Nature-Inspired Learning Framework for Cyberattack Classification in IoT Networks

Ishwarya K.[1,*], Saraswathı S.[2]

[1]Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, Kattankulathur, 603203, Tamilnadu, India

[2]Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India

Emails: ishwaryk3@srmist.edu.in; saraswathi.s.cse@sathyabama.ac.in

Abstract

Due to the massive data and communication progress, the usage of Internet of Things (IoT) devices has developed significantly. The extensive use of IoT systems heightens the complex interactions among devices and increases the data traffic, generating numerous possibilities for cyber challengers. Therefore, identifying and alleviating cyber-attacks focusing on IoT systems has appeared as an essential obligation in the context of cybersecurity. Academics and enterprises are contemplating means of machine learning (ML) and deep learning (DL) for cyberattack prevention because ML and DL exhibit great potential in numerous domains. Various DL teachings are executed to extract several patterns from multiple annotated datasets. DL is a beneficial tool for identifying cyberattacks. Timely network data detection and segregation become more fundamental than alleviating cyberattacks. Therefore, this paper proposes a novel Brown Bear Optimization method with an Ensemble of Machine Learning-based Cyber Attack Detections (BBOA-EMLCADs) method for secure IoT environment. The main aim of the BBOA-EMLCAD method relies on the automatic classification of the cyber threats in the IoT environment. Initially, the brown bear optimization (BBO) method is utilized for feature selection (FS). Moreover, an ensemble of two ML approaches namely XGBoost and least square support vector machine (LSSVM) are employed for the automatic identification of the cyber-attacks. Lastly, the salp swarm algorithms (SSAs) is implemented for the optimal hyperparameter tuning of the two ML techniques. The simulation validation of the BBOA-EMLCAD approach is performed under the WSN-DS dataset. The comparison assessment of the BBOA-EMLCAD approach portrayed a superior accuracy value of 99.62% over existing models.

Keywords: Cyber Attack Detection; Brown Bear Optimization; Machine Learning; Hyperparameter Tuning; Internet of Things

1. Introduction

A smart city is a metropolitan region that employs the IoT and remote sensors allowing techniques to gather information from various places and utilize it to improve the value of people's lives [1]. Real-time monitoring and tracking applications are included [2]. The architecture is subjective in its usage, the surroundings that are authorized, hardware, cost, and objective. Currently, cyber-attacks are growing in size and scope nowadays, resulting in huge damages and dangerous effects [3]. With information processing and Internet accessibility costs falling, agencies have become progressively at risk of potential cyber-threat network cyber-attacks [4]. Hence, there is a requirement to offer safe and secure transactions through firewall usage, cyber-attack detection (CAD), authentication, encryption, and diverse software and hardware solutions [5]. CAD's goal is to identify cyber-

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

attacks when these are performed on a network and computer process [6]. A solitary device that security predictors severely lie on is the intrusion detection system (IDS). These tools can recognize network misuse and network intrusions to match patterns of well-known attacks versus preceding network activities. Nowadays, a novel revolutionary approach called AI and ML is developing for the future of completely automatic IoT implementations [7]. ML a subset of AI, wherein, the computer techniques are learned by us to improve from previous understandings. To develop security, it is important to accept a multi-faceted and comprehensive tactic that influences developed ML techniques [8]. The ML approach used to detect the mistakes in the system represents the artificial neural network (ANN) methodology using IoT. DL is the innovation of AI work in the image processing fields, computer vision, and pattern recognition [9]. The DL techniques are used in the IoTs for identifying the mischievous nodes and is utilized for improving the detection efficiency [10].

This paper proposes a novel Brown Bear Optimization method with Ensemble of Machine Learning-based Cyber Attack Detections (BBOA-EMLCADs) methods for secure IoT environment. The main aim of the BBOA-EMLCAD method relies on the automatic classification of the cyber threats in the IoT environment. Initially, the brown bear optimization (BBO) method is utilized for feature selection (FS). Moreover, an ensemble of two ML approaches namely XGBoost and least square support vector machine (LSSVM) are employed for the automatic identification of the cyber-attacks. Lastly, the salp swarm algorithms (SSAs) is implemented for the optimal hyperparameter tuning of the two ML techniques. The simulation validation of the BBOA-EMLCAD approach is performed under the WSN-DS dataset. The key contributions are:

- The BBOA-EMLCAD model employs BBO technique for robust and efficient FS, reducing redundant and irrelevant attributes while improving the in formativeness of input features, thereby improving model interpretability, accelerating training, and enhancing the detection accuracy.
- The BBOA-EMLCAD technique integrates an ensemble-learning framework incorporating XGBoost and LSSVM method to improve robustness and detection accuracy across varied cyber-attack scenarios, enabling better generalization, reducing false positives, and ensuring reliable classification in dynamic threat environments.
- The BBOA-EMLCAD methodology implements SSA model for hyperparameter tuning to improve model accuracy and stability, mitigate manual effort through intelligent parameter search, lowers computational overhead, and ensures consistent performance in identifying complex cyber-attack behaviors.
- The integration of BBO-based FS with an XGBoost–LSSVM ensemble, further refined through SSA-based hyperparameter tuning, presents a streamlined and highly adaptive framework for cyber-attack detection. This novel combination effectively tackles key challenges such as feature redundancy, classification precision, and automated tuning, presenting a distinct and underexplored methodology in the domain of intelligent threat detection and response.

2. Related Works

The authors [11] recommended technique incorporates goal-based artificial intelligence (AI) agents (GAIAs) with an autoencoder (AE) structure, producing an AE-based A (AE-A). This combined model aims to boost the efficacy of detecting the attacks of botnets, with a particular focus on the developing security threat associated with cloud computing (CC). This idea is about generating an accurately measured, goal driven AI agent personalized specifically for healthcare applications. The agent exactly examines network data and efficiently incorporates AE improved anomaly recognition methods to expose difficult patterns suggestive of botnet activity. In [12], cyberattack recognition systems are developed based on an intelligent hybrid method that utilizes DL and ML techniques. The presented method enhances the cyberattack recognition speeds. Additionally, a feature reduction method is presented by utilizing ML techniques (SVD and PCA) to choose the most relevant features to the adoptive attack kinds. Almajed et al. [13] presented a new cyber-attack detection method by combining ML and AI techniques. Now, first, the dataset is gathered from the CPS database and pre-process the data by utilizing standardization for redundant data and error removal. The features are mined by utilizing LDA. In [14], an emphatic farmland fertility incorporated deep perceptron networks (EFDPNs) is presented to improve the WSN IoT security. This advantage proposes the farmland fertility FSs (F3Ss) method for classification. In addition, the DPNs classifier is utilized for precise intrusion classifications, attaining outstanding performances of metrics. In the identification stage, the tunicate swarm optimizer (TSO) method is used with improving forecast precision. Ismail et al. [15] proposed a light weight multilayer ML detection system to alleviate cyberattacks that target WSN. The multilayer recognition model contains 2 ML methods used at the base station (BS) and monitor nodes. The

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

naïve bayes (NB) method is utilized as the first layer of recognition for binary classifications, and a LightGBM method as the subsequent layer of detections for multi-class classifications. The presented method could identify 4 network layers inside DoS attack noticed in WSN DS datasets. Surbhi, Chauhan, and Dahiya [16] improved cyber threat detection by optimizing Extreme Gradient Boosting (XGBoost) hyperparameters using the dragonfly algorithm (DA) technique. The approach improves detection accuracy and efficiency, emphasizing superior performance over conventional optimization methods.

Vaiyapuri et al. [17] proposed a model to accurately detect cyberattacks by utilizing improved reptile search optimization (IRSO) technique for FS integrated with an ensemble of DL models with tuning via a modified gray wolf optimizer (MGWO) methodology to improve detection performance. Alamro et al. [18] improved cyber threat detection and classification by utilizing white shark optimizer (WSO)-based FS and stacked autoencoder (SAE) classification. The SMOTE is also used, enhancing detection accuracy and robustness. Alkhafaji, Viana, and Al-Sherbaz [19] presented a technique by integrating genetic algorithms (GA) for optimized FS with DL models for accurate attack classification. Chen et al. [20] improved IoT network attack detection by integrating CNN with an enhanced pelican optimization algorithm (EPOA) model for optimized hyperparameter tuning and FS. The presented hybrid model illustrates higher exactness and efficiency across multiple benchmark datasets, improving cyberattack identification in resource-constrained IoT systems. Alamro et al. [21] introduced a method utilizing an advanced ensemble transfer learning (AETL) model integrating gated recurrent unit (GRU), deep CNN (DCNN), and stacked sparse autoencoder (SSAE) classifiers. The model leverages an improved coati optimization algorithm (ICOA) model for FS. Ultimately, the bayesian optimization algorithm (BOA) technique is employed for tuning. Wani, Aziz, and Raut [22] proposed a model to improve cyberattack detection in IoT and wireless sensor networks (WSN) by employing a stacking ensemble model integrating gradient boosting optimized with Optuna, AdaBoost tuned with RF, and XGBoost. Malathi and Begum [23] improved cyberattack identification in IoT by integrating gorilla troop's optimization (GTO) method for FS with an ensemble DL model. The approach utilizes Z-Score normalization for preprocessing and achieves improved detection. Gupta et al. [24] suggested an ensemble DL that incorporates FS and hyperparameter tuning using the ant lion optimization (ALO) approach. This model achieves efficient and lightweight detection, outperforming standard ML and DL approaches. Alrefaei and Ilyas [25] utilized a deep ensemble model integrating multilayer perceptron (MLP), CNN, and long short-term memory (LSTM) for multi-class classification.

Despite the improvements in ML, DL, and optimization techniques for cyberattack detection in IoT, WSN, and CPS environments, various limitations still exist. Several techniques encounter threats in handling imbalanced datasets effectively, often relying on oversampling techniques that may introduce bias. Few studies face difficulty due to computational complexity and resource constraints, particularly in real-time detection scenarios within resource-limited devices. Additionally, while ensemble and hybrid models improve accuracy, they tend to increase training time and model interpretability becomes difficult. Most approaches concentrate on specific datasets, restricting generalizability across diverse network environments. There is a research gap in developing lightweight, scalable models that balance detection accuracy, efficiency, and adaptability across heterogeneous IoT and CPS networks while addressing data imbalance and computational overhead.

3. Proposed Method

In this study, the BBOA-EMLCAD method for secure IoT environment is proposed. The key aim of BBOA-EMLCAD method relies on the automated classification of the cyberthreats in the IoT. Fig. 1 depicts the architecture of the BBOA-EMLCAD approach.

A. FS Process

Initially, the BBOA-EMLCAD approach performs FS by utilizing the BBO method [26]. This approach is chosen for its superior capability in balancing exploration and exploitation, enabling it to efficiently identify optimal feature subsets with minimal redundancy. This model replicates the adaptive foraging behavior of brown bears, allowing it to navigate complex search spaces effectively unlike conventional models. The model also indicates robust convergence characteristic and robustness across high-dimensional datasets, which is crucial in cybersecurity where data can be vast and noisy. BBO also mitigates the dimensionality of the feature set without compromising classification accuracy. Compared to standard models like GAs or particle swarm optimization (PSO), BBO presents an enhanced stability and faster convergence. Its adaptive search strategy ensures the selected features are both relevant and non-redundant, improving the overall performance. The capability to tackle the

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

approach of FS is based on numerous phases, containing initialization, upgrading location utilizing the BBO method, and altering locations at random.

Initial values population

A primary N population of possible locations was formed at this phase at random. Every position is signified by the D dimension vector, which is equivalent to the features number in the main benchmark, and displays a possible solution within its upper and lower limits. The resultant arbitrary location is restrained among the range of $[-1, 1]$ at every position vector.

Update the locations

The BBO method is an effective metaheuristic proposed for upgrading the locations. The pedal smell differentiation (PSD) and sniffing pedal differentiation (SPD) are the dual behaviours performed by bears. The exploration and exploitation abilities of the optimizer technique prescribe its efficiency. The exploitation stage depends upon the PSD behaviour, while the behaviour of SPD manages the exploration stage. These dual behaviors and the computations are described below in brief:

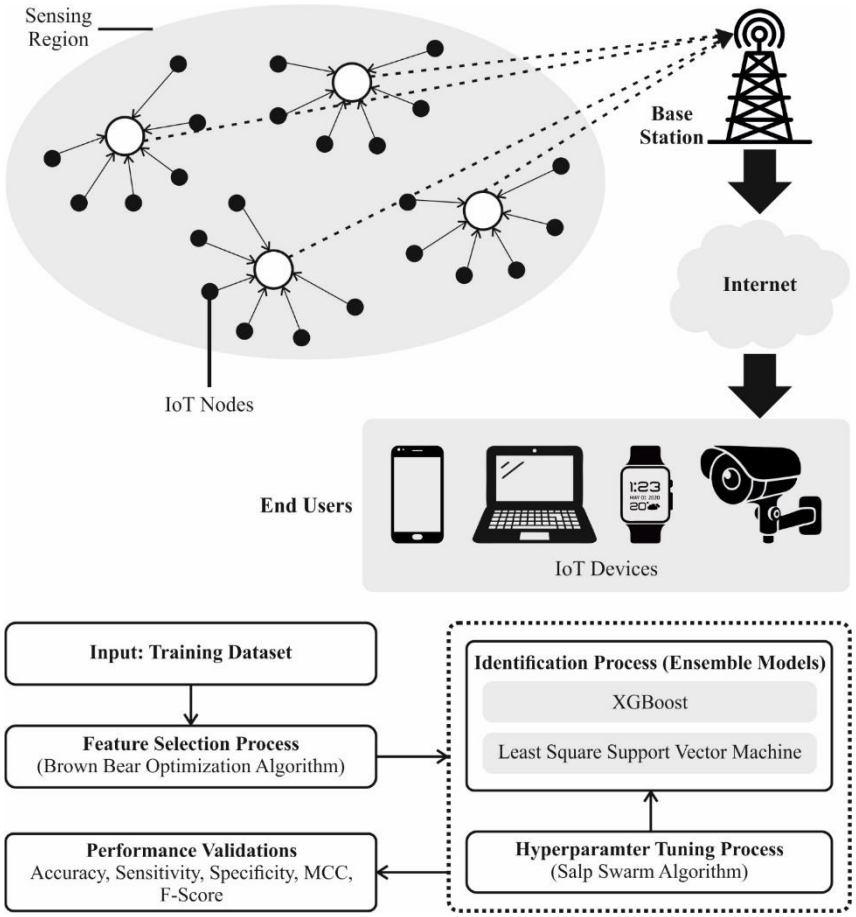

Figure 1. Overall flow of BBOA-EMLCAD technique

PSD behaviour (exploitation stage): Cautious stepping, foot twisting on land depressions and unique walking patterns are the three features. By utilizing these characteristics, numerical methods are presented to illuminate this behaviour. The mathematical formulation for this behaviour is given below:

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

$$X_i^{g+1} = \begin{cases} X_i^g - \left(P^g \cdot rand_w \cdot X_i^g\right), & if\ P^g > 0\ and\ P^g \le \frac{1}{3}, \\ X_i^g + Q^g \cdot \left(X_{Best}^g - S^g \cdot X_{Worst}^g\right), & if\ P^g > \frac{1}{3}\ and\ P^g \le \frac{2}{3}, \\ X_i^g + \omega^g \cdot \left(X_{Best}^g - \left|X_i^g\right|\right) - \omega^g \\ \cdot \left(X_{Worst}^g - \left|X_i^g\right|\right), & otherwise, \end{cases} \quad (1)$$

$$P^g = \frac{g}{G_{\max}}. \quad (2)$$

$$Q^g = rand_q \cdot P^g, \quad (3)$$

$$S^g = ronnd(1 + rand_s). \quad (4)$$

$$\omega^g = 2 \cdot P^g \cdot \pi \cdot rand_t. \quad (5)$$

Whereas X_i^{g+1} and X_i^g denotes the subsequent advanced pedal differentiation of jth brown bears cluster at $(g+1)th$ and gth iteration, respectively. $rand_w$ indicates a randomly produced value among [0,1]. P^g designates the occurrence factor for $g \uparrow h$ iteration, G_{mox} denotes a maximum iteration count. X_{Worst}^g and X_{Best}^g represents the present worst and best pedal differences across complete brown bears' clusters at the existing $g \uparrow h$ iteration, correspondingly. Q^g denotes a step factor at gth iteration. P^g is a factor of occurrence. $rand_q$ and $rand_s$ portray randomly produced numbers spread similarly in [0, 1]. At every iteration g, S^g specifies the length of the step, which is allotted as both 1 or 2 values; ω^g is a twist's jth angular speed at the $g \uparrow h$ iteration. $rand_t$ represents a randomly produced value dispersed similarly within [0,1].

SPD behavior (exploration stage): Generally, the brown bears can organize their gesture and interact in a group by sensing the pedal scent differentiation. If brown bears begin to travel, they smell arbitrarily selected pedal differentiation in the nearby area. Its mathematical expression is given below:

$$X_i^{g+1} = \begin{cases} X_i^g + rand_n \cdot \left(X_i^g - X_k^g\right), if\ f\left(X_i^g\right) < f\left(X_k^g\right), \\ X_i^g + rand_n \cdot \left(X_k^g - X_i^g\right), otherwise. \end{cases} \quad (6)$$

Here, $rand_n$ signifies a randomly produced value in the range of [0,1]. X_k^g is a pedal differentiation of the kth brown bears' cluster at the gth iteration, as $j \neq k$. $f(X_i^g)$ and $f(X_k^g)$ denotes the fitness function (FF) values for the jth and kth groups at the $g \uparrow h$ iteration.

Altering the locations randomly

This phase is vital for fitting the impossible search space, which cracks the search scope and moves out of limits in position update. This method enhances the exploitation procedure of the BBO method. Its formulation is computed below:

$$X_{i,d}^{adjust} = \begin{cases} X_{i,d}, & if\ x_d^{LB} \le x_{i,d} \le x_d^{UB} \\ rand\ (X_d^{LB}, X_d^{UB}), & if\ X_d^{LB} > X_{i,d}\ or\ X_{i,d} > X_d^{UB}. \end{cases} \quad (7)$$

Here, $X_{i,d}^{adjust}$ denotes the decision variable, $X_{i,d}$ refers to the value that surpasses the variable limits, X_d^{UB} and X_d^{LB} signifies upper and lower limits for search space d, correspondingly, and $rand\ (X_d^{LB}, X_d^{UB})$ is a randomly produced number among X_d^{LB} and X_d^{UB} with even distribution, correspondingly.

In the BBO method, objectives are integrated into a single FF using weighted importance. The FF that integrates both FS objectives is utilized, as shown in (8).

$$Fitness(X) = \alpha \cdot E(X) + \beta * \left(1 - \frac{|R|}{|N|}\right) \quad (8)$$

$Fitness(X)$ represent the fitness of feature subset X, where $E(X)$ is the classification error, $|R|$ is the number of chosen features, $|N|$ is the total original features, with weights $\alpha \in [0,1]$ for error and $\beta = (1-\alpha)$ for reduction.

B. Ensemble ML Process

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

Next, an ensemble of two ML approaches namely XGBoost and LSSVM are employed for the automatic identification of the cyber-attacks. This integration is chosen for its complementary strengths in classification tasks. Due to its gradient boosting framework, the model outperforms in handling large-scale data with high accuracy and efficiency, which effectively mitigates bias and variance. The model also presents robust performance in nonlinear classification with faster training times compared to conventional SVMs, making it appropriate for real-time cyber-attack detection. Altogether, the ensemble highlights efficiency in capturing complex feature interactions and robust generalization capabilities, resulting in an enhanced detection accuracy and mitigated overfitting. This hybrid approach outperforms individual models and other conventional classifiers like decision trees (DTs) or k-nearest neighbors (kNN) by providing higher robustness and adaptability to diverse attack patterns in cybersecurity datasets. Its ability to utilize complementary strengths of multiple algorithms improves accuracy while mitigating false positives. Moreover, the flexibility of the technique allows it to generalize well across diverse network environments and growing threats.

1) XGBoost Model

XGBoost is a popular approach in ML for large datasets to reach classification and regression performance [27]. In the subsequent structures, a DT is generated and adds weight to autonomous variable that is inputted to the DTs for providing the predictive outcomes. Incorrectly predicted weighted variable has increments in the DTs and are fed into the second DTs. These unique classifiers are defined as an ensemble to generate an effective model. The normalization of the objective function (OF) relies on real factors that regulate the convergence and limit the overfitting. XGBoost is enormously beneficial over other models in the FS. This approach can quickly pick the essential attributes at an early selection stage. The unknown features are classified according to the determined and predicted parameters. The data is given in the dimension form. These CART sets are generated in the DTs form. The sample mapping is fed into the leaf nodes of the classified DTs. Then, note the score and node count that should be organized into an optimum model for identifying its factors. Then, this model is applied to the modeling of XGBoost. The OF has the complexities and errors of the model that is formulated by:

$$Obj(t) = \sum_{a=1}^{n} B\,(x_i, x_i') + \sum_{a=1}^{t} \Omega\,(y_i) \tag{9}$$

$$\Omega(y_i) = \alpha T + \frac{1}{2}\beta \sum_{a=1}^{t} W_j^2 \tag{10}$$

The deviance of square loss function with the predicted and real values is represented as (x_i, x_i'). The regularization term is denoted as $\Omega(y)$. α and β are the complications in the coefficient of the splitting tree for the tree form and decrease the over-fitting problems. The predicted value is evaluated when the iteration is completed.

$$C_i'(t) = C_i'(t-1) + d_t(b_i) \tag{11}$$

The OF is formulated by:

$$Obj(t) = \sum_{i=1}^{n} B\,(x_{i'} x_i') + f_i(b_i) + \Omega(y_i) \tag{12}$$

The loss function is based on Taylor's formulation, with a higher accuracy and convergence speed than real DTs. The last OF is formulated by

$$Obj(t) = \sum_{i=1}^{n} [\,E_i - [E_i'(t-1) + ft(x_i)]]2 + h \tag{13}$$

Eq. (13) assesses the nodes and reduces the loss function.

2) LSSVM Model

Unlike conventional SVM, LSSVM attains dimensionality decrease by changing the new optimizer issue with difference constraints into a single with equivalence constraints [28]. This transformation permits LSSVM to use non-linear kernel function, efficiently projecting the data of input into a high-dimension feature space. In this research work, LSSVM is leveraged to build an error compensation method, which reduces the dissimilarity between actual and predicted energy consumption. Let signify the training data as $= \{(x_1, y_1), (x_2, y_2), (x_n, y_n)\}$

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

where x signifies the ith input vector and y_i denotes the equivalent output (actual energy consumption). The function of LSSVM for this method, mapped to the higher-dimension space, is stated as:

$$y(x) = \omega^T \varphi(x) + b \tag{14}$$

Whereas, $\varphi(x)$ denotes the non-linear map function that predicts the input vector (x) into a high- dimension space, b refers to the term of bias that influences the complete forecast, and ω denotes the weight vector that defines the power of every feature in the higher-dimension space. This makeover over the non-linear map function permits the LSSVM to capture difficult relations among the input and the output variables, finally foremost to a function optimizer issue that is stated below:

$$\min \mathrm{J}(\omega, e) = \frac{1}{2}\|w\|^2 + \frac{1}{2}\gamma \Sigma_{k=1}^{N} e_k^2 \tag{15}$$

This makeover is attained over the below given equality constraint:

$$y_k = \omega^T \varphi(x_k) + b + e_k \tag{16}$$

Where $\varphi(x_k)$ denotes the non-linear map of the kth input vector (estimated to higher-dimension space) while e_k refers to the lack of variables. To resolve the LSSVM function optimizer issue, a model named Lagrangian transformation is used. This technique presents Lagrange multipliers () to change the equality constraints (Eq. (16)) into an unconstrained Lagrangian function (L), definite Eq. (17):

$$L(\omega, b, e, a) = \mathrm{J}(\omega, e) - \sum_{k=1}^{N} a_k \left[\omega^T \varphi(x_k) + b + e_k - y_k\right] \tag{17}$$

By using the Karush-Kuhn-Tucker (KKT) condition in Eq. (18), the Lagrangian function is diminished to get the optimum values of ω, b, e_k, and a_k. This optimizer procedure finally mains to the LSSVM method for error compensation in predicting.

$$\begin{cases} \dfrac{\partial L}{\partial \omega} = 0 \rightarrow \omega = \displaystyle\sum_{k=1}^{N} a_k \varphi(x_k) \\ \dfrac{\partial L}{\partial b} = 0 \rightarrow \displaystyle\sum_{k=1}^{N} a_k = 0 \\ \dfrac{\partial L}{\partial e_{\mathrm{k}}} = 0 \rightarrow a_k = \gamma e_k, k = 1,2,N \\ \dfrac{\partial L}{\partial a_k} = 0 \rightarrow \omega^T \varphi(x_k) + b + e_k - y_k, k = 1,2,..,N \end{cases} \tag{18}$$

Eq. (19) is attained after a full calculation as below:

$$\begin{bmatrix} 0 & l^T \\ l & \Omega + \gamma^{-1} l \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \tag{19}$$

Whereas, $a = [a_1, a_2, \ldots, a_N], y = [y_1, y_2, \ldots, y_N], \Omega_{ki} = \varphi(x_k)^T \varphi(x_i), k, i = 1,2, \ldots, N_o$

Owing to the higher, dimensional of the feature space afterward the non-linear map straight functioning with it is computationally costly. To find out this, LSSVM uses the kernel trick. This model influences a kernel function $k(x, x_k)$ that functions on the new input space (x) to calculate the internal product in the higher-dimension space. The Mercer condition certifies that such a kernel function occurs. Arithmetically, this relationship is stated in Eq. (20) and the last regression function of LSSVM is attained in Eq. (21).

$$k(x_k, x_i) = \varphi(x_k)^T \varphi(x_i) \tag{20}$$

$$y(x) = \sum_{k=1}^{N} a_k \, k(x, x_k) + b \tag{21}$$

In the LSSVM structure, the Radial Basis Function (RBF) kernel is selected owing to its well-established generalization capability and extensive convergence field. The RBF kernel function is expressed in Eq. (22):

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

$$K(x_k, x_i) = \exp\left(-\frac{\|x_k - x_i\|^2}{2\sigma^2}\right) \quad (22)$$

Whereas, σ (sigma) signifies the breadth. Enhancing the parameter of the RBF kernel, chiefly sigma (σ) and gamma (γ), is vital for attaining optimum performance in the LSSVM method.

C. SSA-based Tuning

Lastly, the SSA is adapted for tuning the two ML techniques. This model is chosen for its robust capability in balancing exploration and exploitation, and avoids local optima and ensures effective global search. Compared to optimizers like GAs or PSO, SSA has a simpler mathematical model and fewer parameters to adjust, mitigating computational complexity and tuning time. Its bio-inspired nature allows efficient convergence and adaptability to dynamic problem landscapes, making it appropriate for optimizing ML methods in cybersecurity. The ability to quickly find optimal or near-optimal hyperparameters improves model performance while minimizing manual intervention, providing a practical and efficient solution for tuning complex ensemble systems. Fig. 2 represents the flowchart of the SSA model.

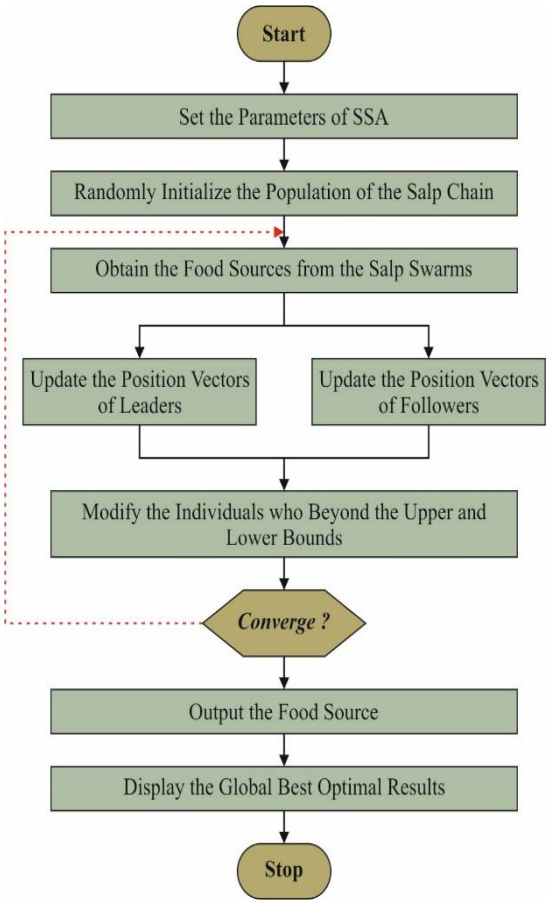

Figure 2. Flowchart of SSA model

SSA is a new metaheuristic model for swarm intelligence optimization, which simulates the swarming of salp chains, attained for better displacement by feeding, reproduction, and environment [29]. The salp chain consists of a leader at the front and followers forming a connected chain. As in other swarm methods, initial salp and prey locations are set within the search space. The position of leader is updated by

$$x_1(k) = \begin{cases} F(i,k) + c_1\left(\left(ub(k) - lb(k)\right)c_2 + lb(k)\right) \leftarrow c_3 \geq 0 \\ F(i,k) - c_1\left(\left(ub(k) - lb(k)\right)c_2 + lb(k)\right) \leftarrow c_3 < 0 \end{cases} \quad (23)$$

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

Where, $x_1(k)$, $F(i,k)$, refers to the leader position and prey location (initial salp) for the k^{th} dimension, $ub(k)$ and $lb(k)$ are upper as well as lower boundaries of the space solution for the k^{th} dimension, c_1 shows a space solution, and c_2, c_3 are uniform distribution random numbers.

$$c_1 = 2e^{-(\frac{4i}{N})^2} \quad (24)$$

Where$i$ denotes the existing iteration, N shows the overall iteration number. c_2 and c_3 are parameters that define the route of the leader movement and, therefore, the chain. The follower salps positions ($j \geq 2$) are determined by Newton's laws of motion in discrete domains:

$$x_j(k) = \frac{x_j(k) + x_{j-1}(k)}{2} \quad (25)$$

Algorithm 1 illustrates the SSA methodology.

Algorithm 1: SSA model
Initialize Salp populace randomly within the search space.
Identify the best Salp (leader) based on the current fitness (objective function).
Move towards the food source to update the leader's position (optimal solution found so far) with some randomness for exploring the search space.
Follow the position of the Salp in front of them to update the position of the follower Salp.
Compute the fitness of all Salp's in their new positions.
Update the food source if a better solution is found.
Repeat steps 3-6 until the optimum iteration or stopping criteria is met.
Return the best solution found.

The optimal solution relates to the food location F, and the process is repeated. SSA utilizes the FF to improve classification by reducing the error rate, as defined in Eq. (26).

$$fitness(x_i) = ClassifierErrorRate(x_i) = \frac{No.of\ misclassified\ samples}{Overall\ samples} \times 100 \quad (26)$$

4. Result Analysis

In this section, the validation of the BBOA-EMLCAD approach is examined under the WSN-DS dataset [30]. Table 1 describes the dataset.

Table 1: Dataset description

Classes	Sample Numbers
"Normal"	"340066"
"Blackhole"	"10049"
"Grayhole"	"14596"
"Flooding"	"3312"
"Scheduling Attacks"	"6638"
Overall Samples	374661

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

Fig. 3 presents the classifier outputs of the BBOA-EMLCAD system on the TR dataset under 70:30 of TRAP/TESP. Figs. 3a-3b displays the confusion matrix under all 5 classes. Fig. 3c and 3d shows the PR and ROC study indicating efficient outputs with better values.

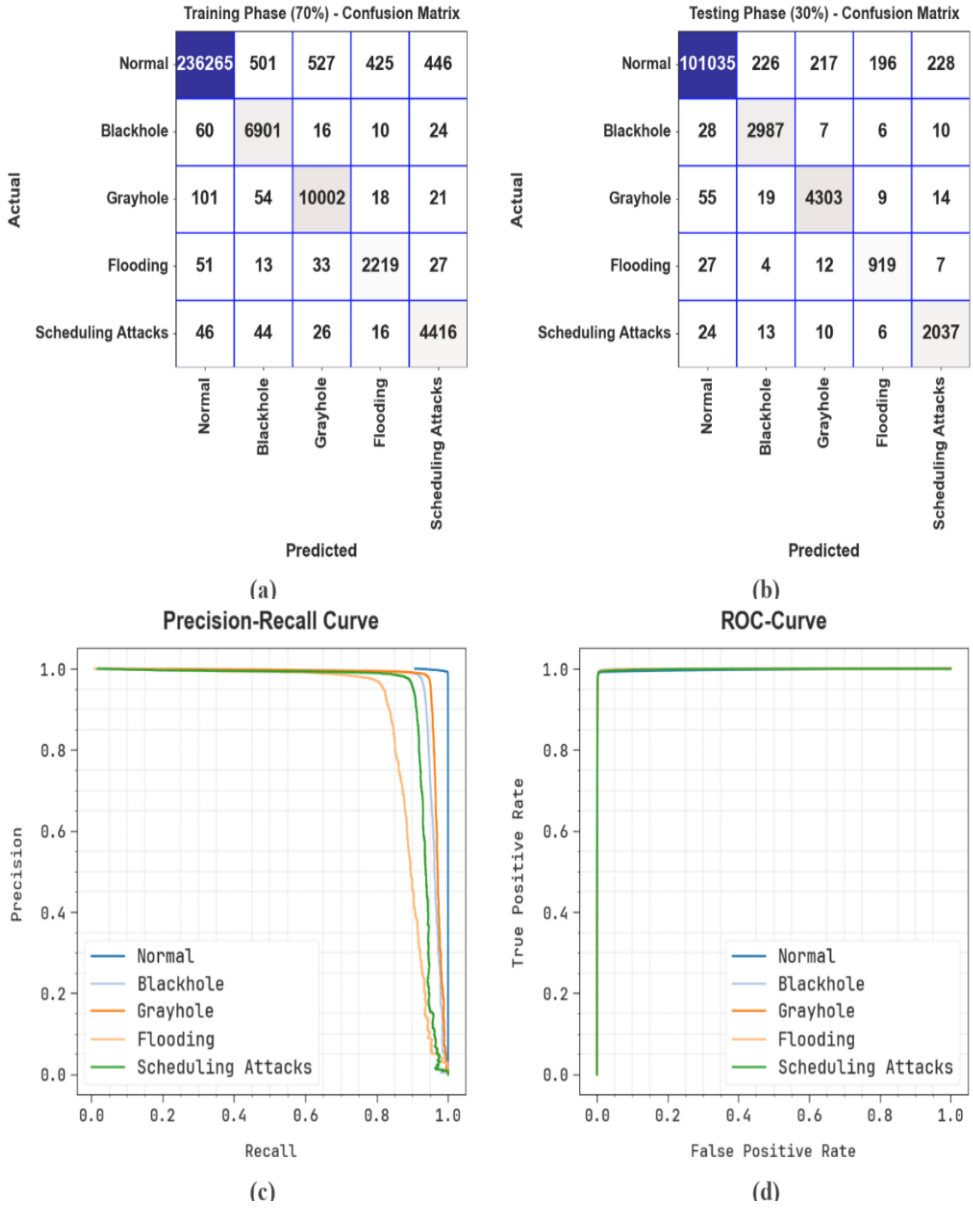

Figure 3. Classifier outputs of (a-b) confusion matrices and (c-d) PR and ROC curves under 70:30 of TRAP/TESP

Table 2 and Fig. 4 demonstrates the overall attack detection outputs of the BBOA-EMLCAD approach with 70:30 of TRAP/TESP under five classes. With 70%TRAP, the BBOA-EMLCAD approach offers an average $accu_y$, $sens_y$, $spec_y$, F_{score}, and MCC of 99.62%, 97.51%, 99.61%, 94.42%, and 93.53%, respectively. Moreover, with 30%TESP, the BBOA-EMLCAD techniques gain average $accu_y$, $sens_y$, $spec_y$, F_{score}, and MCC of 99.60%, 97.51%, 99.57%, 94.18%, and 93.25%, accordingly.

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

Table 2: Attack detection outcome of BBOA-EMLCAD approach under 70:30 of TRAP/TESP

Class	$Accu_y$	$Sens_y$	$Spec_y$	F_{Score}	MCC
TRAP (70%)					
Normal	99.18	99.20	98.93	99.55	95.28
Blackhole	99.72	98.43	99.76	95.03	94.95
Grayhole	99.70	98.10	99.76	96.17	96.04
Flooding	99.77	94.71	99.82	88.21	88.31
Scheduling Attacks	99.75	97.10	99.80	93.14	93.10
Average	99.62	97.51	99.61	94.42	93.53
TESP (30%)					
Normal	99.11	99.15	98.72	99.51	94.97
Blackhole	99.72	98.32	99.76	95.02	94.93
Grayhole	99.69	97.80	99.77	96.17	96.02
Flooding	99.76	94.84	99.81	87.32	87.48
Scheduling Attacks	99.72	97.46	99.77	92.89	92.85
Average	99.60	97.51	99.57	94.18	93.25

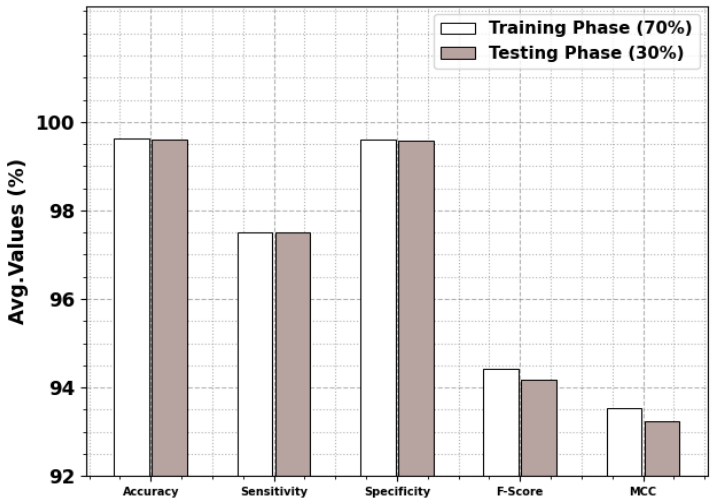

Figure 4. Attack detection outcome of BBOA-EMLCAD approach under 70:30 of TRAP/TESP

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

In Fig. 5, the training (TR) and validation (TS) accuracy results of the BBOA-EMLCAD method over 1-10 epochs is demonstrated under 70:30 of TRAP/TESP. The figure exhibits those TR/TS accuracy steadily increases over iterations, illustrating the robust performance of the BBOA-EMLCAD technique. Their close alignment across epochs indicates minimal overfitting and consistent accuracy on unseen data.

In Fig. 6, the TR/TS loss of the BBOA-EMLCAD approach over 0-10 epochs is portrayed under 70:30 of TRAP/TESP. The decreasing trend in TR/TS accuracy reflects the robust capability of the BBOA-EMLCAD technique to balance generality as well as data fitting. The persistent loss additionally confirms the enhanced performance and prediction over time.

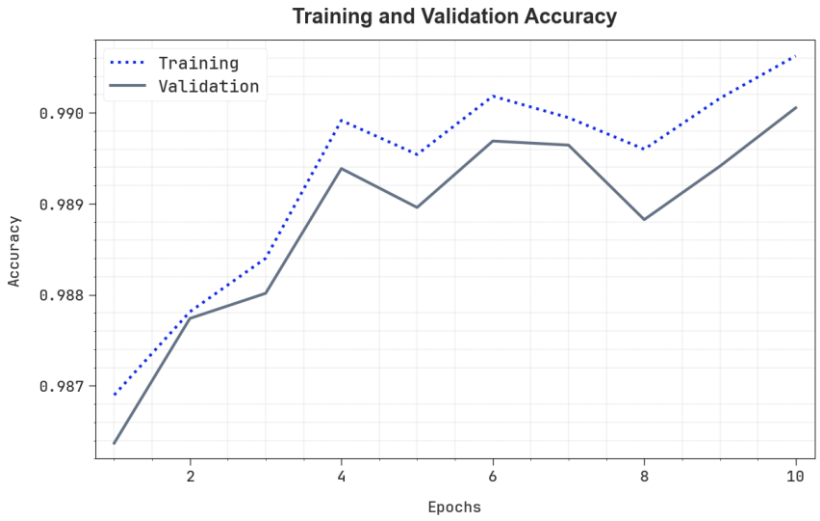

Figure 5. $Accu_y$ Curve of the BBOA-EMLCAD method under 70:30 of TRAP/TESP

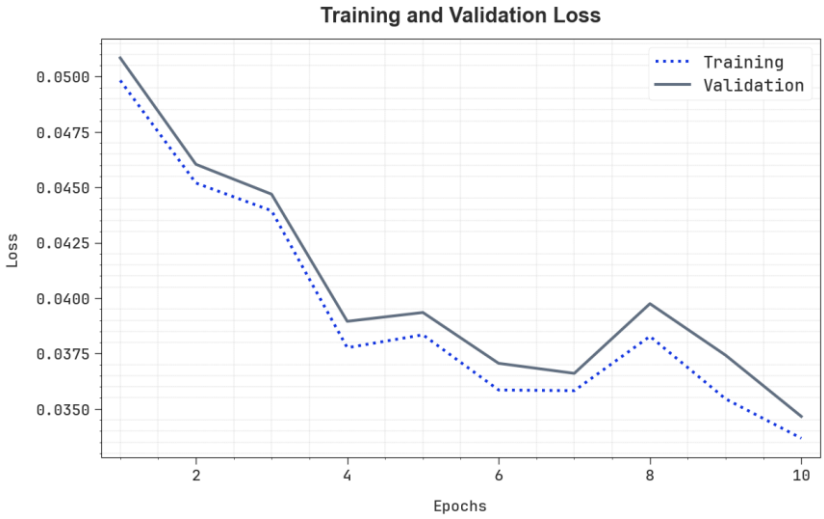

Figure 6. Loss curve of the BBOA-EMLCAD method under 70:30 of TRAP/TESP

Fig. 7 specifies the confusion matrix classifier outputs of the BBOA-EMLCAD approach on the TR dataset under 80:20 of TRAP/TESP. Figs. 7a-7b exhibits the confusion matrix with precise detection of all 5 classes. Fig. 7c and 7d illustrate the PR and ROC study indicating efficient outputs with better values under diverse classes.

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

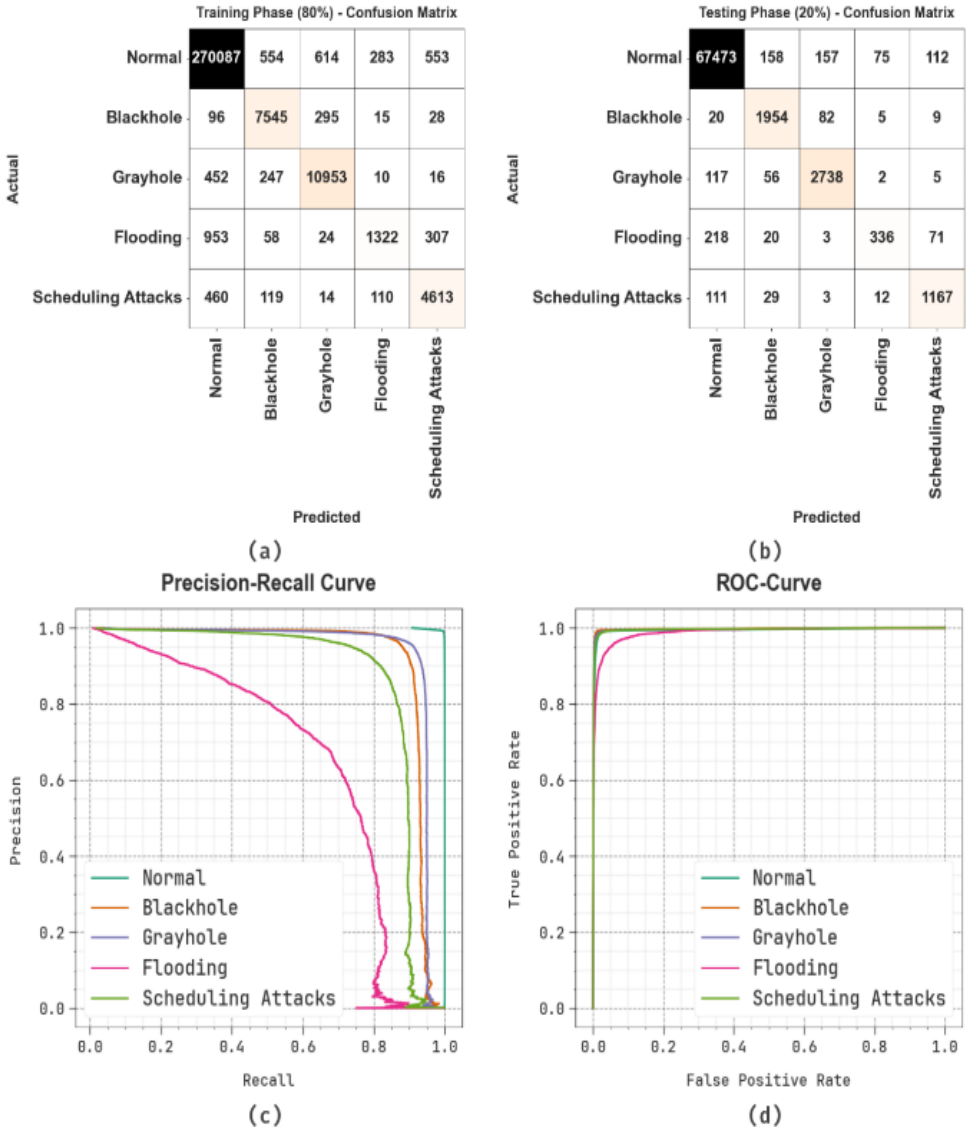

Figure 7. Classifier outputs of (a-b) confusion matrices and (c-d) PR and ROC curves under 80:20 of TRAP/TESP

Table 3 and Fig. 8 demonstrates the attack detection output of the BBOA-EMLCAD approach under 80:20 of TRAP/TESP. The -EMLCAD approach attains an $accu_y$ of 99.30%, $sens_y$ of 87.89%, $spec_y$ of 84.80%, and F_{score} of 85.77% on the TR set, and an $accu_y$ of 99.32%, $sens_y$ of 88.59%, $spec_y$ of 85.52%, and F_{score} of 86.50% on the TS set. For individual attack classes, normal traffic exhibits the highest $sens_y$ and $spec_y$, with an $accu_y$ of 98.7% and F_{score} of 99.3%. Blackhole and grayhole attacks achieve an $accu_y$ above 99.4%, with $sens_y$ and $spec_y$ values mostly in the 88% to 94% range. Flooding attacks exhibit relatively lower $sens_y$ and F_{score}, $spec_y$ of 75% to 78% and 60% to 62% F-Score, indicating a greater challenge in detection. Scheduling attacks perform well with an $accu_y$ close to 99.5%, $sens_y$ above 83%, and F_{score} above 85%. Overall, the MCC values confirm robust and consistent prediction capability across all classes for both TR/TS phases.

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

Table 3: Attack detection outcome of BBOA-EMLCAD approach under 80:20 of TRAP/TESP

Class Labels	$Accu_y$	$Sens_y$	$Spec_y$	F_{Score}	MCC
TRAP (80%)					
Normal	98.68	99.28	99.26	99.27	92.10
Blackhole	99.53	88.53	94.56	91.44	91.25
Grayhole	99.44	92.04	93.79	92.91	92.62
Flooding	99.41	75.98	49.62	60.04	61.13
Scheduling Attacks	99.46	83.61	86.78	85.17	84.91
Average	99.30	87.89	84.80	85.77	84.40
TESP (20%)					
Normal	98.71	99.31	99.26	99.29	92.35
Blackhole	99.49	88.14	94.40	91.16	90.96
Grayhole	99.43	91.79	93.83	92.80	92.51
Flooding	99.46	78.14	51.85	62.34	63.40
Scheduling Attacks	99.53	85.56	88.28	86.90	86.67
Average	99.32	88.59	85.52	86.50	85.18

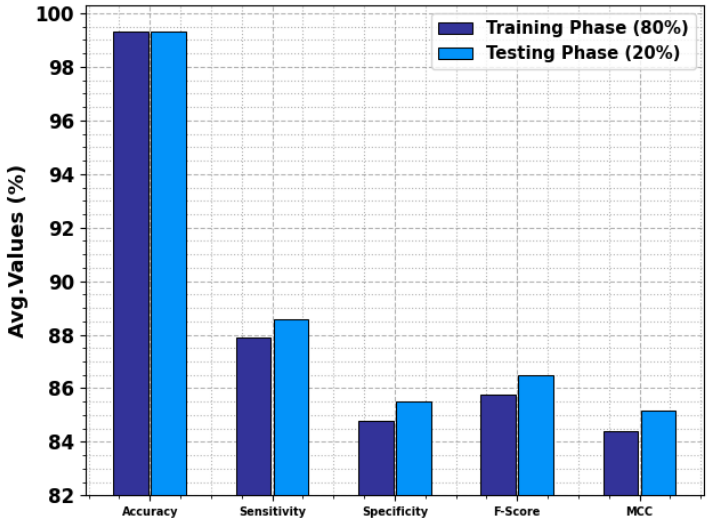

Figure 8. Attack detection outcome of BBOA-EMLCAD approach under 80:20 of TRAP/TESP

Fig. 9 illustrates the TR/TS accuracy outputs of the BBOA-EMLCAD method over 1 to 10 epochs under 80:20 of TRAP/TESP. The outputs portray a consistent increase in accuracy as the epoch's progress, highlighting the efficiency of the BBOA-EMLCAD approach. The close correlation between TR and RS accuracies indicates low overfitting and stable performance on unseen samples.

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

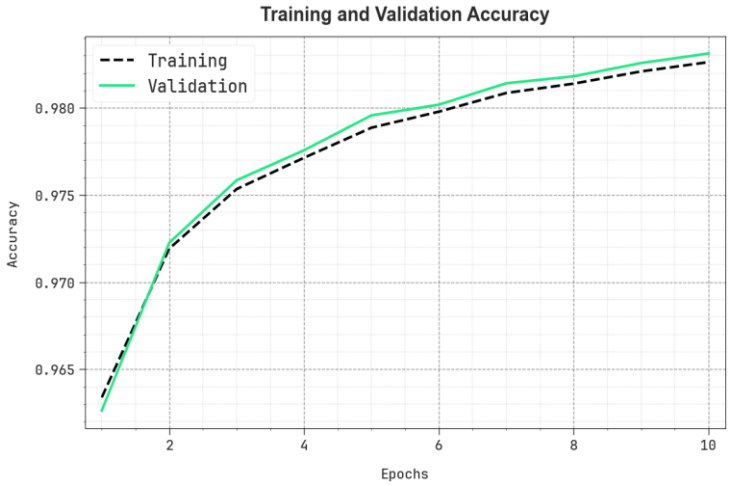

Figure 9. $Accu_y$ Curve of the BBOA-EMLCAD approach under 80:20 of TRAP/TESP

Fig. 10 presents the TR/TS loss curves of the BBOA-EMLCAD approach over 0 to 10 epochs under 80:20 of TRAP/TESP. The continuous decline in loss for both TR and TS sets highlights the robust capability of the BBOA-EMLCAD technique to achieve a robust balance between fitting the data and generalizing to new samples. This persistent decline in loss shows an excellent model performance and increasingly accurate predictions as training progresses.

Figure 10. Loss curve of the BBOA-EMLCAD technique under 80:20 of TRAP/TESP

The comparison study of the BBOA-EMLCAD model is described in Table 4 and Fig. 11 [31-33]. The outputs showed that the XGBoost and GB models displayed least outcomes. Furthermore, the KNN and AdaBoost techniques exhibited slightly nearer results. Moreover, the KNN-PSO technique has presented reasonable performance with $accu_y$ of 96.47%, $sens_y$ of 94.10%, $spec_y$ of 94.21%, and F_{score} of 92.59%. However, the BBOA-EMLCAD model establishes assured performance with $accu_y$ of 99.62%, $sens_y$ of 97.51%, $spec_y$ of 99.61%, and F_{score} of 94.42%.

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

Table 4: Comparison analysis of BBOA-EMLCAD method with existing approaches

Methods	$Accu_y$	$Sens_y$	$Spec_y$	F_{Score}
BBOA-EMLCAD	99.62	97.51	99.61	94.42
AdaBoost	96.30	94.96	94.47	91.09
GB Model	94.23	96.95	94.55	92.43
XGBoost	95.91	94.75	94.14	90.70
KNN	96.40	96.99	96.20	90.79
KNN-PSO	96.47	94.10	94.21	92.59

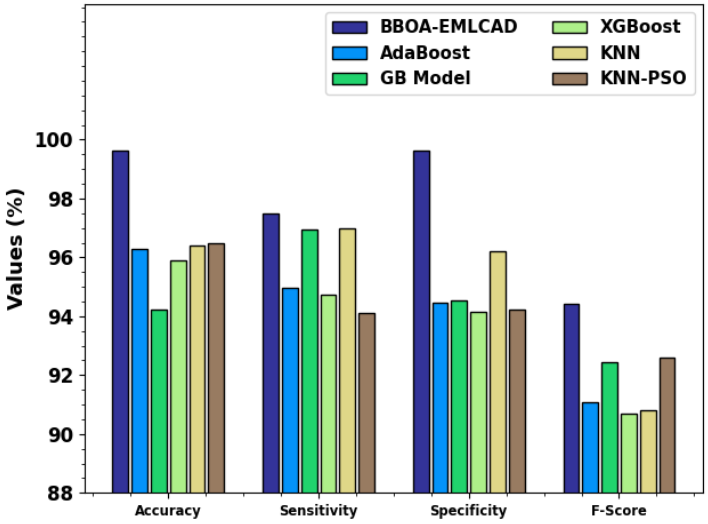

Figure 11. Comparison evaluation of BBOA-EMLCAD method with existing approaches

Table 5 and Fig. 12 indicates the computational time (CT) analysis of the BBOA-EMLCAD approach with present techniques. The BBOA-EMLCAD methodology achieves the lowest CT of 4.82 seconds, illustrating superior efficiency. In contrast, the AdaBoost method requires 13.75 seconds, while the GB model records a slightly higher CT of 14.24 seconds. XGBoost completes its process in 8.81 seconds, followed by KNN with 7.95 seconds. The KNN-PSO technique exhibits moderate improvement, reducing the CT to 7.35 seconds. Overall, the BBOA-EMLCAD method significantly outperforms the existing techniques.

Table 5: CT analysis of BBOA-EMLCAD methodology with existing techniques

Approaches	CT (sec)
BBOA-EMLCAD	4.82
AdaBoost	13.75
GB Model	14.24
XGBoost	8.81
KNN	7.95
KNN-PSO	7.35

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

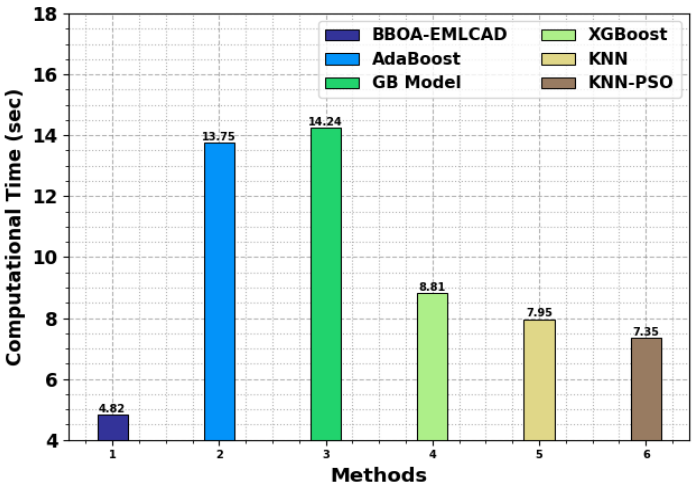

Figure 12. CT analysis of BBOA-EMLCAD methodology with existing techniques

5. Conclusion

In this study, the BBOA-EMLCAD method for secure IoT environment is proposed. The main aim relies on the automated classification of the cyberthreats in IoT. Initially, the BBO method is utilized for FS. Moreover, an ensemble of two ML approaches namely XGBoost and LSSVM are employed for the automatic identification of the cyber-attacks. Finally, the SSA is used for tuning the two ML techniques. The simulation validation of the BBOA-EMLCAD approach is performed under the WSN-DS dataset. The comparison assessment of the BBOA-EMLCAD approach portrayed a superior accuracy value of 99.62% over existing models.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] Al-Abassi, H. Karimipour, A. Dehghantanha, and R. M. Parizi, "An ensemble deep learning-based cyber-attack detection in industrial control system," *IEEE Access*, vol. 8, pp. 83965-83973, 2020.

[2] P. Kumar, G. P. Gupta, R. Tripathi, S. Garg, and M. M. Hassan, "DLTIF: Deep Learning-Driven Cyber Threat Intelligence Modeling and Identification Framework in IoT-Enabled Maritime Transportation Systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2472-2481, 2023.

[3] Bibi, A. Akhunzada, and N. Kumar, "Deep AI-powered cyber threat analysis in IIoT," *IEEE Internet Things J.*, vol. 10, no. 9, pp. 7749-7760, 2023.

[4] A. Khan *et al.*, "A new explainable deep learning framework for cyber threat discovery in industrial IoT networks," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 11604-11613, 2021.

[5] H. M. Rouzbahani, H. Karimiphan, and L. Lei, "Multi-layer defense algorithm against deep reinforcement learning-based intruders in smart grids," *Int. J. Electr. Power Energy Syst.*, vol. 146, p. 108798, 2023.

[6] Ullah *et al.*, "Software defined network enabled fog-to-things hybrid deep learning driven cyber threat detection system," *Secur. Commun. Netw*, vol. 2021, pp. 1-15, 2021.

[7] W. Ding, M. Abdel-Basset, and R. Mohamed, "DeepAK-IoT: An effective deep learning model for cyberattack detection in IoT networks," *Inf. Sci.*, vol. 634, pp. 157-171, 2023.

[8] K. Rahman *et al.*, "Cognitive lightweight logistic regression-based IDS for IoT-enabled FANET to detect cyberattacks," *Mobile Inf. Syst.*, vol. 2023, 2023.

[9] M. Ebrahimi, J. F. Nunamaker Jr, and H. Chen, "Semi-supervised cyber threat identification in dark net markets: A transductive and deep learning approach," *J. Manage. Inf. Syst.*, vol. 37, no. 3, pp. 694-722, 2020.

[10] M. Al-Hawawreh and E. Sitnikova, "Leveraging deep learning models for ransomware detection in the industrial Internet of Things environment," in *Proc. MilCIS*, Canberra, ACT, Australia, 2019, pp. 1-6.

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025

[11] S. C. Amirthabai, U. Malhotra, S. T. Rajeswari, and S. T. Natesan, "Concurrency and Computation: Practice and Experience," *Concurrency Computat: Pract. Exper.*, vol. 34, no. 5, p. e6721, 2022.

[12] S. M. Naser, Y. H. Ali, and D. A. J. Obe, "Hybrid cyber-security model for attacks detection based on deep and machine learning," *Int. J. Online Biomed. Eng.*, vol. 18, no. 11, pp. 84-96, 2022.

[13] R. Almajed *et al.*, "Using machine learning algorithm for detection of cyber-attacks in cyber physical systems," *Period. Eng. Nat. Sci.*, vol. 10, no. 3, pp. 261-275, 2022.

[14] S. Krishnasamy *et al.*, "Development and validation of a cyber-physical system leveraging EFDPN for enhanced WSN-IoT network security," *Sensors*, vol. 23, no. 22, p. 9294, 2023.

[15] S. Ismail, D. Dawoud, and H. Reza, "A lightweight multilayer machine learning detection system for cyber-attacks in WSN," in *Proc. IEEE CCWC*, Las Vegas, NV, USA, 2022, pp. 481-486.

[16] Surbhi, N. R. Chauhan, and N. Dahiya, "Optimizing XGBoost hyperparameters using the dragonfly algorithm for enhanced cyber attack detection in the internet of healthcare things (IoHT)," *Cluster Comput.*, vol. 28, no. 4, p. 230, 2025.

[17] T. Vaiyapuri *et al.*, "Automated cyberattack detection using optimal ensemble deep learning model," *Trans. Emerg. Telecommun. Technol.*, vol. 35, no. 4, p. e4899, 2024.

[18] H. Alamro *et al.*, "Feature enhancement model with up sampling based cyber threat attack detection and classification on imbalanced dataset in Industrial Internet of Things," *Alexandria Eng. J.*, vol. 128, pp. 247-258, 2025.

[19] N. Alkhafaji, T. Viana, and A. Al-Sherbaz, "Integrated genetic algorithm and deep learning approach for effective cyber-attack detection and classification in Industrial Internet of Things (IIoT) environments," *Arab. J. Sci. Eng.*, pp. 1-25, 2024.

[20] Y. Chen, Y. Guo, Y. Gao, and B. Liu, "A novel lightweight deep learning framework using enhanced pelican optimization for efficient cyberattack detection in the Internet of Things environments," *J. Eng. Appl. Sci.*, vol. 72, no. 1, pp. 1-26, 2025.

[21] H. Alamro *et al.*, "Modelling of Bayesian-based optimized transfer learning model for cyber-attack detection in Internet of Things assisted resource constrained systems," *IEEE Access*, vol. 12, pp. 12345-12356, 2024.

[22] R. Wani, A. A. Aziz, and R. Raut, "Enhanced detection of cyberattacks in wireless sensor networks and IoT-networks using powerful stacking ensemble," in *Proc. ICSSES*, 2025, pp. 1-9.

[23] S. Malathi and S. R. Begum, "Enhancing trustworthiness among IoT network nodes with ensemble deep learning-based cyber attack detection," *Expert Syst. Appl.*, vol. 255, p. 124528, 2024.

[24] B. Gupta *et al.*, "A hybrid ant lion optimization algorithm based lightweight deep learning framework for cyber attack detection in IoT environment," *Comput. Electr. Eng.*, vol. 122, p. 109944, 2025.

[25] Alrefaei and M. Ilyas, "Ensemble deep learning model based on multi-class classification technique to detect cyber attacks in IoT environment," in *Proc. Int. Conf. Smart Comput., IoT Mach. Learn. (SIML)*, 2024, pp. 174-179.

[26] S. E. Sorour *et al.*, "Credit card fraud detection using the brown bear optimization algorithm," *Alexandria Eng. J.*, vol. 104, pp. 171-192, 2024.

[27] P. Mani *et al.*, "An efficient real-time vehicle classification from a complex image dataset using eXtreme gradient boosting and the multi-objective genetic algorithm," *Processes*, vol. 12, no. 6, p. 1251, 2024.

[28] F. A. Zaini *et al.*, "A hybrid model based on LSSVM and the improved BFOA for sustainability of daily electricity load forecasting in Malaysia," *Energies*, vol. 17, no. 3, p. 567, 2024.

[29] S. E. W. Case, "Salp swarm algorithm application in excitation controllers for power system rotor angle stability enhancement: WSCC case study," *IEEE Trans. Power Syst.*, vol. 38, no. 2, pp. 1234-1245, 2023.

[30] Almomani, M. Al-Kasasbeh, and M. AL-Akhras, "WSN-DS: A dataset for intrusion detection systems in wireless sensor networks," *J. Sensors*, vol. 2016, pp. 1-16, 2016.

[31] Murugesh and S. Murugan, "Moth search optimizer with deep learning enabled intrusion detection system in wireless sensor networks," *SSRG Int. J. Electr. Electron. Eng.*, vol. 10, no. 4, pp. 77-90, 2023.

[32] G. Liu *et al.*, "An enhanced intrusion detection model based on improved kNN in WSNs," *Sensors*, vol. 22, no. 4, p. 1407, 2022.

[33] M. Alqahtani *et al.*, "A genetic-based extreme gradient boosting model for detecting intrusions in wireless sensor networks," *Sensors*, vol. 19, no. 20, p. 4383, 2019.

DOI: https://doi.org/10.54216/JCIM.170206
Received: March 03, 2025 Revised: May 31, 2025 Accepted: July 06, 2025