sensors

MDPI

Article

Binary Chimp Optimization Algorithm with ML Based Intrusion Detection for Secure IoT-Assisted Wireless Sensor Networks

Mohammed Aljebreen [1], Manal Abdullah Alohali [2], Muhammad Kashif Saeed [3], Heba Mohsen [4], Mesfer Al Duhayyim [5,*], Amgad Atta Abdelmageed [6], Suhanda Drar [6] and Sitelbanat Abdelbagi [6]

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 Computer Science, Applied College, King Khalid University, Muhayil 63311, Saudi Arabia
4 Department of Computer Science, Faculty of Computers and Information Technology, Future University in Egypt, New Cairo 11835, Egypt
5 Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj 16273, Saudi Arabia
6 Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia
* Correspondence: m.alduhayyim@psau.edu.sa

Abstract: An Internet of Things (IoT)-assisted Wireless Sensor Network (WSNs) is a system where WSN nodes and IoT devices together work to share, collect, and process data. This incorporation aims to enhance the effectiveness and efficiency of data analysis and collection, resulting in automation and improved decision-making. Security in WSN-assisted IoT can be referred to as the measures initiated for protecting WSN linked to the IoT. This article presents a Binary Chimp Optimization Algorithm with Machine Learning based Intrusion Detection (BCOA-MLID) technique for secure IoT-WSN. The presented BCOA-MLID technique intends to effectively discriminate different types of attacks to secure the IoT-WSN. In the presented BCOA-MLID technique, data normalization is initially carried out. The BCOA is designed for the optimal selection of features to improve intrusion detection efficacy. To detect intrusions in the IoT-WSN, the BCOA-MLID technique employs a class-specific cost regulation extreme learning machine classification model with a sine cosine algorithm as a parameter optimization approach. The experimental result of the BCOA-MLID technique is tested on the Kaggle intrusion dataset, and the results showcase the significant outcomes of the BCOA-MLID technique with a maximum accuracy of 99.36%, whereas the XGBoost and KNN-AOA models obtained a reduced accuracy of 96.83% and 97.20%, respectively.

Keywords: intrusion detection system; wireless sensor networks; machine learning; chimp optimization algorithm; feature selection

Citation: Aljebreen, M.; Alohali, M.A.; Saeed, M.K.; Mohsen, H.; Al Duhayyim, M.; Abdelmageed, A.A.; Drar, S.; Abdelbagi, S. Binary Chimp Optimization Algorithm with ML Based Intrusion Detection for Secure IoT-Assisted Wireless Sensor Networks. *Sensors* **2023**, *23*, 4073. https://doi.org/10.3390/s23084073

Academic Editor: Achyut Shankar

Received: 4 February 2023
Revised: 24 February 2023
Accepted: 1 March 2023
Published: 18 April 2023

1. Introduction

The Internet of Things (IoT) is commonly known as a network that is made up of many devices that are connected through the internet [1]. Wireless Sensor Networks (WSN) have a crucial role in the IoT, which is helpful to produce seamless data that influence the lifetime of a network. Despite the significant applications of the IoT [2], various challenges, such as storage, security, load balancing, and energy exist. In addition, it is an open network with random and dynamic topology [3]. Thus, it is essential to execute a sequence of targeted studies to guarantee reliability, real-time response, energy-saving, and other operational needs of WSNs. As a data-centric network, a lot of delicate information is transmitted, collected, processed, and stored in WSN [4,5]. Its security problem has become very serious. Owing to the characteristics and limitations of WSN itself, the data can be easily tampered with, ruined, or stolen. How to protect network security effectually in the

face of numerous network attacks becomes a significant research topic [6]. Passive defense via firewalls, access control, and other means is inadequate to thwart every network attack. Intrusion detection (ID) is a proactive security protection technology that is used to observe the functioning condition of the network and find intrusions such as maloperations and internal or external attacks, in such a way that the network can interrupt them and respond as needed [7].

To protect IoT systems from cyber threats, an Intrusion Detection System (IDSs) is another line of defense that must be advanced in IoT networks [8,9]. Many surveys have been performed to describe machine learning (ML)-related IDSs for protection from compromised IoT devices or IoT networks. The surveys have covered studies on IDSs for cloud-related IoT systems, WSNs, mobile ad hoc networks (MANETs), and cyber–physical systems (CPS) [10]. However, conventional IDS techniques are insufficient or less effective for the security of IoT systems because of their peculiar features, for example,, limited bandwidth capacity [11], limited energy, heterogeneity, global connectivity, and ubiquity.

Deep Learning (DL) and Machine Learning (ML) related methods have obtained credibility through a successful implementation in the detection of network attacks, which includes IoT networks. Since WSN includes low computing and communication abilities, conventional network intrusion detection models are not directly used in WSN. Presently, several researchers on WSN intrusion detection can exploit ML models for investigating traffic data. Because of the expansion in both the network's size and its user base, the WSN network produces high-dimensional traffic data, and the classical ML models encounter problems such as poor feature extraction and detection accuracy, which cannot meet the requirements of such an application environment [12]. Compared to ML models for IDS, the DL models can decrease the computation burden and increase the ability to learn the characteristics of data traffic, which can improve the precision of the detection model [13].

This article presents a Binary Chimp Optimization Algorithm with Machine Learning based Intrusion Detection (BCOA-MLID) technique for secure IoT-WSN. In the presented BCOA-MLID technique, data normalization is initially carried out. The BCOA is designed for the optimal selection of features to improve intrusion detection efficacy. To detect intrusions in the IoT-WSN, the BCOA-MLID technique employs a Class-specific Cost Regulation Extreme Learning Machine (CCR-ELM) classification model with a Sine Cosine Algorithm (SCA) as a parameter optimization approach. The design of BCOA feature selection with an SCA-optimized CCR-CLM classifier for intrusion detection shows the novelty of the work. The experimental result of the BCOA-MLID technique was tested on the Kaggle intrusion dataset.

2. Related Works

Kagade and Jayagopalan [14] developed a new intrusion detection system (IDS) that was set up with a DL method. First of all, the optimum cluster head (CH) was chosen from among the SNs, from which SNs with higher energy will be listed to act as CH. In this study, the CH selection was optimally assessed concerning energy variables under limitations such as distance and delay. For the best selection, a new technique called the Self-Improved Sea Lion Optimization (SI-SLnO) method was presented in this study. Krishnan et al. [15] aimed to frame an intrusion prevention protocol and anomalous ID protocol for interruption evasion in the IoT, based on WSN for expanding the information reliability and network time frame. This structure made dissimilar energy-efficient groups reliant on the natural features of nodes. In [16], a smart IDS suitable to finding IoT-related attacks was applied. Specifically, to identify malicious IoT network traffic, a DL technique was utilized. The identity solution has supported the IoT connectivity protocols to interoperate, and it assures the security of operation. An IDS is one common type of network security technology that can be employed to secure the network. Zhiqiang et al. [17] devised an enriched empirical-related component analysis for choosing applicable features. The feature-selecting method compiles the benefits of both PCA and empirical mode decomposition to retain many

appropriate attributes. The classifications of the attack nodes with selective attributes have been executed with LSTM.

Muruganandam et al. [18] developed a DL-related feed-forward ANN method that enables accurate predictions of k-barrier count for potential ID and mitigation. The area of RoI, sensing transmission area, sensor sensing area, and various sensors were the four potential features that can be utilized to assess and learn the feed-forward ANN method. Subramani and Selvi [19] modeled an intelligent IDS to detect intruders in IoT-related WSNs so that it can manage such intrusions. To develop this intelligent IDS, a rule- and multi-objective PSO-based feature selection technique was devised by the author, who even suggested an intellectual rule-based enhanced multiclass SVM classifier method to detect the intruders with a higher level of accuracy. Saba et al. [20] presented a CNN-related algorithm for anomaly-based IDS that uses IoT power, offering the ability to potentially inspect all of the traffic across the IoT. This presented algorithm displays the capability to find any abnormal traffic behavior and possible intrusion.

Sadeghi et al. [21] presented a hybrid method of a new DCNN and multi-objective binary chimp optimization algorithm (MOBChOA) for selecting the feature optimally. Then, for optimal selection of features, a method called MOBChOA is applied. Finally, for classifying the pixels into particular specific land-cover labels, the author trained the fully connected DCNN. In [22], the author presented a method to optimize the network parameters, which combined both GRU and CNN, and distinct CNN–GRU combination sequences were introduced. In [23], the author scrutinized the effect of data imbalance on formulating a potential SCADA-based IDS. CNNs were combined with Long Short-Term Memory (CNN-LSTM) for binary classification.

Abosata et al. [24] modeled a Federated-Transfer-Learning-Based Customized Distributed IDS (FT-CID) approach to identify RPL intrusion in a heterogeneous IoT. Primarily, to construct a local model, the central server initialized the FT-CID with a predefined learning approach and observed the unique attributes of various RPL-IoTs. Then, using the local parameters, the edge IDSs were trained and, through federation, the globally shared parameters generated by the central server were altered and aggregated into diverse local parameters of different edges. In [25], two different approaches were devised. In the first method, a custom CNN was framed and united with LSTM deep network layers. The second model was constructed around each fully connected layer (dense layers) to build an Artificial Neural Network (ANNs).

3. The Proposed Intrusion Detection Model

In this article, an automated BCOA-MLID technique has been developed for accurate intrusion detection to accomplish security tasks in the IoT-WSN. The presented BCOA-MLID technique intends to effectively discriminate different types of attacks to secure the IoT-WSN. In the presented BCOA-MLID technique, a four-stage process is involved, namely, data normalization, FS using BCOA, CCR-ELM classification, and SCA-based parameter optimization. Figure 1 represents the overall flow of the BCOA-MLID approach.

3.1. Data Normalization

In the presented BCOA-MLID technique, data normalization is performed at the initial stage. The data-normalized operation scales the data so that the weighted sum exists in the range of the activation functions [26]. The un-normalized data generates an ill-trained network and delays the convergence. At the same time, normalizing the data accelerate the convergence and attain non-dimensionality. For scaling the data in the range of zero and one, it utilizes the min–max normalized system that is determined as:

$$X_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

where X_{norm} represents normalization data, x signifies the primary value from the database, x_{max} denotes the maximal value, and x_{min} stands for the minimal value.

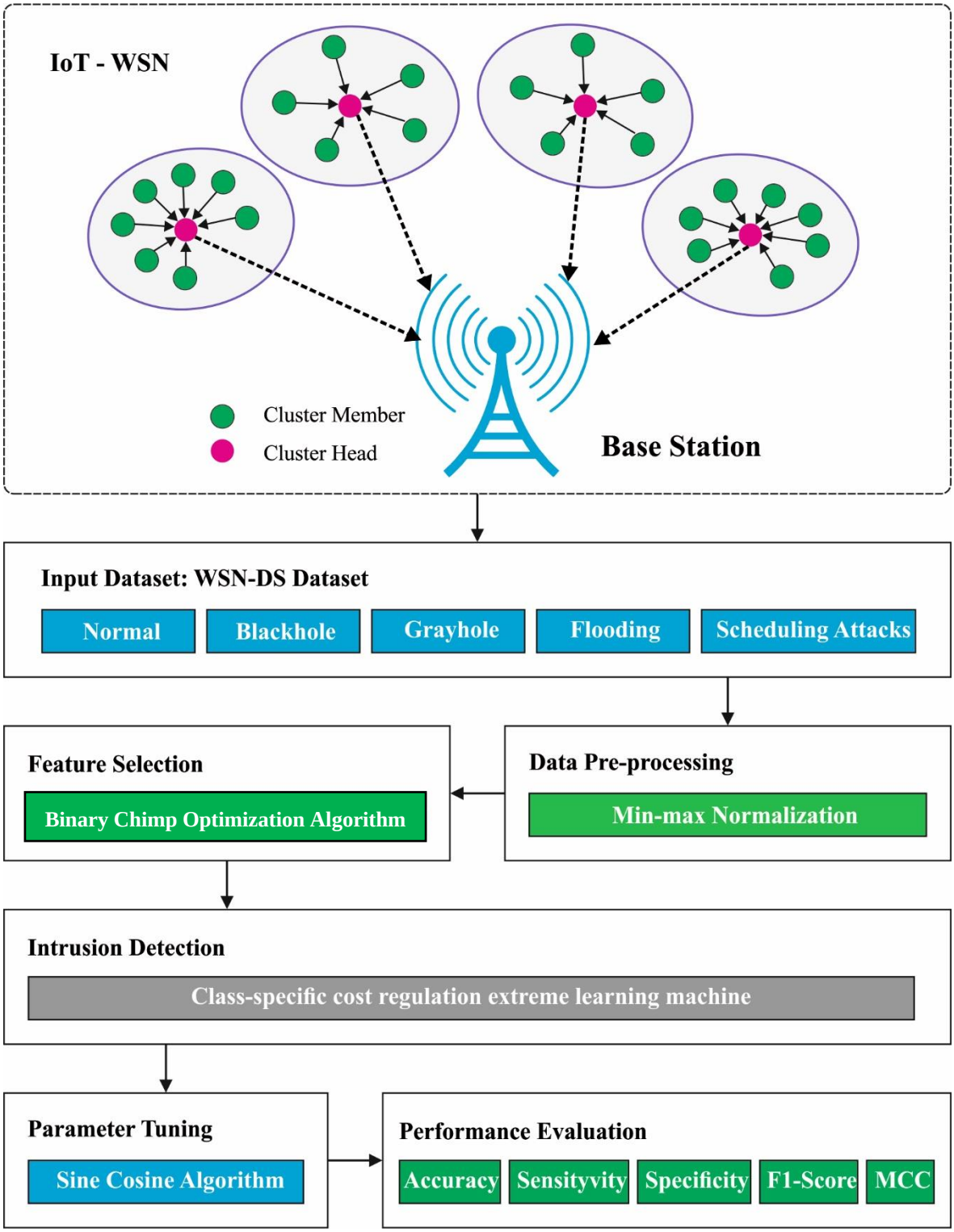

Figure 1. The overall flow of the BCOA-MLID approach.

3.2. Feature Selection Using BCOA

At this stage, the BCOA is designed for the optimal selection of features to improve intrusion detection efficacy. Khishe and Mosavi (2020) introduced a BCOA that was stimulated by the ability of chimpanzees to think individually during group hunting and sexual motivation [21]. The BCOA can recognize optimal solutions by the exploration of the entire search space and avoids the local optima. It is simple to design and does not

require a large number of computational resources. BCOA has a fast convergence rate, which means it can quickly converge to the optimal solution. This makes it suitable for applications where time is a critical factor. In summary, BCOA is a simple and robust optimization algorithm that is capable of finding the global optimal solution in complex and noisy search spaces.

Meanwhile, attacking, driving, blocking, and chasing are the four major stages of BCOA. The BCOA can be initialized by randomly producing several chimps. The attacker chimp prognosticates the breakout path of prey by forcing it back toward the chaser. The driver chimp follows the prey without trying to capture it. The barrier chimp places themselves in trees to generate a barrier during prey development.

The chaser chimp moves faster to catch the prey. Chasing and driving the prey are expressed as follows:

$$d = \left| c \cdot X_{prey}(r) - m \cdot X_{chimp} \right| \tag{2}$$

$$X_{chimp}(r+1) = X_{prey}(r) - a \cdot d \tag{3}$$

$$a = 2 \cdot f \cdot r_1 - f \tag{4}$$

$$c = 2 \cdot r_2 \tag{5}$$

$$m = Chaolic\_value \tag{6}$$

X_{prey} denotes the prey location vector; a, c_t, and m show the coefficient vectors; X_{chimp} symbolizes the chimp location vector; r represents the existing iteration; r_1 and r_2 indicate the random vector $\in$ [0, 1]; f denotes the dynamic vector $\in$ [0, 2.5], and m represents a chaotic vector. First, the chimpanzees search for the prey location during the hunting stage based on the four hunting strategies. Then, the prey position can be evaluated using those hunting strategies, and other chimpanzees update the position of the prey. These steps are expressed as follows:

$$\begin{cases} d_{Attacher} = |c_1 \cdot X_{Attacher} - m_1 \cdot X| \\ d_{Barrier} = |c_2 \cdot X_{Barrier} - m_2 \cdot X| \\ d_{Chaser} = |c_3 \cdot X_{Chaser} - m_3 \cdot X| \\ d_{Driver} = |c_4 \cdot X_{Driver} - m_4 \cdot X| \end{cases} \tag{7}$$

$$\begin{cases} X_1 = X_{Attacher} - a_1(d_{Attacher}) \\ X_2 = X_{Barrier} - a_2(d_{Barrier}) \\ X_3 = X_{Chaser} - a_3(d_{Chaser}) \\ X_4 = X_{Driver} - a_4(d_{Driver}) \end{cases} \tag{8}$$

$$X(t+1) = \frac{X_1 + X_2 + X_3 + X_4}{4} \tag{9}$$

Let $X_{Attacher}$ be the better searching agent, $X_{Barrier}$ represents the second better searching agent, X_{Chaser} represents the third better searching agent, X_{Diver} indicates the fourth better searching agent, and $X(t+1)$ denotes the updated location of every chimp.

Lastly, each chimpanzee attacks the prey. After hunting the prey, they attain sexual motivation, regardless of their duties. Sexual motivation can be represented as follows:

$$x_{chimp(t+1)} = \begin{cases} X_{prey}(l) - a \cdot d & if\ \mu < 0.5 \\ Choatic\_value & if\ \mu \geq 0.5 \end{cases} \tag{10}$$

In Equation (10), μ denotes the randomly generated number $\in [0, 1]$. In the extended version of BCOA, chimpanzees continuously change their location at any point in the search space. In discrete issues, the solution is constrained to binary values. The operator of the binary metaheuristic method moves toward the nearer and farther corners of the hypercube by constantly changing zero to one and one to zero. Thus, in the BBCOA model, the position updating formula needs to be adjusted. For these purposes, a transfer function maps the continuous space to the discrete space. The transfer function symbolizes changing the probability of the location vector from zero to one. Therefore, the transfer function

forces the chimpanzees to move in the discrete space. Here, a newly generated technique used to update the position of a chimpanzee is presented. In the presented technique, the location-updating formula can be given as the following:

$$X_d^{t+1} = \begin{cases} 1 & if\ sigmoid\left(\frac{X_1+X_2+X_3+X_4}{4}\right) \geq R \\ 0 & otherwise \end{cases} \tag{11}$$

$$Sigmoid(x) = \frac{1}{1+e^{-14(x-0.45)}} \tag{12}$$

In the expression, X_d^{t+1} denotes the upgraded binary location at iteration, r, R represents the arbitrary value $\in [0,1$. $Sigmoid(\mathrm{x})$ shows an S-shaped function, $X_{12}X_{22}X_{32}$, and X_4 denotes the chimpanzee's movement towards the four attacking strategies of chimps, correspondingly.

In the presented method, two objective functions have been utilized for feature selection: the minimum number of features and the maximum overall accuracy (OA). The weighted sum has been used for integrating both main functions. Hence, the fitness function is represented as follows:

$$Fitness\ Funclion(i) = \alpha \cdot 0A(i) + (1-\alpha) \cdot log_{10}\left(\frac{N}{n(i)}\right) \tag{13}$$

In Equation (13), objective $Function(i)$ represents the fitness function of *i-th* chimps, $0A(i)$ denotes the total accuracy of *i-th* chimps, $N = 101$ features, and $n(i)$ indicates the number of features chosen at the *i-th* chimps. Moreover, α represents the weight parameter, which can be assumed to be 0.92. The calibration of α has been set by using the trial-and-error technique.

3.3. Intrusion Detection Using Optimal CCR-ELM Model

To detect intrusions in the IoT-WSN, the BCOA-MLID technique utilized the SCA with the CCR-ELM classification model in the ELM model. The input bias and weight of SLFNs can be randomly created [27]. An equal resultant matrix of hidden states was computed, concerning the resultant, weighted with some steps. Therefore, the computation cost of ELM was lower.

Assume that there are N various instances defined as (X_i, y_i), $i = 1,2,\ldots, N$. $X_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T 2R^n$ and $y_i = [y_{i1}, y_{i2}, \ldots, y_{im}]^T 2R^m$. Consider a_j and β_j to be the input and resultant, weighted correspondingly. b_j refers bias of hidden units. The SLFN with L hidden node can be modeled as:

$$\sum_{j=1}^{L} \beta_j g(a_j,\ b_j, X_i) = 0_i, i = 1,\ldots,N \tag{14}$$

where $g(\bullet)$ denotes the activation function and generally utilizes typical non-linear functions, such as radial basis functions, sigmoid, sine, etc. The error amongst evaluated output 0_i and the actual output y_i is zero if the SLFNs exactly estimate the data feature.

$$\sum_{j=1}^{L} \beta_j g(a_j,\ b_j, X_i) = y_i, i = 1,\ldots,N \tag{15}$$

Assume $\beta = [\beta_1^T, \ldots, \beta_L^T]^T$ and $Y = [y_{1^T}, \ldots, y_{N^T}]^T$. The above method is represented as $H\beta = Y$.

$$H = \begin{bmatrix} g(a_1, b_1, X_1) & \ldots & g(a_L, b_L, X_1) \\ \vdots & \ldots & \vdots \\ g(a_1, b_1, X_N) & \ldots & g(a_L, b_L, X_N) \end{bmatrix} \tag{16}$$

H is the supposed resultant matrix of the hidden state. h_{ij} signifies the resultant of jth hidden node equivalent to input X_i. In the trained procedure, the parameters of hidden nodes comprising a_j and b_j, could not be modified then primarily created. The equivalent resultant weighted can be evaluated as:

$$\hat{\beta} = \mathrm{H}^{\dagger}Y = \begin{cases} (\frac{I}{C} + H^T H)^{-1} H^T Y, \ L < N \\ H^T (\frac{I}{C} + H^T H)^{-1} Y, \ L \geq N \end{cases} \tag{17}$$

$\mathrm{H}^{\dagger}$ represents the Moore–Penrose generalization inverse of H. C denotes the pre-set parameter, intending to give a trade-off between minimizing the trained error and maximizing the marginal distance. I denotes the unit matrix. A better resultant weighted can be obtained with the minimized cost function $\|O - Y\|$.

After establishing class-specific regulation cost, CCR-ELM has been projected for solving the class imbalance issues. Two trade-off factors, comprising C^+ for minority positive instances and C^- for most negative instances, can be utilized for rebalancing both classes. Let the count of minority positive instances and most negative instances be formulated as l_1 and l_2, correspondingly. CCR-ELM was modeled as:

$$\min \left(\frac{1}{2}\|\beta\|^2 + \frac{1}{2}C^+ \sum_{i=1|y_{i=+1}}^{l_1} \xi_i^2 + \frac{1}{2}C^- \sum_{i=1|y_{i=-1}}^{l_2} \xi_i^2\right) \tag{18}$$

$$s{\cdot}t{\cdot}h(x_i)\beta = y_i - \xi_i, i = 1, \ldots N.$$

Equivalent resultant weighted $\hat{\beta}$ is calculated as:

$$\hat{\beta} = \mathrm{H}^{\dagger}Y = \begin{cases} (\frac{I}{C^+} + \frac{I}{C^-} + H^T H)^{-1} H^T Y, \ L < N \\ H^T (\frac{I}{C^+} + \frac{I}{C^-} + H^T H)^{-1} Y, \ L \geq N \end{cases} \tag{19}$$

To binary classifier issues, the decision function of the CCR-ELM-based classifier was $f(x) = sign\ h(x)\beta$.

$$f(x) = \begin{cases} sign\ h(x)(\frac{I}{C^+} + \frac{I}{C^-} + H^T H)^{-1} H^T Y, \ L < N \\ sign\ h(x) H^T (\frac{I}{C^+} + \frac{I}{C^-} + H^T H)^{-1} Y, \ L \geq N \end{cases} \tag{20}$$

In CCR-ELM, five key parameters contain direct features of the classifier accuracy, comprising the count of hidden nodes L, input weighted a_j, biases b_j, C^+ for minority positive instances, and C^- for most negative instances. The former three parameters determine the infrastructure of SLFNs and were generally pre-set by humans.

Finally, the SCA is applied to optimally choose the parameters related to the CCR-ELM classifier. SCA is a simple and versatile optimization algorithm that is capable of finding the global optimal solution in complex and noisy search spaces. Its robustness, fast convergence rate, and scalability make it a suitable algorithm for a wide range of optimization problems. The SCA creates several primary random solutions and appeals to them to shift nearby optimum solutions utilizing a mathematical method dependent upon sine and cosine functions [28]. For expressing the functions of SCA, a gathering of random variables can be utilized. Figure 2 illustrates the flowchart of SCA.

- The motion direction;
- The movement place;
- Emphasizing or de-emphasizing the target effect;
- Swapping amongst the sine and cosine elements.

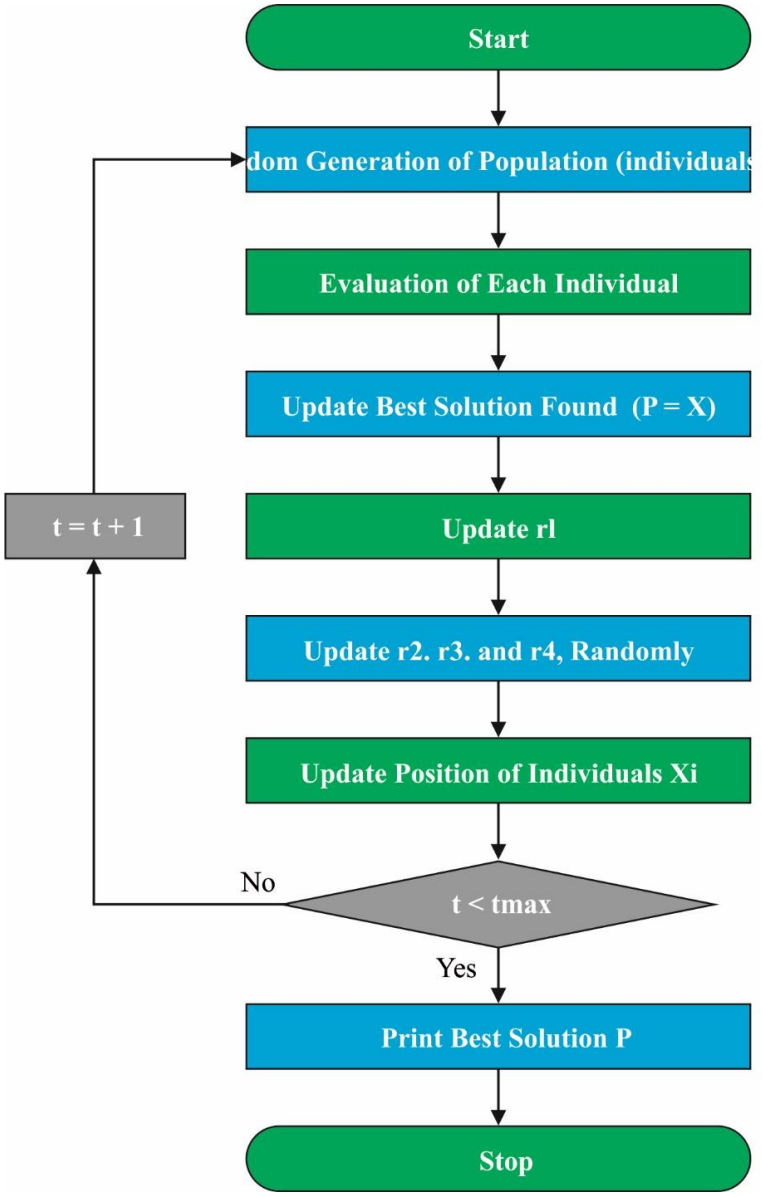

Figure 2. Flowchart of SCA.

The upgrade procedure of candidate solutions can be carried out utilizing the subsequent formula.

$$P(t+1) = \begin{cases} P(t) + r_5 \cdot \sin(r_6) \cdot |r_7 S^*(t) - S(t)| & r_4 < 0.5 \\ P(t) + r_5 \cdot \cos(r_6) \cdot |r_7 S^*(t) - S(t)| & r_4 \geq 0.5 \end{cases} \quad (21)$$

where t refers to the count of searching iterations. Present and better solutions can be indicated as S and S^*. The values of [0, 1] are assigned to random variables r_4, r_6, and r_7. For instance, it is seen in the formula that the places of optimum solutions control the present solution position, generating it more simply to obtaining an ideal solution. The value of r_4 was altered as follows in the running iterations of SCA.

$$r_4 = a - \frac{a \times t}{t_{max}} \quad (22)$$

where a represents the constant, and t and t_{max} signify the present and maximal iterations, correspondingly. The SCA technique is more resilient than a broad range of metaheuristic

techniques from the literature because it utilizes just one better solution to manage the other solution. Fitness selection becomes a vital factor in the SCA method. Solution encrypting was used to evaluate the accuracy of the candidate solution. Here, the accuracy value was the main condition utilized to modchip a fitness function.

$$Fitness = max\,(P) \tag{23}$$

$$P = \frac{TP}{TP + FP} \tag{24}$$

From the expression, FP denotes the false positive value and TP indicates the true positive.

4. Results and Discussion

In this section, the intrusion detection fallouts of the BCOA-MLID technique are examined using the WSN-DS dataset [29], which holds 374661 samples with 5 class labels as defined in Table 1. For experimental validation, we have used 80:20 and 70:30 of training/testing data.

Table 1. Details of the dataset.

Class	No. of Samples
Normal	340,066
Blackhole	10,049
Grayhole	14,596
Flooding	3312
Scheduling Attacks	6638
Total Number of Samples	374,661

The proposed model was simulated using Python 3.6.5 tool on a PC with i5-8600k CPU, GeForce 1050Ti 4 GB, 16 GB RAM, 250 GB SSD, and 1 TB HDD. The parameter settings are given as follows: learning rate: 0.01, dropout: 0.5, batch size: 5, epoch count: 50, and activation: ReLU.

In Figure 3, the confusion matrices of the BCOA-MLID technique are examined under distinct sizes of the Training Phase (TRP) and Testing Phase (TSP). The figures indicate that the BCOA-MLID technique categorizes the attacks and normal samples proficiently.

In Table 2, the entire results of the BCOA-MLID technique received under 80:20 of TRP/TSP are given. In Figure 4, the average intrusion detection results of the proposed model are illustrated under 80:20 of TRP/TSP. The results show that the BCOA-MLID technique reported improved results under every individual class. With 80% of TRP, the BCOA-MLID technique reaches an average $accu_y$ of 99.63%, $sens_y$ of 97.91%, $spec_y$ of 99.67%, F_{score} of 94.52%, and AUC_{score} of 98.79%. Concurrently, with 20% of TSP, the BCOA-MLID approach reaches an average $accu_y$ of 99.63%, $sens_y$ of 97.86%, $spec_y$ of 99.66%, F_{score} of 94.28%, and AUC_{score} of 98.76%.

Table 2. Classifier outcome of the BCOA-MLID approach on TRP/TSP of 80:20.

Class Labels	$Accu_y$	$Sens_y$	$Spec_y$	F_{score}	AUC_{score}
Training Phase (80%)					
Normal	99.19	99.19	99.21	99.55	99.20
Blackhole	99.75	98.44	99.79	95.46	99.11
Grayhole	99.71	98.45	99.76	96.38	99.11
Flooding	99.77	95.64	99.80	87.84	97.72
Scheduling Attacks	99.75	97.84	99.79	93.39	98.81
Average	99.63	97.91	99.67	94.52	98.79

Table 2. *Cont.*

Class Labels	$Accu_y$	$Sens_y$	$Spec_y$	F_{score}	AUC_{score}
Testing Phase (20%)					
Normal	99.17	99.17	99.16	99.54	99.16
Blackhole	99.77	98.72	99.80	95.87	99.26
Grayhole	99.73	98.45	99.78	96.53	99.11
Flooding	99.73	94.79	99.78	86.43	97.28
Scheduling Attacks	99.75	98.18	99.78	93.03	98.98
Average	99.63	97.86	99.66	94.28	98.76

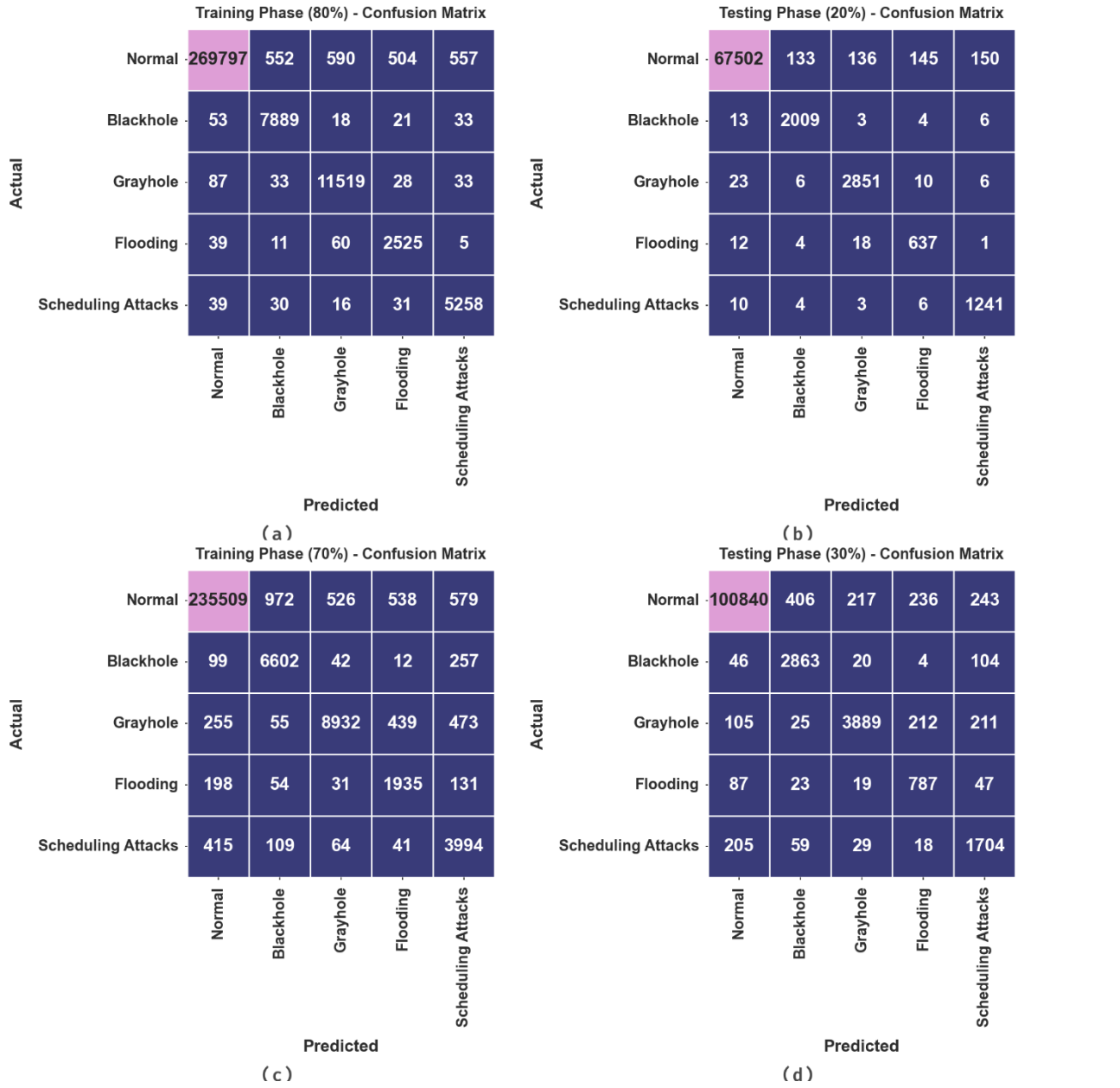

Figure 3. Confusion matrices of the BCOA-MLID approach (**a**,**b**) TRP/TSP of 80:20 and (**c**,**d**) TRP/TSP of 70:30.

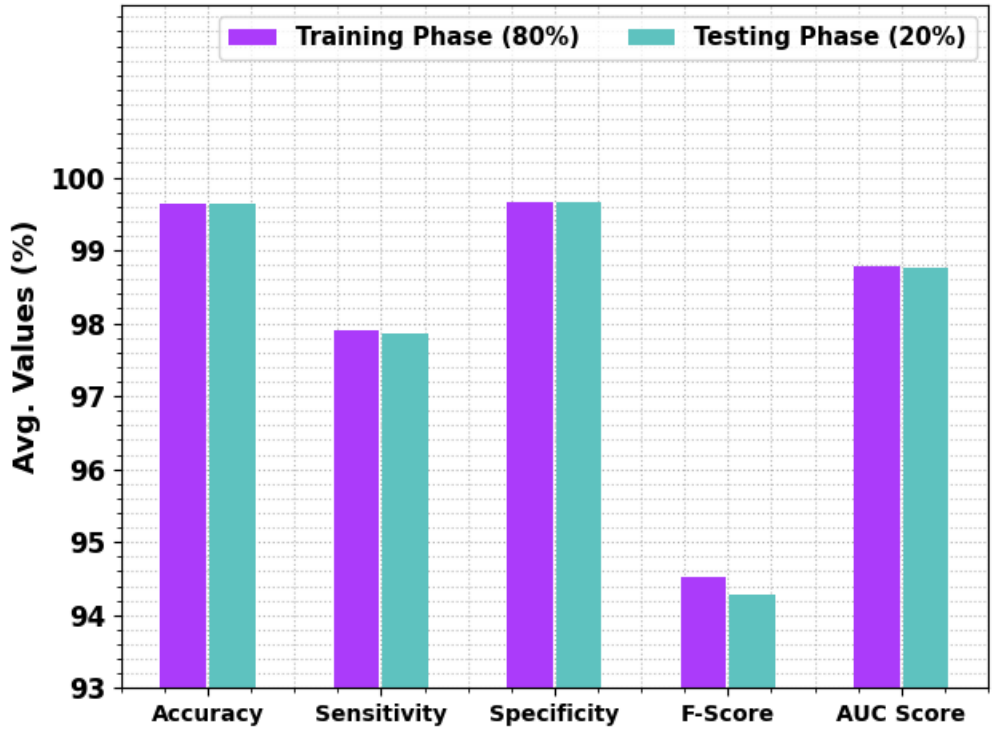

Figure 4. The average outcome of the BCOA-MLID approach on TRP/TSP of 80:20.

Table 3 shows the overall results of the BCOA-MLID technique obtained under 70:30 of TRP/TSP.

Figure 5 demonstrates the average classification outcomes of the BCOA-MLID technique are given under 70:30 of TRP/TSP. The results show that the BCOA-MLID algorithm reported improved results under every individual class. With 70% of TRP, the BCOA-MLID technique reaches an average $accu_y$ of 99.19%, $sens_y$ of 89.96%, $spec_y$ of 98.86%, F_{score} of 86.23%, and AUC_{score} of 94.41%. Concurrently, with 30% of TSP, the BCOA-MLID approach reaches an average $accu_y$ of 99.18%, $sens_y$ of 89.41%, $spec_y$ of 98.81%, F_{score} of 85.70%, and AUC_{score} of 94.11%.

Table 3. Classifier outcome of the BCOA-MLID approach on TRP/TSP of 70:30.

Class Labels	$Accu_y$	$Sens_y$	$Spec_y$	F_{score}	AUC_{score}
Training Phase (70%)					
Normal	98.63	98.90	95.99	99.25	97.45
Blackhole	99.39	94.15	99.53	89.19	96.84
Grayhole	99.28	87.97	99.74	90.46	93.85
Flooding	99.45	82.38	99.60	72.83	90.99
Scheduling Attacks	99.21	86.39	99.44	79.43	92.92
Average	99.19	89.96	98.86	86.23	94.41
Testing Phase (30%)					
Normal	98.63	98.92	95.76	99.24	97.34
Blackhole	99.39	94.27	99.53	89.29	96.90
Grayhole	99.25	87.55	99.74	90.27	93.64
Flooding	99.43	81.72	99.58	70.90	90.65
Scheduling Attacks	99.19	84.57	99.45	78.82	92.01
Average	99.18	89.41	98.81	85.70	94.11

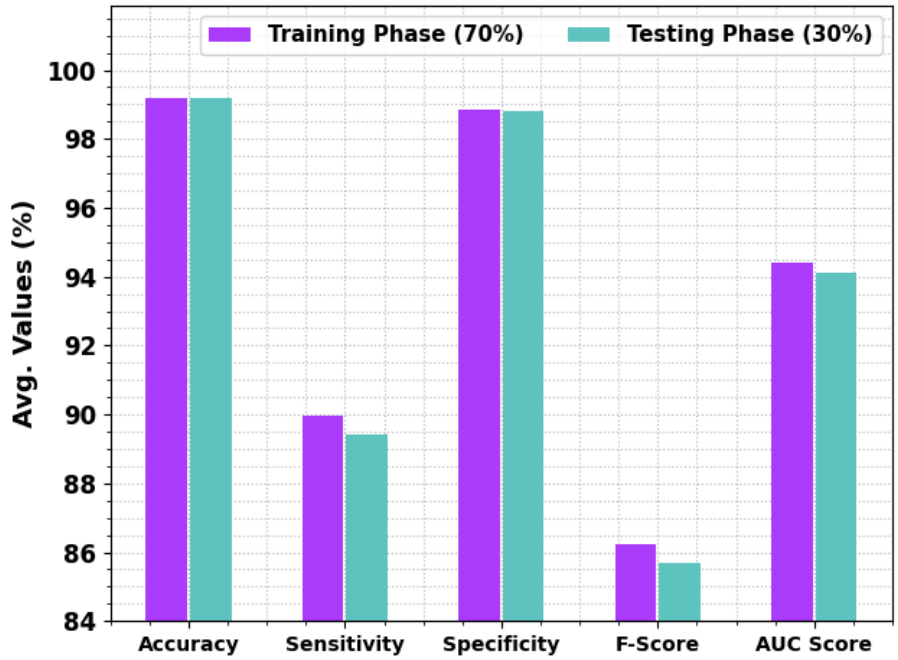

Figure 5. The average outcome of the BCOA-MLID approach on TRP/TSP of 70:30.

The TACY and VACY of the BCOA-MLID model were used to investigate the IoT-WSN detection performance in Figure 6. The figure shows that the BCOA-MLID model has shown improved performance with increased values of TACY and VACY. To be specific, the BCOA-MLID method has attained maximum TACY valued outcomes.

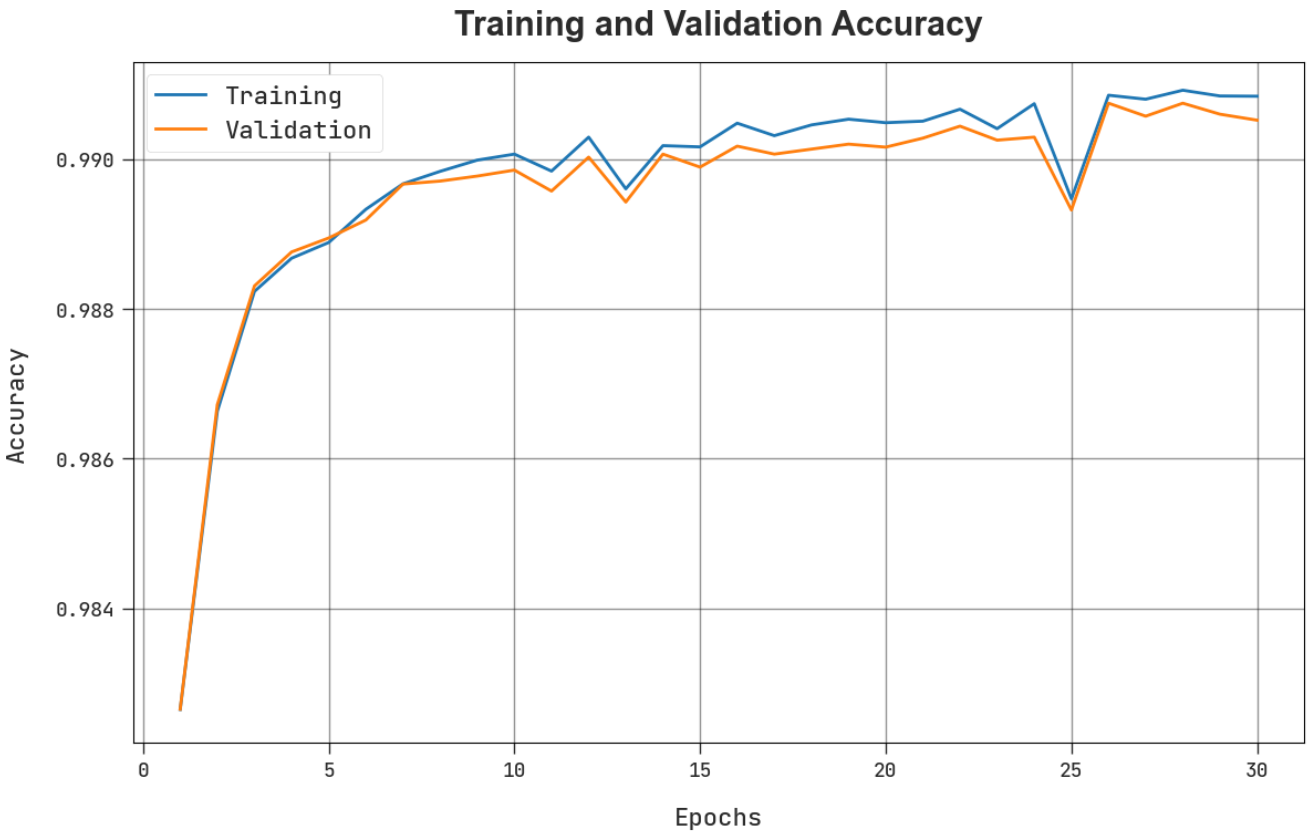

Figure 6. TACY and VACY outcome of the BCOA-MLID approach.

The TLOS and VLOS of the BCOA-MLID approach were tested on IoT-WSN detection performance in Figure 7. The figure shows that the BCOA-MLID approach has superior performance with menial values of TLOS and VLOS. The BCOA-MLID model has resulted in reduced VLOS-valued outcomes.

Figure 7. TLOS and VLOS outcome of the BCOA-MLID approach.

A brief, clear precision–recall analysis of the BCOA-MLID system under the test database is shown in Figure 8. The figure shows the BCOA-MLID approach has enhanced values of precision–recall values for each class label.

In Table 4, the classification results of the BCOA-MLID technique compared with recent methods are examined briefly [30,31]. The results indicate that the AdaBoost, GB, and KNN-PSO algorithms result in the worst performance compared other models. Next, the XGBoost model manages to demonstrate moderately improved results. Meanwhile, the KNN model results in somewhat considerable performance, with an $accu_y$ of 97.2%, $sens_y$ of 96.49%, $spec_y$ of 96.34%, and F_{score} of 90.23%. In contrast, the BCOA-MLID technique attains a maximum performance $accu_y$ of 99.63%, $sens_y$ of 97.91%, $spec_y$ of 99.67%, and F_{score} of 94.52%.

Table 4. Comparative outcome of the BCOA-MLID approach with recent systems [30,31].

Methods	$Accu_y$	$Sens_y$	$Spec_y$	F_{score}
BCOA-MLID	99.63	97.91	99.67	94.52
AdaBoost	95.69	95.77	95.00	90.31
GB Model	94.58	95.25	94.09	93.31
XGBoost	96.83	96.10	94.43	91.52
KNN-AOA	97.20	96.49	96.34	90.23
KNN-PSO	92.89	95.63	95.08	92.99

In Table 5 and Figure 9, the computation time (CT) outcomes of the BCOA-MLID technique compared with existing techniques are investigated. The experimental outcomes demonstrate that the AdaBoost, KNN, and KNN-PSO algorithms led to ineffectual results,

with higher CT values over other models. Moreover, the XGBoost model tried to exhibit somewhat reduced CT values. In addition, the BG model results in somewhat considerable performance, with a CT of 12.75 s. In contrast, the BCOA-MLID technique attains better results, with a lower CT of 7.26 s. These results ensure the improved detection performance the of BCOA-MLID technique in the IoT-WSN environment. The enhanced performance of the proposed model is due to the inclusion of BCOA for feature subset selection and SCA based parameter tuning.

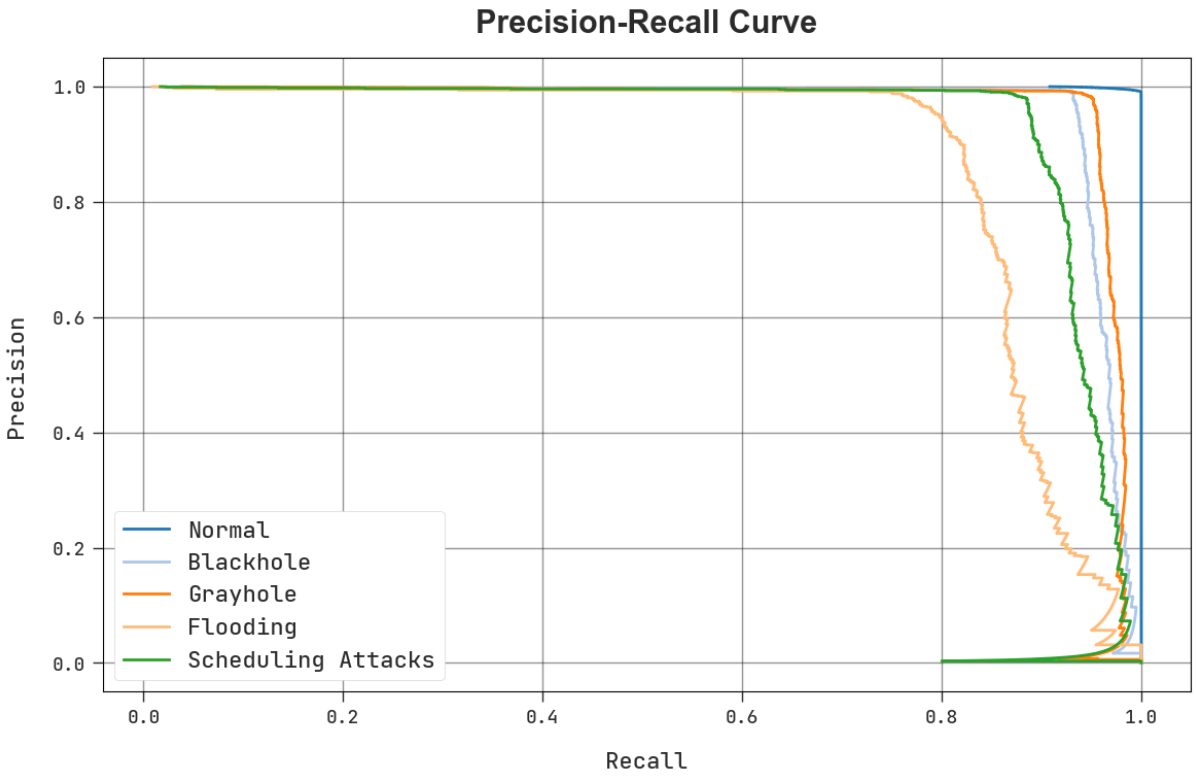

Figure 8. The precision-recall outcome of the BCOA-MLID approach.

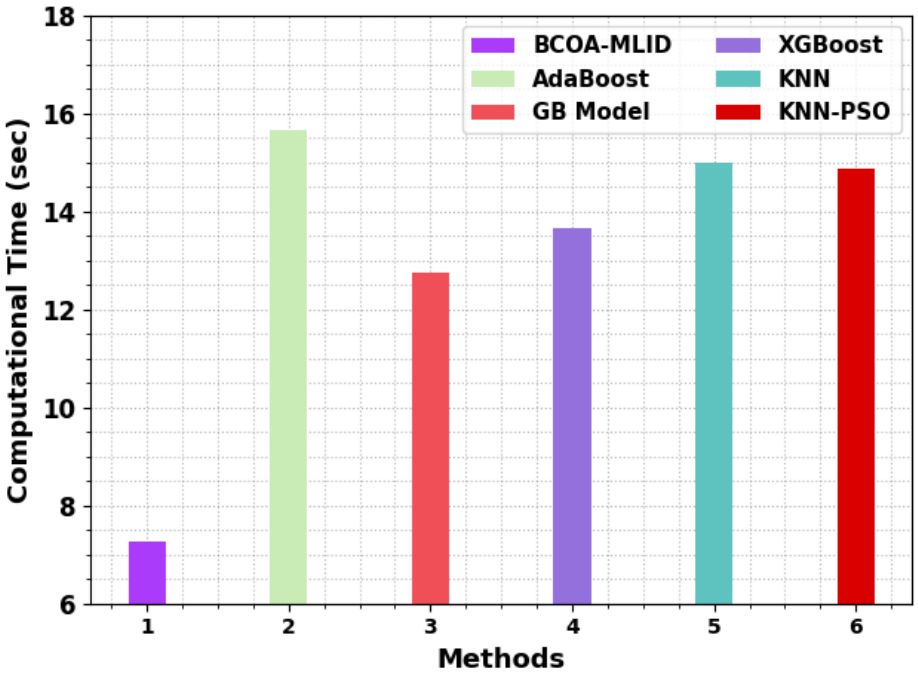

Figure 9. CT outcome of the BCOA-MLID approach with recent systems.

Table 5. CT outcome of the BCOA-MLID approach with recent systems.

Methods	Computational Time (s)
BCOA-MLID	7.26
AdaBoost	15.65
GB Model	12.75
XGBoost	13.67
KNN	15.01
KNN-PSO	14.87

5. Conclusions

In this article, an automated BCOA-MLID technique has been developed for accurate intrusion detection to accomplish security tasks in the IoT-WSN. The presented BCOA-MLID technique identifies intrusions using a series of processes: data normalization, BCOA-based feature subset selection, CCR-ELM classification, and SCA-based parameter tuning. The experimental result of the BCOA-MLID technique was tested on the Kaggle intrusion dataset, and the results showcase the significant outcomes of the BCOA-MLID technique with a maximum accuracy of 99.63%. In the future, the performance of the proposed technique can be improved by the use of an unsupervised or semi-supervised WSN intrusion detection model. These models will not only target a particular type of DoS attack, but also strive to cover Sybil attacks, routing attacks, and other possible attacks.

Author Contributions: Conceptualization, M.A.; Methodology, M.A.A.; Software, A.A.A. and S.D.; Validation, M.A. and M.K.S.; Formal analysis, H.M. and A.A.A.; Investigation, M.A. and S.D.; Resources, M.A.D.; Data curation, M.A.D. and A.A.A.; Writing—original draft, M.A., M.A.A., M.K.S., H.M., M.A.D. and S.A.; Writing—review & editing, M.K.S., H.M., A.A.A., S.D. and S.A.; Visualization, S.D.; Supervision, M.A.A.; Project administration, M.A.D.; Funding acquisition, M.A.A, M.A. and M.K.S. The manuscript was written through the contributions of all authors. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Large Groups Project under grant number (117/44). Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R330), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Research Supporting Project number (RSP2023R459), King Saud University, Riyadh, Saudi Arabia. This study is supported via funding from Prince Sattam bin Abdulaziz University project number (PSAU/2023/R/1444).

Institutional Review Board Statement: This article does not contain any studies with human participants performed by any of the authors.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing does not apply to this article as no datasets were generated during the current study.

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

References

1. Ramana, K.; Revathi, A.; Gayathri, A.; Jhaveri, R.H.; Narayana, C.L.; Kumar, B.N. WOGRU-IDS—An intelligent intrusion detection system for IoT-assisted Wireless Sensor Networks. *Comput. Commun.* **2022**, *196*, 195–206. [CrossRef]
2. Rajan, D.A.J.; Naganathan, E.R. Trust-based anonymous intrusion detection for cloud-assisted WSN-IOT. *Glob. Transit. Proc.* **2022**, *3*, 104–108. [CrossRef]
3. Ramana, T.V.; Thirunavukkarasan, M.; Mohammed, A.S.; Devarajan, G.G.; Nagarajan, S.M. Ambient intelligence approach: IoT-based decision performance analysis for intrusion detection. *Comput. Commun.* **2022**, *195*, 315–322. [CrossRef]
4. RM, B.; K Mewada, H.; BR, R. Hybrid machine learning approach based intrusion detection in the cloud: A metaheuristic assisted model. *Multiagent Grid Syst.* **2022**, *18*, 21–43.

5. Mohan, P.; Subramani, N.; Alotaibi, Y.; Alghamdi, S.; Khalaf, O.I.; Ulaganathan, S. Improved metaheuristics-based clustering with multihop routing protocol for underwater wireless sensor networks. *Sensors* **2022**, *22*, 1618. [CrossRef] [PubMed]
6. Abuqaddom, I.; Mahafzah, B.; Faris, H. Oriented stochastic loss descent algorithm to train very deep multi-layer neural networks without vanishing gradients. *Knowl. Based Syst.* **2021**, *230*, 107391. [CrossRef]
7. Al-Shaikh, A.; Mahafzah, B.; Alshraideh, M. Hybrid harmony search algorithm for social network contact tracing of COVID-19. *Soft Comput.* **2021**, *27*, 3343–3365. [CrossRef]
8. Quincozes, S.E.; Passos, D.; Albuquerque, C.; Mossé, D.; Ochi, L.S. An extended assessment of metaheuristics-based feature selection for intrusion detection in CPS perception layer. *Ann. Telecommun.* **2022**, *77*, 457–471. [CrossRef]
9. Tabash, M.; Abd Allah, M.; Tawfik, B. Intrusion detection model using naive bayes and deep learning technique. *Int. Arab J. Inf. Technol.* **2020**, *17*, 215–224. [CrossRef]
10. Fatani, A.; Abd Elaziz, M.; Dahou, A.; Al-Qaness, M.A.; Lu, S. IoT intrusion detection system using deep learning and enhanced transient search optimization. *IEEE Access* **2021**, *9*, 123448–123464. [CrossRef]
11. Qaiwmchi, N.A.H.; Amintoosi, H.; Mohajerzadeh, A. Intrusion detection system based on gradient-corrected online sequential extreme learning machine. *IEEE Access* **2020**, *9*, 4983–4999. [CrossRef]
12. Pandey, J.K.; Kumar, S.; Lamin, M.; Gupta, S.; Dubey, R.K.; Sammy, F. A Metaheuristic Autoencoder Deep Learning Model for Intrusion Detector System. *Math. Probl. Eng.* **2022**, *2022*, 3859155. [CrossRef]
13. Almomani, O. A hybrid model using bio-inspired metaheuristic algorithms for network intrusion detection system. *Comput. Mater. Contin* **2021**, *68*, 409–429. [CrossRef]
14. Kagade, R.B.; Jayagopalan, S. Optimization-assisted DL-based intrusion detection system in a wireless sensor network with two-tier trust evaluation. *Int. J. Netw. Manag.* **2022**, *32*, e2196. [CrossRef]
15. Krishnan, R.; Krishnan, R.S.; Robinson, Y.H.; Julie, E.G.; Long, H.V.; Sangeetha, A.; Subramanian, M.; Kumar, R. An intrusion detection and prevention protocol for internet of things based wireless sensor networks. *Wirel. Pers. Commun.* **2022**, *124*, 3461–3483. [CrossRef]
16. Yadav, N.; Pande, S.; Khamparia, A.; Gupta, D. Intrusion detection system on IoT with 5G network using deep learning. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 9304689. [CrossRef]
17. Zhiqiang, L.; Mohiuddin, G.; Jiangbin, Z.; Asim, M.; Sifei, W. Intrusion detection in wireless sensor network using enhanced empirical-based component analysis. *Future Gener. Comput. Syst.* **2022**, *135*, 181–193. [CrossRef]
18. Muruganandam, S.; Joshi, R.; Suresh, P.; Balakrishna, N.; Kishore, K.H.; Manikanthan, S.V. A deep learning-based feed-forward artificial neural network to predict the K-barriers for intrusion detection using a wireless sensor network. *Meas. Sens.* **2022**, *25*, 100613. [CrossRef]
19. Subramani, S.; Selvi, M. Multi-objective PSO-based feature selection for intrusion detection in IoT-based wireless sensor networks. *Optik* **2023**, *273*, 170419. [CrossRef]
20. Saba, T.; Rehman, A.; Sadad, T.; Kolivand, H.; Bahaj, S.A. Anomaly-based intrusion detection system for IoT networks through deep learning model. *Comput. Electr. Eng.* **2022**, *99*, 107810. [CrossRef]
21. Sadeghi, F.; Larijani, A.; Rostami, O.; Martín, D.; Hajirahimi, P. A Novel Multi-Objective Binary Chimp Optimization Algorithm for Optimal Feature Selection: Application of Deep-Learning-Based Approaches for SAR Image Classification. *Sensors* **2023**, *23*, 1180. [CrossRef]
22. Henry, A.; Gautam, S.; Khanna, S.; Rabie, K.; Shongwe, T.; Bhattacharya, P.; Sharma, B.; Chowdhury, S. Composition of Hybrid Deep Learning Model and Feature Optimization for Intrusion Detection System. *Sensors* **2023**, *23*, 890. [CrossRef] [PubMed]
23. Balla, A.; Habaebi, M.H.; Elsheikh, E.A.A.; Islam, M.R.; Suliman, F.M. The Effect of Dataset Imbalance on the Performance of SCADA Intrusion Detection Systems. *Sensors* **2023**, *23*, 758. [CrossRef] [PubMed]
24. Abosata, N.; Al-Rubaye, S.; Inalhan, G. Customised Intrusion Detection for an Industrial IoT Heterogeneous Network Based on Machine Learning Algorithms Called FTL-CID. *Sensors* **2023**, *23*, 321. [CrossRef] [PubMed]
25. Salman, E.H.; Taher, M.A.; Hammadi, Y.I.; Mahmood, O.A.; Muthanna, A.; Koucheryavy, A. An Anomaly Intrusion Detection for High-Density Internet of Things Wireless Communication Network Based Deep Learning Algorithms. *Sensors* **2023**, *23*, 206. [CrossRef]
26. Hafeez, G.; Khan, I.; Jan, S.; Shah, I.A.; Khan, F.A.; Derhab, A. A novel hybrid load forecasting framework with intelligent feature engineering and optimization algorithm in smart grid. *Appl. Energy* **2021**, *299*, 117178. [CrossRef]
27. Cheng, J.; Chen, J.; Guo, Y.N.; Cheng, S.; Yang, L.; Zhang, P. Adaptive CCR-ELM with variable-length brainstorm optimization algorithm for class-imbalance learning. *Nat. Comput.* **2021**, *20*, 11–22. [CrossRef]
28. Abdelhamid, A.A.; El-Kenawy, E.S.M.; Khodadadi, N.; Mirjalili, S.; Khafaga, D.S.; Alharbi, A.H.; Ibrahim, A.; Eid, M.M.; Saber, M. Classification of monkeypox images based on transfer learning and the Al-Biruni Earth Radius Optimization algorithm. *Mathematics* **2022**, *10*, 3614. [CrossRef]
29. Almomani, I.; Al-Kasasbeh, B.; Al-Akhras, M. WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks. *J. Sens.* **2016**, *2016*, 4731953. [CrossRef]

30. Alqahtani, M.; Gumaei, A.; Mathkour, H.; Maher Ben Ismail, M. A genetic-based extreme gradient boosting model for detecting intrusions in wireless sensor networks. *Sensors* **2019**, *19*, 4383. [CrossRef]
31. Liu, G.; Zhao, H.; Fan, F.; Liu, G.; Xu, Q.; Nazir, S. An enhanced intrusion detection model based on improved kNN in WSNs. *Sensors* **2022**, *22*, 1407. [CrossRef] [PubMed]