An Efficient DDoS Attack Detection Method Using Optimized Long Short-Term Memory based on an Enhanced Salp Swarm Optimization Approach

P. Sakthivel[1], Dr. A. Manikandan[2]

[1]Department of Computer Applications, Muthayammal College of Arts and Science, Namakkal, Affiliated to Periyar University, Salem-636001, Tamil Nadu, India.
Email: *sakthivelmettur[at]gmail.com*

[2]Department of Computer Science and Engineering, Paavai Engineering College, Namakkal (Dt), Tamilnadu, India.
Email: *manikandanarumugampec[at]paavai.edu.in*

Abstract: DDoS attacks are currently rapidly evolving and represent a significant challenge to the stability and security of contemporary network infrastructures. Higher-dimensional traffic data, optimization inefficiencies, class imbalance, and slow convergence are the primary causes of performance degradation of existing deep learning-based Intrusion Detection Systems (IDS), which leads to high false alarm rates and low generalization. In order to address these shortcomings, this study offers a new Long Short-Term Memory model that is optimized by a Centroid Opposition-Based Learning and an Enhanced Simplex Search Salp Swarm Algorithm (CSSSA-LSTM). The integrated CSSSA improves search capability on a global scale, converges faster, and avoids premature stagnation through the combination of the exploratory opposition learning with the exploitative simplex refinement. Four standard IDS datasets, such as NSL-KDD, CICIDS2017, CICIDS2019, and UNSW-NB15, are experimentally validated following powerful preprocessing procedures, including normalization, feature encoding, imbalance management, and dimensionality reduction. The proposed model has been compared with LSTM, GA-LSTM, PSO-LSTM, SSA-LSTM, and COSSA-LSTM; the model has a superior performance with a maximum accuracy of 98.76, better F-score, sensitivity, specificity, and AUC, and faster convergence in terms of minimizing the MSE. As the results confirm, CSSSA-LSTM is an effective tool in terms of improving the ability of detecting attacks and minimizing the number of false positives as well as providing excellent stability in both binary and multi-class DDoS situations.

Keywords: DDoS Detection, Intrusion Detection System, Deep Learning, Long Short-Term Memory, Salp Swarm Algorithm, Centroid Opposition-Based Learning, Simplex Search, Swarm Intelligence Optimization, Network Security, Cyber-attack Detection.

1. Introduction

DDoS attacks have been among the most disruptive and dominant threats to current network infrastructure. As cloud computing, 5G, and large-scale IoT systems have emerged, cyber attackers use large numbers of devices controlled by bots to overwhelm targeted systems with massive volumes of devices, resulting in devastating network congestion, service disruption, and financial damage [1]. The growing sophistication and frequency of DDoS attacks complicate their identification by traditional systems, particularly with low-rate, multi-vector attacks that follow the appearance of legitimate traffic patterns [2]. The rule-based and signature-based detection methods do not generalize to unknown or zero-day attack variations, as they lack flexibility and depend on pre-existing signatures. The rise of machine learning (ML) [3, 4] and deep learning (DL)-based IDS solutions has become one of the promising alternatives because these solutions can automatically extract discriminative patterns in the network traffic[5]. Long Short-Term Memory (LSTM) networks have demonstrated good performance among them with regard to the capability to model temporal patterns that are very important in detecting dynamic DDoS trends. Nevertheless, the success of LSTM is highly determined by the best hyperparameter settings and appropriate feature representation [6]. Slow convergence, over-fitting, and low detection accuracy are common characteristics of high-dimensional traffic datasets, imbalanced classes, and inefficient parameter search, and are encountered in real large-scale situations. To address this limitation, a COBL and simplex method-based optimized LSTM (CSSSA-LSTM) intrusion detection model developed in response to accurate and efficient DDoS attack detection is presented in this research. The proposed CSSSA optimizer builds upon the standard SSA and COBL incorporated to increase the diversity of global searches, and a simplex search strategy to speed up the local convergence. [7]. The given hybrid optimization mechanism guarantees the efficient hyperparameter optimization of LSTM that will allow high detection rates and reduced false alarms, as well as accelerated convergence. The main objective of the current study is to develop a smarter and more efficient IDS capable of detecting DDoS attacks with great accuracy and a low rate of false alarms, and a high rate of real-time detection in the contemporary network context.

- A new hybrid framework of detection is suggested based on the integration of the LSTM and the improved CSSSA optimization algorithm in the case of efficient hyperparameter tuning and accelerated convergence.
- The LSTM model is implemented to effectively learn long-term time dependencies in network traffic so that it can

detect the high-rate as well as the changing low-rate DDoS attacks.

- It has a sound data preprocessing and feature engineering pipeline, such as normalization, dimensionality reduction, and imbalance, to be able to guarantee stable model performance.
- CICIDS2017, CICIDS2019, NSL-KDD, and UNSW-NB15 are some of the benchmark datasets on which the suggested CSSSA-LSTM model is rigorously tested to guarantee high generalization ability. The analysis of the experimental results showed that there were considerable enhancements in classification accuracy, precision, recall, F1-score, and AUC using the method compared to its state-of-the-art counterparts in terms of the use of ML/DL and metaheuristic-based detection methods.
- The model has a low FP rate, which justifies its applicability in real-time use in a network infrastructure.

The remaining part of the paper is organized as follows: the section 2 discusses the related works. Section 3 discusses the research methods. Section 4 discusses experimental results. Section conclusions are discussed in Section 5.

2. Literature Survey

Recent developments of DDoS and intrusion detection have focused on the techniques of hybrid deep learning and meta-heuristic optimization to enhance the accuracy and robustness. The proposed DBSCAN-SMOTE-ANN-XGBoost-PSO model by Efendi et al. (2025) [8] increases the reliability of the classifier against complex cyber-attacks, whereas Albashayreh et al. (2025) [9] suggested a Fireworks-GWO-based feature selection model that is optimized by using the Random Forest in detecting anomalies in the IoT. To enhance the efficiency of the BoT-IoT detection, Yang et al. (2025) [10] designed FPODL-DDoSAD that combines IPOA, SDAE, and FMO. Comparably, SD-FCIoT was developed by Chaudhary et al. (2025) [11] to use entropy-based measurements and Random Forest to generate high multi-class attack detection in fog-IoT systems. It was shown that GNN ensembles can decrease overfitting but improve accuracy on CICIDs datasets by Bakar et al. (2024) [12], and Ramzan et al. (2023) [13] emphasized the computational benefits of GRU and the high detection rates of the model in comparison to LSTM.

Sathishkumar et al. (2024) [14] used CSA to FL-LSTM to aid an ideal feature reduction, whereas Sathish et al. (2024) [15] combined IBFO with COBL-enhanced LSTM to hasten the convergence. Ali et al. (2024) [16] applied GWO to enhance the MLP weight training, and Deore et al. (2022) [17] took advantage of CNN and ChCSO-DLSTM to learn spatial and temporal patterns of flow. Chen et al. (2022) [18] used DBN together with LSTM to lower the computational cost of IDS, and Yang et al. (2023) [19] applied NRS-SSA together with weighted voting to enhance the reliability of its classification. Mahesh et al. (2025) [20] optimized DCNN using cosine learning to enhance the resilience of DDoS, whereas Hacılar et al. (2024) [21] used parallel ABC using DAE to overcome ANN local minima. Dharmapuri et al. [22] developed IFLO-assisted ML-GNU to enhance multi-attack recognition, and Alazab et al. [23] Modified the MLP using HHO to achieve enhanced performance on generalization. Moreover, Alkanhel (2023) [24] proposed GWDTO to bring a balance in exploration and exploitation in network threat detection, Hemnath (2023) [25] used autoencoder-LSTM fusion to enhance cloud intrusion prediction, and Daund et al. (2023) [26] used HHO-DBN to classify attacks better. Bandarupalli (2025) [27] prioritized lightweight IDS models as the real-time response, whereas Dalmaz et al. (2023) [28] combined GWO and MFO in order to accelerate the search capacity of intrusion recognition. Chen et al. (2024) [29] has suggested GQBWSSA-LightGBM, which is used to select adaptive features quickly, Althobaiti et al. (2024) [30] have proposed WSSADL-CTDC based on SFLA to achieve a high threat detection rate in the IoT, and Chu et al. (2024) [31] have suggested SATSSA to minimize false alarms, whereas Althubiti et al. (2022) [32] Collectively, the studies support the claim that hybrid optimization-based deep learning can substantially grow the accuracy of intrusion detection and decrease the number of computations at the expense of optimization instability and slower convergence, which are still open research problems.

3. Research Methods

3.1 LSTM

LSTM [33] was designed to eliminate the gradient vanishing and gradient exploding. It also processes the information in the forward propagation; however, it possesses rather dissimilar cell structure, which allows it to use useful information and transfer it through the long chain of sequences. The LSTM layers are primarily built with three gates, including the input, forget, and output gate and a cell state. The cell state can be regarded as the memory of the network, which can be able to carry relevant information to the sequence. The gates are neural networks that are able to identify the information that is to be kept. The rest of the information is forgotten in these gates as well. The forget gate is applied to forget redundant information. The input gate picks out the pertinent information to be added, and the output gate picks out the succeeding hidden state information. The forget gate will determine whether to retain or forget the information. The current input and the previous hidden state values are propagated through a sigmoidal process, which translates to a value of $0-1$ whereby the values that are nearer to 1 are retained, and the ones that are nearer to 0 are forgotten. h_{t-1} Is the previous hidden state information and xt is the current input. f_t The output of the forget gate can be computed as below equation.

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

In this case, W_f is the weight of the forget gate and b_f is the forget gate bias. The memory in LSTM is the cell state, and the input gate is used to update it. Initially, the current input and the hidden state information are input into a sigmoid function to

Table 1: Comparative analysis of existing detection methods

Ref. No.	Method	Dataset	Key Contribution	Limitations
[9]	Hybrid RNN + GRU + LSTM	CIC-DDoS2019, UNSW-NB15	Better temporal learning & higher accuracy	High computational complexity; lacks real deployment
[10]	LSTM-CLOUD	CICDDoS2019	99.83% accuracy with a defense mechanism	Weak for zero-day attacks due to signature dependency
[11]	LSTM	NSL-KDD, CICIDS, Bot-IoT	SSA-LSTM highest among compared	Expensive tuning; scalability unclear
[12]	LSTM	UNSW-NB15, NSL-KDD	Better than conventional ML	No comparison with the latest DL frameworks
[13]	LSTM in SDN	CICDDoS2019	Early detection (<250 packets)	SDN dependency; controller latency ignored
[14]	LS-DRNN + LAE	Bot-IoT	Better IoT multi-class detection	SMOTE may cause bias; the dataset is limited
[16]	Deep CNN	CICIDS2019, CICIDS2018	High performance	Too complex for real-time
[18]	CNN-GRU	Flow-rule data	Accurate LDoS attack detection	Limited to SDN flow rules
[19]	Softmax Deep Spectral RNN	IoT-23	Efficient cloud intrusion detection	Not validated on complex DDoS in 5G IoT
[20]	ILSTM	NSL-KDD, LITNET-2020	Better binary & multi-class IDS	Higher computational cost
[21]	CNN-LSTM	CIC-IDS2017	Higher accuracy than single models	No handling of dataset imbalance
[23]	DIWGAN	NSL-KDD, CICIDS2017	Improved WSN threat detection	GAN instability and long training
[24]	HMHSA	Multiple benchmark datasets	Good SYN-flood detection	Limited attack diversity
[25]	CNN-CBAM-GRU	UNSW-NB15, NSL-KDD	Strong multi-class classification	Not tested in real-time SDN/cloud
[26]	DALCNN + SDN	Testbed simulation	Efficient detection in SDN	Very small dataset (177 records)
[27]	RNN	Benchmark	High detection accuracy	Not scalable; lacks stress testing
[28]	Analytical Model	CTU-13 & Real logs	Effective application-layer DDoS detection	Manual features; no DL learning
[29]	DCGAN + ResNet-50 + AlexNet	CICIDS2019, UNSW-NB15	High classification performance	Very heavy computation
[30]	RNN-LSTM	Cloud traffic	Respectable DoS detection	High False Positives; no multi-class
[31]	MTL (MLP+LSTM+CNN)	NF-CSE-CIC-IDS2018-V2, BoT-IoT-V2	Strong generalization across datasets	Multi-tasking slows performance
[32]	Enhanced AE + DNN	WSN-DS, CICIDS2017, NSL-KDD	Good multi-dataset performance	Limited zero-day attack resilience

convert it to the range of 0 to 1, meaning the significance of information. The tanh function is also applied to the current input and hidden state information, which is also used to control the network. The result of these functions is multiplied, and in case the result of the sigmoid is zero, the information is discarded. At the production level of 1, The cell retains it. $\tilde{C}_t$ Is the output of the $tanh$ activation and i_t is the output of the sigmoid function. The following equation can be used to compute them.

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

$$\tilde{C}_t = \tanh(W_c \times [h_{t-1}, x_t] + b_c) \quad (3)$$

In this case, W_i and W_c are the weights of the input gate and the state of the cell, whereas b_i and b_c are the biases of the input gate and the cell state, respectively. h_{t-1} Is the past information of the hidden state and x_t is the present input. The point-wise multiplication of the forget gate output with the input gate output is then added using the point-wise addition. During point-wise multiplication, values dropped when multiplied by 0. Where C_{t-1} is previous state information and C_t is present state information, then, C_t can be determined by using the equation below.

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \quad (4)$$

The output gate is the final gate of the structure, and it can be used to arrive at the next hidden state. A sigmoid activation is used to transform the previous concealed data and the current inputs into a value ranging between is 0 and 1.

The new cell state representation is transmitted by an $tanh$ activation, and the result of this cell is multiplied by the result of the sigmoid activation function. The message that will be carried by the hidden state is the multiplication. When o_t the sigmoid output of this gate, and h_t is the output of this gate, then they may be obtained with the help of the following equation.

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

$$h_t = o_t \times \tanh(C_t) \quad (6)$$

In this case, W_o is the weight of the output gate, and b_o is the bias of the output gate. Initially, the input and the hidden state information are summed with the current and is referred to as the combination. The hidden layer receives the input of the combination and picks the information that is needed. The information to be added to the cell state is formed in a candidate layer. The input gate receives the combination as the input and chooses the data that would be inserted in the new cell state as part of the candidate. The forget and input gate outputs are then used to update the cell state and give the new hidden state. The output takes it into account and determines the information that will be included in the next state. In this manner, LSTM generates output based on the past states.

3.2 Salp Swarm Algorithm (SSA)

The latest swarm approach was the SSA, and Algorithm 1 demonstrated its procedure. The salps have the body of jellyfish. SSA utilizes the moving water pressures in order to find food. These are organisms that occur in swarms. The two mathematically different groups that constitute the salp chains are the head salp and the followers. It considers their food-seeking behavior to be a major indicator of behavior. F is the food source that the swarm is heading to. The equation shows the revised position of the leader.

$$x_j^1 = \begin{cases} F_j + c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right) & c_3 \geq 0 \\ F_j - c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right) & c_3 < 0 \end{cases} \quad (7)$$

$\mathrm{F_j}$ is the food supply location, x_j^1 is the location of the leader in the j-th dimension, lb_j and $\mathrm{ub_j}$ are the upper limits and lower limits, respectively. The SSO coefficient c_1 equals exploration and exploitation. The exploitation improves the local search. The bit repetition is eventually used to determine c_1, which is as follows:

$$c_1 = 2exp^{-\frac{4l^2}{L}} \quad (8)$$

L Is the best number of iterations and l is the running iteration. The parameters $\mathrm{c_2, c_3}$ are constituted of random numbers in the interval $[-1,1]$. c_3 Implies that the direction of the salp leader must be$+\infty$ or $-\infty$. The rest of the salps in the swarm revise their positions with the following law of motion of Newton:

$$x_j^i = \frac{1}{2}at^2 + v_0 t \quad (9)$$

With $v = \frac{(x-x_0)}{t}$, with t being time, v_0 being the initial speed and $\mathrm{i} \geq 2, \mathrm{x_j^i}$ being the position of follower i in dimension j ,$a = \frac{v_{final}}{v_0}$. Considering that $\mathrm{v_0} = 0$ and the conflict between iterations (time) =1, then the above equation can be rewritten as follows:

$$x_j^i = \frac{1}{2}(x_j^i + x_j^{i-1}) \quad (10)$$

Where i denotes the position of the follower i in dimension j,$i \geq 2, x_j^i$. Although the global optimum may not be known in the optimization problem, it is assumed that the optimal solution is attainable by moving with the leader towards this optimal solution. The leader is also tracked by the followers. Consequently, it will lead to the salp chain being able to evolve through multiple iterations to accomplish the global optimum.

3.3 COBL

One of the most effective approaches of the OBL scheme, the one that recorded impressive success in the DE algorithm in all competitions of its type, is the COBL [34]. The algorithm takes the whole population into consideration when it calculates the centroid of opposite points in the metaheuristic algorithm. $(X_1 \dots X_N)$ Let $(X_1 \dots X_N)$ be N points in a D dimensional search space. What are the carrying unit mass of the search space? The centroid of the body may then be defined in the following way:

$$M = \frac{x_1+x_2+x_3+\cdots+x_N}{N} \quad (11)$$

the centroid point at the j^{th} dimension can be computed in the following way:

$$M_j = \frac{1}{N}\sum_{i=1}^{N} x_i, j \quad (12)$$

The opposite-point $\breve{X_t}$ of a point X_i of a body has been defined as M, the centroid of a discrete uniform body, and is obtained in the following way:

$$\breve{X_\iota} = 2 \times M - X_i \quad (13)$$

Optimization problems can be applied to the centroid method, which can perform better than the min-max method.

3.4 Simplex method

Spendley et al. (1962) [35] suggested the simplex method. It is also defined by a set of points, which is one greater than the set of dimensions of the search space. The simplex technique possesses numerous properties, including speed of search, insignificant calculation size, and high local search [36, 37]. The stepwise action of the SM method is explained as follows,
Step 1: Screen all of the solutions (bacteria) of the population. Choose the global best X_g and second best X_b , where X_s means that the bacteria must be exchanged. These three values of the fitness are set as $f(X_g), f(X_b)$, and$f(X_s)$.

Step 2: to calculate the middle position X_c between two positions X_g, X_b , The formula below is applied:

$$X_c = \frac{X_g+X_b}{2} \quad (14)$$

Step3: Determination of the reflection point X_r , it is as follows,

$$X_c + \alpha(X_c - (X_s) \quad (15)$$

In this case, α is the reflection coefficient that will be equal to 1.

Step 4: A comparison operation is carried out between the point of reflection and the global best. Suppose that$f(X_r) < f(X_g)$, Then compute the following equation,

$$X_e = X_c + \gamma(X_r - (X_c) \quad (16)$$

Where γ is the extension coefficient, and that normally equals 2. In turn, equalize the fitness of the extension point X_e and global best X_g . When $(X_e < X_g)$ there is a then X_s there ought to be a should X_e . Then, X_r will be replaced by X_s .

Step 5: Comparisons between X_r and X_s are carried out. In the case that $f(X_r) > f(X_s)$, Then, the comparison operation would be obtained through the following equations,

$$X_t = X_c + \beta(X_s - X_c) \quad (17)$$

Where, β is the condense coefficient and it is normally taken as 0.5. The comparison is then conducted on the fitness values of the condense point X_t in which the point X_s was the condense point. When then X_s , should be used in place of X_t . Otherwise, X_r will be changed by the X_s.

Step 6: the shrink operations are reached to obtain the condense point $X_w f(X_g) < f(X_r) < f(X_s)$

This is defined as follows,

$$X_w = X_c - \beta(X_s - X_c) \quad (18)$$

In this case, β is the shrink coefficient. In case of the necessary exchange of X_r by will be substituted by X_w instead of X_s .

3.5 Centroid opposition-based simplex SSA (CSSSA)

The latest development in swarm-based optimization is the CSSSA that combines the capabilities of the COBL with the Simplex local search mechanism to the original SSA module. Salps in CSSSA continue to be organized in a chain of representation of the swarm, comprising of a leader and several followers. Nevertheless, as opposed to traditional SSO, the diversity of search and convergence accuracy has been improved greatly with the help of two adaptive operators. First, COBL helps to obtain high-quality opposite solutions in the centroid of the population, which is more likely to find promising regions in the search landscape. Mathematically, this is presented as shown below:

$$c_j = \frac{1}{N}\sum_{i=1}^{N} x_j^i \quad (19)$$

In which c_j is the centroid of the jth dimension and $x_{j,opp}^i$ is the centroid-opposite candidate position of the ith solution. The most successful salps are picked out of the collective population of original and reverse populations, which is then repeated to sustain the optimization process and, therefore, to achieve successful exploration. Then, to enhance the exploitation process in the neighborhood of optimal areas, Simplex technique is used in elite salps to improve local refinement. The fine-tuning salp positions relying on the reflection, expansion and contraction transformations are provided as:

$$x_r = C_{s+}\alpha(C_s - x_w) \quad (20)$$
$$x_e = C_{s+}\beta(x_r - C_s) \quad (21)$$
$$x_c = C_{s+}\gamma(x_w - C_s) \quad (22)$$

The x_r , x_e and x_c are the reflected, expanded, and contracted points, respectively. C_s is the centroid of elite solutions, α , β and γ are the coefficients of reflections, expansion, and contraction. In every single cycle, the leader salp adapts its position to the food source by the above equation, and the follower salps adapt by using the above equation as in regular SSO. Then COBL proposes better candidate posts, and the Simplex mechanism locally optimizes the best-performing persons. This hybridization of synergy contributes greatly to the acceleration of convergence as well as avoiding local minima entrapment. Consequently, CSSSA is good at balancing both search and local exploitation over the globe, and offers better optimization and better stability as compared to traditional SSO when the swarm progresses to the superior global solution through numerous attempts.

3.6 Proposed optimized CSSSA-based LSTM

LSTM networks are very effective in sequence learning, but their effectiveness highly relies on the optimal choice of hyperparameters and proper weight initialization. Manual tuning tends to provide a sub-optimal level of accuracy and augment calculation. In response to this, a new hybrid optimization framework is suggested and named Optimized CSSSA-LSTM that combines the CSSSA to enhance the training efficiency and the model generalization of the LSTM. CSSSA is used to optimize key trainable parameters of LSTM in this proposed model, which include learning rate, number of hidden units, and network weights. The salp of the swarm corresponds to a set of candidate parameters of the performance of the LSTM, and the fitness value is determined on the basis of the error of the model classification. The proposed hybrid design enables CSSSA to appropriately enhance the LSTM training process by increasing the speed of convergence, classification accuracy, and preventing the stagnation of local optima. In this way, the suggested Optimized CSSSA-LSTM model proves to be more efficient in performance and strong learning.

4. Experimental results and discussion

The present section discusses the performance of developed methods, including dataset details, performance metrics, results, and discussion. The developed detection methods are implemented using MATLAB2019R on Windows 11 with an i5 processor and 16 GB RAM.

4.1 Datasets

- NSL-KDD: The dataset [38] (https://www.kaggle.com/datasets/hassan06/nslkdd) consists of both normal and malicious network connections, categorized in four major attacks, which are Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L) attacks. The connection records are 41-feature characterized features of the various network traffic characteristics.
- To evaluate our work, the set of CIC-IDS2017 (https://www.unb.ca/cic/datasets/ids-2017.html) was collected in 2017 over 5 days of attack and non-attack and non-attacks, and it includes regular and DDoS attack data [39].The collection contains 225742 samples of attack and normal data, 85 network flow features, and label features. The balance in this dataset is rather negative; therefore, in the current instance, we balanced the training dataset, not only by the attack and normal samples, but also by the number of instances reduction, and we selected 80 features. Two kinds of datasets exist, among which are normal and malicious networks (SSH, FTP, HTTP, and email protocols).

- CIC-DDoS2019(https://www.unb.ca/cic/datasets/ddos-2019.html) is a fully labelled dataset consisting of normal and malicious (DNS, LDAP, MSSQL, NetBIOS, NTP, SNMP, SSDP, SYN, TFTP, and UDP) [40, 41]. This entire dataset consists of 50,063,112 examples with 80 features and 11 classifications to predict the DDoS attacks. CICFlowMeter was one of the most popular to utilize in generating network traffic characteristics that was employed in the creation of CIC-IDS2017 and CIC-DDoS2019.
- The UNSW developed the BoT-IoT by creating a fair network. The botnet traffic was separated from the normal traffic. The data sets were sorted into an attack category and subcategory according to the protocols to aid in tagging (TCP, HTTP, and UDP) [42] (“https://research.unsw.edu.au/projects/unsw-nb15-dataset”). The Bot-IoT dataset that we used to test our proposed framework consists of normal network traffic of regular IoT networks and various types of attacks. The rationale of selecting this dataset is that it is a simulation of an environment of an actual IoT ecosystem. The dataset has DDoS, DoS, OS, and Service Scan, Keylogging, and Data exfiltration attacks.

4.2 Performance metrics

In this paper, the prediction algorithm has been evaluated using five measures of its performance accuracy, sensitivity, specificity, precision, and F-score that are calculated with respect to a confusion matrix [43]. This leads to the following outcome of the use of the classes Benign as negative and attack as positive: The number of attack incidences that are classified as attack as should be known as True Positive (TP); the number of benign incidences that are classified as benign as should be known as True Negative (TN); the number of benign incidences that are classified as attack as should be known as False Positive (FP); the number of attack incidences that are classified as benign as should be known as False Negative (FN).

The accuracy of a classifier is only a description of the number of times a classifier makes accurate predictions and may be a ratio between correct and incorrect predictions. The overall behaviour of a prediction algorithm can be said to be based on its accuracy, and can be measured as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \qquad (23)$$

Sensitivity (recall) is the ability to predict positive cases (including the real positive rate) that can be represented in the following manner:

$$Sensitivity = \frac{TP}{TP+FN} \times 100\% \qquad (24)$$

The number of relevant samples in the examples retrieved is known as precision, also termed as a positive predictive value, defined as follows,

$$Pr\,e\,cision = \frac{TP}{TP+FP} \times 100\% \qquad (25)$$

The Sensitivity and precision have their harmonic mean calculated to provide the F-score that is estimated to weigh the same to either of them. It allows the comparison of prototypes, the description of the performance of a prototype and combines both the accuracy and Sensitivity into one score.

$$F-score = 2 \times \frac{Pr\,ecision \times Re\,call}{Pr\,ecision + Re\,call} \times 100\% \qquad (26)$$

4.3 Result Analysis

In order to determine how well the proposed LSTM with COBL-based SSA (LSTM+CSSSA) classifies, a lot of experiments were conducted on four standard NIDS datasets: NSL-KDD, CICIDS 2017, CICIDS 2019, and UNSW-NB15. The comparative analysis consists of LSTM, LSTM+GA, LSTM+PSO, LSTM+SSA, and LSTM+CSSSA variables based on standard IDS measures: Accuracy, Precision, Recall, F-score, and AUC.It can be seen that the proposed LSTM+CSSSA has clearly surpassed the other two reported in Table 1 and Figure 1 with an absolute highest accuracy of 95.78 and 95.58 F-score. The higher accuracy and the recall value underline the lower false alarm and enhancement of the attack detecting rate. LSTM+CSSSA has the second-highest accuracy of 94.45 per cent, compared to PSO, GA, and classical LSTM. As the convergence plot in Figure 2 shows, CSSSA is much more stable and achieves the optimum MSE at a significantly lower rate, meaning that it has improved global search efficiency and local minimum resistance. Table 2 and Figure 3 also show a similar trend of improvement. CSSSA optimization has the most accurate result of 96.54% compared to the popular metaheuristics.

The F-score of 96.38 percent ensures the strengthened robustness of large-scale traffic variability. CSSSA is next with the accuracy of 95.12 percent, and SSA, PSO, and GA all provide moderate detection rates. Figure 4 convergence behavior shows that CSSSA uses fast exploitation ability with the help of opposition learning with simplex refinement with the ability to reduce MSE rapidly as opposed to benchmark optimizers. On the newer CICIDS 2019 dataset (Table 3 and Figure 5), the suggested LSTM+CSSSA achieves 97.65 percent accuracy and 0.9761 AUC value, which is characterized by good generalization and discriminative capacity in a variety of advanced intrusion scenarios. Figure 6 reaffirms that CSSSA achieves consistent and regular convergence, which strengthens its capacity to deal with very nonlinear attack patterns and optimize the parameters of deep learning in the most efficient way possible.

As Table 4 and Figure 7 show, the high generalization capability of the proposed LSTM+CSSSA model is evident in the performance evaluation on the UNSW-NB15 dataset. Having the maximum recorded accuracy of 98.76, the precision value of 98.62, the recall value of 98.71, the F-score value of 98.66, and the excellent AUC value of 0.9845, the model demonstrates its superiority to LSTM optimized with COSSA, SSA, PSO, and GA. These are better values that underline the fact that CSSSA is more effective in distinguishing between legitimate and malicious traffic, particularly when it comes to the identification of advanced and unidentified attack behaviors that are evident in modern networks. Also, in Figure 8, the convergence curve indicates that the MSE decreases rapidly with little variation, which proves that centroid opposition learning and simplex improvement are effective in inhibiting premature convergence and cannot improve optimization stability. On the whole, the findings support the idea that LSTM+CSSSA provides highly reliable, accurate, and stable detection results on the UNSW-NB15 data, which makes it a good choice to be deployed in the real-world network security monitoring setting

Table 2: Performance analysis of NSL-KDD dataset.

Methods	Accuracy	Precision	Recall	F-Score
LSTM+ CSSSA	95.78	95.52	95.64	95.58
LSTM+COSSA	94.45	94.19	94.31	94.25
LSTM+SSA	93.12	92.86	92.98	92.92
LSTM+PSO	91.87	91.54	91.72	91.63
LSTM+GA	90.53	90.21	90.39	90.3
LSTM	89.34	89.05	89.22	89.13

Table 3: Performance analysis of CICIDS 2017 dataset

Methods	Accuracy	Precision	Recall	F-Score
LSTM+ CSSSA	96.54	96.32	96.45	96.38
LSTM+COSSA	95.12	94.89	95.01	94.95
LSTM+SSA	93.87	93.56	93.72	93.64
LSTM+PSO	92.45	92.13	92.31	92.22
LSTM+GA	90.98	90.65	90.83	90.74
LSTM	89.34	89.02	89.21	89.11

Table 4: Performance analysis of the CICIDS 2019 dataset

Methods	Accuracy	Precision	Recall	F-Score
LSTM+ CSSSA	97.65	97.42	97.58	97.56
LSTM+COSSA	96.32	96.08	96.25	96.16
LSTM+SSA	95.18	94.92	95.05	94.98
LSTM+PSO	93.87	93.61	93.74	93.67
LSTM+GA	92.45	92.19	92.32	92.25
LSTM	91.12	90.87	90.99	90.93

Table 5: Performance analysis of UNSW NB-15 dataset

Methods	Accuracy	Precision	Recall	F-Score
LSTM+ CSSSA	98.76	98.62	98.71	98.66
LSTM+COSSA	98.35	98.21	98.29	98.25
LSTM+SSA	97.12	96.98	97.05	97.01
LSTM+PSO	95.89	95.72	95.81	95.76
LSTM+GA	94.56	94.39	94.48	94.43

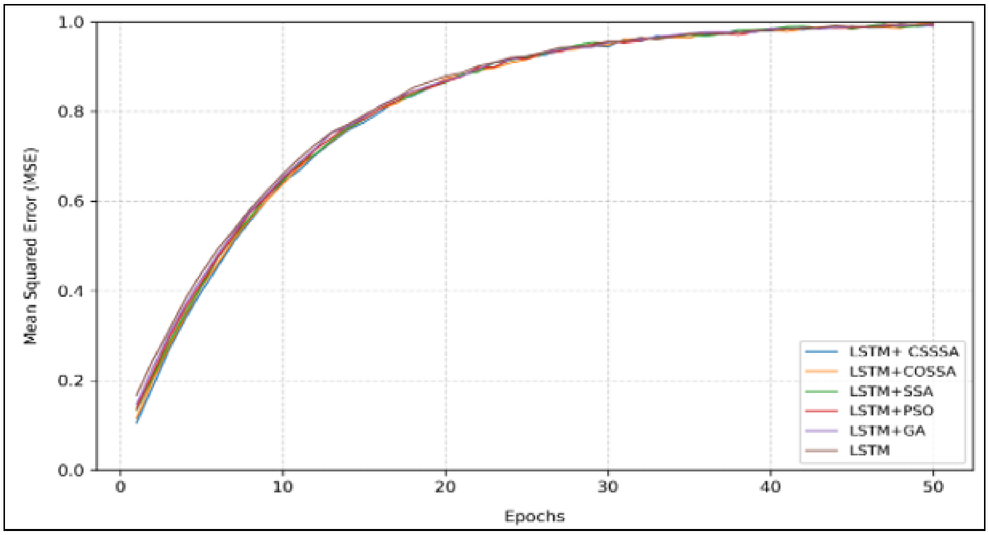

Figure 1: Convergence analysis of NSL-KDD dataset

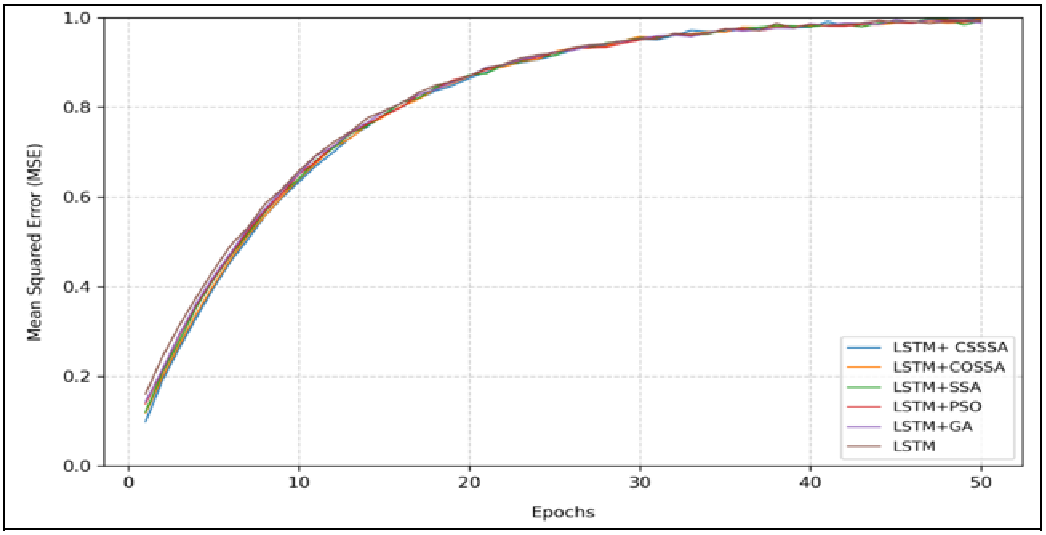

Figure 2: Convergence analysis of CICIDS2017 dataset

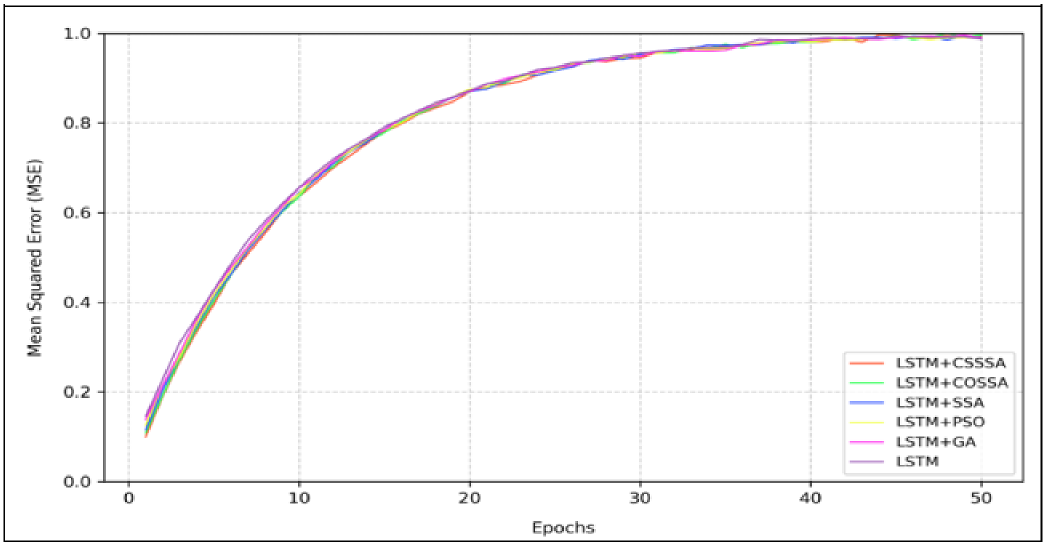

Figure 3: Convergence analysis of CICIDS 2019 dataset

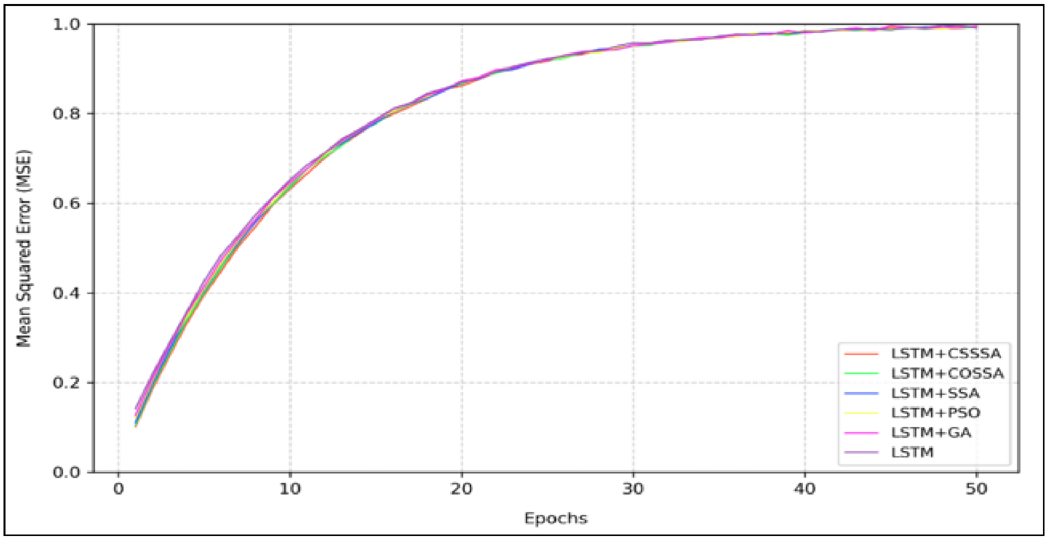

Figure 4: Convergence analysis of UNSW NB-15 dataset

5. Conclusions and Future Work

A new hybrid optimization-based deep learning model, called CSSSA-LSTM, was proposed to improve the reliability and accuracy of the detection method in heterogeneous network setups. CSSSA allowed SSA to be enhanced with COBL in addition to Simplex Search, which enhanced the global exploration and local exploitation of SSA capabilities in the face of feature optimization and thus minimized convergence stagnation that is common to swarm-based meta-heuristics. Together with the time learning capability of LSTM, the proposed approach was capable of capturing non-trivial traffic behavior and the dependency of cyber-attacks. A series of experiments performed on widely benchmarked and varied data sets, such as NSL-KDD, and CICIDS2017, CICIDS2019, and UNSW-NB15, have consistently demonstrated that CSSSA-LSTM has been able to significantly improve its performance when compared to traditional models, such as standalone LSTM, GA-LSTM, PSO-LSTM, and SSA-LSTM. The high detection accuracy, improved recall of minority attack classes, high values of AUC, and accelerated convergence using the MSE analysis show its strength in contemporary situations of designing an intrusion detector.

References

[1] L. Chen, Z. Wang, R. Huo, and T. Huang, "An adversarial DBN-LSTM method for detecting and defending against DDoS attacks in SDN environments," *Algorithms,* vol. 16, no. 4, p. 197, 2023.

[2] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Anomaly detection IDS for detecting DoS attacks in IoT networks based on machine learning algorithms," *Sensors,* vol. 24, no. 2, p. 713, 2024.

[3] R. Kavitha, K. Saravanan, S. A. Jebakumari, and K. Velusamy, "Machine learning algorithms for IoT applications," in *Artificial Intelligence for Internet of Things*: CRC Press, 2022, pp. 185-214.

[4] P. Velmurugan, A. Kannagi, M. Varsha, and K. Velusamy, "Data analytics techniques and tools in smart city applications," in *Artificial Intelligence for Internet of Things*: CRC Press, 2022, pp. 163-183.

[5] B. Fathimamary, M. Nasreen, and K. Velusamy, "An Efficient DDoS Attack Detection Using Optimized Long Short-Term Optimization Based on Improved Brainstorm Optimization," *Indian Journal of Science and Technology,* vol. 19, no. 5, pp. 298-312, 2026.

[6] P. Sakthivel and A. Manikandan, "An Optimized LSTM Based DDoS Attack Detection using Centroid Opposition Based Salp Swarm Optimization Algorithm," 2025.

[7] Z. Yang, Y. Jiang, and W.-C. Yeh, "Self-learning salp swarm algorithm for global optimization and its application in multi-layer perceptron model training," *Scientific Reports,* vol. 14, no. 1, p. 27401, 2024.

[8] R. Efendi, "Optimizing Neural Network Architecture for Detecting DDOS Attacks using ANN and XGBoost in Imbalanced Networks," *Engineering, Technology & Applied Science Research,* vol. 15, no. 3, pp. 22518-22526, 2025.

[9] A. Albashayreh, H. Mosleh, and A. Sharieh, "Hybrid Optimization-Based DDoS Attack Detection Using Fireworks and Grey Wolf Algorithms with Deep Learning," in *2025 International Conference on New Trends in Computing Sciences (ICTCS)*, 2025: IEEE, pp. 307-313.

[10] E. Yang, S. Jeong, and C. Seo, "Harnessing feature pruning with optimal deep learning based DDoS cyberattack detection on IoT environment," *Scientific Reports,* vol. 15, no. 1, p. 17516, 2025.

[11] P. Chaudhary, A. Singh, and B. Gupta, "Dynamic multiphase DDoS attack identification and mitigation framework to secure SDN-based fog-empowered consumer IoT Networks," *Computers and Electrical Engineering,* vol. 123, p. 110226, 2025.

[12] R. A. Bakar, L. De Marinis, F. Cugini, and F. Paolucci, "FTG-Net-E: A hierarchical ensemble graph neural network for DDoS attack detection," *Computer Networks,* vol. 250, p. 110508, 2024.

[13] M. Ramzan *et al.*, "Distributed denial of service attack detection in network traffic using deep learning algorithm," *Sensors,* vol. 23, no. 20, p. 8642, 2023.

[14] P. Sathishkumar, A. Gnanabaskaran, M. Saradha, and R. Gopinath, "Dos attack detection using fuzzy temporal deep long Short-Term memory algorithm in wireless sensor network," *Ain Shams Engineering Journal,* vol. 15, no. 12, p. 103052, 2024.

[15] D. Sathish and A. Kavitha, "DDoS Attack Detection Using Optimized Long Short-Term Memory Based on Improved Bacterial Foraging Optimization," in *2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, 2024: IEEE, pp. 568-573.

[16] A. Ali *et al.*, "An optimized multilayer perceptron-based network intrusion detection using Gray Wolf Optimization," *Computers and Electrical Engineering,* vol. 120, p. 109838, 2024.

[17] B. Deore and S. Bhosale, "Hybrid optimization enabled robust CNN-LSTM technique for network intrusion detection," *Ieee Access,* vol. 10, pp. 65611-65622, 2022.

[18] A. Chen, Y. Fu, X. Zheng, and G. Lu, "An efficient network behavior anomaly detection using a hybrid DBN-LSTM network," *Computers & security,* vol. 114, p. 102600, 2022.

[19] Z. Yang, Z. Liu, X. Zong, and G. Wang, "An optimized adaptive ensemble model with feature selection for network intrusion detection," *Concurrency and Computation: Practice and Experience,* vol. 35, no. 4, p. e7529, 2023.

[20] D. Mahesh and T. S. Kumar, "OptCosineDCNN: An Advanced Network Attack Detection and Mitigation in SDN Environments Utilizing Enhanced Deep Learning Based Approaches," *Cybernetics and Systems,* pp. 1-43, 2025.

[21] H. Hacılar, B. K. Dedeturk, B. Bakir-Gungor, and V. C. Gungor, "Network anomaly detection using Deep Autoencoder and parallel Artificial Bee Colony

algorithm-trained neural network," *PeerJ Computer Science,* vol. 10, p. e2333, 2024.

[22] V. Dharmapuri and S. R. Dutta, "A novel cluster of improved frilled lizard optimization and multi-ladder gated networks for the detection of cyber-attacks in computer networks," *Discover Computing,* vol. 28, no. 1, p. 120, 2025.

[23] M. Alazab, R. A. Khurma, P. A. Castillo, B. Abu-Salih, A. Martín, and D. Camacho, "An effective networks intrusion detection approach based on hybrid Harris Hawks and multi-layer perceptron," *Egyptian Informatics Journal,* vol. 25, p. 100423, 2024.

[24] R. Alkanhel *et al.*, "Network Intrusion Detection Based on Feature Selection and Hybrid Metaheuristic Optimization," *Computers, Materials & Continua,* vol. 74, no. 2, 2023.

[25] R. Hemnath, "Adaptive Intrusion Detection for Cloud Platforms Using LSTM and Autoencoder Networks," *Journal of Techno Social,* vol. 15, no. 1, 2023.

[26] R. P. Daund, D. Kumar, P. Charan, R. S. K. Ingilela, and R. Rastogi, "Intrusion detection in wireless sensor networks using hybrid deep belief networks and harris hawks optimizer," in *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2023: IEEE, pp. 1631-1636.

[27] G. Bandarupalli, "Efficient deep neural network for intrusion detection using CIC-IDS-2017 dataset," in *2025 First International Conference on Advances in Computer Science, Electrical, Electronics, and Communication Technologies (CE2CT)*, 2025: IEEE, pp. 476-480.

[28] H. Dalmaz, E. Erdal, and H. Ünver, "A new hybrid approach using GWO and MFO algorithms to detect network attack," *Computer Modeling in Engineering & Sciences,* vol. 136, no. 2, p. 1277, 2023.

[29] W. Chen, H. Yang, L. Yin, and X. Luo, "Large-scale IoT attack detection scheme based on LightGBM and feature selection using an improved salp swarm algorithm," *Scientific Reports,* vol. 14, no. 1, p. 19165, 2024.

[30] M. M. Althobaiti and J. Escorcia-Gutierrez, "Weighted salp swarm algorithm with deep learning-powered cyber-threat detection for robust network security," *AIMS Math.,* vol. 9, no. 7, pp. 17676-17695, 2024.

[31] S.-C. Chu, X. Yuan, J.-S. Pan, T.-Y. Wu, and F. Yan, "An efficient surrogate-assisted Taguchi salp swarm algorithm and its application for intrusion detection," *Wireless Networks,* vol. 30, no. 4, pp. 2675-2696, 2024.

[32] S. A. Althubiti, "A trust aware authentication scheme for wireless sensor networks optimized by salp swarm optimization and deep belief networks," *Mathematical Problems in Engineering,* vol. 2022, no. 1, p. 7842287, 2022.

[33] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: hybrid deep neural network for network intrusion detection system," *IEEE Access,* vol. 10, pp. 99837-99849, 2022.

[34] L. Zhou, M. Ma, L. Ding, and W. Tang, "Centroid opposition with a two-point full crossover for the partially attracted firefly algorithm," *Soft Computing-A Fusion of Foundations, Methodologies & Applications,* vol. 23, no. 23, 2019.

[35] W. Spendley, G. R. Hext, and F. R. Himsworth, "Sequential application of simplex designs in optimisation and evolutionary operation," *Technometrics,* vol. 4, no. 4, pp. 441-461, 1962.

[36] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal,* vol. 7, no. 4, pp. 308-313, 1965.

[37] Y. Zhou, Y. Zhou, Q. Luo, and M. Abdel-Basset, "A simplex method-based social spider optimization algorithm for clustering analysis," *Engineering Applications of Artificial Intelligence,* vol. 64, pp. 67-82, 2017.

[38] https://www.kaggle.com/datasets/hassan06/nslkdd (accessed.

[39] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSp,* vol. 1, pp. 108-116, 2018.

[40] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*, 2019: IEEE, pp. 1-8.

[41] R. J. Alzahrani and A. Alzahrani, "Security analysis of ddos attacks using machine learning algorithms in networks traffic," *Electronics,* vol. 10, no. 23, p. 2919, 2021.

[42] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems,* vol. 100, pp. 779-796, 2019.

[43] J. C. Obi, "A comparative study of several classification metrics and their performances on data," *World Journal of Advanced Engineering Technology and Sciences,* vol. 8, no. 1, pp. 308-314, 2023.