AIMS Mathematics

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

AIMS Mathematics, 9(7): 17676–17695.
DOI: 10.3934/math.2024859
Received: 05 March 2024
Revised: 23 April 2024
Accepted: 25 April 2024
Published: 22 May 2024

Research article

Weighted salp swarm algorithm with deep learning-powered cyber-threat detection for robust network security

Maha M. Althobaiti[1] **and José Escorcia-Gutierrez**[2,*]

[1] Department of Computer Science, College of Computing and Information Technology, Taif University, Taif, 21944, Saudi Arabia

[2] Department of Computational Science and Electronics, Universidad de la Costa, CUC, Barranquilla, 080002, Colombia

* **Correspondence:** Email: jescorci56@cuc.edu.co; Tel: +573005864207.

Abstract: The fast development of the internet of things has been associated with the complex worldwide problem of protecting interconnected devices and networks. The protection of cyber security is becoming increasingly complicated due to the enormous growth in computer connectivity and the number of new applications related to computers. Consequently, emerging intrusion detection systems could execute a potential cyber security function to identify attacks and variations in computer networks. An efficient data-driven intrusion detection system can be generated utilizing artificial intelligence, especially machine learning methods. Deep learning methods offer advanced methodologies for identifying abnormalities in network traffic efficiently. Therefore, this article introduced a weighted salp swarm algorithm with deep learning-powered cyber-threat detection and classification (WSSADL-CTDC) technique for robust network security, with the aim of detecting the presence of cyber threats, keeping networks secure using metaheuristics with deep learning models, and implementing a min-max normalization approach to scale the data into a uniform format to accomplish this. In addition, the WSSADL-CTDC technique applied the shuffled frog leap algorithm (SFLA) to elect an optimum subset of features and applied a hybrid convolutional autoencoder (CAE) model for cyber threat detection and classification. A WSSA-based hyperparameter tuning method can be employed to enhance the detection performance of the CAE model. The simulation results of the WSSADL-CTDC system were examined in the benchmark dataset. The extensive analysis of the accuracy of the results found that the WSSADL-CTDC technique exhibited a better value of 99.13% than comparable methods on different measures.

Keywords: cyber threat; intrusion detection systems; salp swarm algorithm; network security, convolutional autoencoder
Mathematics Subject Classification: 68T05, 68T40, 90C59

1. Introduction

The continuous development and wide-ranging utilization of the internet benefits several network users from several points of view [1]. In the meantime, security is highly significant with the vast exploitation of the network. Network security has undoubtedly been connected with programs, different information, networks, computers, and more, where the need for defense is to avoid adaptation and unauthorized access [2]. However, the increasing number of systems connected to the internet in e-Commerce, military and finance make them targets of network attacks, contributing to a lot of damage and risks [3]. Providing efficient approaches for identifying and protecting from attacks and maintaining network security is essential. Additionally, various categories of attacks have been frequently needed and processed in numerous ways [4]. Recognizing various types of network attack is the main problem that needs to be solved in network security, particularly those that have not been noticed. Researchers have recently employed different types of machine learning (ML) techniques for classifying network attacks without previous knowledge of their comprehensive features [5].

Internet of things (IoT) networks have continuously exposed and modified the topology by connecting nodes and exiting the network in real time [6]. The lack of centralized network control tools makes them susceptible to security attacks. IoT devices can have superior features, including limited data storage, connection bandwidth, smaller memory size, and restricted power supply [7]. These limitations considerably affect the efficiency of security protocols for IoT platforms concerning development and effectiveness. Consequently, a competent intrusion detection system (IDS) for IoT networks will probably emerge due to the lack of the computational power needed [8]. Cyberattacks are highly difficult to recognize, as hackers utilize cutting-edge methods to take sensitive data while avoiding identification by IDS. The interaction between internetworks has also been subject to cyber-security vulnerabilities. Consequently, a novel technique can be essential for attack prevention activities and earlier intrusion detection [9].

Machine learning and deep learning (DL) approaches are currently used for intrusion detection, network abnormality detection, and avoidance. The DL technique should be utilized to increase the accuracy and proficiency of IDS in IoT devices [10]. Since hyperparameter tuning of the DL model remains a challenging process, metaheuristic algorithms can be used, which in turn improves the overall detection performance. Some of the recently developed metaheuristics are Liver cancer algorithm (LCA) [11], slime mould algorithm (SMA) [12, 13], moth search algorithm (MSA) [14], hunger games search (HGS) [15], Runge Kutta method (RUN) [16], colony predation algorithm (CPA) [17], weighted mean of vectors (INFO) [18], Harris Hawks optimization (HHO) [19], Rime optimization algorithm (RIME) [20, 21], etc.

This article presents a weighted Salp swarm algorithm with a DL-powered cyber threat detection and classification (WSSADL-CTDC) technique for robust network security, utilizing a min-max normalization approach to scale data into even formats. In addition, it applies a shuffled frog leap algorithm (SFLA) to elect an optimum subset of features. The WSSADL-CTDC technique applies a hybrid convolutional autoencoder (CAE) model to identify and classify cyber threats. A WSSA-based hyperparameter tuning process can be employed to enhance the detection performance of the CAE model. The experimental analysis can be examined on a benchmark dataset.

- Uses SFLA to choose an optimum feature subset from the network security dataset. This process is crucial to mitigate the input data's dimensionality, enhance the technique's effectualness, and concentrate on the most appropriate features for recognizing cyber threats.

- Proposes a novel fusion CAE technique to detect and classify cyber threats within network traffic data. This method incorporates the robustness of convolutional neural networks (CNNs) and autoencoders (AEs), allowing for productive feature extraction and anomaly recognition in network data.
- The WSSADL-CTDC model augments the accomplishment of the CAE technique by implementing a WSSA-based hyperparameter tuning procedure.

The remaining sections of the article are arranged as follows: Section 2 offers the literature review, and Section 3 represents the proposed method. Then, Section 4 elaborates on the evaluation of the results, and Section 5 completes the work.

2. Related works

In [22], an ensemble DL algorithm was developed that implements the gains of the long short-term memory (LSTM) and AE technique. The LSTM was implemented to create an architecture on a standard time series of data. Khan et al. [23] projected an AE-based identification model using recurrent and convolutional networks. A two-phase sliding window (SW) was employed. Malicious activities in the raw time series were converted to fixed-length series using a first-phase SW. All series were transformed into constant time-dependent sub-series through additional small SW. In classification, fully connected networks employ the removed temporal-spatial features. Bibi et al. [24] developed a well-organized, self-learning, autonomous multi-vector attack intelligence and identification method. An innovative convolutional LSTM2D technique powered by the computational integrated device model (ConvLSTM2D) is developed. Das et al. [25] introduced a novel transient search optimization (TSO) method enabled by the optimum stacked sparse AE (OSSAE) method. This technique followed the TSA-based feature selection method. Furthermore, the Transient search optimization-optimum stacked sparse AE (TSA-OSSAE) method implements the SSAE method. Lastly, the hyperparameters of the SSAE technique were selected using a multi-versus optimizer (MVO) approach.

In [26], the authors presented a graph pattern based on a technique equivalent to a deep hunter graph neural network (GNN) model. A GNN model was developed mainly with two new networks: feature embedding networks integrated with the indicators of compromise (IOC) data and graph embedding networks that took connections among IOCs. Ndife et al. [27] developed a Bayesian neural networks (BNNs) model called SGtechNet. Spatio-temporal feature engineering and uncertainty evaluation in Bayesian modeling are exploited. In [28], a two-stage IDS was developed. Initially, an adversarial training method was introduced employing the generative adversarial networks (GAN) method. In addition, a DL method was presented for the secondary detector to recognize intrusions. In [29], an IDS depended on DL, and a DL method for IDS was developed, employing the efficient recurrent neural networks (ERNN) method. Additionally, the effectiveness of the system in binary and multi-class classifications, as well as the count of neurons and diverse learning rate, has implications under the efficiency of the developed method. This technique relates the method to recurrent neural networks (RNN) and support vector machine (SVM) models.

Zhou et al. [30] proposed a model to build a continuous leakage-resilient identity-based encryption (IBE) model with tight security. In [31], a bioinspired motion-sensitive technique is presented, which comprises spatial and temporal perception of motion cues, employing simple visual signal processing. Li et al. [32] presented a model employing the multi-scale radial basis function (MRBF) and Fisher vector (FV) encoding. The high-resolution time-frequency approach gray level co-ocurrence matrix (GLCM) texture descriptors and FV extraction are also utilized. Chen et al. [33] introduced a Lung-Dense attention network (LDANet) method. The residual spatial attention (RSA) method is employed to weigh spatial data, and the weight of every channel is calibrated utilizing the gated channel attention (GCA) technique. A dual attention guidance module (DAGM) technique is constructed for maximization. A lightweight dense block (LDB) technique is also utilized. In [34], a deep neural network (DNN) model is proposed, which excels in a localized and sparse approximation. Savanovi et al. [35] proposed an optimized ML model employing an altered firefly model. In [36], a novel feature selection (FS) model is introduced, which improves the gorilla troops optimizer (GTO) by incorporating the bird swarm modeling (BSA) technique. Jovanovic et al. [37] proposed an improved firefly model to tune the extreme gradient boosting (XGBoost) technique.

3. Proposed method

His article introduces an innovative method for robust network security WSSADL-CTDC. The technique aims to detect the presence of cyber threats and accomplish network security using metaheuristics with DL models. Figure 1 demonstrates the entire procedure of this approach.

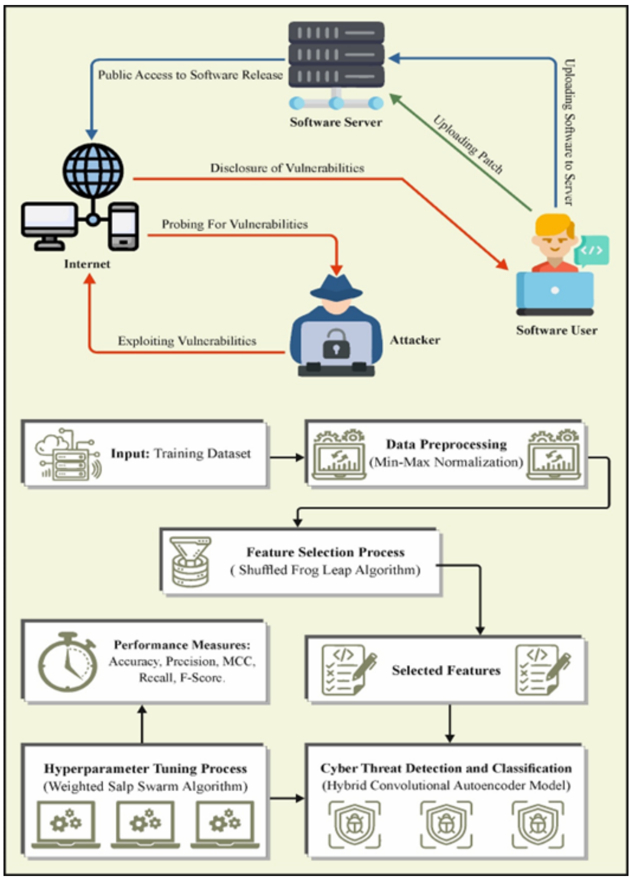
Public Access to Software Release
Software Server
Uploading Software to Server
Uploading Patch
Disclosure of Vulnerabilities
Internet
Probing For Vulnerabilities
Software User
Exploiting Vulnerabilities
Attacker
Input: Training Dataset
Data Preprocessing (Min-Max Normalization)
Feature Selection Process (Shuffled Frog Leap Algorithm)
Performance Measures: Accuracy, Precision, MCC, Recall, F-Score.
Selected Features
Hyperparameter Tuning Process (Weighted Salp Swarm Algorithm)
Cyber Threat Detection and Classification (Hybrid Convolutional Autoencoder Model)

Figure 1. Overall process of the WSSADL-CTDC technique.

3.1. Min-max normalization

Primarily, the technique WSSADL-CTDC employs a min-max normalization approach to normalize data [38]. In this study, the input data is standardized from the ranges 0 and +1 utilizing the specified Eq (3.1),

$$X_n = \frac{X_i - X_{min}}{X_{max} - X_{min}}. \tag{3.1}$$

Here, X_i is the observed input value, X_{min} and X_{max} refer to the maximum and minimum values of the descriptive variables, and X_n represents the normalized input values at intervals of $[-1, +1]$.

3.2. Feature selection

At this stage, the WSSADL-CTDC technique applies SFLA to elect an optimum feature subclass. SFLA has a new population-based metaheuristic approach [39] that Integrates the social behaviors of the particle swarm optimization (PSO) technique and the benefits of the memetic algorithm [40]. In this approach, a population can be a group of frogs that find optimum food by employing search strategies relevant to the PSO technique. The search process is performed by alternating frogs' intra- and inter-cluster transmission to find food. Intra-cluster transmission is performed within the memeplex for local invention, and inter-cluster transmission is performed among the frogs belonging to the dissimilar memeplexes for global exploration. In classical SFLA, consider that the early population was randomly generated with a P number of frogs. Evaluate fitness values for every individual frog. Subsequently, arrange P in descending sequence of the fitness values. The frogs are then distributed into an M count of memeplexes, and every memeplex has N frogs. Here, the frog distribution can be performed thus: The initial frog moves to the initial memeplex, the next frog moves to the initial memeplex, frog M goes to the M^{th} memeplex and frog $M + 1$ goes to the initial memeplex, up to the last frog. In every memeplex, X_b and X_w are the best and worst frogs based on fitness. X_g represents the frog with global fitness X. The position of the worst frog has been updated according to the random position or position of the global or local best frog so that the frog can move to the optimum solution. This can be mathematically modeled as the set of Eqs (3.2) and (3.3).

$$X_w(new) = X_w + Stp, \tag{3.2}$$

$$Stp = rand(X) \times (X_b - X_w) - Stp_{max} \leq Stp. \tag{3.3}$$

In Eq (3.3), Stp denotes the size of the leaping step of the frog with $[-1, 1]$. $rand()$ represents the random integer. If $X_w(new)$ generates the best solution, it replaces X_w. Otherwise, the computations in Eqs (3.2) and (3.3) are reiterated by replacing X_b with X_g. The new random solution can be replaced with X_w if there is no improvement. Thus, each memeplex is shuffled together to exchange data and generate a new population for the following search ranges.

In this SFLA method, the objectives could be incorporated into a single objective equation, such as a preset weight that recognizes the significance of all the objectives [41]. In this study, an fitness function (FF) is implemented to integrate these FS objectives, as represented in Eq (3.4),

$$Fitness(X) = \alpha \cdot E(X) + \beta * \left(1 - \frac{|R|}{|N|}\right). \tag{3.4}$$

Here, $Fitness(X)$ denotes the fitness value of X, $|R|$ and $|N|$ which have the number of particular features α, and β describes the weights of the classification error. $E(X)$ signifies the classification error rate employing the preferred features in the X subset, as well as the number of new features in the dataset and the reduction ratio $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$.

3.3. Classification using CAE model

The WSSADL-CTDC technique applies a hybrid CAE model for the detection and classification of cyber threats. AE is a self-directed learning procedure that utilizes a neural network (NN) for representation learning [42]. It has a method for a model to learn how to encrypt input data. AEs have been used to map input data to low-dimension spaces or representations of reduced areas. To fix this, a bottleneck is presented that implements a method to learn the condensed area of the input data. Figure 2 illustrates the CAE infrastructure.

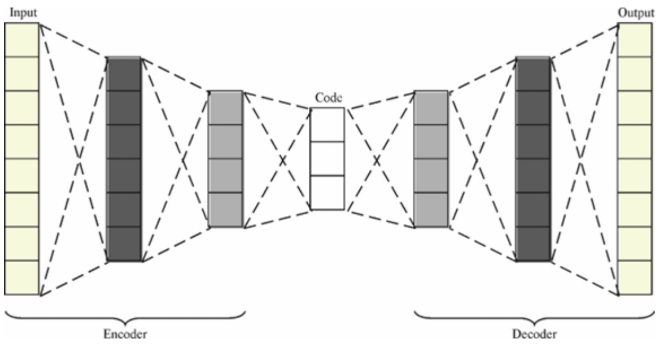

Figure 2. Structure of CAE.

An AE includes four modules: bottleneck layer (BL), encoder network (EN), reconstruction loss (RL), and decoder network (DN). EN is an NN that encrypts input data in a condensed area. The BL is the latter layer; the output is called encoded input data. N layers in an EN in which the latter layer (i.e., N^{th} layer) is BL. The arithmetical calculation used to signify the functioning of every layer existing in the EN can be revealed in Eq (3.5),

$$X_{e_i+1} = f_{ei}\Big(WV_{e_i}^T X_{e_i} + b_{e_i}\Big), \quad \forall i = 0, 1, 2, \ldots, N, \tag{3.5}$$

where X_{e_i} denotes the input for the EN layer i^{th}, f_{e_i} denotes the activation function for the layer i^{th}, b_{e_i} is the bias for the layer i^{th}, WV_{e_i} signifies the weight vector for the i^{th} layer, and X_{e_i+1} refers to the output of the layer i^{th}. DN is also known as an NN that consumes the output of the BL as input and attempts to renovate the original data. The number of layers in DN is similar to the number in the EN, in the opposite order. The last layer of the DN offers a loud regeneration of input data. The scientific expression used to signify the functioning of every layer existing in the DN is presented in Eq (3.6),

$$X_{d_i+1} = f_{d_i}\Big(W_{d_i}^T X_{d_i} + b_{d_j}\Big), \quad \forall i = 0, 1, 2, \ldots, N, \tag{3.6}$$

where X_{d_i} denotes the input for the DN layer i^{th}, X refers to the output of the layer i^{th}, f_{d_i} refers to the activation function for the i^{th} layer, b_{d_i} represents the bias for the i^{th} layer, and WV_{d_i} signifies the weight vector for the layer i^{th}. The dissimilarity between the original data X^O and the renovated data X^R is termed RL. The AE is trained by employing the backpropagation algorithm to reduce the RL.

Mean squared error (MSE) and binary cross-entropy (BCE) losses are dual, highly employed loss functions to calculate RL. These dual loss function formulas have been exposed in Eqs (3.7) and (3.8), while D denotes the number of samples in a dataset used by the AE,

$$MSE\left(X^0, X^R\right)= \frac{1}{D}\sum_{j=1}^{D}\left(X_j^0 - X_j^R\right)^2, \tag{3.7}$$

$$BCE\left(X^0, X^R\right)= -\frac{1}{D}\sum_{j=1}^{D} X_j^0 \cdot \log X_j^0 + \left(1 - X_j^0\right)\cdot \log\left(1 - X_j^R\right). \tag{3.8}$$

3.4. WSSA-based hyperparameter tuning

Eventually, the WSSA-based hyperparameter tuning process can be employed. SSA is a recent metaheuristic optimization technique introduced by Mirjalili et al. [43]. SSA must be stimulated by the navigation and swarming behaviors of salps in the oceans for food. The individual salp belongs to the Salpidae family and has a bottle-shaped body. Salps form chains, similar to flocks or swarms, in the deep ocean for foraging and navigation. The leader and followers are two levels of the salp chain. During navigation, the leaders are guided by the swarm (followers) in a multidimensional search range, resulting in the optimum solution (food source) for the optimizer problems. SSA begins with a random solution and later reinvents the search for optimum fitness of salps by exploiting and exploring the search range. The salp moves in the N-dimension search range and updates the leader location according to the distance between a food source and the salp as Eq (3.9),

$$x_i^1 = \begin{cases} F_i + r_1\left((ub_i - lb_i) * r_2 + lb_i\right), & r_3 \geq 0, \\ F_i + r_1\left((ub_i - lb_i) * r_2 + lb_i\right), & r_3 < 0. \end{cases} \tag{3.9}$$

- X_i^1 - denotes the leader's location in the i^{th} location.
- ub_i, lb_i - shows the upper and lower bounds of i^{th} location.
- F_i - represents the food source location in i^{th} location.
- r_1 parameter - balances the exploration and exploitation in searching space.
- r_1, r_2, r_3 - refer to random numbers (see Eq (3.10)), where L refers to the maximal iteration count, and l denotes the existing iteration,

$$r_1 = 2e^{\left(\frac{4l}{L}\right)^2}. \tag{3.10}$$

Newton's law of motion updates the follower's location in Eq (3.11),

$$x_i^j = \frac{1}{2}at^2 + V_0 t, \tag{3.11}$$

- v_0 represents the initial speed;
- x_i^j denotes the location of the j^{th} follower at the i^{th} dimension;
- t is the iteration;

$$x_i^j = \frac{1}{2}\left(x_i^j + x_i^{j-1}\right). \tag{3.12}$$

In SSA, salp forms chains for the foraging process. Salp navigates through the jet propulsion process. Based on the available information from the neighborhood, the leader updates the location with food sources. In a typical SSA approach, the follower salp updates the area according to Newton's law of motion. The model's ability to find solutions for the global search process is compromised due to its lost inertia parameter. Inertia controls the speed and direction of the object. However, such an approach only works well for multimodal and unimodal issues of smaller dimensions. However, this provides poor outcomes for high-dimension difficulties, representing poor convergence. Thus, the location update formula should be modified in all iterations to achieve the optimum solution.

Here, a WSSA has been introduced according to weighted distance location updating to enhance the efficiency of traditional SSA. This model boosts the variety of the original SSA. The weighted location updating parameter was developed based on the findings of Shi and Eberhart (1998). The author initially established the idea of inertia weight and stated that it provides the best control through exploitation and exploration processes. The position updating approach can be adapted as a weighted sum of optimum positions rather than implementing averages to improve the search capability of a typical SSA. This approach can be obtained by evaluating the weight according to the coefficient vectors C_1, C_2, and C_3. Location updating is modeled as the set of Eqs (3.13) to (3.16),

$$w_1 = \overrightarrow{C_1} * \overrightarrow{C_2}, \tag{3.13}$$

$$r_2 = \overrightarrow{C_1} * \overrightarrow{C_3}, \tag{3.14}$$

$$x_i^1 = \begin{cases} w_1 * F_i + r_1\Big((ub_i - lb_i) * r_2 + lb_i\Big), & r_3 \geq 0, \\ w_2 * F_i + r_1\Big((ub_i - lb_i) * r_2 + lb_i\Big), & r_3 < 0, \end{cases} \tag{3.15}$$

$$x_i^j = \frac{1}{2(w_1 + w_2)}\Big(w_2 * x_i^j + w_1 * x_i^{j-1}\Big). \tag{3.16}$$

This weighted location-update approach balances searching agents' exploitation and exploration capabilities. Furthermore, it improves the convergence rate and leads to better SSA performance.

The WSSA algorithm has obtained an FF to achieve increased classification effectiveness. It evaluates a positive integer to denote the higher efficiency of the candidate solutions. The decrease in the classifier error rate can be measured as FF, as shown in Eq (3.17),

$$fitness(x_i) = Classifier\ Error\ Rate = 100 * \frac{Number\ of\ Misclassified\ Instances}{Total\ Number\ of\ Instances}. \tag{3.17}$$

4. Performance validation

The performance analysis of the WSSADL-CTDC system can be examined using the network-based detection of IoT botnet attacks (N-BaIoT) dataset [44]. It includes 11 classes with 1000 samples for each class, as defined in Table 1. The suggested technique is simulated using the Python 3.6.5 tool on PC i5-8600k, 250GB SSD, GeForce 1050Ti 4GB, 16GB RAM, and 1TB HDD. The parameter settings are provided: learning rate: 0.01, activation: ReLU, epoch count: 50, dropout: 0.5, and batch size: 5.

Figure 3 shows the confusion matrices acquired by the WSSADL-CTDC system with 80:20 and 70:30 of true positive rate per hour / true negative rate per hour (TRPH/TSPH) under the N-BaIoT dataset. The completed findings show effective recognition with 11 classes.

Table 1. N-BaIoT dataset.

Class	No. of Instances
Mirai udppalain	1000
Mirai udppalain	1000
Mirai syn	1000
Mirai scan	1000
Mirai ack	1000
Gafgyt udp	1000
Gafgyt tcp	1000
Gafgyt scan	1000
Gafgyt junk	1000
Gafgyt combo	1000
Benign	1000
Total instances	11000

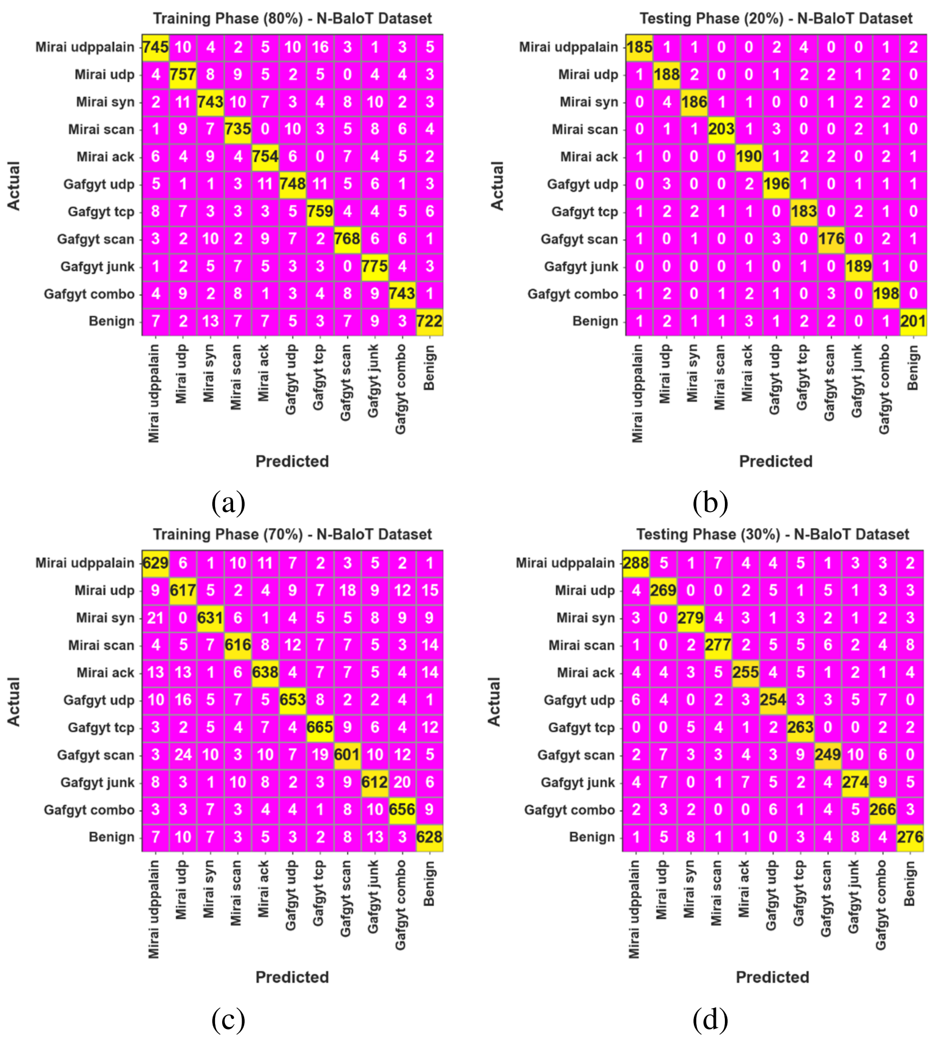

Figure 3. N-BaIoT dataset (a):(b) 80:20 TRPH/TSPH, and (c):(d) 70:30 TRPH/TSPH.

The results of cyber threat detection of the WSSADL-CTDC technique in the N-BaIoT dataset are shown in Table 2 and Figure 4. The results obtained show that the WSSADL-CTDC system achieves effective findings in each class. According to 80% of TRPH, the WSSADL-CTDC method obtains an average Acc of 98.86%, $Prec$ of 93.76%, $Recall$ of 93.73%, $F-Score$ of 93.74%, and Matthews correlation coefficient (MCC) of 93.12%. In addition, on 20% of TSPH, the WSSADL-CTDC algorithm gains an average Acc of 99.13%, $Prec$ of 95.23%, $Recall$ of 95.25%, $F-Score$ of 95.23%, and MCC of 94.76%, respectively.

The cyber threat detection analysis of the WSSADL-CTDC technique in the N-BaIoT dataset can be seen in Table 3 and Figure 5. The results obtained show that the WSSADL-CTDC system obtains adequate results in all classes. According to 70% of TRPH, the WSSADL-CTDC algorithm receives an average Acc of 98. 22%, $Prec$ of 90. 22%, $Recall$ of 90. 21%, $F-Score$ of 90. 20%, and MCC of 89.23%. Furthermore, based on 30% of TSPH, the WSSADL-CTDC method acquires an average Acc of 98. 07%, $Prec$ of 89. 39%, $Recall$ of 89. 44%, $F-Score$ of 89. 39%, and MCC of 88. 34%, respectively.

Table 2. Cyber threat detection outcomes of the WSSADL-CTDC model under the N-BaIoT.

Classes	Acc	Prec	Recall	F-Score	MCC
		TRPH (80%)			
Mirai udppalain	98.86	94.78	92.66	93.71	93.09
Mirai udp	98.85	93.00	94.51	93.75	93.12
Mirai syn	98.61	92.30	92.53	92.41	91.65
Mirai scan	98.77	93.04	93.27	93.16	92.48
Mirai ack	98.86	93.43	94.13	93.78	93.16
Gafgyt udp	98.85	93.27	94.09	93.68	93.05
Gafgyt tcp	98.88	93.70	94.05	93.88	93.26
Gafgyt scan	98.92	94.23	94.12	94.18	93.58
Gafgyt junk	98.93	92.70	95.92	94.28	93.71
Gafgyt combo	99.00	95.01	93.81	94.41	93.86
Benign	98.93	95.88	91.97	93.89	93.33
Average	98.86	93.76	93.73	93.74	93.12
		TSPH (20%)			
Mirai udppalain	99.23	96.86	94.39	95.61	95.19
Mirai udp	98.82	92.61	94.47	93.53	92.89
Mirai syn	99.14	95.88	94.42	95.14	94.67
Mirai scan	99.41	98.07	95.75	96.90	96.58
Mirai ack	99.09	94.53	95.48	95.00	94.50
Gafgyt udp	99.05	94.23	95.61	94.92	94.39
Gafgyt tcp	99.00	93.85	94.82	94.33	93.78
Gafgyt scan	99.18	94.62	95.65	95.14	94.69
Gafgyt junk	99.50	95.94	98.44	97.17	96.91
Gafgyt combo	98.91	93.40	95.19	94.29	93.69
Benign	99.14	97.57	93.49	95.49	95.04
Average	99.13	95.23	95.25	95.23	94.76

Table 3. Cyber threat detection analysis of WSSADL-CTDC algorithm with N-BaIoT dataset.

Classes	Acc	Prec	Recall	F-Score	MCC
			TRPH (70%)		
Mirai udppalain	98.32	88.59	92.91	90.70	89.81
Mirai udp	97.77	88.27	87.27	87.77	86.54
Mirai syn	98.48	92.79	90.27	91.52	90.69
Mirai scan	98.36	91.94	89.53	90.72	89.83
Mirai ack	98.22	91.01	89.61	90.30	89.33
Gafgyt udp	98.49	92.10	91.58	91.84	91.01
Gafgyt tcp	98.48	91.60	92.23	91.91	91.08
Gafgyt scan	97.68	88.77	85.37	87.04	85.78
Gafgyt junk	98.14	89.34	89.74	89.54	88.52
Gafgyt combo	98.38	89.99	92.66	91.30	90.42
Benign	98.09	87.96	91.15	89.52	88.49
Average	98.22	90.22	90.21	90.20	89.23
			TSPH (30%)		
Mirai udppalain	98.12	91.43	89.16	90.28	89.25
Mirai udp	98.21	88.49	91.81	90.12	89.15
Mirai syn	98.61	92.08	92.69	92.38	91.62
Mirai scan	98.12	91.12	88.78	89.94	88.91
Mirai ack	98.18	90.43	88.54	89.47	88.48
Gafgyt udp	97.94	87.89	88.50	88.19	87.07
Gafgyt tcp	98.39	87.67	94.27	90.85	90.04
Gafgyt scan	97.67	89.25	84.12	86.61	85.38
Gafgyt junk	97.55	88.10	86.16	87.12	85.77
Gafgyt combo	97.97	86.64	91.10	88.81	87.73
Benign	98.03	90.20	88.75	89.47	88.38
Average	98.07	89.39	89.44	89.39	88.34

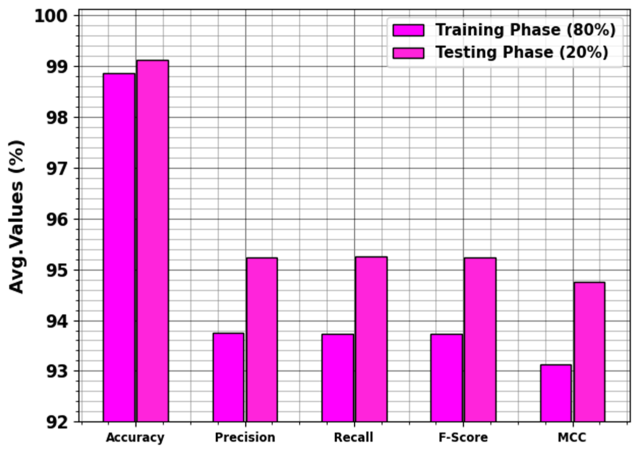

Figure 4. Average outcome of the WSSADL-CTDC system under N-BaIoT dataset.

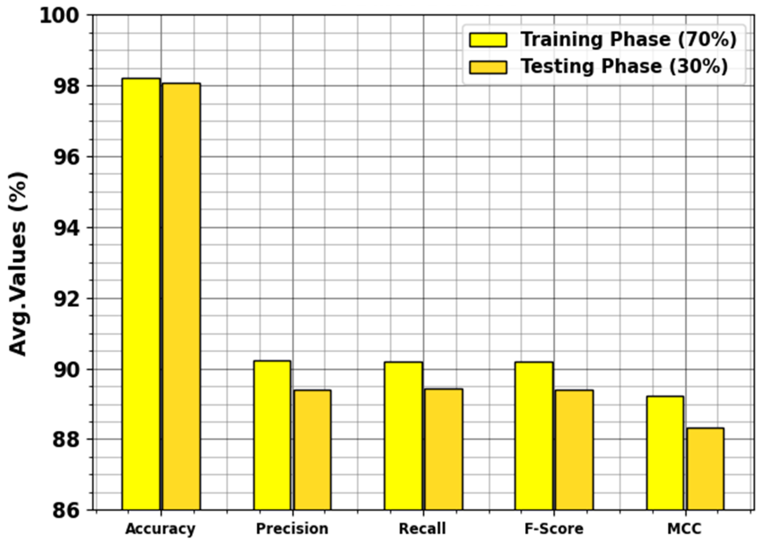

Figure 5. Average outcome of WSSADL-CTDC algorithm with N-BaIoT dataset.

The *Acc* curves for training (TR) and validation (VL) are illustrated in Figure 6 for the WSSADL-CTDC methodology with the N-BaIoT dataset providing valuable insight into its performance on various epochs. Specifically, it shows constant improvement at both TR and testing (TS) *Acc* with improving epochs, demonstrating the model's proficiency in learning and recognizing patterns in both TR and TS data. The upward trend in TS *Acc* emphasizes the model adaptability to the TR dataset and its ability to generate correct predictions on unnoticed data, underscoring the robust generalization capabilities.

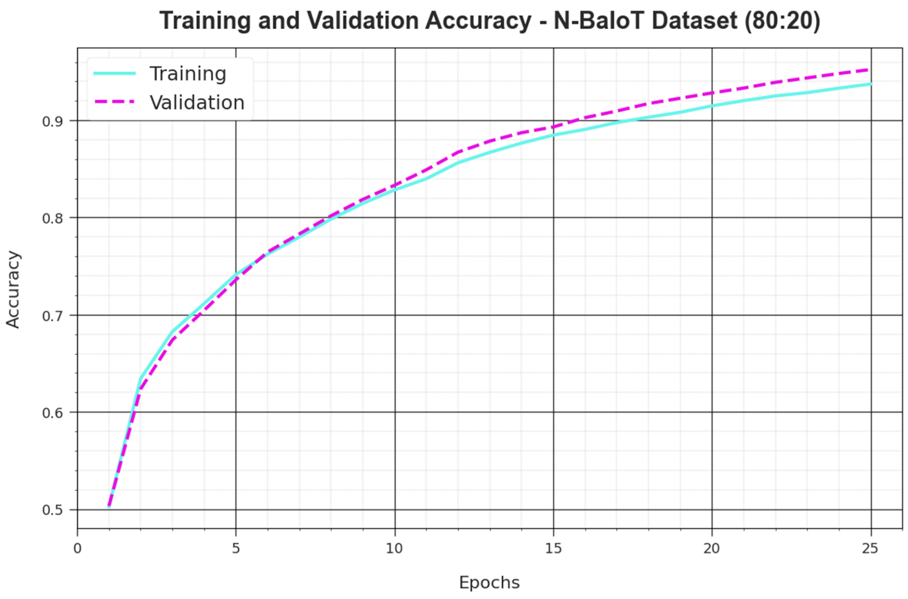

Figure 6. *Acc* curve of WSSADL-CTDC system under N-BaIoT dataset.

Figure 7 displays a wide-ranging overview of the TR and TS loss values for the WSSADL-CTDC algorithm with the N-BaIoT dataset at different epochs. The TR loss is reliably minimized as the model refines its weights to decrease classification errors in both datasets. The loss curves exemplify the alignment of the model with TR data, highlighting its ability to capture patterns efficiently. Consider the continuous improvement of parameters in the WSSADL-CTDC technique, which aims to diminish the discrepancies between predictions and actual TR labels.

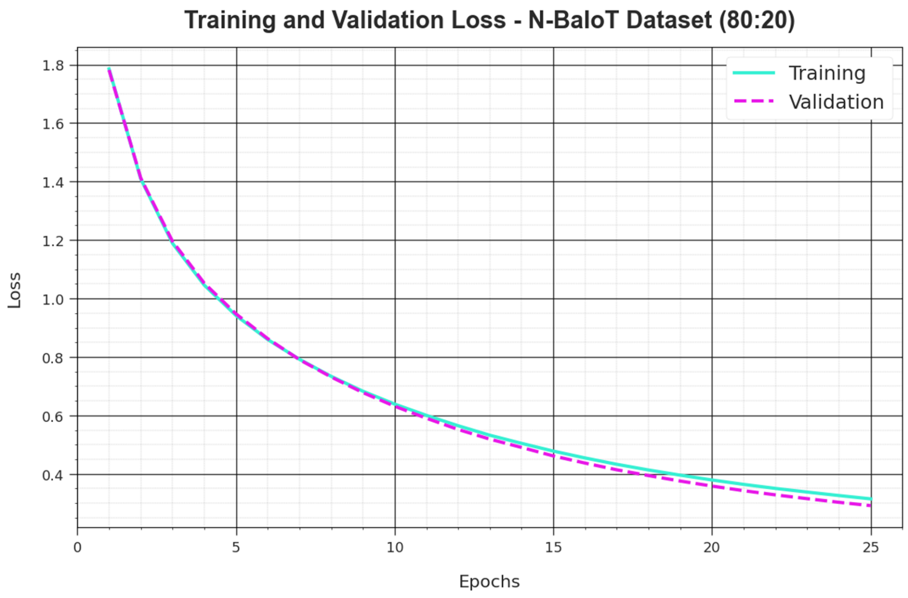

Figure 7. Loss curve of WSSADL-CTDC model with N-BaIoT database.

Regarding the precision-recall (PR) curve in Figure 8, the findings confirm that the WSSADL-CTDC system in the N-BaIoT dataset consistently achieves increased PR values in every class. These findings emphasize the model's better capacity to discriminate among diverse classes, underscoring its efficiency in accurately recognizing class labels.

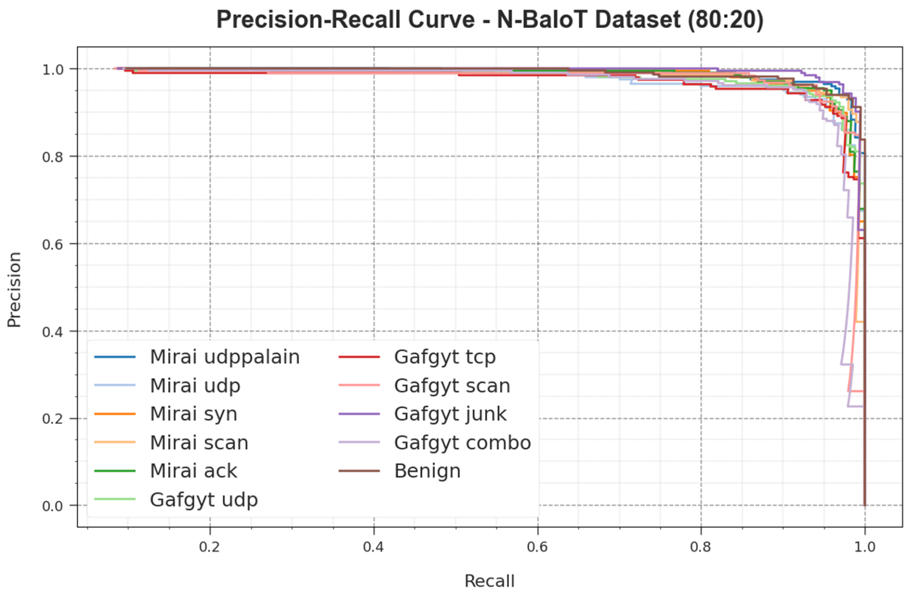

Figure 8. PR curve of WSSADL-CTDC model at N-BaIoT dataset.

Furthermore, in Figure 9, receiver operating characteristic (ROC) curves produced by the

WSSADL-CTDC technique are denoted under the N-BaIoT dataset, representing its proficiency in differentiating between classes. These curves offer valuable insights into how the trade-off between false positive rate (FPR) and true positive rate (TPR) differs through diverse classification epochs and thresholds. The results underscore the accuracy of the model's classification on numerous class labels, emphasizing its efficacy in covering diverse classification challenges.

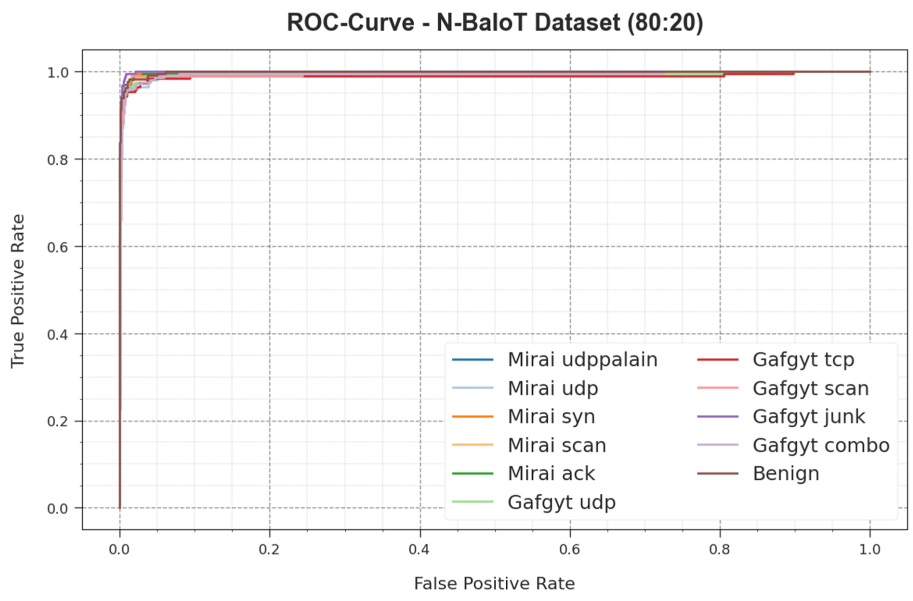

Figure 9. ROC curve of WSSADL-CTDC model with N-BaIoT dataset.

The detailed comparative statement of the WSSADL-CTDC technique with existing ones under the N-BaIoT dataset is given in Table 4 and Figure 10 [42]. These outcomes infer that the WSSADL-CTDC system performs better than other systems. It is noted that the deep belief network (DBN), LSTM, bat algorithm (BA)-NN, and PSO-NN models perform poorly, whereas the lightweight gradient-based algorithm (LGBA)-NN and multi-factor optimization random error linear model (MFO-RELM) models achieve reasonable results. However, the WSSADL-CTDC technique performs better with a maximum Acc of 99.13%, $Prec$ of 95.23%, $Recall$ of 95.25%, and $F-Score$ of 95.23%.

Table 4. Comparison analysis of the WSSADL-CTDC model with other algorithms under N-BaIoT dataset.

Methods	Acc	Prec	Recall	F-Score
DBN Algorithm	89.47	89.48	90.03	89.40
LSTM Model	89.88	89.63	89.92	89.61
BA-NN Model	91.20	84.53	88.02	84.68
PSO-NN Model	90.30	83.94	88.81	84.31
LGBA-NN Model	96.35	87.34	92.42	89.18
MFO-RELM	98.91	94.93	93.95	93.94
WSSADL-CTDC	99.13	95.23	95.25	95.23

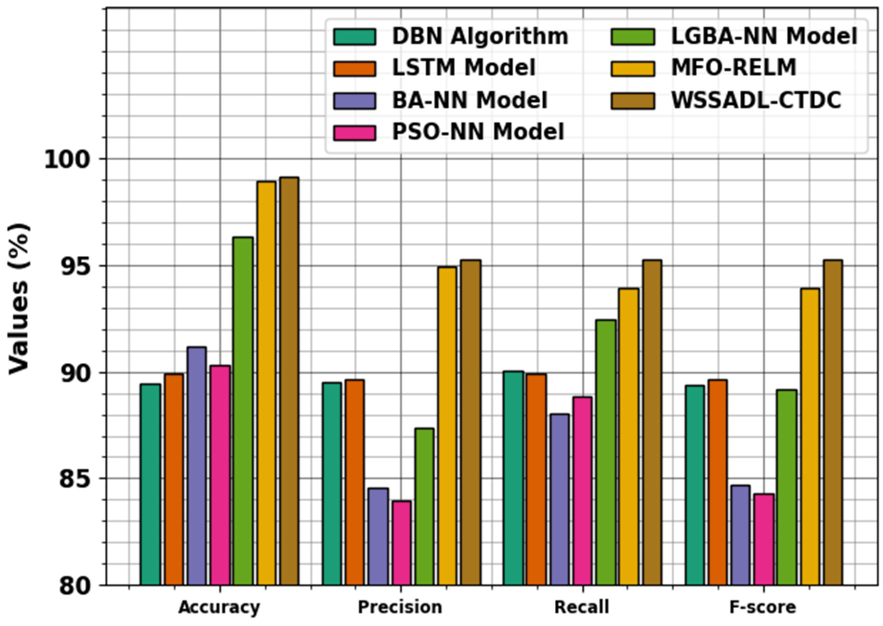

Figure 10. Comparison analysis of the WSSADL-CTDC model under the N-BaIoT dataset.

The computation time (CT) analysis of the WSSADL-CTDC methodology can be compared with other systems, as described in Table 5 and Figure 11. The findings show that the WSSADL-CTDC technique obtains a CT decrease of 0.46 s. Alternatively, the DBN, LSTM, BA-NN, PSO-NN, LGBA-NN, and MFO-RELM algorithms gain improved CT expressed in seconds units 4.67, 3.65, 2.64, 2.89, 2.71, and 1.94, respectively. Consequently, the WSSADL-CTDC method has been employed for superior performance over other compared algorithms.

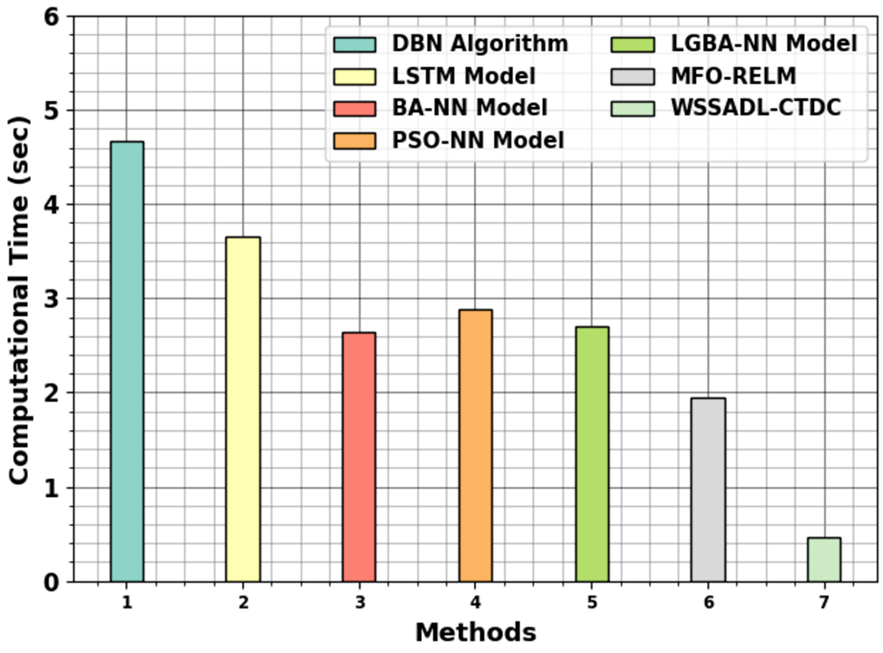

Figure 11. Computation time of the WSSADL-CTDC model with other systems.

Table 5. Computation time analysis of the WSSADL-CTDC model with other systems.

Methods	Computational Time (sec)
DBN Algorithm	4.67
LSTM Model	3.65
BA-NN Model	2.64
PSO-NN Model	2.89
LGBA-NN Model	2.71
MFO-RELM	1.94
WSSADL-CTDC	0.46

These results state that the WSSADL-CTDC technique improves the network security performance.

5. Conclusions

This article introduces an innovative WSSADL-CTDC method for robust network security. The WSSADL-CTDC technique aims to detect the presence of cyber threats and accomplish network security using metaheuristics with DL models. To achieve this, the WSSADL-CTDC technique implements a min-max normalization approach to scale the data into a uniform format. In addition, the WSSADL-CTDC technique applies SFLA to elect an optimal subset of features. The WSSADL-CTDC technique applies a hybrid CAE model for cyber threat detection and classification. A WSSA-based hyperparameter tuning method can be employed to enhance the detection performance of the CAE model. The simulation result of the WSSADL-CTDC technique can be evaluated under the benchmark dataset. The extensive analysis of the results found that the WSSADL-CTDC technique performed better than comparable methods on different measures. The WSSADL-CTDC technique may encounter scalability threats with more extensive datasets and benefit from real-time utilization optimizations in dynamic network environments. Future studies may focus on improving scalability and enabling real-time deployment.

Author contributions

Maha M. Althobaiti: Conceptualization, validation, formal analysis, investigation, resources, software, writing–original draft, writing–review & editing; José Escorcia-Gutierrez: Methodology, validation, formal analysis, investigation, visualization, writing–original draft, writing–review & editing. All authors have read and approved the final version of the manuscript for publication.

Use of AI tools declaration

During the revision of this work, the authors used the writefull language check embedded in Overleaf to improve English grammar, proofread and rephrase certain sentences in order to minimize the iThenticate similarity index. Following the use of these tools and services, the authors reviewed and edited the content as necessary, assuming full responsibility for the content of the publication.

Acknowledgments

The authors would like to acknowledge the Deanship of Graduate Studies and Scientific Research, Taif University for funding this work.

Conflict of interest

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

References

1. M. A. Ferrag, O. Friha, L. Maglaras, H. Janicke, L. Shu, Federated deep learning for cyber security in the internet of things: Concepts, applications, and experimental analysis, *IEEE Access*, **9** (2021), 138509–138542. https://doi.org/10.1109/ACCESS.2021.3118642
2. Y. Li, Y. Zuo, H. Song, Z. Lv, Deep learning in security of internet of things, *IEEE Internet Things J.*, **9** (2022), 22133–22146. https://doi.org/10.1109/JIOT.2021.3106898
3. A. Salih, S. T. Zeebaree, S. Ameen, A. Alkhyyat, H. M. Shukur, A survey on the role of artificial intelligence, machine learning and deep learning for cybersecurity attack detection, In: *2021 7th International engineering conference"Research & innovation amid global pandemic" (IEC)*, IEEE, 2021, 61–66. https://doi.org/10.1109/IEC52205.2021.9476132
4. Z. Z. Xian, F. Zhang, Image real-time detection using LSE-Yolo neural network in artificial intelligence-based internet of things for smart cities and smart homes, *Wirel. Commun. Mob. Com.*, **2022** (2022), 2608798. https://doi.org/10.1155/2022/2608798
5. A. D. Raju, I. Y. Abualhaol, R. S. Giagone, Y. Zhou, S. Huang, A survey on cross-architectural IoT malware threat hunting, *IEEE Access*, **9** (2021), 91686–91709. https://doi.org/10.1109/ACCESS.2021.3091427
6. B. Jothi, M. Pushpalatha, Wils-trs—A novel optimized deep learning based intrusion detection framework for IoT networks, *Pers. Ubiquit. Comput.*, **27** (2023), 1285–1301. https://doi.org/10.1007/s00779-021-01578-5
7. P. Dixit, S. Silakari, Deep learning algorithms for cybersecurity applications: A technological and status review, *Comput. Sci. Rev.*, **39** (2021), 100317. https://doi.org/10.1016/j.cosrev.2020.100317
8. D. Chen, P. Wawrzynski, Z. Lv, Cyber security in smart cities: A review of deep learning-based applications and case studies, *Sustain. Cities Soc.*, **66** (2021), 102655. https://doi.org/10.1016/j.scs.2020.102655
9. R. Ahmad, I. Alsmadi, Machine learning approaches to iot security: A systematic literature review, *Internet Things*, **14** (2021), 100365. https://doi.org/10.1016/j.iot.2021.100365
10. E. Bout, V. Loscri, A. Gallais, How machine learning changes the nature of cyberattacks on iot networks: A survey, *IEEE Commun. Surv. Tutor.*, **24** (2022), 248–279. https://doi.org/10.1109/COMST.2021.3127267

11. E. H. Houssein, D. Oliva, N. A. Samee, N. F. Mahmoud, M. M. Emam, Liver cancer algorithm: A novel bio-inspired optimizer, *Comput. Biol. Med.*, **165** (2023), 107389. https://doi.org/10.1016/j.compbiomed.2023.107389

12. S. Li, H. Chen, M. Wang, A. A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, *Future Gener. Comp. Syst.*, **111** (2020), 300–323. https://doi.org/10.1016/j.future.2020.03.055

13. X. Zhou, Y. Chen, Z. Wu, A. A. Heidari, H. Chen, E. Alabdulkreem, et al., Boosted local dimensional mutation and all-dimensional neighborhood slime mould algorithm for feature selection, *Neurocomputing*, **551** (2023), 126467. https://doi.org/10.1016/j.neucom.2023.126467

14. G. Wang, Moth search algorithm: A bio-inspired metaheuristic algorithm for global optimization problems, *Memetic. Comp.*, **10** (2018), 151–164. https://doi.org/10.1007/s12293-016-0212-3

15. Y. Yang, H. Chen, A. A. Heidari, A. H. Gandomi, Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts, *Expert Syst. Appl.*, **177** (2021), 114864. https://doi.org/10.1016/j.eswa.2021.114864

16. J. C. Butcher, G. Wanner, Runge-kutta methods: Some historical notes, *Appl. Numer. Math.*, **22** (1996), 113–151. https://doi.org/10.1016/S0168-9274(96)00048-7

17. J. Tu, H. Chen, M. Wang, A. H. Gandomi, The colony predation algorithm, *J. Bionic Eng.*, **18** (2021), 674–710.

18. I. Ahmadianfar, A. A. Heidari, S. Noshadian, H. Chen, A. H. Gandomi, INFO: An efficient optimization algorithm based on weighted mean of vectors, *Expert Syst. Appl.*, **195** (2022), 116516. https://doi.org/10.1016/j.eswa.2022.116516

19. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Future Gener. Comp. Syst.*, **97** (2019), 849–872. https://doi.org/10.1016/j.future.2019.02.028

20. H. Su, D. Zhao, A. A. Heidari, L. Liu, X. Zhang, M. Mafarja, et al., RIME: A physics-based optimization, *Neurocomputing*, **532** (2023), 183–214. https://doi.org/10.1016/j.neucom.2023.02.010

21. Y. Li, D. Zhao, C. Ma, J. Escorcia-Gutierrez, N. O. Aljehane, X. Ye, CDRIME-MTIS: An enhanced rime optimization-driven multi-threshold segmentation for covid-19 X-ray images, *Comput. Biol. Med.*, **169** (2024), 107838. https://doi.org/10.1016/j.compbiomed.2023.107838

22. A. Yazdinejad, M. Kazemi, R. M. Parizi, A. Dehghantanha, H. Karimipour, An ensemble deep learning model for cyber threat hunting in industrial internet of things, *Digit. Commun. Netw.*, **9** (2023), 101–110. https://doi.org/10.1016/j.dcan.2022.09.008

23. I. A. Khan, N. Moustafa, D. Pi, K. M. Sallam, A. Y. Zomaya, B. Li, A new explainable deep learning framework for cyber threat discovery in industrial iot networks, *IEEE Internet Things J.*, **9** (2022), 11604–11613. https://doi.org/10.1109/JIOT.2021.3130156

24. I. Bibi, A. Akhunzada, N. Kumar, Deep AI-powered cyber threat analysis in IIOT, *IEEE Internet Things J.*, **10** (2023), 7749–7760. https://doi.org/10.1109/JIOT.2022.3229722

25. S. Das, Y. Manchala, S. K. Rout, S. K. Panda, *Deep learning and metaheuristics based cyber threat detection in internet of things enabled smart city environment*, 2023. http://dx.doi.org/10.21203/rs.3.rs-3141258/v1

26. R. Wei, L. Cai, L. Zhao, A. Yu, D. Meng, DeepHunter: A graph neural network based approach for robust cyber threat hunting, In: *Security and privacy in communication networks*, Springer, **398** (2021), 3–24. https://doi.org/10.1007/978-3-030-90019-9_1

27. A. N. Ndife, Y. Mensin, W. Rakwichian, P. Muneesawang, Cyber-security audit for smart grid networks: An optimized detection technique based on bayesian deep learning, *J. Internet Serv. Inf. Secur.*, **12** (2022), 95–114. https://dx.doi.org/10.22667/JISIS.2022.05.31.095

28. M. A. Ferrag, D. Hamouda, M. Debbah, L. Maglaras, A. Lakas, Generative adversarial networks-driven cyber threat intelligence detection framework for securing internet of things, In: *2023 19th International conference on distributed computing in smart systems and the internet of things (DCOSS-IoT)*, IEEE, 2023, 196–200. https://doi.org/10.1109/DCOSS-IoT58021.2023.00042

29. T. Elangovan, S. Sukumaran, S. Muthumarilakshmi, An efficient recurrent neural network based classification method for cyber threat detection analysis, *J. Alebr. Stat.*, **13** (2022), 5514–5520.

30. Y. Zhou, B. Yang, H. Hou, L. Zhang, T. Wang, M. Hu, Continuous leakage-resilient identity-based encryption with tight security, *Comput. J.*, **62** (2019), 1092–1105. https://doi.org/10.1093/comjnl/bxy144

31. J. Xu, S. H. Park, X. Zhang, A bio-inspired motion sensitive model and its application to estimating human gaze positions under classified driving conditions, *Neurocomputing*, **345** (2019), 23–35. https://doi.org/10.1016/j.neucom.2018.09.093

32. Y. Li, W. G. Cui, H. Huang, Y. Z. Guo, K. Li, T. Tan, Epileptic seizure detection in EEG signals using sparse multiscale radial basis function networks and the fisher vector approach, *Knowledge Based Syst.*, **164** (2019), 96–106. https://doi.org/10.1016/j.knosys.2018.10.029

33. Y. Chen, L. Feng, C. Zheng, T. Zhou, L. Liu, P. Liu, et al., LDANet: Automatic lung parenchyma segmentation from CT images, *Comput. Biol. Med.*, **155** (2023), 106659. https://doi.org/10.1016/j.compbiomed.2023.106659

34. S. B. Lin, Generalization and expressivity for deep nets, *IEEE Trans. Neural Netw. Learn. Syst.*, **30** (2019), 1392–1406. https://doi.org/10.1109/TNNLS.2018.2868980

35. Q. Pham, B. Mohammadi, R. Moazenzadeh, S. Heddam, R. P. Zolá, A. Sankaran, et al., Prediction of lake water-level fluctuations using adaptive neuro-fuzzy inference system hybridized with metaheuristic optimization algorithms, *Appl. Water Sci.*, **13** (2023), 13. https://doi.org/10.1007/s13201-022-01815-z

36. R. Dash, R. Dash, R. Rautray, An evolutionary framework-based microarray gene selection and classification approach using binary shuffled frog leaping algorithm, *J. King Saud Univ. Comput. Inf. Sci.*, **34** (2022), 880–891. https://doi.org/10.1016/j.jksuci.2019.04.002

37. M. Mafarja, T. Thaher, M. A. Al-Betar, J. Too, M. A. Awadallah, I. A. Doush, et al., Classification framework for faulty software using enhanced exploratory whale optimizer-based feature selection scheme and random forest ensemble learning, *Appl. Intell.*, **53** (2023), 18715–18757. https://doi.org/10.1007/s10489-022-04427-x

38. P. Bedi, P. Gole, Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network, *Artif. Intell. Agric.*, **5** (2021), 90–101. https://doi.org/10.1016/j.aiia.2021.05.002

39. M. A. Syed, R. Syed, Weighted salp swarm algorithm and its applications towards optimal sensor deployment, *J. King Saud Univ. Comput. Inf. Sci.*, **34** (2022), 1285–1295. https://doi.org/10.1016/j.jksuci.2019.07.005

40. G. D. Singh, V. Tripathi, A. Dumka, R. S. Rathore, M. Bajaj, J. Escorcia-Gutierrez, et al., A novel framework for capacitated sdn controller placement: Balancing latency and reliability with pso algorithm, *Alex. Eng. J.*, **87** (2024), 77–92. https://doi.org/10.1016/j.aej.2023.12.018

41. Y. Meidan, M. Bohandana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, et al., N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders, *IEEE Pervas. Comput.*, **17** (2018), 12–22. https://doi.org/10.1109/MPRV.2018.03367731

42. F. Alrowais, S. Althahabi, S. S. Alotaibi, A. Mohamed, M. A. Hamza, R. Marzouk, Automated machine learning enabled cyber security threat detection in the internet of things environment, *Comput. Syst. Sci. Eng.*, **45** (2023), 687–700. https://doi.org/10.32604/csse.2023.030188

43. N. Savanović, A. Toskovic, A. Petervoic, M. Zivkovic, R. Damaševičius, L. Jovanovic, Intrusion detection in healthcare 4.0 internet of things systems via metaheuristics optimized machine learning, *Sustainability*, **15** (2023), 12563. https://doi.org/10.3390/su151612563

44. S. S. Kareem, R. R. Mostafa, F. A. Hashim, H. M. El-Bakry, An effective feature selection model using hybrid metaheuristic algorithms for IoT intrusion detection, *Sensors*, **22** (2022), 1396. https://doi.org/10.3390/s22041396

AIMS Press