Received: October 24, 2024. Revised: November 29, 2024.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Multivariate Long Short-Term Memory with Spark Module for an Intrusion Detection System

Teena Kodapalu Balakrishna[1]* **Swati Sharma[1]**

[1]*Department of School of Computer Science & Engineering and IS, Presidency University, Bangalore, India*
* Corresponding author's Email: teenakb1@gmail.com

Abstract: Moving towards much digital and intelligent world equipped with Internet of Things (IoT) devices develops various security problems. In that, the Distributed Denial of Service (DDoS) attacks are highly complex and challenging security issues. The existing intrusion detection system has limitations like huge dimensionality, multi-feature dimension, less accuracy in classification and a huge false positive rate in raw traffic data. This research proposed a Multivariate Long Short-Term Memory with Spark module (MLSTM with spark) for an intrusion detection system. The dataset used for evaluation is CIC-DDoS 2019 and it is pre-processed by feature selection and min-max normalization techniques. After pre-processing, the intrusion is detected and classified by using the proposed MLSTM with spark. The LSTM layer's weight is optimized using the Black Widow Optimization (BWO) algorithm. The developed technique achieved 99.82% accuracy, 99.52% precision, 99.32% recall, 99.41% f1-score and a computation time of 14 secs which is more effective than existing approaches like Convolutional Neural Network – Bidirectional LSTM (CNN – BiLSTM) and Deep Hierarchical Machine Learning Method (DHMLM).

Keywords: Black widow optimization, Distributed denial of service, Intrusion detection, Multivariate long short-term memory, Spark.

1. Introduction

Distributed Denial of Service (DDoS) attacks are challenging to defend against and can cause severe damage to the services and reputation of the targeted victims [1-4]. With the rapid digitalization across various sectors like banking, education, healthcare, communication systems, entertainment, advertising and investment, the digital era has significantly increased security concerns [5-9]. According to a report by Huawei on botnets and DDoS attacks, common DDoS attacks result in substantial economic loss for governments and enterprises, with over 65% of DDoS attacks leading to financial damage [10-12]. As network technologies continue to update, the count, frequency and effect of DDoS attacks are increasing dramatically. This makes it especially critical to detect how these attacks are carried out and to differentiate between normal and DDoS attack traffic [13-15]. Hence, detecting DDoS attacks that differentiate between normal and malicious attacks has gained significant attention and has been majorly researched both internationally and domestically [16]. The DDoS attack techniques are separated into statistical, machine and deep learning-based techniques [17]. In recent times, various organizations have reported a wide range of DDoS attacks in their systems [18].

DDoS attacks aim to make services unavailable or restrict them temporarily, resulting in revenue losses and increased costs for service restoration [19, 20]. The Digital Attack Map tracks DDoS attacks and generates historical data on past attacks, which have severe effects on victim organizations [21-23]. The lack of practical solutions for detecting and preventing DDoS attacks has created a strong motivation to develop effective DDoS detection methods [24, 25]. There has been enormous research investigating the effects of using classification techniques for detecting and preventing DDoS attacks [26]. However, existing research has several drawbacks including practical performance rates of

detection devices, detection delay and the ability to handle large datasets effectively [27-29]. A practical method is required for detecting and preventing DDoS attacks when keeping the components for consideration [30]. Various techniques have been employed to maintain different data aspects which is projected into lower-dimensional spaces. As data size continued to increase, the reduction of dimensionality maximized significance [31-32]. The significant contributions of the research are given below:

- The dataset is pre-processed by feature selection and min-max normalization techniques which reduce the unnecessary features and scale the features between 0 and 1.
- The MLSTM with Spark module is proposed for detecting an intrusion which detects the intrusion with high accuracy and reduces computation time.
- The Black Widow Optimization (BWO) algorithm is utilized for updating a weight to the LSTM layer which reduces overall loss and improves accuracy.

This research paper is organized as follows: Section 2 gives details of the literature review. Section 3 explains the process of the proposed methodology. Section 4 gives results and a discussion of the proposed method and the conclusion of this research is given in Section 5.

2. Literature review

Zhao [33] presented self-attention with Convolutional Neural Network – Bidirectional Long Short-Term Memory (CNN – BiLSTM) for an intrusion detection system. Initially, Random Forest (RF) was integrated with Pearson Correlation Analysis for choosing significant features which minimizes redundancy of input data. Next, 1D-CNN and Bi-LSTM were utilized to extract spatial and temporal features, which were processed in parallel to obtain a combined feature set. Next, an attention mechanism was implemented to ensure significant input data features were completely expressed. At last, the softmax classifier was utilized for acquiring classification outcomes. The implemented method monitored network and host events, but the method was sensitive to imbalanced data in a network.

Mahadik [34] introduced an Intelligent Intrusion Detection System (IDS) by CNN which was named HetloT-CNN for an intrusion detection system. The introduced method identified and mitigated different DDoS attacks. The feasibility of the introduced method was assessed by taking binary and multiple class classifications. The performance of the introduced method was compared with DL techniques and outputs show that the introduced method performed well. The introduced method solved the issue of class imbalance. However, the introduced method has overfitting problems and huge training time.

Rani [35] presented a Deep Hierarchical Machine Learning Method (DHMLM) for an intrusion detection system. The presented method concentrated on finding many essential issues like DDoS and dictionary attacks. The presented method was essential due to the identification of whole issues and classification with less time for execution and training. Moreover, robust and IDS adopted exceptional light which operated data set hierarchical architecture which process on mobile devices. The presented method provided huge performance on multiple datasets and superior performance on finding attacks. However, the presented method was complex in execution.

Wu [36] suggested Robust Transformer-based Intrusion Detection System (RTIDS) method for an intrusion detection system. The suggested method used locational embedding to capture the relationships in sequential data among features. Then, a stacked encoder-decoder neural network was utilized to learn lower-dimensional feature representations from high-dimensional actual data. Moreover, the assigned self-attention mechanism facilitated network traffic type classification. The suggested method was unable to recognize patterns in network traffic. However, the suggested method attained less detection accuracy with huge false alert rates.

Varma [37] developed an Enhanced Elman Spike Neural Network based Intrusion Attack Detection in Software-defined Internet of Things Network (EESNN-IAD-SDN) for an intrusion detection system. In the developed method, SDN secured defense devices detected intrusion and DDoS attacks on controllers by Multiple dimensional Internet Protocol (IP) investigation. EESNN method classified DDoS and intrusion attacks as anomaly and normal. The developed method was robust and stable in intrusion detection. However, the developed method was computationally high because it needed an excessive quantity of data to acquire effective performance.

Alazab [38] introduced a Harris Hawks Optimization (HHO) algorithm - Multilayer Perceptron (MLP) for an intrusion detection system. The HHO optimized MLP by optimizing weight and bias parameters. The introduced method concentrated on selecting optimum parameters in their learning procedure for reducing errors in intrusion detection.

The introduced methods have been implemented by the EvoloPy NN framework which was an open-source Python tool to train MLPs by evolutionary mechanisms. However, the method has a high training time.

Wu [39] implemented a Feature - Weighted Naive Bayesian (NB) for an intrusion detection system. The feature engineering was implemented for assigning context-dependent weights for various feature terms, the NB technique was improved by implementing Jensen - Shannon (JS) divergence, FW and Inverse Category Frequency (ICF). The enhanced NB technique was combined with an intrusion detection system and classification. However, the method has computation and training time.

Selvana [40] suggested a Spider Monkey Social Optimization Algorithm (SMSOA) depended on the Deep Q network for an intrusion detection system. The suggested method has three stages like pre-processing, fusion of features and detection. The evaluation score was changed by imputing missing values in the stage of pr-processing. Jaccard measure integrated with a Deep Belief Network (DBN) was employed for the fusion of features. DQN was utilized to detect intrusion where SMSOA was used for training. However, the proposed method has less detection accuracy.

3. Proposed method

This research proposed a Multivariate Long Short-Term Memory with Spark (MLSTM with Spark) for an intrusion detection system. The dataset used for intrusion detection is CIC-DDoS 2019 and it is pre-processed by using feature selection which reduces features and normalization which normalizes features between 0 to 1. Then, the intrusion is detected by using the proposed MLSTM with Spark. Fig. 1 describes the process of implementing MLSTM with Spark method.

3.1 Dataset

The dataset used in the research is CIC-DDoS 2019 [41] which is shared by the Canadian Institute for Cybersecurity (CIC), It is arranged in a suitable testing context and includes the outcome of actual network traffic analysis.

The dataset includes 30,480,823 files in that, 30,423,960 samples are DDoS attacks and 56,863 samples are benign files. Moreover, these files are separated into eleven subtypes and every record is represented through 86 features. The dataset includes two networks such as Attack and Victim Network. This incorporates a highly secure structure with firewalls, switches and various operating devices. Every device is equipped with an agent which simulates benign behavior on every PC, ensuring a secure and controlled environment. An attack network is a fully divided third-party structure which executed various types of DDoS attacks. Table 1 represents the dataset description of CICIDS2019 dataset.

The next dataset used in this research is CICIDS 2017 [42] dataset which includes 79 distinct features separated to 15 traffic types such as Benign, FTP-Patator, SSH-Patator, DoS Hulk, DoS GoldenEye, DoS Slowhttptest, DoS slowloris, Hearbleed, Web Attack-XSS, Web Atack-SQL Injection, PortScan, DDoS and Web Attack-Brute Force.

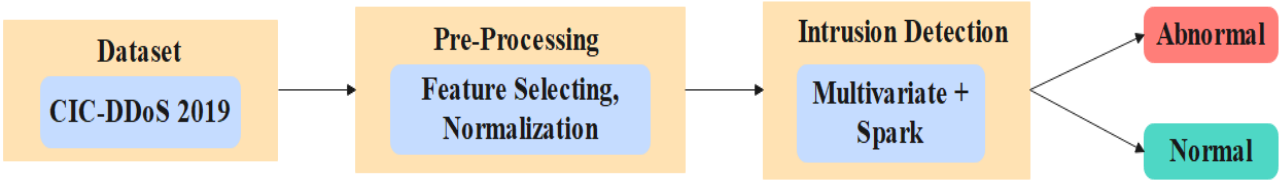

Figure. 1 Process of Implemented MLSTM with Spark method

Table 1. Dataset description

Type of Class	Flow Count
DDoS-WebDDoS	439
Benign	56,863
DDoS UDP-Lag	366,461
DDoS_NTP	1,202,642
DDoS_TFTP	2,082,580
DDoS_LDAP	2,179,930
DDoS_SSDP	2,610,611
DDoS_UDP	3,134,645
DDoS_NetBIOS	4,093,279
DDoS_MSSQL	4,522,492
DDoS_DNS	5,071,011
DDoS_SNMP	5,159,870
Total	30,480,823

3.2 Pre-processing

The pre-processing is an essential stage used in a network that cleans, reduces, normalizes and converts the data into a suitable format for further processing. In this research, Mutual Information based Feature Selection (MIFS) and min-max normalization are two pre-processing techniques used.

3.2.1. Feature Selection

This section provides data on selection of optimal features for training deep learning methods. The aim is to identify many relevant features to improve the method's performance and accuracy. The statistically dependent feature selection process involves estimating the associations between every input and destination variable by statistical techniques. Input variables that exhibit high correlation are chosen to ensure that the most significant features are used for training [43]. The selection of statistical metrics is grounded in the types of data integrated with both input and output variables. These techniques are efficient in identifying relevant features, enables efficient training and enhancing overall performance. Feature selection in a predictive method is the procedure to reduce the count of input variables. Minimizing the count of input variables reduces the execution cost of the model and in certain stages maximizes the performance of the method.

Two-stage feature selection technique is utilized for selecting a suitable feature set for training the model and detecting an attack. This method includes correlation-dependent feature selection through the Mutual Information based Feature Selection (MIFS) method. In particular, the matrix serves as a feature selection method that evaluates correlation among features. It identifies pairs of features with high correlation and removes one of them to prevent redundancy, as both features have the same effect on the dependent variable. As well as Mutual Information depended (MI) assessed arbitrary requirements among two variables. This feature selection technique utilizes MI for ranking good features within a given group of features.

3.2.2. Min-Max Normalization

Data normalization minimizes variance or characteristics of traffic within a certain range and minimizes the influence of outliers. Data is encoded using one-hot encoding, and min-max normalization is applied to scale the feature values between 0 and 1. The mathematical formula for min-max normalization is given in Eq. (1).

$$x = \frac{x - min_x}{max_x - min_x} \quad (1)$$

Where, x represents eigen values of i row and j column in the dataset. By using feature selection, it reduces unnecessary features and helps to improve the detection accuracy. The min-max normalization is used for scaling features between 0 and 1. By selecting the relevant features from raw data minimized the feature dimensionality and improved the detection performance. The min-max normalization preserved the relationship among actual data. After pre-processing, the pre-processed data is given as input to MLSTM for further processing.

3.3 Multivariate Long Short-Term Memory (LSTM)

The pre-processed data is taken as input for LSTM network for detecting intrusion in network. LSTM is a type of Recurrent Neural Network (RNN) which has many difficult neurons. While dealing with time series data that includes huge intervals and delays, LSTM is generally more effective than RNN. The architecture of multivariate LSTM is same as classical LSTM, but with changes in handling multiple input features. Multivariate LSTM has input, hidden and output layer. Every neuron accepted cell state addition to sample input at present state and result at final state. Neuron architecture of LSTM is represented in Fig. 2 in which, X shows scaled data, $+$ shows added data, sigmoid layer, tanh shows tanh layer (hyperbolic tangent), h_{t-1} shows outcome of past LSTM unit, c_{t-1} shows memory of past LSTM, x_t shows input, c_t shows recent memory and h_t shows outcome. There are three gates in LSTM neurons such as forget, input, output and subscript t shows a time. Fig. 2 represents neuron architecture of LSTM.

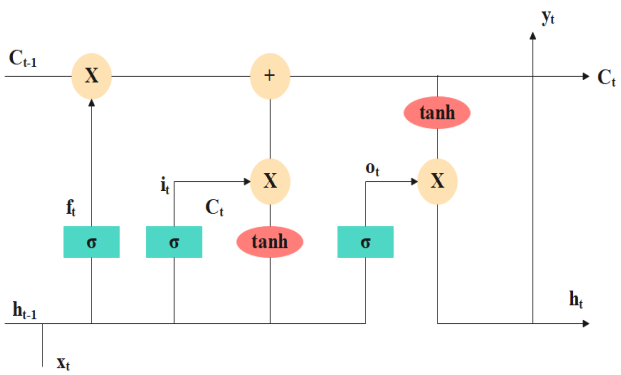

Figure. 2 Neuron Architecture of LSTM

Forget gate (f_t):

In forget gate, x_t and h_{t-1} represents input and output. The score among 0 and 1 is utilized for determining how much to recollect cell state (C_{t-1}) at the final stage, here 1 represents completely reserved and 0 represents completely abandoned. Considering this, when the LSTM method is used to fit and predict displacement information, the predicted displacement values typically fall within the range of 0.9 – 1. However, when predicting displacement information, the subsequent displacement value determined by the gating mechanism must be infinitely near to 0. During that time, the LSTM method discards information or rapidly removes it as it passes by the forget gate. The mathematical formula for forget gate is given as Eq. (2).

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1}\right) \tag{2}$$

In the above equation (2), the f_t represents forget gate at time t, the σ represents sigmoid activation function, W_f represents weight matrix integrated in input, x_t represents input vector at time t, the U_f represents weight matrix integrated with past hidden state and the h_{t-1} represents hidden state from past time $t-1$.

Input gate ($\mathbf{i_t}$):

Initially, utilize sigmoid layer which is input gate for deciding that data requires time updation and next develop vector in tanh layer that included how many data about an input value (X_t) of network is saved in cell state (C_t) at instant. Next, integrate these two parts for updating data of cell state (C_t) at moment. At that period, it is required for judging if each information is recollected and next updated to present cell state by information is recollected. The mathematical formula for the input gate is given in Eqs. (3) – (5).

$$i_t = \sigma(W_i x_t + U_i h_{t-1}) \tag{3}$$

$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1}) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

In the above equation (3), the i_t represents input gate at time t, the W_i represents weight matrix integrated with input and the U_i represents weight matrix integrated with past hidden state. In the above equation (4), the $\tilde{C}_t$ represents candidate cell state at time t, the tanh represents hyperbolic tangent activation function and the W_c represents weight matrix integrated with input x_t. In the above equation (5), C_t represents cell state in time t, the $f_t * C_{t-1}$ represents effect of forget gate in past cell state and the C_{t-1} represents cell state from past time state.

Output gate (o_t):

Sigmoid layer knows as output gate is utilized for determining how many cells state (C_t) at the present period is recollect in present result (h_t) and next tanh is utilized for processing cell state. The mathematical formula for the output gate is given in Eqs. (6) and (7).

$$o_t = \sigma(W_o x_t + U_o h_{t-1}) \tag{6}$$

$$y_t = h_t = o_t * \tanh(C_t) \tag{7}$$

In the above equation (6), the o_t represents output gate, the W_o represents weight matrix integrated with input, the U_o represents weight matrix with past output state. In the above equation (7), the y_t represents result of LSTM cell at time t, the h_t represents hidden state at time t, the o_t represents output gate at time t, the $\tanh(C_t)$ represents tanh function to cell state.

Forget gate controlled how much historical data has an effect on current and further which is how many continued to recollected in long-state data. The input gate controls how much input data is updated to long-state data and the output gate controls how much-aggregated data was utilized as the present result. The weight of the layer is updated by using BWO algorithms which is explained in the following section.

3.3.1. Black Widow Optimization (BWO)

BWO is a new and efficient conceptual optimization algorithm for the issues of non-linear optimization.

3.3.1.1. Population initialization

Population of spiders has count of N widow spiders that is described as $W_{N\times D} = [X_1, X_2, \dots, X_N]$. The dimension of optimization issue is described through variable D. $X_i = \left[x_{i,1}, x_{i,2}, \dots, x_{i,D}\right] (1 \le i \le N)$ describes population of ith widow.

Mathematical formula for initializing value of every component in individual X_i is given in Eq. (8).

$$x_{i,j} = l_j + rand(0,1) + . + \left(u_j - l_j\right), 1 \le j \le D \tag{8}$$

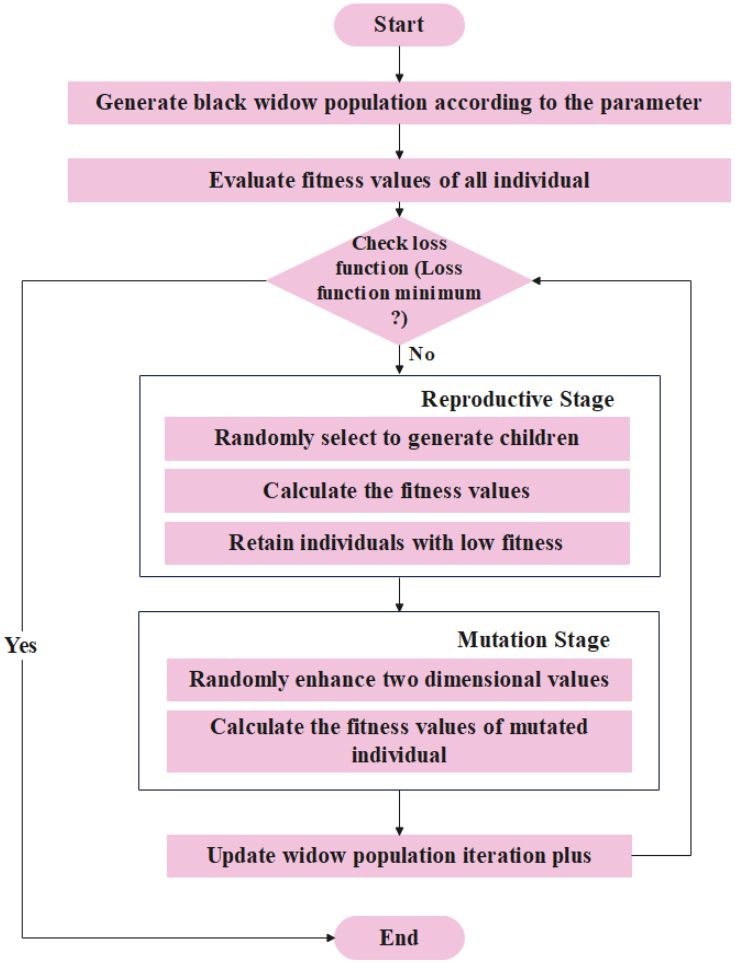

Figure. 3 represents the process of the BWO algorithm

Where, $L = [l_1, l_2, \ldots, l_D]$ represents lower bound and $U = [u_1, u_2, \ldots, u_D]$ represents upper bound variables in the optimization method.

3.3.1.2. Procreate

A Peculiar mating action of black widows resulted in the production of a new generation. When mating is initiated, pairs of spiders are developed from maternal and paternal spiders, randomly chosen from the mating population. This process is performed based on spiders' procreating rate (PR). The mathematical formula of offspring is given in Eqs. (9) and (10).

$$c_1 = a \times p_1 + (1 - a) + p_2 \quad (9)$$

$$c_2 = a \times p_2 + (1 - a) + p_1 \quad (10)$$

Where, p_1 and p_2 represents parents, c_1 and c_2 represents generation of new spiders and a represents D-dimensional array that contains random numbers.

3.3.1.3. Cannibalism

In this phase, there are three basic ways in which the destruction of the population happens. The initial form of population depletion occurs when female spiders consume male spiders. In the next stage, essential spiders prey on those of less significance. The third stage involves offspring consuming their mothers. In BWO, Cannibalism Rating (CR) is utilized for evaluating the survival rate of the population.

3.3.1.4. Mutation

In the mutation stage, the mutation score of every individual is arbitrarily chosen and that leads to maximizing the population. The solution is chosen randomly from the array. Mutation Rate (MR) is utilized for assessing that.

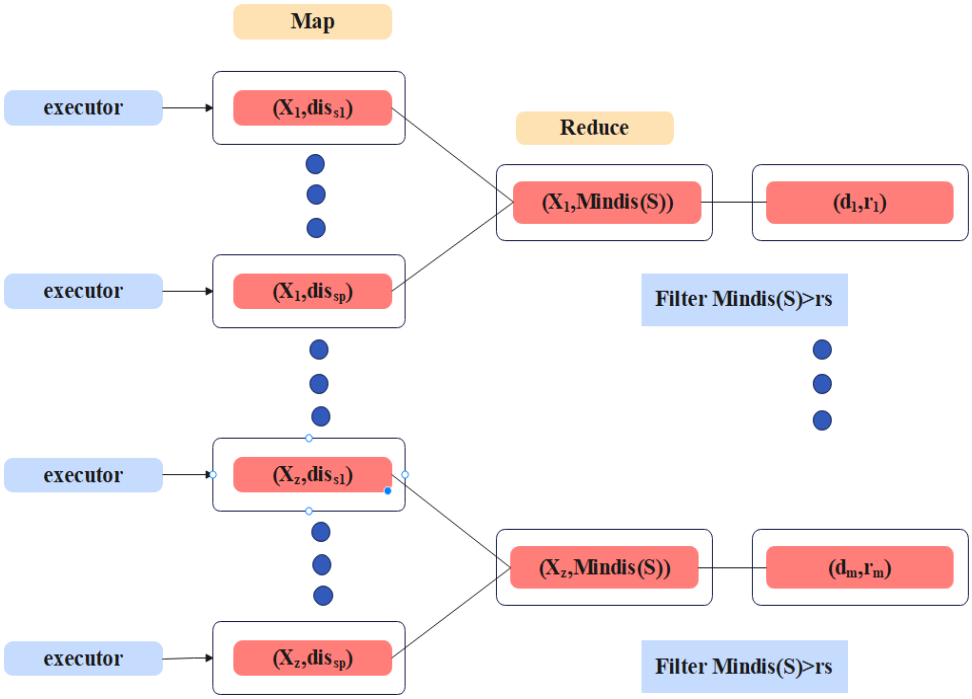

Figure. 4 Process of Spark

3.3.2. Spark

This subsection is described about an implementation of spark with proposed multivariate LSTM. This module is utilized for classifying new traffic data (NT) after pre-processing and storing in a Hadoop Distributed File System (HDFS). Fig. 4 represents the process of spark.

The optimal set of detectors is identified through the module of fuzzy optimized detection, which differentiates between intrusion and normal actions based on observed network information by Euclidean distance. Initially, Cartesian product (NT x Bestd) is done, next, the decision is taken by using matching rules. The mathematical formula for implemented mapper is given in Eq. (11).

$$Y = \begin{cases} abnormal & if\ Ed(x,d) \le r \\ normal & otherwise \end{cases} \quad (11)$$

Next, the minimized function considers a new sample x as key and Y as value (i.e abnormal or normal). Next, the system checks whether the entire status of x sample contains at least one abnormal condition, if any abnormality is detected, the sample is classified as an intrusion or else it is classified as normal.

4. Experimental analysis

The implemented technique is simulated with Python environment and system requirements of Windows 10, i7 internal processor and 64 Gb RAM. The performance of the implemented technique is analyzed with measures of accuracy, precision, recall, f1-score and computation time. The mathematical formula for performance metrics is given in Eqs. (12) – (15).

$$Accuracy = \frac{(True\ Positive + True\ Negative)}{Total\ Instances} \quad (12)$$

$$Precision = \frac{True\ Positive}{(Predicted\ Instances = True)} \quad (13)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (14)$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (15)$$

Here, TP - True Positives, TN - True Negatives, FP - False Positives and FN - False Negatives.

4.1 Quantitative and Qualitative analysis

The performance of the implemented technique is analyzed with various measures on CIC-DDoS 2019 dataset. The existing techniques taken for evaluating the implemented method are Multilayer Perceptron (MLP), Artificial Neural Network (ANN), Recurrent Neural Network (RNN) and CNN. Various tables and graphical representations are provided in this section.

Table 2 and Fig. 5 describe the performance of the optimization algorithm on CIC-DDoS 2019 dataset. The BWO algorithm achieves an accuracy of

Table 2. Performance of optimization algorithm

Optimization Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
WOA	90.44	90.48	89.17	88.27
GWO	91.59	91.04	89.69	89.13
ROA	92.03	91.92	90.49	90.28
RSA	92.82	92.72	91.24	91.03
BWO	94.57	93.18	92.37	91.62

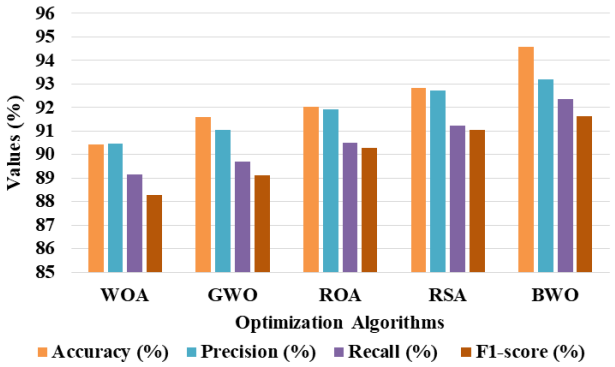

Figure. 5 Performance of optimization algorithm

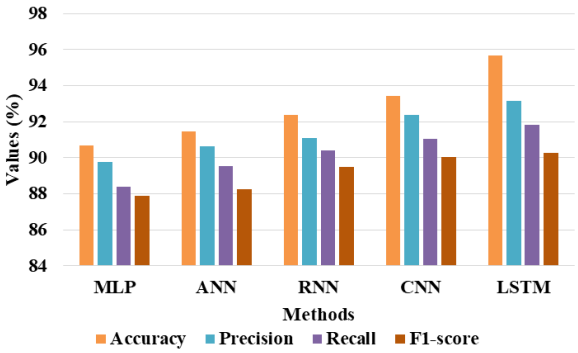

Figure. 6 Performance of LSTM

Table 3. Performance of LSTM

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
MLP	90.69	89.77	88.39	87.91
ANN	91.48	90.62	89.53	88.27
RNN	92.39	91.10	90.41	89.48
CNN	93.45	92.38	91.07	90.03
LSTM	95.67	93.17	91.83	90.28

Table 4. Performance of proposed Multivariate LSTM

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Multivariate - MLP	90.69	89.77	88.39	87.91
Multivariate - ANN	91.48	90.62	89.53	88.27
Multivariate - RNN	92.39	91.10	90.41	89.48
Multivariate - CNN	93.45	92.38	91.07	90.03
Multivariate - LSTM	95.67	93.17	91.83	90.28

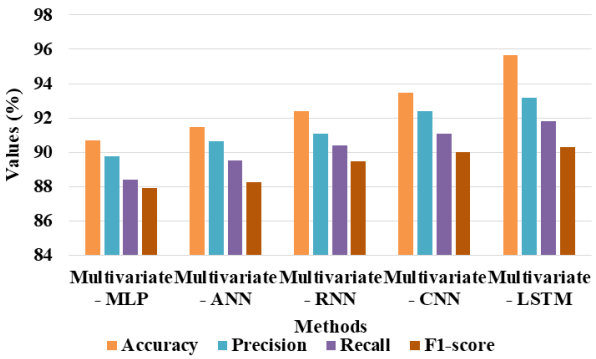

Figure. 7 Performance of proposed Multivariate LSTM

94.57%, precision of 93.18%, recall of 92.37% and f1-score of 91.62% which is more efficient than other existing techniques like Whale Optimization Algorithm (WOA), Grey Wolf Optimization (GWO), Remora Optimization Algorithm (ROA) and Reptile Search Algorithm (RSA).

Table 3 and Fig. 6 describes the performance of LSTM on CIC-DDoS 2019 dataset. The LSTM attained an accuracy of 95.67%, precision of 93.17%, recall of 91.83% and f1-score of 90.28% which is more efficient than other existing techniques like MLP, ANN, RNN and CNN.

Table 4 and Fig. 7 describes the performance of the proposed MLSTM on the CIC-DDoS 2019 dataset. The LSTM attained an accuracy of 95.67%, precision of 93.17%, recall of 91.83% and f1-score of 0.28% which is more efficient than existing techniques like Multivariate - MLP, Multivariate - ANN, Multivariate - RNN and Multivariate - CNN.

Table 5. Performance of Proposed Method with Spark

Performance metrics	Without Spark	With Spark
Computation time taken	34 secs	14 secs
Validation score	0.9927	0.9923
Evaluation score	[0.0023, 0.9927]	[0.0024, 0.9923]
ROC-AUC score	0.9258	0.9042

In Table 5, the performance of the proposed method with spark is described with the CIC-DDoS 2019 dataset. From Table 5, it is clear that the proposed method reduces the computation time by 14 seconds after utilizing the spark module. The spark module reduces the computation time and improves the detection accuracy. The below figure 8 represents the confusion matrix of CICIDS2019 dataset.

4.2 Comparative analysis

The performance of implemented Multivariate LSTM with Spark is compared with existing approaches like CNN – BiLSTM [33], HetloT-CNN [34], DHMLM [35], RTIDS [36] and EESNN-IAD-SDN [37] with CIC-IDS 2019 and CICIDS 2017 datasets. The proposed method achieves an accuracy of 99.82%, precision of 99.52%, recall of 99.32% and f1-score of 99.41% which is better than existing approaches. Table 6 describes a comparative analysis of the implemented approach.

4.3 Discussion

The advantages of the proposed MLSTM with spark module and the drawbacks of existing research are analyzed in this section. The CNN – BiLSTM [33] method has limitations like it is sensitive to imbalanced information in a network. The HetloT-CNN [34] method has drawbacks like overfitting problems and huge training time. The DHMLM [35] method has limitations like complexity in execution. The RTIDS [36] method has disadvantages like attaining less detection accuracy with huge false alert rates and the EESNN-IAD-SDN [37] method has limitations like being computationally high because it needs an excessive quantity of data to acquire effective performance.

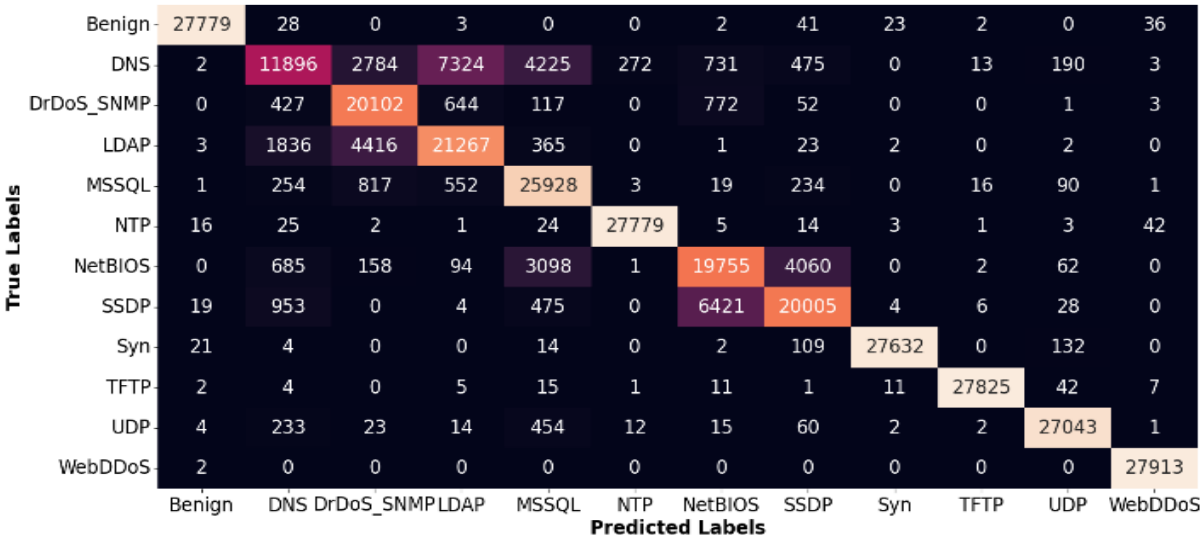

Figure. 8 Confusion matrix of CICIDS2019 dataset

Table 6. Comparative Analysis

Dataset	Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
CICIDS2019	CNN – BiLSTM [33]	92.501	92.809	92.451	92.630
	HetloT-CNN [34]	99.75	NA	NA	NA
	DHMLM [35]	99.07	98.91	98.95	98.93
	RTIDS [36]	98.58	98.82	98.66	98.48
	EESNN-IAD-SDN [37]	98.9	98.5	97.5	99.6
	Proposed Multivariate LSTM with Spark	99.82	99.52	99.32	99.41
CICIDS2017	RTIDS [36]	98.45	98.32	98.73	98.02
	Proposed Multivariate LSTM with Spark	98.61	98.48	98.27	98.35

To overcome these drawbacks, this research implemented an MLSTM with a spark module that reduces computation time and provides high intrusion detection accuracy. After pre-processing, the intrusion is detected and classified by using the proposed MLSTM with spark. The LSTM layer's weight is optimized using the BWO algorithm. The developed technique achieved 99.82% accuracy, 99.52% precision, 99.32% recall, 99.41% f1-score and a computation time of 14 secs on CICIDS 2019 dataset.

5. Conclusion

The proposed MLSTM with spark module maximizes security and detects the attacks in the network to eliminate malicious nodes in the network. The previous intrusion detection techniques have drawbacks like high computation and training time. The research proposed an MLSTM with a spark module which reduces computation time and provides high detection accuracy. The dataset used in the research is CIC-DDoS 2019, CICIDS2017 and it is pre-processed by feature selection which reduces unnecessary features and helps to improve detection accuracy. The BWO algorithm is used for updating weights of LSTM which reduces overall loss and improves the detection accuracy. The proposed method attained accuracy of 99.82%, precision of 99.52%, recall of 99.32% and f1-score of 99.41% and computation time of 14 secs on CICIDS 2019 dataset which is effective than existing methods like CNN – BiLSTM and DHMLM. In future, various neural networks can be developed to further improve the detection accuracy.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, have been done by 1st author. The supervision and project administration, have been done by 2nd author.

References

[1] V. Hnamte, H. Nhung-Nguyen, J. Hussain, and Y. Hwa-Kim, "A Novel Two-Stage Deep Learning Model for Network Intrusion Detection: LSTM-AE", *IEEE Access*, Vol. 11, pp. 37131-37148, 2023.

[2] M. Bakro, R.R. Kumar, M. Husain, Z. Ashraf, A. Ali, S.I. Yaqoob, M.N. Ahmed, and N. Parveen, "Building a Cloud-IDS by Hybrid Bio-Inspired Feature Selection Algorithms along with Random Forest Model", *IEEE Access*, Vol. 12, pp. 8846-8874, 2024.

[3] M. Najafimehr, S. Zarifzadeh, and S. Mostafavi, "A hybrid machine learning approach for detecting unprecedented DDoS attacks", *Journal of Supercomputing*, Vol. 78, pp. 8106-8136, 2022.

[4] O. Sbai, and M. Elboukhari, "Deep learning intrusion detection system for mobile ad hoc networks against flooding attacks", *International Journal of Artificial Intelligence*, Vol. 11, No. 3, p. 893-885, 2022.

[5] P.C.S. Reddy, P.S. Muller, S.N. Koka, V. Sharma, N. Sharma, and S. Mukherjee, "Detection of Encrypted and Malicious Network Traffic using Deep Learning", In: *Proc. of 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE)*, Ballari, India, pp. 1-6, 2023.

[6] S. Baskar, M.L. Prasad, N. Sharma, I. Nandhini, T. Katale, and P.C.S. Reddy, "An Accurate Prediction and Diagnosis of Alzheimer's Disease using Deep Learning", In: *Proc. of 2023 IEEE North Karnataka Subsection Flagship International Conference (NKCon)*, Belagavi, India, pp. 1-7, 2023.

[7] N. Sharma, S. Sharma, and A. Sindgi, "Solidity Smart Contract Vulnerabilities, Attack Scenarios, and Mitigation—A Survey", In: *Proc. of International Conference on Communication and Computational Technologies. ICCCT 2023*, Algorithms for Intelligent Systems, pp. 901-910, 2023.

[8] P. C. S. Reddy, S. Nithyapriya, N. Sharma, S. Maheswari, B. Jayaram, and T. Katale, "A Novel Ensemble Deep Learning Framework for Breast Cancer Prediction", In: *Proc. of 2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, Chennai, India, pp. 1-6, 2023.

[9] L.C.S. Reddy, R. Jayakarthik, A. Kiran, N. Sharma, S. Sharma, and P.C.S. Reddy, "An Uncertainty-Aware Deep Learning-Based Model for COVID-19 Diagnosis", In: *Proc. of 2023 3rd International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, Tumkur, India, pp. 1-6, 2023.

[10] J. Fuhr, F. Wang, and Y. Tang, "MOCA: A Network Intrusion Monitoring and

Classification System", *Journal of Cybersecurity and Privacy*, Vol. 2, No. 3, pp. 629-639, 2022.

[11] T.T. Khoei, and N. Kaabouch, "A Comparative Analysis of Supervised and Unsupervised Models for Detecting Attacks on the Intrusion Detection Systems", *Information*, Vol. 14, No. 2, p. 103, 2023.

[12] A. Berguiga, A. Harchay, A. Massaoudi, M.B. Ayed, and H. Belmabrouk, "GMLP-IDS: A Novel Deep Learning-Based Intrusion Detection System for Smart Agriculture", *Computers, Materials & Continua*, Vol. 77, No. 1, pp. 379-402, 2023.

[13] B. Yang, M.H. Arshad, and Q. Zhao, "Packet-Level and Flow-Level Network Intrusion Detection Based on Reinforcement Learning and Adversarial Training", *Algorithms*, Vol. 15, No. 12, p. 453, 2022.

[14] Y. Shewale, S. Kumar, and S. Banait, "Machine Learning Based Intrusion Detection in IoT Network Using MLP and LSTM", *International Journal of Intelligent Systems and Applications in Engineering*, Vol. 11, No. 7S, pp. 210-223, 2023.

[15] M.B. Anley, A. Genovese, D. Agostinello, and V. Piuri, "Robust DDoS attack detection with adaptive transfer learning", *Computers & Security*, Vol. 144, p.103962, 2024.

[16] C.S. Shieh, T.T. Nguyen, and M.F. Horng, "Detection of Unknown DDoS Attack Using Convolutional Neural Networks Featuring Geometrical Metric", *Mathematics*, Vol. 11, No. 9, p. 2145, 2023.

[17] V. Hnamte, and J. Hussain, "Dependable intrusion detection system using deep convolutional neural network: A novel framework and performance evaluation approach", *Telematics and Informatics Reports*, Vol. 11, p. 100077, 2023.

[18] Kamaldeep, M. Malik, and M. Dutta, "Feature engineering and machine learning framework for DDoS attack detection in the standardized internet of things", *IEEE Internet of Things Journal*, Vol. 10, No. 10, pp. 8658-8669, 2023.

[19] G.N. Tikhe, and P.S. Patheja, "A Wrapper Feature Selection Based Hybrid Deep Learning Model for DDoS Detection in a Network with NFV Behaviors", *Wireless Personal Communications*, Vol. 133, No. 1, pp. 481-506, 2023.

[20] H.S. Sharma, and K.J. Singh, "A feed forward deep neural network model using feature selection for cloud intrusion detection system", *Concurrency and Computation: Practice and Experience*, Vol. 36, No. 9, p. e8001, 2023.

[21] H.M. Saleh, H. Marouane, and A. Fakhfakh, "Stochastic Gradient Descent Intrusions Detection for Wireless Sensor Network Attack Detection System Using Machine Learning", *IEEE Access*, Vol. 12, pp. 3825-3836, 2024.

[22] H. Yang, J. Xu, Y. Xiao, and L. Hu, "SPE-ACGAN: A Resampling Approach for Class Imbalance Problem in Network Intrusion Detection Systems", *Electronics*, Vol. 12, No. 15, p. 3323, 2023.

[23] M.A. Hossain, and M.S. Islam, "Ensuring network security with a robust intrusion detection system using ensemble-based machine learning", *Array*, Vol. 19, p. 100306, 2023.

[24] S. Sivamohan, and S.S. Sridhar, "An optimized model for network intrusion detection systems in industry 4.0 using XAI based Bi-LSTM framework", *Neural Computing and Applications*, Vol. 35, No. 15, pp. 11459-11475, 2023.

[25] J. Han, and W. Pak, "Hierarchical LSTM-Based Network Intrusion Detection System Using Hybrid Classification", *Applied Sciences*, Vol. 13, No. 5, p. 3089, 2023.

[26] V. Graveto, T. Cruz, and P. Simões, "A Network Intrusion Detection System for Building Automation and Control Systems", *IEEE Access*, Vol. 11, pp. 7968-7983, 2023.

[27] A. Abdelkhalek, and M. Mashaly, "Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning", *The Journal of Supercomputing*, Vol. 79, pp. 10611-10644, 2023.

[28] Y.G. Damtew, and H. Chen, "SMMO-CoFS: Synthetic Multi-minority Oversampling with Collaborative Feature Selection for Network Intrusion Detection System", *International Journal of Computational Intelligence Systems*, Vol. 16, p. 12, 2023.

[29] R. Alkanhel, D.S. Khafaga, E.S.M. El-kenawy, A.A. Abdelhamid, A. Ibrahim, R. Amin, M. Abotaleb, and B.M. El-den, "Hybrid Grey Wolf and Dipper Throated Optimization in Network Intrusion Detection Systems", *CMC-Computers Materials & Continua*, Vol. 74, No. 2, pp. 2695-2709, 2023.

[30] H. Yan, X. Li, W. Zhang, R. Wang, H. Li, X. Zhao, F. Li, and X. Lin, "Automatic Evasion of Machine Learning-Based Network Intrusion Detection Systems", *IEEE Transactions on*

Dependable and Secure Computing, Vol. 21, No. 1, pp. 153-167, 2024.

[31] T. Kim, and W. Pak, "Integrated Feature-Based Network Intrusion Detection System Using Incremental Feature Generation", *Electronics*, Vol. 12, No. 7, p. 1657, 2023.

[32] N. Sharma, and S. Sharma, "Optimization of t-SNE by Tuning Perplexity for Dimensionality Reduction in NLP", In: *Proc. of International Conference on Communication and Computational Technologies*, pp. 519-528, 2023.

[33] J. Zhao, Y. Liu, Q. Zhang, and X. Zheng, "CNN-AttBiLSTM Mechanism: A DDoS Attack Detection Method Based on Attention Mechanism and CNN-BiLSTM", *IEEE Access*, Vol. 11, pp. 136308-136317, 2023.

[34] S. Mahadik, P.M. Pawar, and R. Muthalagu, "Efficient Intelligent Intrusion Detection System for Heterogeneous Internet of Things (HetIoT)", *Journal of Network and Systems Management*, Vol. 31, p. 2, 2023.

[35] S.V.J. Rani, I.I. Ioannou, P. Nagaradjane, C. Christophorou, V. Vassiliou, H. Yarramsetti, S. Shridhar, L.M. Balaji, and A. Pitsillides, "A Novel Deep Hierarchical Machine Learning Approach for Identification of Known and Unknown Multiple Security Attacks in a D2D Communications Network", *IEEE Access*, Vol. 11, pp. 95161-95194, 2023.

[36] Z. Wu, H. Zhang, P. Wang, and Z. Sun, "RTIDS: A robust transformer-based approach for intrusion detection system", *IEEE Access*, Vol. 10, pp. 64375-64387, 2022.

[37] P.R.K. Varma, R.R. Sathiya, and M. Vanitha, "Enhanced Elman spike neural network based intrusion attack detection in software defined Internet of Things network", *Concurrency and Computation: Practice and Experience*, Vol. 35, No. 2, p. e7503, 2023.

[38] 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.

[39] H. Wu, "Feature-Weighted Naive Bayesian Classifier for Wireless Network Intrusion Detection", *Security and Communication Networks*, Vol. 2024, No. 1, p. 7065482, 2024.

[40] G.S.R.E. Selvan, T. Daniya, J.P. Ananth, and S.K. Kumar, "Network intrusion detection and mitigation using hybrid optimization integrated deep Q network", *Cybernetics and Systems*, Vol. 55, No. 1, pp. 107-123, 2024.

[41] CICIDS2019 Dataset link: https://www.unb.ca/cic/datasets/ddos-2019.html

[42] CICIDS2017 Dataset link: https://www.unb.ca/cic/datasets/ids-2017.html

[43] K.B. Teena, and S. Sharma, "Anomaly based Intrusion Detection System using Hybrid ResNet50 and 3D Convolutional Neural Network", *International Journal of Intelligent Systems and Applications in Engineering*, Vol. 12, No. 3, pp. 673-683, 2024.