RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt26424

Optimized Long Short-Term Memory based on Partial Opposition-based Firefly Algorithm for Detecting DDoS Attacks in a Cloud Environment

M.C Malini [a, *], N. Chandrakala [a]

[a] Department of Computer Science, SSM College of Arts & Science, Komarapalayam, Namakkal, India.
* Corresponding Author Email: malinimarianesan@gmail.com
DOI: https://doi.org/10.54392/irjmt26424
Received: 05-09-2024; Revised: 30-06-2026; Accepted: 12-07-2026; Published: 30-07-2026

Abstract: Cloud computing environments are vulnerable to Distributed Denial of Service (DDoS) attacks, which can disrupt services and cause substantial financial and reputational damage. The dynamic nature of cloud traffic makes it difficult for traditional detection methods to remain effective, creating a need for more robust and adaptive detection strategies. To detect DDoS attacks accurately and efficiently, this study proposes an optimized Long Short-Term Memory (LSTM) model combined with a Partial Opposition-Based Firefly Algorithm (POFA). LSTM is used because of its ability to process sequential data and capture long-term dependencies, both of which are important for identifying network-traffic patterns associated with DDoS attacks. POFA is employed to optimize the LSTM hyperparameters, thereby reducing false-positive rates and improving detection accuracy. Experimental results indicate that the proposed method improves detection accuracy, processing speed, and scalability compared with the evaluated baseline approaches. These findings suggest that the proposed POFA-LSTM model can strengthen security in cloud-computing environments.

Keywords: Cloud Computing, Long-Short Term Memory, Firefly Algorithm, DDos Attack Detection, Partial Opposition-Based Learning.

1. Introduction

Cloud computing is an Internet-based computing model that provides users and organizations with on-demand access to resources such as servers, storage, and databases. DDoS attacks represent a major security threat in cloud environments because attackers can prevent legitimate users from accessing services [1]. The scale and frequency of DDoS attacks have increased significantly in recent years. A Google Cloud Armor customer was reportedly targeted by an attack that initially generated more than 10,000 requests per second (RPS). After eight minutes, the attack rate increased to approximately 100,000 RPS and subsequently peaked at 46 million RPS. DDoS attacks are among the most damaging forms of Internet-based cyberattack and reportedly account for more than 65% of the attacks considered in the cited source [2].

This large volume of requests can exhaust server and network capacity, making the targeted services unavailable to legitimate users. DDoS attacks must therefore be detected promptly to prevent disruption and damage to systems and computational resources [3]. Deep-learning (DL) methods play an important role in DDoS detection because they can analyse large-scale traffic data and identify complex patterns [4]. These methods are increasingly applied to DDoS detection because they can adapt to dynamic cloud traffic and overcome some limitations of traditional detection approaches. Deep learning therefore provides an adaptive and scalable approach to protecting cloud infrastructure from the growing threat of DDoS attacks [5]. Long Short-Term Memory (LSTM) is a type of recurrent neural network designed to learn long-term dependencies in sequential data. However, the LSTM architecture and its hyperparameters are typically selected empirically by the practitioner [6]. Hyperparameter selection strongly influences the model's learning capacity and predictive performance. To address this issue, several metaheuristic optimization methods have been developed for parameter tuning [7].

To determine suitable LSTM parameters for DDoS detection, this study combines partial opposition-based learning (POBL) with the Firefly Algorithm (FA). The resulting hybrid method benefits from the population diversity introduced by POBL and the global search capability of FA. The POBL-FA technique explores the hyperparameter space more comprehensively and

thereby increases the likelihood of identifying an effective LSTM configuration. To distinguish legitimate network traffic from attack traffic, the proposed technique examines a limited number of packets from each flow. The principal contributions are as follows:

- A novel hybrid optimization strategy that combines the FA with the POBL method to improve the hyperparameters of the LSTM.
- The hybrid technique helps the LSTM model for DDoS attack detection be tuned more precisely and effectively by improving the optimization process's exploration and exploitation capabilities.
- The strength of the recommended detection system was demonstrated by comparisons with other methods found in the literature.
- An optimized DDoS-detection method is developed to achieve high classification accuracy.
- Four benchmark datasets are used to evaluate the performance of the optimized LSTM model.
- The performance of POFA-LSTM is evaluated using five performance metrics.

The study's additional sections are arranged as follows: While sections 3, 4, and 5 discuss LSTM, FA, and POBL, respectively, section 2 discusses related works. The recommended detection technique is covered in Section 6, the findings analysis is covered in Section 7, and the conclusion is included in Section 8.

2. Related Works

The literature on deep-learning-based DDoS detection provides the theoretical and empirical foundation for identifying current research gaps and developing practical detection systems. Previous studies describe the evolution of DDoS attacks and the techniques developed to identify and mitigate them. Understanding this progression is essential for recognizing limitations in existing detection methods. Poornachander et al. [8] developed a CNN-LSTM anomaly-detection model for identifying abnormal traffic patterns. The model uses controller flow-rule information to detect anomalous traffic and applies IP tracing to identify the attacker. Devendiran et al. [9] proposed a deep-learning-based DDoS-detection method that incorporates chaotic optimization. The preprocessing stage applies data cleaning and M-squared normalization, after which the datasets are balanced using a synthetic sampling method. Kernel-assisted principal component analysis is used for feature extraction, followed by Chaotic Honey Badger Optimization for feature selection. A gated-attention dual LSTM then classifies the traffic using the selected features. The method was evaluated on the NSL-KDD and TON-IoT datasets. Sughanthini et al. [10] developed an optimized LSTM method based on partial opposition-based elephant herding optimization, termed POEHO-LSTM, and reported improved detection performance.

Vibhute *et al*. (2024) [11] developed a real-time detection method using LSTM that can identify abnormal network traffic. The developed method reduces dimensionality and selects optimal features using a random forest (RF) approach. An LSTM approach formed for recognizing network intrusions also made use of the optimal features that were chosen. Sathish *et al.* (2024) [12] developed an efficient detection model based on LSTM to detect abnormalities. Improved bacterial foraging optimization (IBFO) is used to tune the hyperparameters in an LSTM. The IBFO is used for population initialization and position updates, which is enhanced by the centroid opposition-based learning (COBL). Comparing the modified LSTM to a few well-used detection methods, the experimental result disclosed that it attained outstanding detection accuracy and fast convergence. Thangasamy *et al*. (2023) [13] suggested a new detection method based on a hybrid deep belief network (DBN) and LSTM. The prediction error is reduced in the hybrid model by combining Particle Swarm Optimization (PSO) with LSTM. By extracting IP packet attributes, this DBN technique uses PSO-LSTM to detect DDoS attacks. Furthermore, it distinguishes anomalies and predicts malicious network traffic with high accuracy. The proposed PSO-LSTM performs better in terms of detection performance.

Issa et al. [14] developed a seven-layer hybrid CNN-LSTM architecture and reported an accuracy of approximately 99%. Ramzan et al. [15] compared GRU, LSTM, and RNN models for DDoS detection to identify methods that provide accurate detection with lower computational complexity and execution time. Xu et al. [16] developed a hybrid one-dimensional CNN-GRU method for detecting low-rate denial-of-service attacks. Real legitimate traffic was collected from a data-centre website, while attack traffic was generated on a laboratory mirror of the website. The method detected low-rate attacks in fluctuating HTTP traffic without manual feature extraction. Bhardwaj et al. [17] proposed a method combining a deep neural network with a well-posed stacked sparse autoencoder. The model was designed to reduce overfitting, reconstruction error, and exploding or vanishing gradients, and it was compared with eleven existing approaches on the CICIDS2017 and NSL-KDD datasets. Priya et al. [18] proposed a recurrent neural-network autoencoder optimized using a genetic algorithm to detect network intrusions in cloud services. Deniz et al. [19] created a cloud-based intrusion dataset and evaluated LSTM, CNN, and ANN models, reporting the best performance for LSTM. Yin et al. [20] proposed a multi-scale CNN and bidirectional LSTM architecture, termed MSCBL-ADN, in which an arbitration network reweights the extracted features. Sathishkumar et al. [21] proposed a fuzzy intrusion-detection system integrating LSTM, temporal constraints, fuzzy rules, and Crow Search Algorithm-based feature selection.

Mazumder *et al. [*22] used an LSTM-based recurrent neural network to model temporal correlations in network traffic and compared it with DNN and random-forest models. Kumar *et al.* (2025) [23] compared vanilla LSTM, stacked LSTM, DNN, and other machine-learning methods to identify the model producing the best predictive results. LSTM models were selected because of their ability to model sequential and time-series data, which are intrinsic characteristics of network-traffic and cybersecurity datasets. As contemporary DDoS attacks can generate traffic volumes beyond the capacity of traditional methods, scalable machine-learning, artificial-intelligence, and cloud-based solutions are increasingly being investigated.

3. Long-Short Term Memory

Long Short-Term Memory, originally introduced by Hochreiter and Schmidhuber, is designed to analyse long and non-stationary sequences using memory cells and nonlinear gating mechanisms [24]. Figure 1 shows the LSTM architecture. An LSTM network consists of an input layer, one or more hidden LSTM layers, and an output layer. Each LSTM cell contains three principal gates: the input gate, output gate, and forget gate. These gates regulate the flow of information through the memory cell. At each time step, they determine which information is retained, updated, or discarded. The input gate determines how much new information is added to the cell state:$(i_t)(o_t)(f_t)t$

$$i_t = \sigma(W_{xi}x_t + W_{ih}h_{t-1} + b_i) \quad (1)$$

The input x_t and the previous hidden state h_{t-1} are read into the memory cell. The following equation then generates the forgetting gate:

$$f_t = \sigma(W_{fi}x_t + W_{fh}h_{t-1} + b_f) \quad (2)$$

The following equation produces a new candidate vector C_t' for the cell state:

$$C_t' = tanh(W_{c'i}x_t + W_{c'h}h_{t-1} + b_{c'}) \quad (3)$$

The following changes are made to the cell state C_{t-1}' stands for the previous cell state kept in the memory cell:

$$C_t = f_t.C_{t-1}.I_t.C_t' \quad (4)$$

The output gate O_t is then updated, and the final output h_t is determined using the formulae below:

$$o_t = \sigma(W_{oi}x_t + W_{oh}h_{t-1} + b_o) \quad (5)$$

$$h_t = o_t \odot tanh(c_t) \quad (6)$$

Here, W stands for the synaptic weight matrix; x_t is the real input; b stands for the bias vectors; and vector c'_t holds potential additions to the cell's present state. The prior output of the LSTM at moment $t-1$ is represented by h_{t-1}. $\sigma(.)$ and $tanh()$ are the related activation functions.

4. Firefly Algorithm (FA)

The Firefly Algorithm is based on variations in light intensity and attractiveness among candidate solutions [25, 26]. Brightness represents the quality of a candidate solution and is related to the objective-function value. Assume that the population consists of a swarm of fireflies and that each firefly represents a candidate solution. The brightness of a firefly at its current position is defined as follows:$x_i f(x_i)(x)$

$$I_i = f(x_i), 1 \le i \le n \quad (7)$$

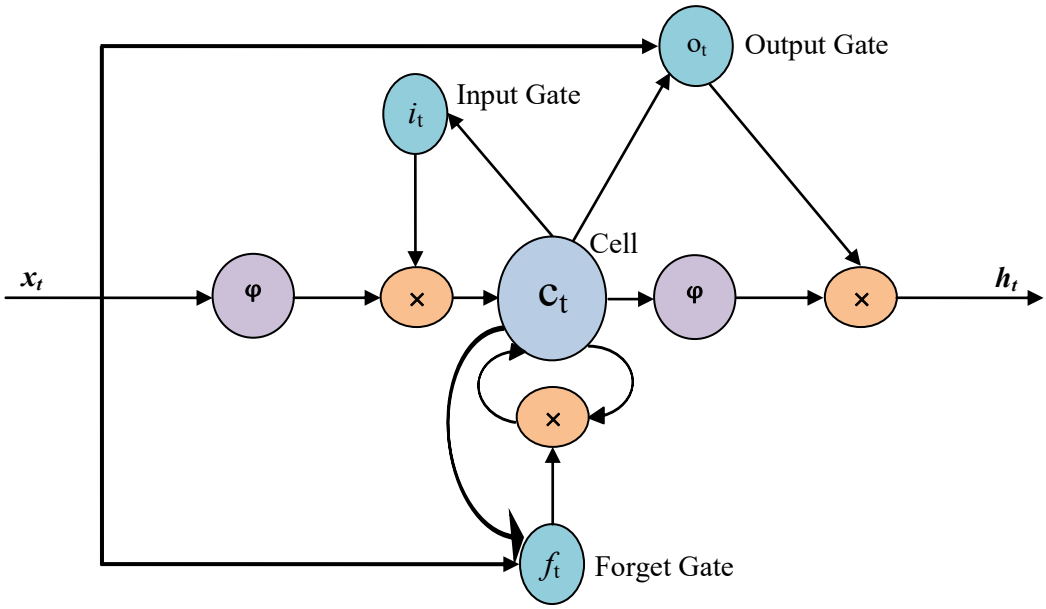

Figure 1. Architecture of LSTM.

The perceived light intensity and attractiveness decrease with distance. Each firefly therefore has an attractiveness value that determines its influence on other members of the swarm. For fireflies located at different positions, attractiveness varies according to their separation distance:$(\beta)ab$

$$r_{ij} = \|x_i - x_j\| \quad (8)$$

The attractiveness of a firefly is defined as follows:

$$\beta(r) = \beta_0 e^{-\gamma r^2} \quad (9)$$

Here, the attractiveness parameter represents the initial attractiveness, while the light-absorption coefficient controls the decrease in attractiveness with distance. A brighter firefly at position x_j attracts a less bright firefly at position x_i. The corresponding position-update rule is expressed as follows:$\beta_0(r=0)\gamma j x_j i x_i$

$$x_i(t+1) = x_i(t) + \beta_0 e^{-\gamma r^2}(x_i - x_j) \quad (10)$$

5. Partial Opposition-Based Learning

5.1 Concept of POBL

POBL is an updated version of the opposition-based learning scheme used in adaptive differential evolution. In a multidimensional search space, the complete opposite point is constructed from the opposite values of all dimensions. If a subset of dimensions is selected, the corresponding partial opposite points can be defined as follows:$\begin{bmatrix} x_1 & \acute{x}_2 & \acute{x}_3 \\ \acute{x}_1 & x_2 & \acute{x}_3 \end{bmatrix} XDp\acute{\chi}$

$$p\acute{\chi}1 = \begin{bmatrix} p\acute{\chi}_1^1 \\ p\acute{\chi}_2^1 \\ \cdots \\ p\acute{\chi}_d^1 \end{bmatrix}_{D\times 1} = \begin{bmatrix} x_1 & \acute{x}_2 & \acute{x}_3 & \ldots & \acute{x}_D \\ \acute{x}_1 & x_2 & \acute{x}_3 & \ldots & \acute{x}_D \\ \vdots & \vdots & \vdots & \ddots & \ldots \\ \acute{x}_1 & \acute{x}_2 & \acute{x}_3 & \ldots & x_D \end{bmatrix}_{D\times D} \quad (11)$$

The superscript 1 in $p\acute{\chi}1$ indicates the degree of partial opposition. The above-mentioned partial opposing points are of order or degree one, because each point only has one original number in one dimension. Therefore, a point χ in D-dimensional space possesses partial opposite points.

5.2 FA with POBL

This study proposes a modified Firefly Algorithm incorporating partial opposition-based learning, termed POFA. The proposed method extends the conventional FA by introducing three principal stages: opposition-based initialization, generation jumping, and best-individual jumping.

5.2.1 Opposition-Based Initialization

The initialization of the i^{th} firefly's j^{th} dimension is set as follows:

$$X_{ij}(t) = X_j^{min} + \left(X_j^{max} - X_j^{min}\right).rand_{ij}^{u}(t) \quad (12)$$

where the random value is uniformly distributed in the interval [0,1]. After the initial firefly positions are generated, their opposite positions are also evaluated. Finally, the highest-fitness candidates are selected from the combined original and opposite populations.$rand_{ij}^{u}(t)i^{th}\ X_i(t)$

5.2.2 Generation Jumping

In the dynamic search space, the j^{th} element $\acute{x_{ij}}(t)$ of the opposite position $\acute{x_i}(t)$ of the i^{th} firefly is determined as:

$$\acute{x_{ij}}(t) = a_j(t) + b_j(t) - \alpha_{ij}(t).x_{ij}(t), \quad (13)$$

The following is the derivation of $a_j(t), b_j(t)$ where $a_j(t), b_j(t)$ is the dynamic search space range:

$$a_j(t) = \min_{\forall i}\{x_{ij}(t)\} \quad (14)$$

$$b_j(t) = \max_{\forall i}\{x_{ij}(t)\} \quad (15)$$

Furthermore, the dynamic constriction factor for the i^{th} firefly in j^{th} dimension is represented by $\alpha_{ij}(t)$, which is employed to improve convergence speed and successfully break out of local minima, hence enhancing the capacity for global search. Here is the definition of α_{ij}:

$$\alpha_{ij}(t) = 1 - \eta.r_{ij}^{c}(t) \quad (16)$$

where $r_{ij}^{c}(t)$ is the origin-centered, scale-parameter-one random number that is Cauchy distributed as

$$f(x) = \frac{1}{\pi}.\frac{s}{s^2+x^2}, -\infty < x < \infty \quad (17)$$

S is the scaling parameter in this case. The Cauchy distributed function is

$$F_s(x) = \frac{1}{2} + \left(\frac{1}{\pi}\right) arctan\left(\frac{x}{s}\right) \quad (18)$$

Every 50 generations, $\eta = \beta \times \eta$, where β is a random number, and η starts at 1.0. Upon computing the opposite position, distinct orders are obtained. The partial opposite positions:

$$k = \lceil rand() \times (D-1) \rceil \quad (19)$$

Here, the parameter specifying full opposition corresponds to the complete opposite point, whereas the parameter specifying the original location retains the initial coordinate. A uniformly distributed random number in the interval [0,1] determines the selected dimensions. The following expression is used to obtain the indices of the elements replaced during partial opposition:$1 < k < Dk = 0k = Drand()l(1 \le l \le D)x_{il}(t)$

$$l = \lceil rand() \times D \rceil \quad (20)$$

The partial opposite vector $p\acute{\chi}_i^k(t)$is obtained by substituting the $x_{il}(t)$ of the original vector $x_i(t)$for the l^{th} element of the opposing vector $\acute{x}_i(t)$.

$$\acute{x}_{il}(t) = x_{il}(t) \quad (21)$$

5.2.3 Best Individual Jumping

In the current swarm, create the difference-offspring $y^{gbest}(t)$ of the global $x^{gbest}(t)$.

$$y^{gbest}(t) = \chi^{gbest}(t) + F.(\chi_{r1}(t) - \chi_{r2}(t)) \quad (22)$$

where two mutually exclusive random integers chosen from the range [1, NP] are denoted by r1 and $r_2(\neq r_1)$. Two mutually exclusive randomly picked particles are $\chi_{r1}(t)$ And$\chi_{r2}(t)$. The scale factor, F, is used to scale the difference $(\chi_{r1}(t) - \chi_{r2}(t))$. It is bounded between 0 and 0.2.

6. Proposed POFA-LSTM

The performance of a deep-learning algorithm is strongly influenced by its hyperparameter configuration. Optimal values are identified by evaluating model performance across candidate configurations. Hyperparameters are configuration variables specified before training and are not directly learned from the training data. Hyperparameter optimization therefore seeks a configuration that minimizes the model's objective function. For LSTM models, important hyperparameters include the learning rate, number of hidden units, weights, and biases. This study combines POBL and FA to exploit the complementary strengths of both methods. POBL and FA are optimization techniques used to search for near-global solutions to complex engineering and scientific problems. However, each method has strengths and limitations arising from its search mechanism. The hybrid method is intended to balance global exploration and local exploitation. FA primarily supports global exploration, while POBL improves population diversity and local refinement. These characteristics can reduce premature convergence, improve convergence speed, and support the identification of more effective solutions. POFA is used to optimize the LSTM model for distinguishing cyberattacks in a cloud environment. The workflow begins with data preparation, including data partitioning and normalization. The number of hidden neurons is selected to reduce the risk of underfitting or overfitting, while the learning rate controls the step size used to update model weights during training. The trained LSTM classifies each traffic instance as either normal or malicious.

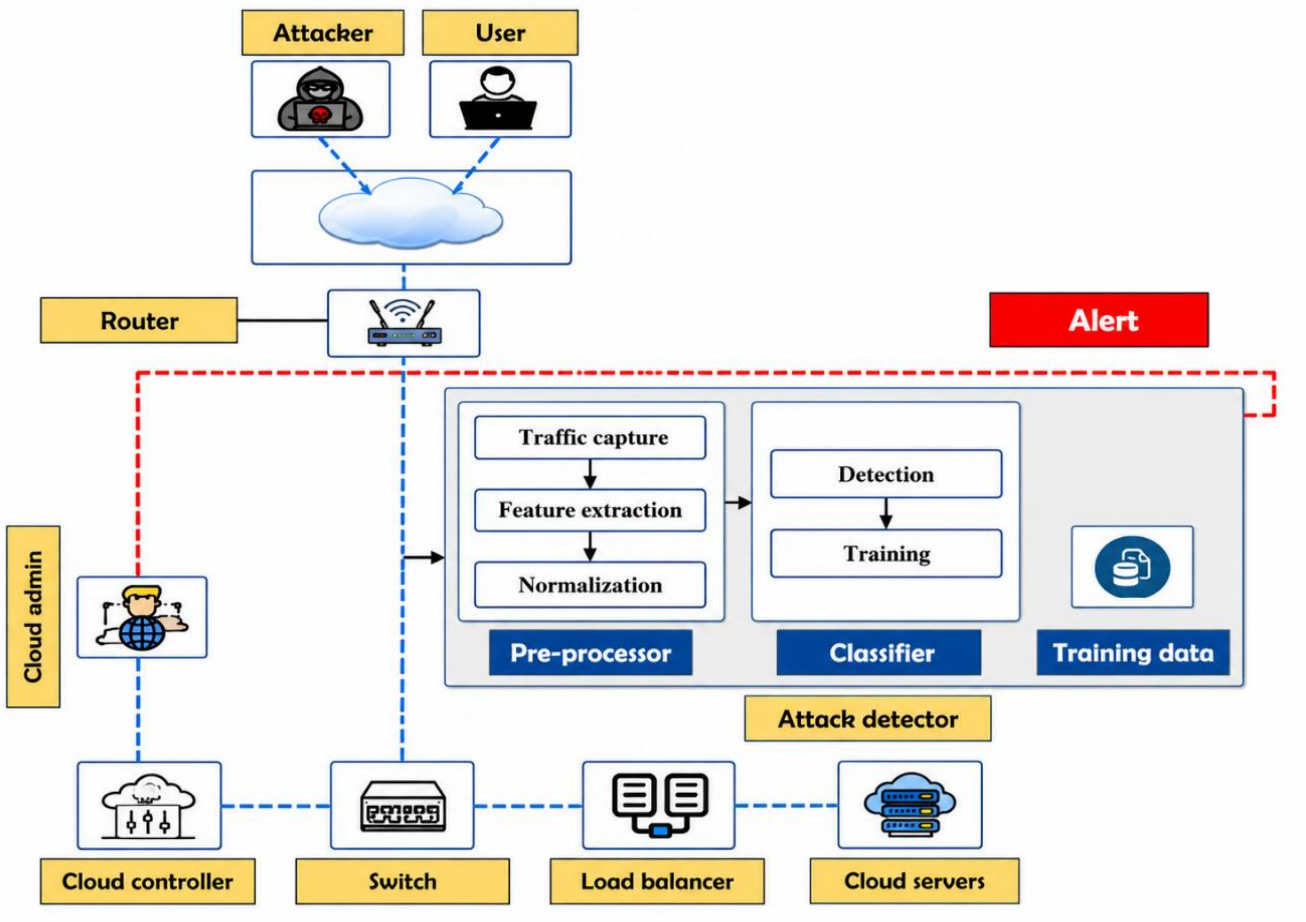

Figure 2. Schematic representation of a proposed method.

The hybrid method can increase detection accuracy while accelerating calculations. The definition of the objective function is as follows:

$$MSE = \frac{1}{N}(y_i - \hat{y}_i)^2 \quad (23)$$

where the forecasted and actual values are used to compute the objective function. Algorithm 1 summarizes the development of the optimized LSTM method. The normalized data are supplied to the LSTM model, while POFA searches for suitable hyperparameter values by minimizing the mean-squared error. Each firefly is treated as a search agent representing a candidate solution. During each iteration, the position of every search agent is updated using the FA movement rule and the partial opposition mechanism. When the stopping criterion is satisfied, the best LSTM hyperparameter configuration is retained; otherwise, the iterative search continues. The proposed approach has three potential advantages: (i) LSTM can model temporal and sequential dependencies in network traffic; (ii) FA can improve model generalization by optimizing the hyperparameters; and (iii) POBL can improve search efficiency and reduce the likelihood of premature convergence. By combining LSTM for temporal modelling, FA for optimization, and POBL for population diversity, the proposed method is designed for complex and dynamic detection problems. Compared with PSO and GA, POFA evaluates partially opposite candidate solutions during the search process, which may accelerate convergence.$y_i \hat{y}_i$

GA may take longer since it depends on genetic operations like crossover and mutation, whereas PSO may have trouble with sluggish convergence in high-dimensional situations. In contrast to a full opposition-based method, POFA concentrates on a subset of opposed solutions (partial opposition), which lowers the computational overhead and increases efficiency without sacrificing solution quality. A schematic representation is shown in Figure 2.

Algorithm 1. Optimized LSTM

Step 1: Normalize the data samples to values between 0 and 1 using min-max normalization. Divide the normalized samples into training and testing sets.

Step 2: Initialize the required POBL, FA, and LSTM parameters.

Step 3: Use POFA to optimize the selected LSTM hyperparameters.

Step 3.1: Construct the LSTM model and train it iteratively while updating the loss function.

Step 3.2: Evaluate the fitness of each firefly and identify both the best individual solution and the global best solution.

Step 3.3: Update the position of each firefly using the FA movement rule and the partial opposition mechanism.

Step 3.4: When the stopping criterion is satisfied, retain the optimal LSTM hyperparameters. Otherwise, return to Step 3.1 and continue the iterative search.

Step 4: Evaluate the trained model using the testing dataset.

Step 5: Select the LSTM configuration associated with the best POFA solution.

Step 6: Use the optimized LSTM model for DDoS-attack detection and assess its performance using the five selected metrics.

7. Experimental Results and Analysis

Cyberattack detection involves identifying unauthorized or malicious activities in interconnected devices, networks, and systems within a cloud-computing environment. The diverse devices, constrained resources, and heterogeneous communication protocols used in cloud systems create specific challenges for attack detection. Experimental evaluation is therefore necessary to assess the effectiveness of cyberattack-detection methods. To improve detection accuracy and convergence speed, this study proposes a DDoS-detection technique based on an LSTM model optimized using POFA. This section compares the proposed method with LSTM-IFA, LSTM-FA [29], LSTM-APSO [30], PSO-LSTM [31], GA-LSTM [32], conventional LSTM [33], BPNN [34], and SVM [35].

The proposed and comparative methods were implemented in MATLAB R2019 on a Windows 11 computer equipped with an Intel Core i5 processor and 8 GB of RAM. This environment was used to obtain the reported experimental results. The methods were evaluated on four benchmark cybersecurity datasets using five performance metrics: accuracy, sensitivity, specificity, precision, and F-score.

7.1 Datasets

Four datasets were used for experimental evaluation. Their principal characteristics are summarized in Table 1.

NSL-KDD: The “DOS, U2R, R2L, and Probe” attacks are represented, and it has 160,367 samples, and each sample has 41 features [36].

ISCXIDS2012 [37]: The traffic was collected over seven days, between Friday and Thursday. On the

other days, samples with normal and attack are present in the traffic, while only normal examples are present on Friday.

UNSW-NB15 [38]*:* A subset of 257,673 examples and 2,540,044 examples with 48 features and nine attacks is used. The training sample has 175,341 examples, whereas the testing set contains 82,332 examples.

CIC-IDS2017 [39]: *The dataset contains traffic collected over five days, from Monday to Friday. Monday contains only normal traffic, whereas the remaining days contain both attack and normal traffic.*

7.2 Preprocessing

Categorical features are converted into numerical form using one-hot encoding. This method represents each categorical value as a binary vector. For example, if the feature "protocol_name" has the possible values "ICMP," "UDP," and "TCP," then TCP, UDP, and ICMP can be represented as (1,0,0), (0,1,0), and (0,0,1), respectively. Because the dataset contains both numerical and categorical variables, categorical features are first encoded and numerical features are subsequently scaled using min-max normalization [40, 41]:

$$z = \frac{x - min(x)}{max(x) - min(x)} \quad (24)$$

The current value is normalized using the minimum and maximum values of the corresponding feature. The samples were divided into 70% for training and 30% for testing. The selected hyperparameter settings are shown in Table 2.$x\ max(x)\ min(x)$

7.3 Performance Indicators

Performance metrics are used to assess how effectively a detection system distinguishes legitimate traffic from malicious traffic, reduces false positives, and responds to threats. These quantitative measures support the evaluation and optimization of DDoS-detection systems.

The detection algorithm was investigated using five performance measures: accuracy, precision, sensitivity, specificity, and F-measure.

Accuracy is the proportion of all instances that are correctly classified as either attacks or normal traffic. A high accuracy value indicates that the detection system correctly classifies a large proportion of both benign and malicious traffic. Accuracy is calculated as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (25)$$

Precision is the proportion of predicted positive instances that are true positives. It indicates the system's ability to limit false alarms and avoid incorrectly blocking legitimate users:

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (26)$$

Specificity is the proportion of actual negative instances correctly classified as negative:

$$Specificity = \frac{TN}{TN+FP} \times 100\% \quad (27)$$

Sensitivity, also termed recall or the true-positive rate, is the proportion of actual positive instances correctly identified by the model:

$$Sensitivity = \frac{TP}{TP+FN} \times 100\% \quad (28)$$

Table 1. Data explanation

Datasets	Attacks	Features	Training	Testing	Total
NSL-KDD	4	41	125973	34394	160367
ISCXIDS2012	4	19	97035	41701	139006
CIC-IDS 2017	8	78	1744184	747505	2491689
UNSW-NB15	9	48	175341	82332	257673

Table 2. parameter settings

LSTM				FA	
Parameters	**Values**	**Parameters**	**Values**	**Parameters**	**Values**
Activation	TanH and Sigmoid	Expected error	0.005	N	50
Loss	MSE	Weight range	[0.5, -0.5	β_0	1
Learning rate	0.001	Hidden neurons	100	λ	1
Epochs	300	Dropout	0.1	T	200

The F-measure is the harmonic mean of precision and recall and provides a single measure that balances both quantities:

$$F - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100\% \quad (29)$$

True positive (TP) indicates that the model correctly identifies DDoS traffic as an attack. False positive (FP) indicates that normal traffic is incorrectly classified as an attack. True negative (TN) indicates that normal traffic is correctly classified as normal. False negative (FN) indicates that DDoS traffic is incorrectly classified as normal. By examining these metrics, researchers can select and improve DDoS-detection methods according to specific security requirements while balancing detection accuracy and network efficiency.

7.4 Results Analysis

Experimental data are required to validate the effectiveness and efficiency of the proposed POFA-optimized LSTM method. The proposed and baseline methods were evaluated during both the training and testing phases. Tables 3-6 present the training performance of the methods on the NSL-KDD, ISCXIDS2012, UNSW-NB15, and CIC-IDS2017 datasets, respectively. Figures 3-6 show the corresponding training results, while Figures 7-10 present the testing results.

The experimental results show that the proposed method achieved higher accuracy than the evaluated baseline algorithms. The LSTM-POFA model obtained accuracy values of 0.9945, 0.9866, 0.9822, and 0.9892 on the NSL-KDD, ISCXIDS2012, UNSW-NB15, and CIC-IDS2017 datasets, respectively. For NSL-KDD, the model achieved a sensitivity of 0.9879, specificity of 0.9956, precision of 0.9944, and F-score of 0.9948. For ISCXIDS2012, it achieved a sensitivity of 0.9999, specificity of 0.9995, precision of 0.9988, and F-score of 0.9910.

Table 3. Results of training performance for the NSL-KDD

Methods	Accuracy	Sensitivity	Specificity	Precision	F-Score
LSTM-POFA	0.9945	0.9879	0.9956	0.9944	0.9948
LSTM-IFA	0.9928	0.9867	0.9866	0.9804	0.9900
LSTM-FA	0.9792	0.9835	0.9677	0.9744	0.9833
LSTM-APSO	0.9673	0.9688	0.9466	0.9563	0.9798
LSTM-PSO	0.9362	0.9473	0.9266	0.9366	0.9693
LSTM-GA	0.9156	0.9206	0.9089	0.9155	0.9573
LSTM	0.8945	0.9144	0.8825	0.8965	0.9364
BPNN	0.8564	0.8843	0.8597	0.8573	0.9115
SVM	0.7888	0.8610	0.8397	0.8467	0.8744

Table 4. Results of training performance for the ISCXIDS-2012

Methods	Accuracy	Sensitivity	Specificity	Precision	F-Score
LSTM-POFA	0.9866	0.9999	0.9995	0.9988	0.9910
LSTM-IFA	0.9848	0.9981	0.9945	0.9956	0.9890
LSTM-FA	0.9800	0.9799	0.9745	0.9918	0.9677
LSTM-APSO	0.9645	0.9566	0.9567	0.9799	0.9455
LSTM-PSO	0.9598	0.9345	0.9355	0.9745	0.9043
LSTM-GA	0.9320	0.9106	0.9143	0.9623	0.8355
LSTM	0.9122	0.8803	0.8890	0.9478	0.8144
BPNN	0.8890	0.8640	0.8610	0.9255	0.7843
SVM	0.8588	0.8590	0.8244	0.8788	0.7682

Table 5. Results of training performance for the UNSW-NB15

Methods	Accuracy	Sensitivity	Specificity	Precision	F-Score
LSTM-POFA	0.9822	0.9940	0.9867	0.9899	0.9933
LSTM-IFA	0.9810	0.9899	0.9833	0.9863	0.9843
LSTM-FA	0.9745	0.9799	0.9793	0.9795	0.979
LSTM-APSO	0.9719	0.9677	0.9699	0.9577	0.9683
LSTM-PSO	0.9588	0.9478	0.9466	0.9355	0.9588
LSTM-GA	0.9376	0.9299	0.9255	0.9166	0.9366
LSTM	0.9168	0.9133	0.9155	0.8577	0.9155
BPNN	0.8289	0.8856	0.8793	0.8033	0.8833
SVM	0.8043	0.8645	0.8689	0.7744	0.8693

Table 6. Results of training performance for the CICIDS-2017

Methods	Accuracy	Sensitivity	Specificity	Precision	F-Score
LSTM-POFA	0.9892	0.9933	0.9928	0.9888	0.9868
LSTM-IFA	0.9880	0.9899	0.9890	0.9833	0.9844
LSTM-FA	0.9856	0.9645	0.9736	0.9783	0.9789
LSTM-APSO	0.9780	0.9532	0.9589	0.9654	0.9578
LSTM-PSO	0.9689	0.9344	0.9366	0.9478	0.9367
LSTM-GA	0.9534	0.9289	0.9155	0.9279	0.9178
LSTM	0.9422	0.9154	0.8599	0.9034	0.8833
BPNN	0.9155	0.8673	0.8155	0.8822	0.8590
SVM	0.8744	0.8566	0.8022	0.8689	0.8478

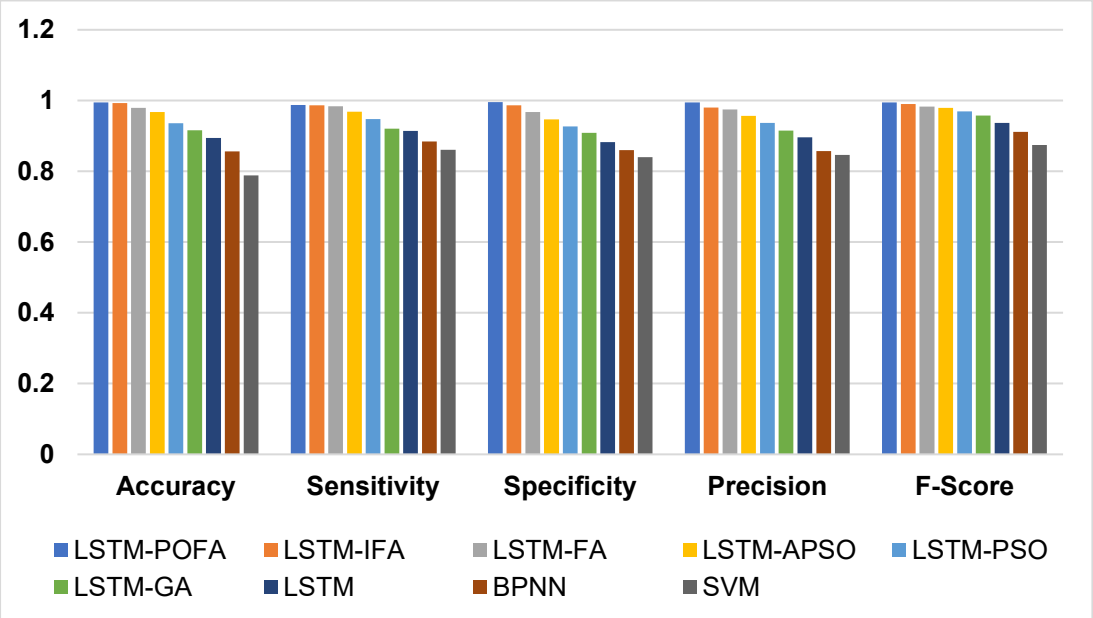

Figure 3. Training performance for NSL-KDD

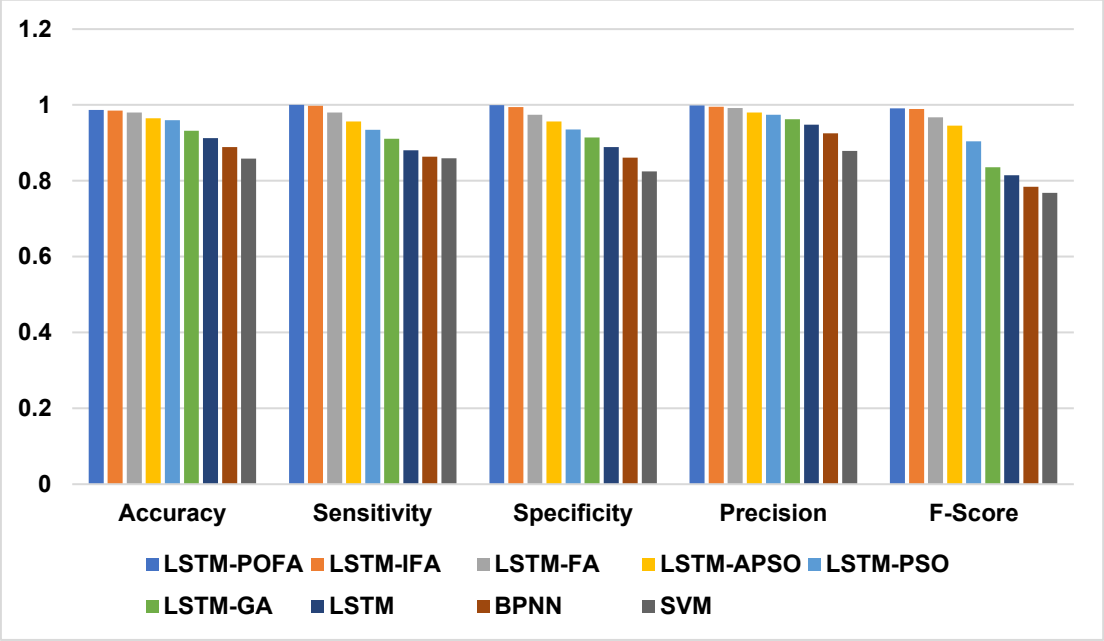

Figure 4. Training performance for ISCXIDS-2012.

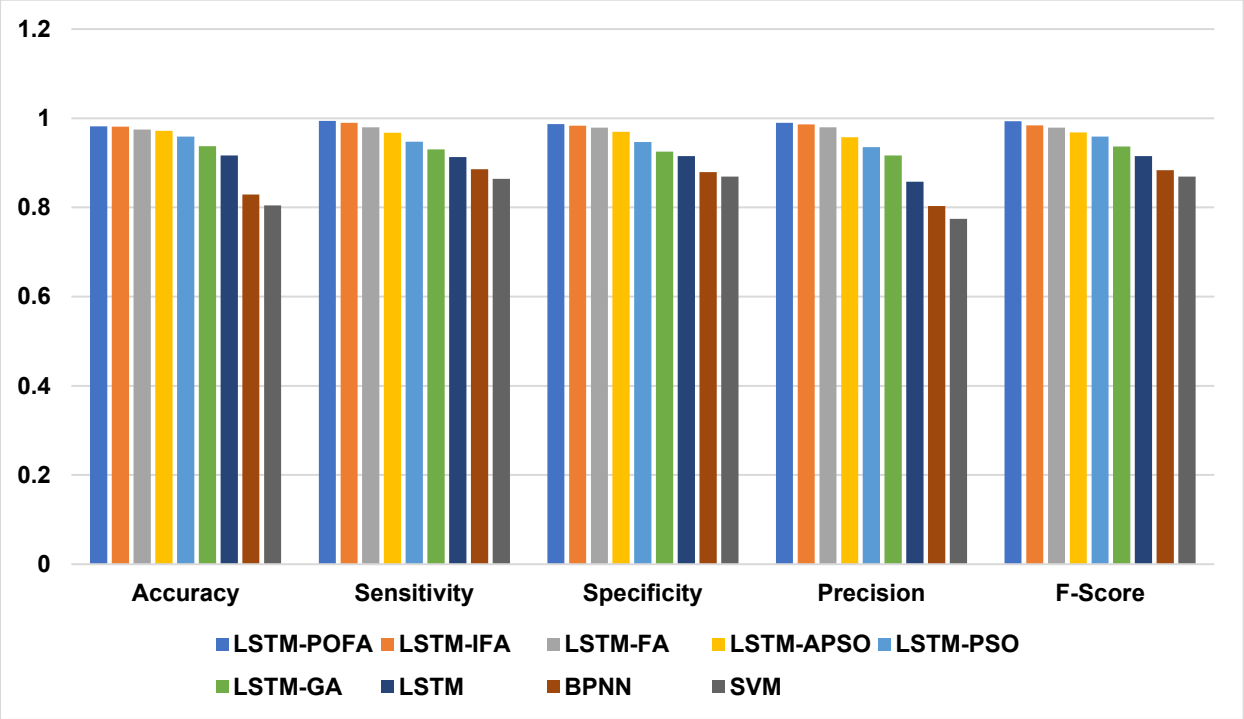

Figure 5. Training performance for UNSW-NB15.

For UNSW-NB15, LSTM-POFA achieved a sensitivity of 0.9940, specificity of 0.9867, precision of 0.9899, and F-score of 0.9933. For CIC-IDS2017, it achieved a sensitivity of 0.9933, specificity of 0.9928, precision of 0.9888, and F-score of 0.9868. Box plots can be used to compare the distribution of success rates across models, datasets, and experimental configurations. Figures 11-14 show the average accuracy of LSTM, LSTM-IFA, and LSTM-POFA on the four datasets. Previous studies have also investigated FA [29], APSO [30], PSO [31], and GA [32] for improving LSTM performance.

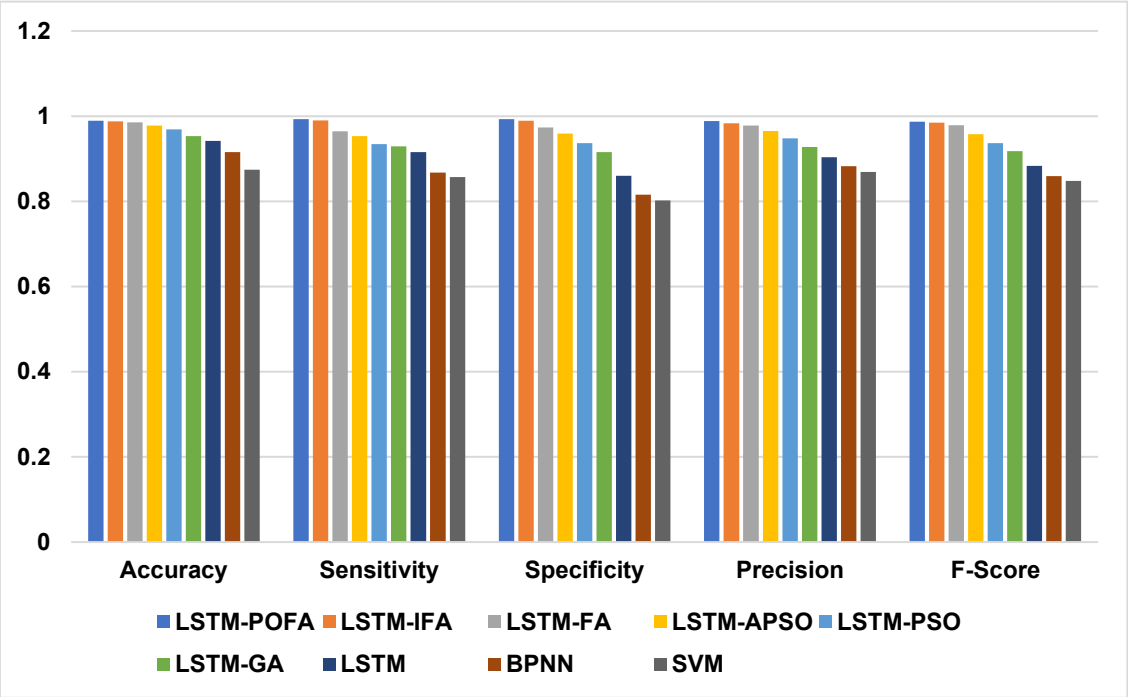

Figure 6. Training performance for CICIDS-2017

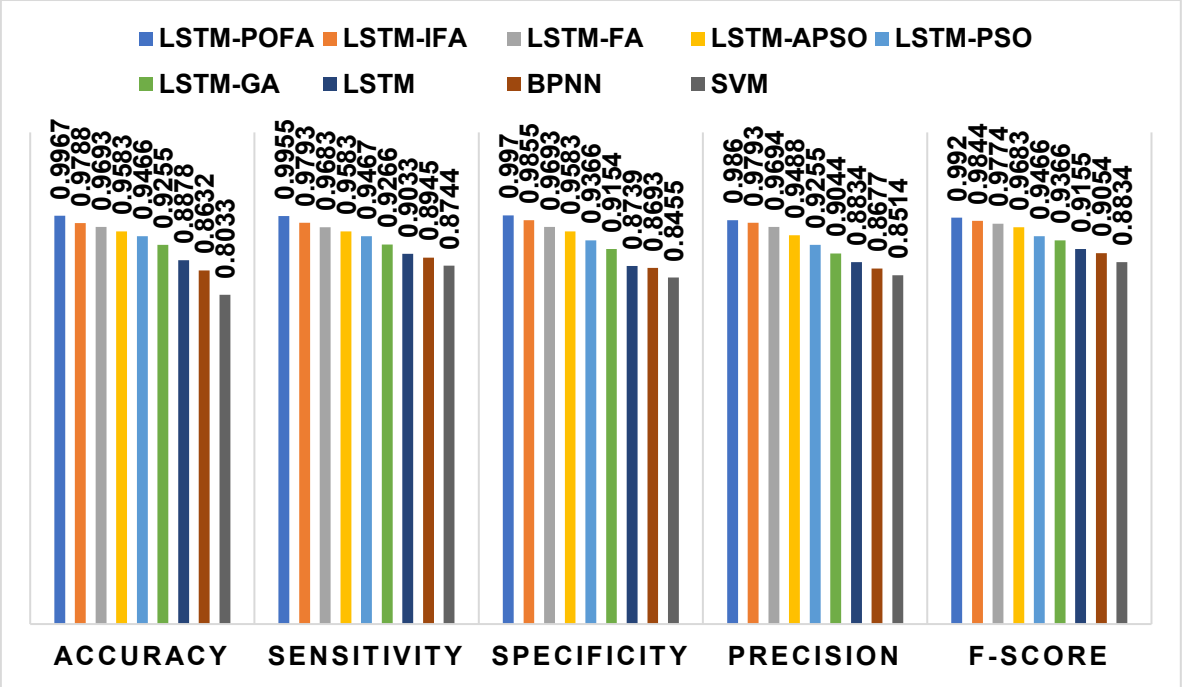

Figure 7. Results of testing performance for the NSL-KDD.

Swarm-intelligence algorithms, which are inspired by the collective behaviour of natural systems, have demonstrated effectiveness in global optimization problems by balancing exploration and exploitation. They can be flexible in high-dimensional search spaces; however, despite their effectiveness, they have several limitations when used for LSTM hyperparameter optimization. GA, PSO, and FA may converge prematurely to a local optimum when population diversity is insufficient. The Firefly Algorithm has been applied to the hyperparameter optimization of deep-learning models, including LSTM [26]. However, FA may suffer from insufficient population diversity because of random population initialization and may consequently converge to a local optimum [42]. The present study therefore incorporates POBL into FA to improve population diversity and support escape from local optima.

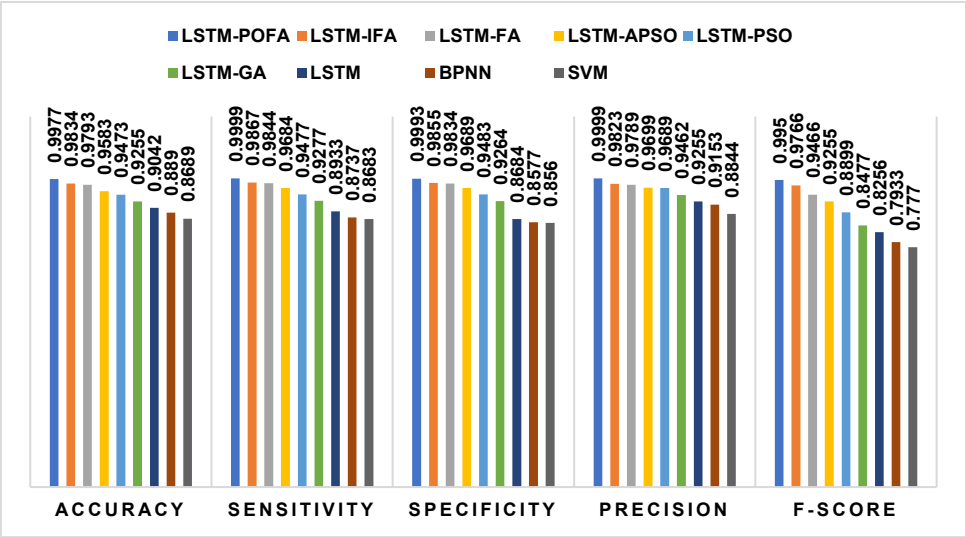

Figure 8. Results of testing performance for the ISCXIDS-2012.

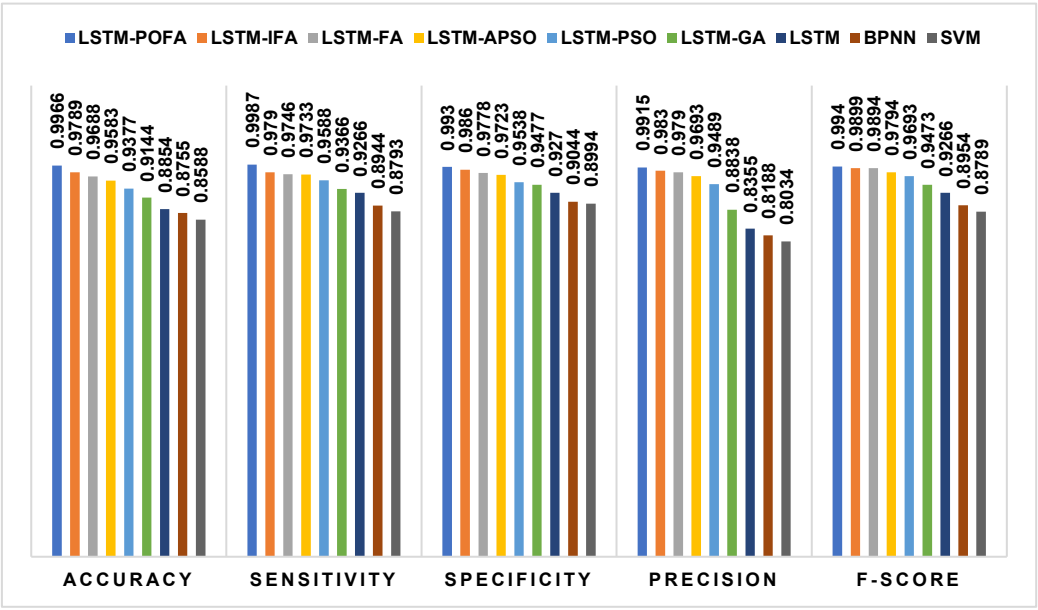

Figure 9. Results of testing performance for the UNSW-NB15.

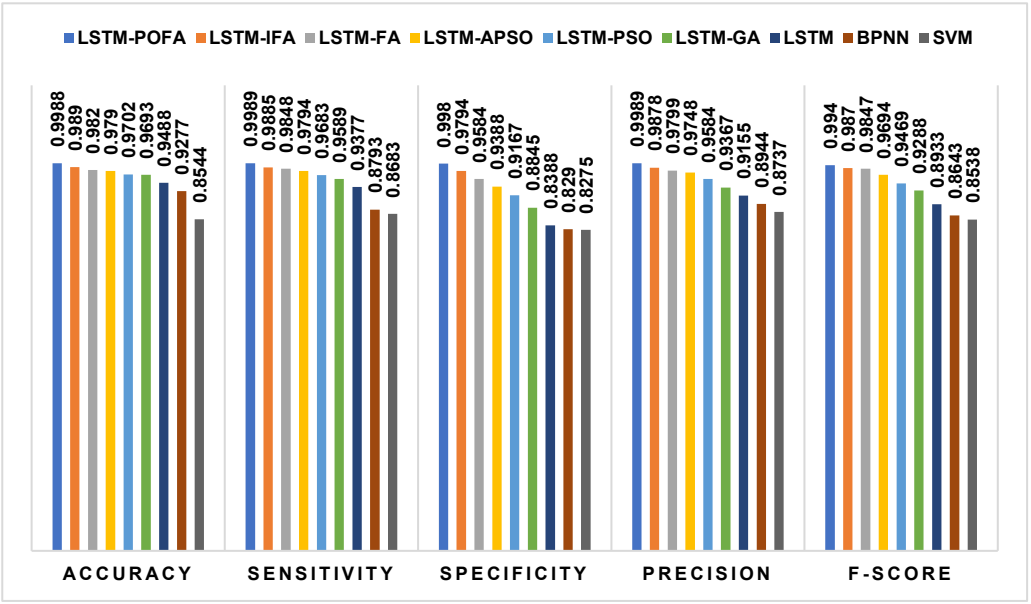

Figure 10. Testing performance results for the CICIDS-2017.

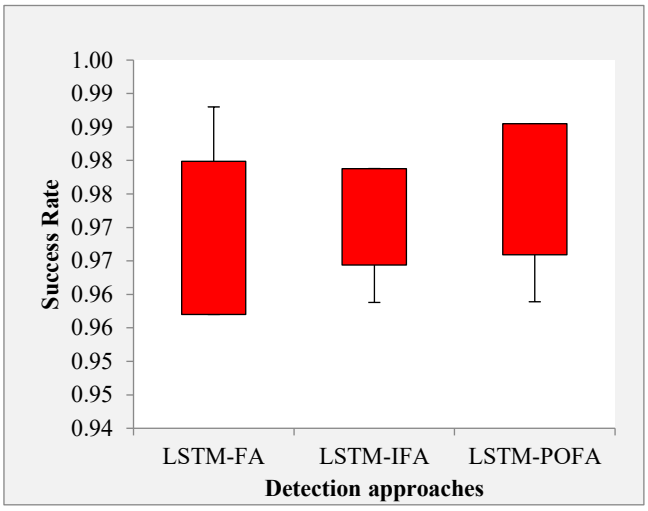

Figure 11. The success rate for the NSL-KDD dataset.

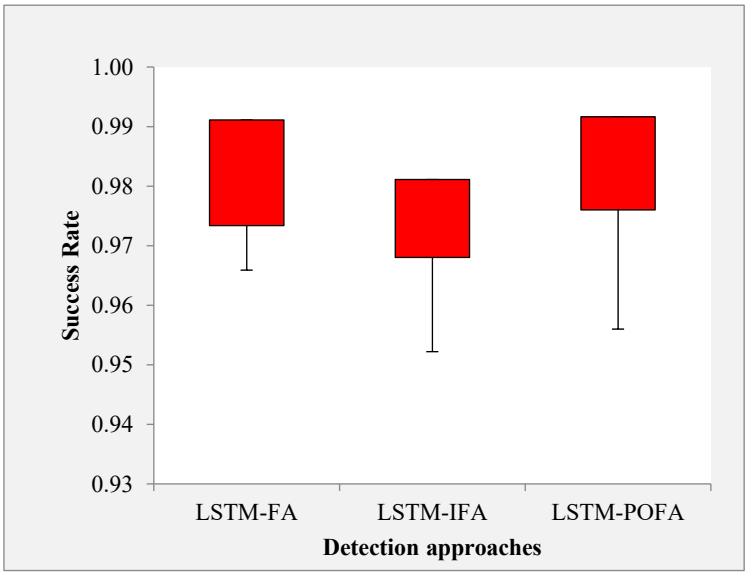

Figure 12. The success rate for the ISCXIDS-2012 dataset.

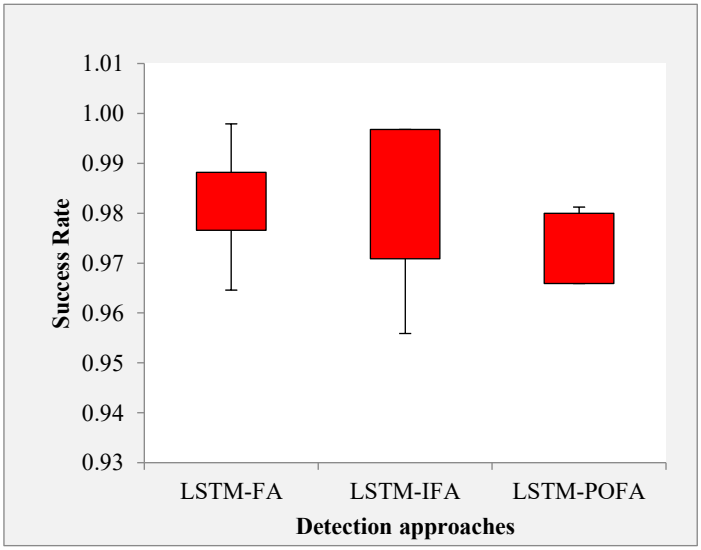

Figure 13. The success rate for the UNSW-NB15 dataset.

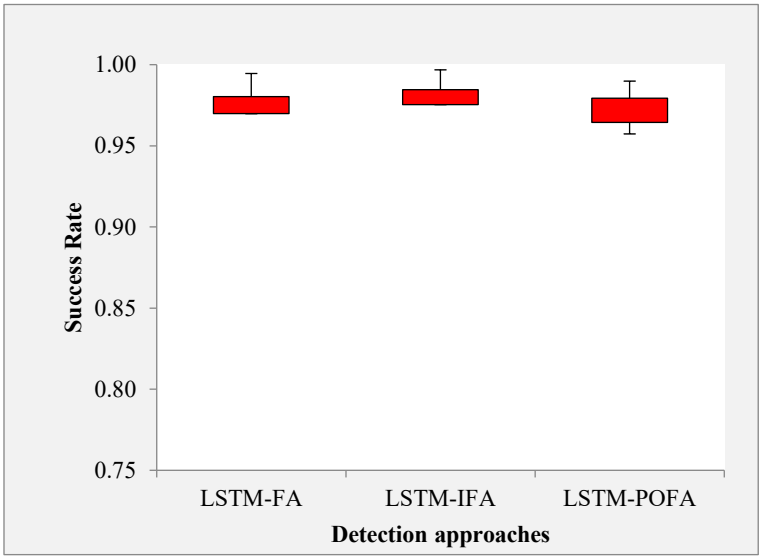

Figure 14. The success rate for the CICIDS-2017 dataset.

The optimized LSTM provides an adaptive detector capable of analysing incoming network traffic in near real time. The experiments on benchmark datasets indicate that the proposed system can detect attacks with high accuracy; however, its real-time performance must still be validated in a live cloud environment. The evaluated LSTM-based models produced predictions within comparable execution times, although inference latency may vary according to the rate and volume of incoming requests.

8. Conclusions

This study proposed a POFA-enhanced LSTM network for detecting DDoS attacks. The approach combines the sequential-data modelling capability of LSTM with the hyperparameter-search capability of POFA. The experimental results indicate that POFA improved the LSTM hyperparameters and increased detection accuracy across the evaluated benchmark datasets. The proposed model detected DDoS attacks in complex and dynamic cloud-traffic conditions more accurately than the evaluated conventional machine-learning and LSTM models. The findings suggest that the optimized LSTM may be suitable for near-real-time applications, although computational overhead and inference latency require further validation. Future studies should investigate combinations of deep-learning models with additional optimization strategies and extend the model to other types of cyberattacks. Deployment and testing across different real-world cloud architectures would also provide evidence regarding operational robustness and scalability. The combined approach may be computationally intensive because it requires repeated LSTM training during FA-based hyperparameter optimization and POBL operations. In addition, extensive hyperparameter search may increase the risk of overfitting the validation data. Future work should therefore develop lightweight LSTM variants, evaluate computational cost explicitly, and investigate hybrid neural-network architectures that improve feature extraction and processing efficiency.

References

[1] G. Somani, M.S. Gaur, D. Sanghi, M. Conti, R. Buyya, DDoS attacks in cloud computing: Issues, taxonomy, and future directions. Computer Communications, 107, (2017) 30-48. https://doi.org/10.1016/j.comcom.2017.03.010

[2] F.S.D. Lima Filho, F.A. Silveira, A. de Medeiros Brito Junior, G. Vargas-Solar, L.F. Silveira, Smart detection: an online approach for DoS/DDoS attack detection using machine learning. Security and Communication Networks, 2019(1), (2019) 1574749. https://doi.org/10.1155/2019/1574749

[3] N.V. RajaSekhar Reddy, N. SreeDivya, B.N. Jagadesh, R. Gandikota, K.K. Lella, B. Pydala, R.Vatambeti, Enhancing anomaly detection: a comprehensive approach with MTBO feature selection and TVETBOOptimized Quad-LSTM classification. Computers and Electrical Engineering, 119, (2024) 109536. https://doi.org/10.1016/j.compeleceng.2024.109536

[4] M. Mittal, K. Kumar, S. Behal, DL-2P-DDoSADF: Deep learning-based two-phase DDoS attack detection framework. Journal of Information Security and Applications, 78, (2023) 103609. https://doi.org/10.1016/j.jisa.2023.103609

[5] R. Priyadarshini, R.K. Barik, (2019) A deep learning based intelligent framework to mitigate DDoS attack in fog environment. Journal of King Saud University-Computer and Information Sciences. 1-7. https://doi.org/10.1016/j.jksuci.2019.04.010

[6] O. Pandithurai, C. Venkataiah, S. Tiwari, N. Ramanjaneyulu, DDoS Attack Prediction using a

honey badger optimization algorithm based feature selection and Bi-LSTM in cloud environment. Expert Systems with Applications, 241, (2024) 122544. https://doi.org/10.1016/j.eswa.2023.122544

[7] N.K. Muthunambu, S. Prabakaran, B. PrabhuKavin, K.S. Siruvangur, K. Chinnadurai, J. Ali, A Novel Eccentric Intrusion Detection Model Based on Recurrent Neural Networks with Leveraging LSTM. Computers, Materials & Continua, 78(3), (2024) 3089-3127. https://doi.org/10.32604/cmc.2023.043172

[8] V. Poornachander, K.S. Kumar, S. Jagadish, DDoS Attack Intrusion Detection System with CNN and LSTM Hybridization. (2024) International Conference on Sustainable Computing and Smart Systems (ICSCSS), IEEE, India. https://doi.org/10.1109/ICSCSS60660.2024.10625330

[9] R. Devendiran, A.V. Turukmane, Dugat-LSTM: Deep learning based network intrusion detection system using chaotic optimization strategy. Expert Systems with Applications, 245, (2024) 123027. https://doi.org/10.1016/j.eswa.2023.123027

[10] V. Sughanthini, P. Bharathisindhu, (2024) DDoS Attack Detection Using Optimized Long Short-Term Based on Partial Opposition-Based Swarm Intelligence Algorithm. International Conference on Sustainable Computing and Smart Systems (ICSCSS), IEEE, India. https://doi.org/10.1109/ICSCSS60660.2024.10624861

[11] A.D. Vibhute, M. Khan, A. Kanade, C.H. Patil, S.V. Gaikwad, K.K. Patel, J.R. Saini, An LSTM-based Novel Near-real-time multiclass network Intrusion Detection System for Complex Cloud Environments. Concurrency and Computation: Practice and Experience, 36(11), (2024) e8024. https://doi.org/10.1002/cpe.8024

[12] D. Sathish, A. Kavitha, (2024) DDoS Attack Detection Using Optimized Long Short-Term Memory Based on Improved Bacterial Foraging Optimization. International Conference on Sustainable Computing and Smart Systems (ICSCSS), IEEE, India. https://doi.org/10.1109/ICSCSS60660.2024.10624895

[13] A. Thangasamy, B. Sundan, L. Govindaraj, A Novel Framework for DDoS Attacks Detection Using Hybrid LSTM Techniques. Computer Systems Science & Engineering, 45(3), (2023) 2553-2567. https://doi.org/10.32604/csse.2023.032078

[14] A.S.A. Issa, Z. Albayrak, DDoS attack intrusion Detection System based on Hybridization of CNN and LSTM. Acta Polytechnica Hungarica, 20(2), (2023)1-19. https://doi.org/10.12700/APH.20.2.2023.2.6

[15] M. Ramzan, M. Shoaib, A. Altaf, S. Arshad, F. Iqbal, Á.K. Castilla, I. Ashraf, Distributed Denial of Service Attack Detection in Network Traffic using Deep Learning Algorithm. Sensors, 23(20), (2023) 8642. https://doi.org/10.3390/s23208642

[16] C. Xu, J. Shen, X. Du, Low-rate DoS attack Detection Method based on Hybrid Deep Neural Networks. Journal of Information Security and Applications, 60, (2021) 102879. https://doi.org/10.1016/j.jisa.2021.102879

[17] A. Bhardwaj, V. Mangat, R. Vig, Hyperband tuned Deep Neural Network with well Posed Stacked Sparse Autoencoder for detection of DDoS attacks in Cloud. IEEE Access, 8, (2020) 181916-181929. https://doi.org/10.1109/ACCESS.2020.3028690

[18] S. Priya, R. Ponmagal, Network Intrusion Detection System based Security System for Cloud Services using Novel Recurrent Neural Network-Autoencoder (nrnn-ae) and Genetic. Advances in Science and Technology, 124, (2023) 729-737. https://doi.org/10.4028/p-076960

[19] E. Deniz, S. Serttaş, Deep learning-based Distributed Denial of Service Detection System in the cloud network. Journal of Scientific Reports-A, 055, (2023) 16-33. https://doi.org/10.59313/jsr-a.1333839

[20] X. Yin, W. Fang, Z. Liu, D. Liu, A novel multi-scale CNN and Bi-LSTM Arbitration Dense Network model for Low-Rate Ddos Attack Detection. Scientific Reports, 14(1), (2024) 5111. https://doi.org/10.1038/s41598-024-55814-y

[21] P. Sathishkumar, A. Gnanabaskaran, M. Saradha, R. Gopinath, DoS Attack Detection using Fuzzy Temporal Deep Long Short-Term Memory Algorithm in Wireless Sensor Network. Ain Shams Engineering Journal, 15(12), (2024) 103052. https://doi.org/10.1016/j.asej.2024.103052

[22] S. Mazumder, S. Neogy, T. Sur, S. Das, A Comparative Assessment of Deep Learning for adaptable DDoS threat detection in cloud computing systems. SN Computer Science, 6(1), (2025) 80. https://doi.org/10.1007/s42979-024-03643-1

[23] P. Kumar, C. Kushwaha, D. Sethi, D. Ghosh, P. Gupta, A. Vidyarthi, Investigating the performance of multivariate LSTM models to predict the occurrence of Distributed Denial of Service (DDoS) attack. PLoS ONE 20(1) (2025) e0313930. https://doi.org/10.1371/journal.pone.0313930

[24] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory. Neural Computation, 9(8), (1997) 1735-

1780. https://doi.org/10.1162/neco.1997.9.8.1735

[25] X.-S. Yang, Firefly algorithms for multimodal optimization. In: Watanabe, O., Zeugmann, T. (eds) Stochastic Algorithms: Foundations and Applications. SAGA 2009. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 5792, (2009) 169-178. https://doi.org/10.1007/978-3-642-04944-6_14

[26] X.-S. Yang, A. Slowik, (2020) Firefly Algorithm. In: Swarm Intelligence Algorithms, CRC Press, 163-174.

[27] J. Revathy, S.K. Jayanthi, (2024) Optimized Long Short-Term Memory Approach using Partial opposition-based Particle Swarm Optimization for Diabetes Detection. In: 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), IEEE, Salem, India. https://doi.org/10.1109/ICAAIC60222.2024.10575455

[28] X.-S. Yang, X. He, Firefly Algorithm: Recent Advances and Applications. International Journal of Swarm Intelligence, 1(1), (2013) 36-50. https://doi.org/10.1504/IJSI.2013.055801

[29] A.M. Anter, M. Ali, Feature Selection Strategy based on Hybrid Crow Search Optimization Algorithm Integrated with Chaos Theory and Fuzzy c-Means Algorithm for Medical Diagnosis Problems. Soft Computing, 24(3), (2020) 1565-1584. https://doi.org/10.1007/s00500-019-03988-3

[30] B. Shao, M. Li, Y. Zhao, G. Bian, Nickel price forecast based on the LSTM neural network optimized by the improved PSO algorithm. Mathematical Problems in Engineering, 2019(1), (2019) 1934796. https://doi.org/10.1155/2019/1934796

[31] P. Wang, J. Zhao, Y. Gao, M.A. Sotelo, Z. Li, Lane Work-schedule of Toll Station based on Queuing Theory and PSO-LSTM model. IEEE Access, IEEE, 8, (2020) 84434-84443. https://doi.org/10.1109/ACCESS.2020.2992070

[32] A.S. Santra, J.-L. Lin, Integrating Long Short-Term Memory and Genetic Algorithm for Short-Term Load Forecasting. Energies, 12(11), (2019) 2040. https://doi.org/10.3390/en12112040

[33] X. Liang, T. Znati, (2019)A Long Short-Term Memory Enabled Framework for DDoS detection. In: 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, Waikoloa, HI, USA. https://doi.org/10.1109/GLOBECOM38437.2019.9013450

[34] R.M. Saad, M. Anbar, S. Manickam, E. Alomari, An intelligent ICMPv6 DDoS Flooding-Attack Detection Framework (V6IIDS) using back-Propagation Neural Network. IETE Technical Review, 33(3), (2016) 244-255. https://doi.org/10.1080/02564602.2015.1098576

[35] H.A. Sakr, M.M. El-Tahlawi, A. Abdelhafeez, M.I. El-Afifi, M.R. Abdellah, Machine learning-based detection of DDoS attacks on IoT devices in multi-energy systems. Egyptian Informatics Journal, 28, (2024) 100540. https://doi.org/10.1016/j.eij.2024.100540

[36] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, (2009) A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, IEEE, Ottawa, ON, Canada. https://doi.org/10.1109/CISDA.2009.5356528

[37] A. Shiravi, H. Shiravi, M. Tavallaee, A.A. Ghorbani, Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Computers & Security, 31(3), (2012) 357-374. https://doi.org/10.1016/j.cose.2011.12.012

[38] N. Moustafa, J. Slay, (2015) UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS), IEEE, Canberra, ACT, Australia. https://doi.org/10.1109/MilCIS.2015.7348942

[39] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSP, 1, (2018) 108-116. https://doi.org/10.5220/0006639801080116

[40] K. Velusamy, R. Amalraj, Cascade correlation neural network with deterministic weight modification for predicting stock market price. In: IOP Conference Series: Materials Science and Engineering, 1110(1), (2021) 012005. https://doi.org/10.1088/1757-899X/1110/1/012005

[41] K. Velusamy, R. Amalraj, (2017) Performance of the Cascade Correlation Neural Network for Predicting the Stock Price. In: 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), IEEE, Coimbatore, India. https://doi.org/10.1109/ICECCT.2017.8117824

[42] H. Wang, W. Wang, H. Sun, Firefly Algorithm with Generalised Opposition-based Learning. International Journal of Wireless and Mobile Computing, 9(4), (2015) 370-376. https://doi.org/10.1504/IJWMC.2015.074028

Authors Contribution Statement

Both the authors equally contributed and approved the final version of this work.

Funding

The authors declare that no funds, grants or any other support were received during the preparation of this manuscript.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Data Availability

The data supporting the findings of this study can be obtained from the corresponding author upon reasonable request.

Has this article screened for similarity?

Yes