazzara, GIT-Net: An ensemble deep learning-based GI tract classification of endoscopic images, *Bioengineering*, **10** (2023), 809. https://doi.org/10.3390/bioengineering10070809
22. M. Vellaisamy, L. I. Freitas, Detection of human stress using optimized feature selection and classification in ECG signals, *Math. Probl. Eng.*, 2023. https://doi.org/10.1155/2023/3356347
23. R. Keyimu, W. Tuerxun, Y. Feng, B. Tu, Hospital outpatient volume prediction model based on gated recurrent unit optimized by the modified cheetah optimizer, *IEEE Access*, 2023. https://doi.org/10.1109/ACCESS.2023.3339613

24. J. Sun, R. Cao, M. Zhou, W. Hussain, B. Wang, J. Xue, et al., A hybrid deep neural network for classification of schizophrenia using EEG Data, *Sci. Rep.*, **11** (2021), 4706. https://doi.org/10.1038/s41598-021-83350-6
25. M. Salb, L. Jovanovic, N. Bacanin, M. Antonijevic, M. Zivkovic, N. Budimirovic, et al., Enhancing Internet of Things network security using hybrid CNN and XGBoost model tuned via modified Reptile Search Algorithm, *Appl. Sci.*, **13** (2023), 12687. https://doi.org/10.3390/app132312687
26. Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, et al., N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders, *IEEE Pervas. Comput.*, **17** (2018), 12–22. https://doi.org/10.1109/MPRV.2018.03367731
27. L. Almuqren, H. Alqahtani, S. S. Aljameel, A. S. Salama, I. Yaseen, A. A. Alneil, Hybrid Metaheuristics with Machine Learning based Botnet Detection in Cloud Assisted Internet of Things Environment, *IEEE Access*, 2023. https://doi.org/10.1109/ACCESS.2023.3322369

AIMS Press