Received: May 14, 2023. Revised: July 4, 2023.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

DDoS Attack Detection using Back Propagation Neural Network Optimized by Bacterial Colony Optimization

M. Arunadevi[1]* V. Sathya[1]

[1]*Department of Computer Science, MGR College, Hosur, Tamilnadu, India*
* Corresponding author's Email: asksmart84@gmail.com

Abstract: The on-demand provision of computing resources as services over the internet is known as cloud computing. The distributed denial of service (DDoS) attack is a major security risk that affects cloud services. Because of the computational complexity that must be handled, detecting DDoS attacks is a very difficult operation for cloud computing. The back propagation neural network (BPNN) method is frequently employed for DDoS attack detection due to its great flexibility and straightforward construction. But it has drawbacks such as slow convergence, inconsistencies, and instability during training. In this work, the proposed optimized BPNN uses bacterial colony optimization (BCO) for optimizing the connection weights and biases to enhance the performance of BPNN. The optimized BPNN is developed to identify DDoS attacks in the cloud environment. The performance of the BCO-BPNN detection scheme is assessed using four DDoS attack datasets such as NSL-KDD, ISCXIDS2012, CIC-IDS2017, and UNSW-NB15. Its respective detection accuracy with the NSL-KDD, ISCXIDS2012, CIC-IDS2017, and UNSW-NB15 datasets is 0.9892%, 0.9883%, 0.9341%, and 0.9987%. The results of the studies demonstrate that the suggested BCO-BPNN performs better than existing BPNN variants, conventional BPNN, and support vector machine (SVM) methods.

Keywords: DDoS attacks, Weight optimization, Cloud computing, Back propagation neural network, Bacterial colony optimization.

1. Introduction

Cloud computing is an Internet-based platform that provides customers and organizations with widespread access to computer services like databases, servers, and networking. DDoS is an important security topic in the cloud, and attackers use it to prevent genuine consumers from using the services [1]. The frequency and size of DDoS attacks have greatly increased during the previous few years. On June 1st, 2022, a Google Cloud Armos customer was recently the target of more than ten thousand requests per second (RPS). After eight minutes, the attack's RPS total grew to one lakh. Google claims that for the next two minutes, the attack went from 100,000 RPS to a peak of 46 million RPS. The company claims that this Layer 7 DDoS is the biggest one yet because it is at least 76% greater than the previous record. One of the most dangerous and destructive techniques used on the Internet is the DDoS attack, which represents more than 65% of all such attacks [2]. In this attack, the attackers flood the victim server with queries, significantly straining it. The victim server's bandwidth is fully occupied as a result of the attackers' enormous volume of requests, making it unreachable to legitimate users. To prevent damage to systems and resources, attacks involving DDoS must be identified. Due to their superior capability, machine learning (ML) approaches are frequently used to detect DDoS attacks [3]. The ML technique is proficient in self-adjusting and knowledge from prior calculations to analyze the provided data and spot hidden patterns when provided with data [4].

Artificial neural networks (ANNs) are a subclass of ML that recreate the communication between organic neurons and are motivated by the human brain. The most popular training strategy for ANNs is

the BPNN algorithm, a gradient-based method [5, 6]. However, because the initial connection weights and bias of the conventional BPNN are random, it is simple for a poor choice to harm convergence and the ensuing stable state, leading to a local minimum [7]. The main purpose of the BPNN learning process is to update connection weights and biases, and optimal values are established by continual training. But excessive or insufficient training may limit the network's capacity for generalization, lead to over-fitting, and keep it from attaining the desired result.

As a result, a variety of strategies, are employed to improve the performance of BPNN. These conventional algorithms, however, suffer from several drawbacks, including a slow convergence rate, local optima, and poor detection accuracy. On the other hand, the Escherichia coli (E. coli) food-searching behavior served as the inspiration for the newly suggested SI optimization algorithm known as the bacterial colony optimization (BCO) algorithm, a population-based method derivative from bacterial foraging optimization (BFO), streamlines the computational process used in the original BFO to increase the effectiveness of optimization [8]. Various real-world applications have been effectively solved using BCO, demonstrating a more effective searching capability than conventional population-based algorithms. In this paper, build an efficient DDoS detection scheme based on BPNN optimized by BCO. The major goal is to improve the performance of BPNN by using BCO to obtain appropriate weights and biases and to construct a powerful DDoS attack detection system to achieve higher detection rates and accuracy. The research's contribution is as follows:

- The new detection technique will increase the effectiveness of the system's detection of unusual incoming data.
- To identify DDoS attacks in the framework of cloud computing, an optimized BPNN based on BCO is proposed.
- The BCO algorithm is used in the suggested detection approach to obtain a more suitable weight and bias for BPNN.
- NSL-KDD, ISCX-IDS, CIC-IDS2017, and UNSW-NB15 are used to test the effectiveness of the recommended approach.
- Five performance analyzers are taken into consideration for the performance analysis of the BCO-based BPNN approach.

The remaining sectors of the paper are designed as follows. In section 2, a summary of the related literary works that use DDoS attacks is presented. In section 3, the BPNN algorithm is covered. BCO algorithms are discussed in section 4. In section 5, the suggested BCO-BPNN approach is covered. The experiment evaluations and comparison outcomes are presented in section 6. section 7 covers the paper's conclusion in its last section.

2. Related works

It's incredible how a single attack in a cloud development could cause such serious damage. The entire cloud network will be offline due to the nature of DDoS, though. As a result, prevention is required. Thus, there is an ever-increasing need for DDoS attack detection frameworks. Numerous authors have suggested various DDoS attack detection techniques as a result of this demand. The following section presents a few studies on DDoS attack detection. Z. Chiba et al. (2018) formulate the best procedure using BPNN for building an effective anomalous intrusion detection system (IDS) [9]. First, every possible combination of the most pertinent values of the parameters, such as feature selection, data normalization, the architecture of the neural network, and activation function, needed to build this classifier or determine how well it performs in anomaly detection are constructed. L. Xu et al. (2021) developed a new optimized BPNN based on the IPSO. The prediction of aero-optical imaging deviation is then performed using the model of the modified PSO-BPNN [10]. M. Almiani et al. (2021) [11] provide a DDoS attack detection method to detect the data traffic and transmission on IoT networks, that may be deployed in IoT dynamic contexts. The detection method proposes a Kalman BPNN-based DDoS detection method.

S. Alzughaibi et al. (2023) [12] developed a new IDS approach to improve IDS performance and efficiency in a cloud to address the IDS problem and lessen its negative consequences. For this study, we construct two deep neural network (DNN) models: the first is based on a multi-layer perceptron (MLP) with BPNN, and the second is trained using an MLP with PSO. ANN method for identifying management-frames-based DoS attacks was developed by A. E. Abdallah et al. in 2023 [13]. The suggested method seeks to efficiently identify fake de-authentication/disassociation frames and enhance network performance by preventing the communication hiccups brought on by such assaults. To examine patterns and features in the management frames sent back and forth between wireless devices, the proposed NN approach makes use of machine learning techniques. The system can improve its

Algorithm 1: BCO algorithm

Step 1: Parameters initialization
Step 2: For all bacteria
Step 3: Chemotaxis and communication
Step 4: Reproduction and elimination
Step 5: Migration
Step 7: If the end state cannot be attained, step 2 should be performed; otherwise, the procedure should be discontinued.
Step 8: The best solutions should remain in the final position

ability to recognize probable DoS assaults by training the ANN.

Extreme learning machine (ELM)-based DDoS attack detection has been planned by G. S. Kushwah et al. (2019) [14]. Compared to certain prediction algorithms that can be quickly trained and provide good detection accuracy, the presented method exhibited great generalization performance, according to experiments. An improved anomalous IDS for the hypervisor layer across virtual machines (VMs) was created by A. Rawashdeh et al. (2018) [15]. The proposed approach combines PSO with neural network detection and categorization of traffic transmitted across VMs with an evolutionary neural network. The performance analysis and conclusions of our proposed approach identify and classify DDoS assaults in the cloud environment with a minimal number of false alarms and great detection accuracy. A hybrid ML-based IDS was recommended by S. Sokkalingam et al. (2022) [12]. Support vector machine (SVM) parameters are modified using hybrid Harris Hawks optimization (HHO) and PSO methods. The performance of the proposed IDS model is improved by selecting features on the benchmark NSL-KDD dataset with the help of a 10-fold cross-validation technique.

A. Sagu et al. (2022) [16] present a hybrid vulnerability detection method in an IoT environment that incorporates three stages. For enhancing the classification correctness, the weights of Bi-LSTM are ideally set by a self-upgraded Cat and Mouse Optimizer called SU-CMO. H. Jing et al. (2022) [17] proposed to predict the development of DDoS attacks, a unique ML is provided. To simultaneously extract the properties of the traffic data, a graph theory framework of edges and vertices is first established. As input variables, eight traffic data characteristics are chosen. Second, the PCA model is used to further extract the properties of both DDoS and regular communication. Fuzzy C-means (FCM) are then used to detect DDoS using these attributes.

3. Problem formulation

Our suggested BCO-BPNN model is the classifier utilized in the proposed assault detection system. It accepts groups of samples as input and categorizes each group's sample as normal or attacking. It is a supervised model that needs to be trained on labeled data before being used for the detection of attacks. Attack detection involves applying the gathered data samples X in groups to the trained classifier and calculating output O. The model's output categorizes each sample as either an attack or a normal sample. The sample belongs to the normal class if $O = [0,1]$ and the attack class if $O = [1,0]$. If every sample from the applied groups is normal, then the normal operation is observable. However, if certain samples are identified as attacks, it means that the attack occurred in the cloud.

4. Backpropagation neural networks

In 1986, Rumelhart and McClelland developed the BPNN, a multilayer feed-forward network that uses an error back-propagation method for training [18]. Three levels make up the usual network topology: the input layer, one or more hidden layers, and the output layer. Fig. 1 depicts the BPNN's fundamental structure, and Algorithm 1 illustrates how it works. BPNN can self-learn and self-adapt since each layer of the network has adjustable weights. They exhibit a high degree of self-adaptability to the environment and can be taught by learning how to choose the network's weights. The BPNN includes two processes based on gradient descent: information forward and error backward propagation. The signal is transferred from the input layer to the output layer when the network is learning [19].

The gradient is transmitted back into the network if the output results do not match the anticipated outcomes to regulate the weight and bias and lower the error between the predicted and actual data. In reaction to input data, the BPNN approach continuously updates weights, improving the network's overall accuracy. As a result, BPNN has high nonlinear performance, making it appropriate for the simulation of nonlinear systems and suitable for handling enormous volumes of data concurrently [20].

5. Bacterial colony optimization (BCO)

BCO is a novel swarm intelligence (SI) method introduced by Niu et al. (2012) [21]. The key difference between BCO and other bacteria-inspired heuristic approaches is that, in contrast to other

bacterial-inspired heuristic algorithms, BCO pursuits for nutrients by exchanging information among itself rather than swimming randomly. The three basic steps of the BCO technique are chemotaxis and communication, elimination and reproduction, and migration. Algorithm 1 shows the process of BCO and it has been used to solve a variety of practical problems including fuzzy clustering [22], feature selection [23], data clustering [24, 25], the multi-objective problem [26], scheduling [27], neural network [28] and disease detection [29]. Chemotaxis is constantly accompanied by communication during the whole BCO lifespan. Bacteria have two choices after an extensive time of chemotaxis and communication. They might starve to death or, if they can find food on their own, they might breed. Some people could encounter hazardous situations in a difficult environment by pushing the bounds or looking for space.

The high-energy bacteria will self-replicate to create new individuals throughout the elimination and reproduction phase, and the unhealthy ones will be replaced. Bacteria with high levels of energy are excellent nutrient hunters. In the last phase, known as migration, the bacteria can move within the search range if specific requirements are met. Throughout the entire BCO operation, chemotaxis and communication are used. The other two steps, on the other hand, are only carried out under specific circumstances, such as when a specific number of iterations have been finished, the randomly generated number is below a particular probability, etc. Chemotaxis can be modeled after two different lifetimes, such as swimming and tumbling. A stochastic direction adds to the swimming process in the tumbling process. The search orientation is influenced by both an ideal searching director and a chaotic director together. Each bacterium's current position is as follows:

$$Position_i(T) = Position_i(T-1) + C(i) \times [f_i \times (G_{best} - Position_i(T-1)) + (1-f_i) \times P_{best_i} - Position_i(T-1)) + turb_i] \quad (1)$$

The bacteria will swim toward their ideal position if there is no turbulent disruption in the swimming process, and each bacterium's position will be updated as follows.

$$Position_i(T) = Position_i(T-1) + C(i) \times [f_i \times \left(G_{best} - Position_i(T-1)\right) + (1-f_i) \times \left(P_{best_i} - Position_i(T-1)\right)] \quad (2)$$

Where, $turb_i$ means turbulent direction variance value. $f_i \in \{0,1\}$. P_{best} means the personal best. G_{best}means the global best. $C(i)$ means the value of the chemotaxis step size is defined as follows,

$$C(i) = C_{\min} + \left(\frac{Iter_{\max} - Iter_j}{Iter_{\max}}\right)^n (C_{\max} - C_{\min}) \quad (3)$$

Where, $Iter_{\max}$ - maximum iteration, $Iter_j$ - present iterations, respectively. n - the linearly decreasing method of the chemotaxis step.

6. Proposed BCO-BPNN

The configuration of the network architecture, including the number of nodes in each layer and learning rates, has an important effect on the performance of BPNN. The selection of a network topology is not subject to any standard guidelines. The trial-and-error method, also known as cross-validation, is typically used to make the selection. The conventional BPNN approach can easily get to the local optimum. To optimize the BPNN's parameters, BCO is included in the study. The three main content levels that make up the overall optimization are the improvement of connection weights and bias, the enhancement of neural network architecture, and the improvement of neural network learning parameters. The BCO approach is used in this study to train the weights and biases to get the optimum value for the objective function. The objective function that has been determined to be the mean square error (MSE) is well-defined as follows,

Algorithm 2: Proposed BCO-BPNN

Step 1: Required parameters are initialized
Step 2: Data normalization
Step 3: Both training and test datasets are created after normalizing the data
Step 4: The BCO optimizes the parameters for BPNN.
Step 5: Train the BPNN
Step 6: Compared the fitness values
Step 7: While MSE < stopping condition
  Step 7.1.: Calculated the error
  Step 7.2: Weights and bias updating
  Step 7.3: The best parameters are noted after the termination conditions are met. If not, move on to step 4 to execute the iteration.
  Step 7.4: Move on to step 5 if the stopping condition is met; otherwise, return to step 4.
  Step 7.5: End while
Step 8: The trained model is assessed using the test dataset with the greatest weights and bias.
Step 9: A minimum MSE solution should be kept in storage
Step 10: Make a performance calculation.

Table 1. Details of datasets

Datasets	No of attacks	Features	Training	Testing	Total
NSL-KDD	4	41	1,25,973	34394	1,60,367
ISCXIDS2012	4	19	97,035	41,701	1,39,006
CIC-IDS 2017	8	78	1,744,184	7,47,505	24,91,689
UNSW-NB15	9	48	175,341	82,332	2,57,673

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (4)$$

where y_i and $\hat{y}_i$ are the predicted value and target value. N is the size of the data samples. The best BPNN parameters are chosen using the BCO. Once the BCO algorithm has been completed, the perfect individual is translated into a set of connection weights and biases, which are defined as the initial parameters of the BPNN. The BPNN is then utilized by the algorithm. Based on the starting settings, the output error can be considered using the training dataset. If the termination condition is not satisfied, the weight value and biases will be modified and adjusted, and the error will be handled by the BPNN. When the output error meets the required standards, the network learning is finished, enabling the development of the ultimate computation model. Algorithm 2 shows its process.

7. Experimental analysis

This paper suggests the BCO-based optimized BPNN for detecting DDoS attacks in the cloud framework. With some well-known algorithms, the proposed optimized BPNN's performance is compared including modified PSO-BPNN, MPSO-BPNN [30], PSO-BPNN [31], modified GA-BPNN (MGA-BPNN) [32], GA-BPNN [20], BPNN [9], and SVM [33]. The BPNN literature lists two basic types of fitting issues: under-fitting and over-fitting, which are typically caused by a poor choice of network parameters, such as initial weights and biases, learning rate, momentum, and hidden layer parameters. Therefore, GA and PSO are frequently used in research to choose the parameters of BPNN. To prevent over-fitting or under-fitting issues, the search performance is enhanced to acquire better network parameters. Although many researchers have enhanced the learning approach, topological structure, and updated formula of GA and PSO. In the conventional GA, the single point mutation has little impact on population evolution since single point mutation is only one parameter in the entire optimization process of GA. This is the rationale behind the introduction of several mutation hotspots for MGA optimization. But both GA and MGA have problems with local optima. Although PSO has been widely utilized to find optimum values in many fields, it still has the drawback of early convergence and entrapment in a local optimum. The MPSO solved the shortcomings of the PSO. But, the convergence rate of MPSO has low.

Later, BCO recently developed an SI optimization method that has a faster convergence rate and high solution accuracy [34, 35]. BCO, which was created using the foraging tactics of Escherichia coli bacteria, is one of the comparatively more recent biologically-inspired optimization techniques. The E. coli bacteria always seek areas with high levels of nutrients and stay away from areas with toxic substances. The location with the highest nutritional level is the best option from an optimizing standpoint [36]. Hence, the present article proposed BCO for obtaining optimal parameters for BPNN to enhance detection accuracy. All experiments are conducted on Windows 10, Core i3 processor, and 4 GB RAM, and MATLAB 2015R is used for implementation. Datasets, parameter settings, results, and discussions for evaluating the effectiveness of detection algorithms are covered in the sections below.

A. Datasets

The four datasets, including NSL-KDD, ISCX-IDS 2012, CIC-IDS2017, and UNSW-NB15, detection algorithms are tested for experimental as well as evaluation purposes. The dataset's details are shown in Table 1 and discussed as follows,

1) *NSL-KDD:* The DOS, U2R, R2L, and Probe are denoted attacks in the dataset. This dataset has 160,367 samples, and each sample has 41 characteristics [37]. "(https://www.unb.ca/cic/datasets/nsl.html)".
2) *ISCXIDS2012* [38] "(https://www.unb.ca/cic/datasets/ids.html)": the traffic in the present dataset was together over seven days, between Friday and Thursday. On the other days, samples with attack and normal are present in the traffic, while only normal samples are present on Friday. DoS, DDoS, HTTP, Bruteforce, and Infiltrating are among the 19 features that make up the dataset's four main types of assaults. The total samples are 2,450,324. Normal is 2,381,414, whereas attack is 68,910 samples.

Table 2. Parameter values of BPNN

Name	Values
Training cycle (TC)	2,000
Hidden Neurons ($H_{n)}$	8
Hidden layers (H)	1
Error rate (Err)	0.005
Learning frequency (η)	0.7
Momentum feature (α)	0.5
Activation function	sigmoid (hidden layer) a linear function (output layer)

Table 3. Parameters values of BCO

Name of parameters	Notations	Values
Number of bacteria	S	50
Chemotaxis	Nc	100
Swim step	Ns	5
Reproductive value	N_{re}	5
Elimination and dispersal	N_{ed}	4
Probability of elimination	P_{ed}	0.25
Chemotaxis value (minimum)	C_{min}	0.01
Chemotaxis value(maximum)	C_{max}	0.2
Maximum iterations	Max_Iter	500
Elimination and dispersal steps	N_{ed}	4

3) *UNSW-NB15 dataset* [39] "(https://research.unsw.edu.au/projects/unsw-nb15-dataset)": A subset of 257,673 samples and 2,540,044 samples with 48 characteristics are used. The training data has 175,341, whereas the testing set comprises 82,332 samples. Worms, backdoors, exploits, fuzzers, generic, shell-code, and denial-of-service attacks are nine types of attacks.

4) *CIC-IDS2017 dataset* [40] "(https://www.unb.ca/cic/datasets/ids-2017.html)": Traffic for five days, between Monday and Friday, is involved in this dataset. On the other days, assaults and normal samples are present in the traffic, but only normal samples are present on Monday. The dataset contains eight different types of attacks: Botnet, Bruteforce, DDoS, DoS, Heartbleed, Infiltration, Portscan, and Web. The total samples are 2,491,689, the normal is 2,273,097, and 218,592 samples are attacked with 78 structures.

B. Preprocessing

Raw traffic data must be normalized before the detection phase, in which a detection approaches only receives numerical input because the data is both numerical and categorical, with numerical data spanning pointedly different choices. Data that is categorical will be transformed into numerical data first. Then, using the min-max normalization method, all data will be turned into values between 0 and 1, as seen below [41, 42]

$$z = \frac{x - min(x)}{max(x) - min(x)} \quad (5)$$

The current value is represented by x, whereas the maximum and minimum values are represented by $max(x)$ and $min(x)$. The collected dataset is utilized as a training sample 70 % for the detection model, while the remaining 30% is used as a testing sample to assess performance.

C. Parameter analysis

The determination of the parameters of BPNN is

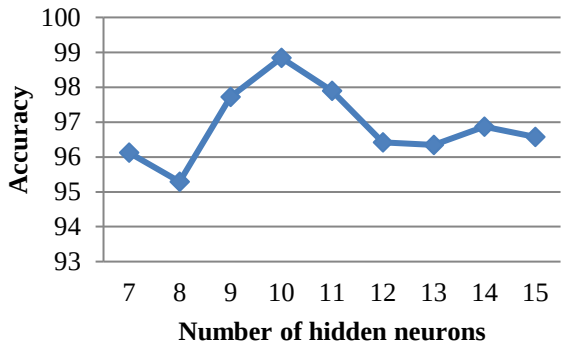

Figure 1. Parameter investigation for hidden neurons

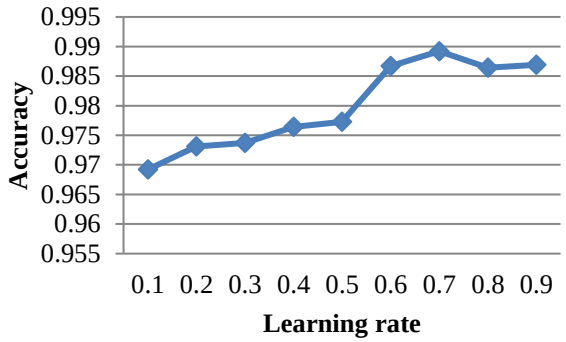

Figure 2. Parameter investigation for learning rate

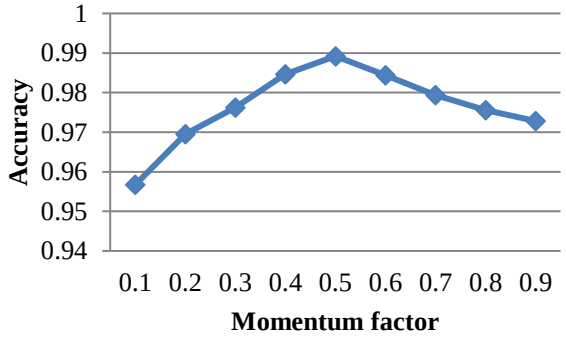

Figure 3. Parameter investigation for momentum factor

needed to obtain the proper solutions. The number of inputs and output neurons in this case is determined by the applications that are defined. The hidden layer neurons, momentum factor, and learning rate are selected through trial and error due to the lack of a useful formula for calculating its parameters [43]. A hyper-parameter called learning rate controls how often our network's weights are modified concerning the loss gradient. The momentum component takes into account information from previous weight adjustments and could expedite training while also minimizing oscillation [44]. The architecture's pace of convergence is significantly impacted by the number of hidden neurons chosen. The largest number of neurons may over-fit the learning process, while the fewest number of neurons may under-fit it [45].

As a result, the best learning rate and momentum factor are chosen based on the accuracy across the NSL-KDD dataset. The parameters investigation of BPNN shows in Figs. 1, 2, and 3 for hidden neurons, learning rate, and momentum factor respectively. The ideal parameters based on Figs. 1, 2, and 3 such as hidden neurons are 8, the learning rate is 0.7, and the momentum factor is 0.5. Finally, Table 2 displays the entire BPNN parameters. In BCO, there are various parameters discussed in Table 3.

D. Performance analyzers

In this study, the performance of the system was assessed using several familiar evaluation metrics, including accuracy, sensitivity, specificity, precision, and F-Score. These metrics were calculated using the following formulas:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (6)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (7)$$

$$Specificity = \frac{TN}{FP+TN} \quad (8)$$

$$Pr\,ecision = \frac{TP}{TP+FP} \quad (9)$$

$$F-Score = 2 \times \frac{Pr\,ecision \times Sensitivity}{Pr\,ecision \times Sensitivity} \quad (10)$$

Here, true positive (TP) denotes attack traffic that was correctly foreseen. True negative (TN) describes the ability to accurately predict typical traffic. False positives (FP) are when regular traffic is predicted inaccurately. Attack flow outcomes that are falsely projected as negative (FN).

E. Results analysis

The multilayer ANN architecture is chosen, and the optimized back-propagation method is utilized as a training algorithm, to validate the detection of DDoS attacks in the cloud environment for the experiment. Since the target function can be measured and observed, supervised learning is appropriate for the learning type. The target function for the attack detection process is suggested to have two values, "normal" and "attack," where the status indicates the type of packets (i.e., 0 signifies normal and 1 means attacks).

The performance of the suggested BCO-BPNN has been assessed using the above-mentioned datasets, structure settings, and system of measurement. The present section displays experimental results for compared detection algorithms for all datasets based on their respective performance indicators. Table 4 and Fig. 4 display experimental results for the NSL-KDD dataset and it revealed that the BCO-BPNN attained better than other detection approaches. The best performance with values of Accuracy of 0.9892, Sensitivity of 0.9933, Specificity of 0.9873, Precision of 0.9895, and F-Score of 0.9912. Performance comparisons for the ISCXIDS2012 dataset are shown in Table 5 and Fig. 5. The ISCXIDS2012 dataset exposed that the BCO-BPNN beat earlier detection techniques. The ISCXIDS2012 dataset, which had the highest performance, had an accuracy of 0.9883, sensitivity of 0.9969, specificity of 0.9938, precision of 0.9987, and F-Score of 0.9791. Table 6 and Fig. 6 display performance comparisons for the CIC-IDS 2017 and it showed that the BCO-BPNN scheme beat earlier detection techniques. The best performance, had an Accuracy of 0.9341, Sensitivity of 0.9592, Specificity of 0.9711, Precision of 0.9692, and F-Score of 0.9684. Table 7 and Fig. 7 display shows the experimental results for the UNSW-NB15 dataset. The UNSW-NB15 dataset showed that the proposed BCO-BPNN beat earlier detection techniques, which had the best results, had an Accuracy of 0.9987, Sensitivity of 0.9876, Specificity of 0.9994, Precision of 0.9965, and F-Score of 0.9927. The overall findings demonstrate that the suggested algorithms perform better than the other systems in comparison. The results of the experiments proved that the BCO-BPNN can give better detection accuracy with a quick convergence rate. In comparison to other techniques, the BCO-BPNN methodology attains the lowermost MSE values. Once more, it is evident from the examination of the experimental findings that the suggested technique, when trained using the BCO algorithm, gives superior detection ability when compared to MPSO-BPN, PSO-BPNN, MGA-BPNN, GA-BPNN, and SVM. The optimized BPNN approach uses BCO's quick convergence rate and

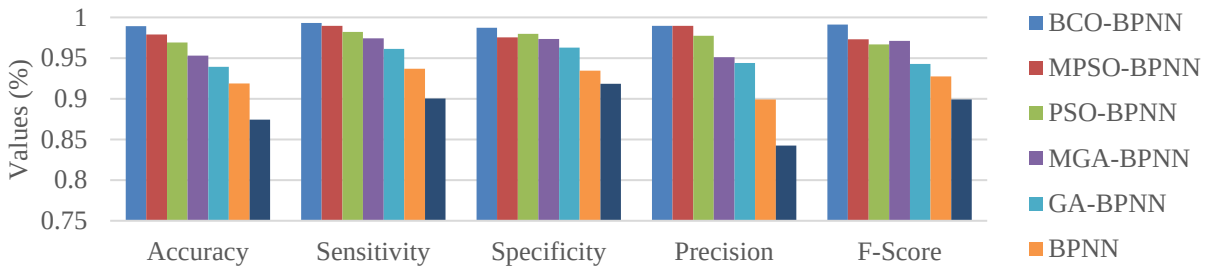

Figure. 4 Performance comparison of the BCO-BPNN for the NSL-KDD dataset

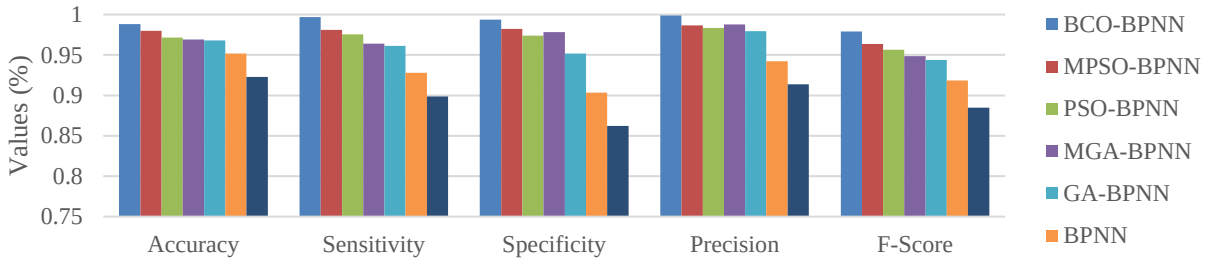

Figure. 5 Performance comparison of the BCO-BPNN for the ISCXIDS2012 dataset

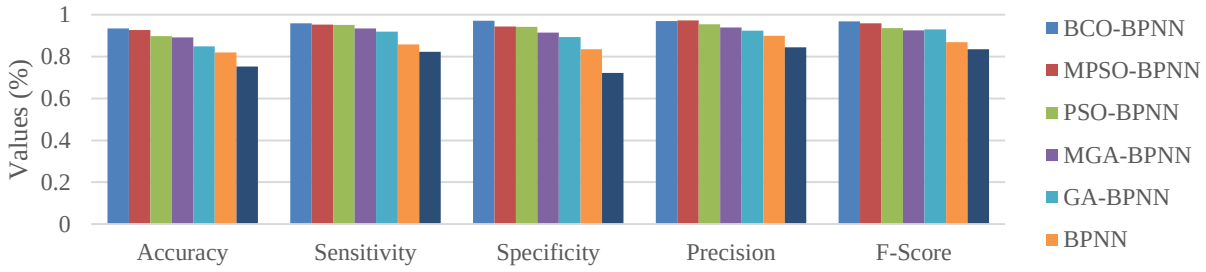

Figure. 6 Performance comparison of the BCO-BPNN for the CIC-IDS 2017 dataset

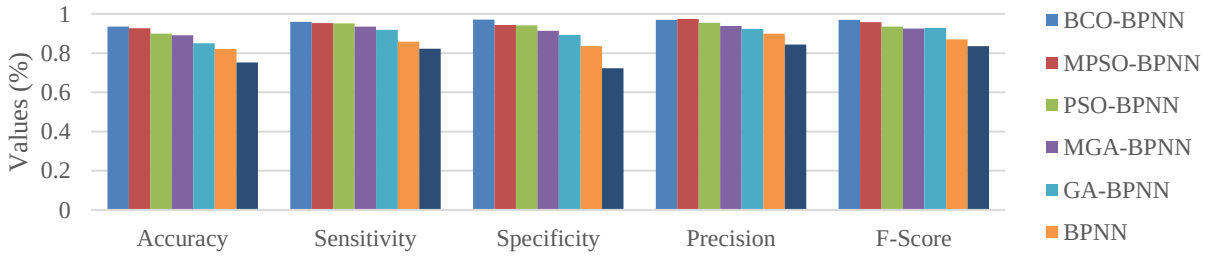

Figure. 7 Performance comparison of the BCO-BPNN for the UNSW-NB15 dataset

Table 4. Results comparison of the NSL-KDD dataset

Methods	BCO-BPNN	MPSO-BPNN	PSO-BPNN	MGA-BPNN	GA-BPNN	BPNN	SVM
Accuracy	**0.9892**	0.9789	0.9692	0.9532	0.9394	0.9189	0.8745
Sensitivity	**0.9933**	0.9898	0.9821	0.9743	0.9613	0.9371	0.9002
Specificity	**0.9873**	0.9754	0.9798	0.9736	0.9629	0.9347	0.9183
Precision	**0.9895**	0.9898	0.9775	0.9512	0.9439	0.8992	0.8423
F-Score	**0.9912**	0.9732	0.9669	0.9711	0.9429	0.9275	0.8993

Table 5. Results comparison of the ISCXIDS2012 dataset

Methods	BCO-BPNN	MPSO-BPNN	PSO-BPNN	MGA-BPNN	GA-BPNN	BPNN	SVM
Accuracy	0.9883	0.9799	0.9716	0.9692	0.9678	0.9519	0.9229
Sensitivity	0.9969	0.9812	0.9756	0.9641	0.9613	0.9281	0.8987
Specificity	0.9938	0.9824	0.9739	0.9781	0.9519	0.9033	0.8623
Precision	0.9987	0.9865	0.9835	0.9878	0.9793	0.9421	0.9139
F-Score	0.9791	0.9637	0.9564	0.9487	0.9438	0.9186	0.8847

Table 6. Results comparison of the CIC-IDS 2017 dataset

Detection Methods	BCO-BPNN	MPSO-BPNN	PSO-BPNN	MGA-BPNN	GA-BPNN	BPNN	SVM
Accuracy	0.9341	0.9273	0.8984	0.8915	0.8492	0.8198	0.7522
Sensitivity	0.9592	0.9526	0.9513	0.9345	0.9189	0.8583	0.8229
Specificity	0.9711	0.9429	0.9419	0.9143	0.8928	0.8355	0.7218
Precision	0.9692	0.9733	0.9539	0.9383	0.9234	0.8986	0.8435
F-Score	0.9684	0.9583	0.9353	0.9246	0.9291	0.8689	0.8358

Table 7. Results comparison of the UNSW-NB15 dataset

Methods	BCO-BPNN	MPSO-BPNN	PSO-BPNN	MGA-BPNN	GA-BPNN	BPNN	SVM
Accuracy	0.9987	0.9904	0.9883	0.9765	0.9698	0.9549	0.9344
Sensitivity	0.9876	0.9819	0.9799	0.9725	0.9482	0.9288	0.9237
Specificity	0.9994	0.9982	0.9897	0.9853	0.9844	0.9789	0.9624
Precision	0.9965	0.9897	0.9891	0.9884	0.9792	0.9724	0.9659
F-Score	0.9927	0.9857	0.9783	0.9746	0.9587	0.9524	0.9198

global searching capability to find more accurate weights and biases, which can improve detection accuracy. In the BCO, Different topologies of communication systems are provided by the partition of the population into different groups, which can greatly accelerate convergence and prevent local optimum.

8. Conclusions

Cloud computing offers a variety of resources in the form of facilities over the Internet. Accessibility to cloud services is crucial for this technology to operate effectively. Attackers may employ DDoS attacks to obstruct the accessibility of cloud services. In the current work, an enhanced DDoS attack detection scheme for cloud computing is provided. The BCO algorithm is used in the suggested way to give the BPNN a weight and bias that are more suitable. The performance of the proposed BCO-based BPNN scheme over the four DDoS attack datasets is examined using five performance analyzers. According to the experimental findings, the BCO-based BPNN outperformed other variants and traditional detection systems in terms of generalization performance.

References

[1] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, and R. Buyya, "DDoS attacks in cloud computing: Issues, taxonomy, and future directions", *Computer Communications*, Vol. 107, pp. 30-48, 2017.

[2] F. S. d. L. Filho, F. A. Silveira, A. D. M. B. Junior, G. V. Solar, and L. F. Silveira, "Smart detection: an online approach for DoS/DDoS attack detection using machine learning", *Security and Communication Networks*, Vol. 2019, 2019.

[3] M. Zekri, S. E. Kafhali, N. Aboutabit, and Y. Saadi, "DDoS attack detection using machine learning techniques in cloud computing environments", In: *Proc. of 2017 3rd*

International Conference of Cloud Computing Technologies and Applications (CloudTech), pp. 1-7, 2017.

[4] K. J. Singh, K. Thongam, and T. De, "Entropy-based application layer DDoS attack detection using artificial neural networks", *Entropy*, Vol. 18, No. 10, p. 350, 2016.

[5] I. Ahmad, A. B. Abdullah, and A. S. Alghamdi, "Application of artificial neural network in detection of DOS attacks", In: *Proc of the 2nd International Conference on Security of Information and Networks*, pp. 229-234, 2009.

[6] L. Cui, Y. Tao, J. Deng, X. Liu, D. Xu, and G. Tang, "BBO-BPNN and AMPSO-BPNN for multiple-criteria inventory classification", *Expert Systems with Applications*, Vol. 175, p. 114842, 2021.

[7] K. Kalaiselvi, K. Velusamy, and C. Gomathi, "Financial prediction using back propagation neural networks with opposition based learning", in *Journal of Physics: Conference Series*, Vol. 1142, No. 1, 2018.

[8] B. Niu and H. Wang, "Bacterial colony optimization: principles and foundations", In: *Proc. of Emerging Intelligent Computing Technology and Applications: 8th International Conference, ICIC 2012, Huangshan, China, July 25-29*, pp. 501-506, 2012.

[9] Z. Chiba, N. Abghour, K. Moussaid, A. E. Omri, and M. Rida, "A novel architecture combined with optimal parameters for back propagation neural networks applied to anomaly network intrusion detection", *Computers & Security*, Vol. 75, pp. 36-58, 2018.

[10] L. Xu, Z. Zhang, Y. Yao, and Z. Yu, "Improved Particle Swarm Optimization-Based BP Neural Networks for Aero-Optical Imaging Deviation Prediction", *IEEE Access*, Vol. 10, pp. 26769-26777, 2021.

[11] M. Almiani, A. A. Ghazleh, Y. Jararweh, and A. Razaque, "DDoS detection in 5G-enabled IoT networks using deep Kalman backpropagation neural network", *International Journal of Machine Learning and Cybernetics*, Vol. 12, pp. 3337-3349, 2021.

[12] S. Alzughaibi and S. E. Khediri, "A Cloud Intrusion Detection Systems Based on DNN Using Backpropagation and PSO on the CSE-CIC-IDS2018 Dataset", *Applied Sciences*, Vol. 13, No. 4, p. 2276, 2023.

[13] A. E. Abdallah *et al.*, "Detection of Management-Frames-Based Denial-of-Service Attack in Wireless LAN Network Using Artificial Neural Network", *Sensors*, Vol. 23, No. 5, p. 2663, 2023.

[14] G. S. Kushwah and S. T. Ali, "Distributed denial of service attacks detection in cloud computing using extreme learning machine", *International Journal of Communication Networks and Distributed Systems*, Vol. 23, No. 3, pp. 328-351, 2019.

[15] A. Rawashdeh, M. Alkasassbeh, and M. A. Hawawreh, "An anomaly-based approach for DDoS attack detection in cloud environment", *International Journal of Computer Applications in Technology*, Vol. 57, No. 4, pp. 312-324, 2018.

[16] A. Sagu, N. S. Gill, P. Gulia, J. M. Chatterjee, and I. Priyadarshini, "A Hybrid Deep Learning Model with Self-Improved Optimization Algorithm for Detection of Security Attacks in IoT Environment", *Future Internet*, Vol. 14, No. 10, p. 301, 2022.

[17] H. Jing and J. Wang, "Detection of DDoS Attack within Industrial IoT Devices Based on Clustering and Graph Structure Features", *Security and Communication Networks*, Vol. 2022, 2022.

[18] Y. Chauvin and D. E. Rumelhart, *Backpropagation: Theory, Architectures, and Applications*. Psychology press, 2013.

[19] T. L. Lee, "Back-propagation neural network for long-term tidal predictions", *Ocean Engineering*, Vol. 31, No. 2, pp. 225-238, 2004.

[20] K. J. Singh and T. De, "MLP-GA based algorithm to detect application layer DDoS attack", *Journal of Information Security and Applications*, Vol. 36, pp. 145-153, 2017.

[21] B. Niu and H. Wang, "Bacterial colony optimization", *Discrete Dynamics in Nature and Society*, Vol. 2012, 2012.

[22] K. Vijayakumari and V. B. Deepa, "Fuzzy C-Means Hybrid with Fuzzy Bacterial Colony Optimization", in *Advances in Electrical and Computer Technologies: Select Proceedings of ICAECT 2020*, 2021: Springer Singapore, pp. 75-87.

[23] H. Wang, L. Tan, and B. Niu, "Feature selection for classification of microarray gene expression cancers using Bacterial Colony Optimization with multi-dimensional population", *Swarm and Evolutionary Computation*, Vol. 48, pp. 172-181, 2019.

[24] K. Tamilarisi, M. Gogulkumar, and K. Velusamy, "Data clustering using bacterial colony optimization with particle swarm optimization", In: *Proc. of 2021 Fourth International Conference on Electrical, Computer and Communication Technologies*, pp. 1-5, 2021.

[25] R. Ramkumar and C. Balasubramanian, "A

novel cluster head selection scheme based on BCO for Internet of Things", In: *Proc. of 2023 Third International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, pp. 1-6, 2023.

[26] B. Niu, Q. Liu, Z. Wang, L. Tan, and L. Li, "Multi-objective bacterial colony optimization algorithm for integrated container terminal scheduling problem", *Natural Computing*, Vol. 20, No. 1, pp. 89-104, 2021.

[27] B. Niu, T. Xie, Y. Bi, and J. Liu, "Bacterial colony optimization for integrated yard truck scheduling and storage allocation problem", In: *Proc. of International Conference on Intelligent Computing*, pp. 431-437, 2014.

[28] P. Vigneshvaran and A. V. Kathiravan, "Heart Disease Prediction using an optimized Extreme Learning Machine with Bacterial Colony optimization", In: *Proc. of 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC)*, pp. 1425-1429, 2022.

[29] S. İlkin, T. H. Gençtürk, F. K. Gülağız, H. Özcan, M. A. Altuncu, and S. Şahin, "hybSVM: Bacterial colony optimization algorithm based SVM for malignant melanoma detection", *Engineering Science and Technology, an International Journal*, 2021.

[30] B. Bai, J. Zhang, X. Wu, G. W. Zhu, and X. Li, "Reliability prediction-based improved dynamic weight particle swarm optimization and back propagation neural network in engineering systems", *Expert Systems with Applications*, Vol. 177, p. 114952, 2021.

[31] A. S. Saljoughi, M. Mehrvarz, and H. Mirvaziri, "Attacks and intrusion detection in cloud computing using neural networks and particle swarm optimization algorithms", *Emerging Science Journal*, Vol. 1, No. 4, pp. 179-191, 2017.

[32] J. Zhang and S. Qu, "Optimization of backpropagation neural network under the adaptive genetic algorithm", *Complexity*, Vol. 2021, 2021.

[33] T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. K. Son, "Performance evaluation of Botnet DDoS attack detection using machine learning", *Evolutionary Intelligence*, Vol. 13, No. 2, pp. 283-294, 2020.

[34] V. Prakash, V. Vinothina, K. Kalaiselvi, and K. Velusamy, "An improved bacterial colony optimization using opposition-based learning for data clustering", *Cluster Computing*, Vol. 25, No. 6, pp. 4009-4025, 2022.

[35] K. Vijayakumari and V. B. Deepa, "Fuzzy C-means hybrid with fuzzy bacterial colony optimization", In: *Proc. of International Conference on Advances in Electrical and Computer Technologies*, pp. 75-87, 2020.

[36] B. Sivasakthi and D. Selvanayagi, "Prediction of Osteoporosis Disease Using Enhanced Elman Recurrent Neural Network with Bacterial Colony Optimization", In: *Proc. of Computational Vision and Bio-Inspired Computing: Proceedings of ICCVBIC 2022*, pp. 211-220, 2023.

[37] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set", In: *Proc. of 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1-6, 2009.

[38] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection", *Computers & Security*, Vol. 31, No. 3, pp. 357-374, 2012.

[39] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)", In: *Proc. of 2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1-6, 2015.

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

[41] K. Velusamy and R. Amalraj, "Cascade correlation neural network with deterministic weight modification for predicting stock market price", *IOP Conference Series: Materials Science and Engineering*, Vol. 1110, No. 1, p. 012005, 2021.

[42] K. Velusamy and R. Amalraj, "Performance of the cascade correlation neural network for predicting the stock price", In: *Proc. of 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1-6, 2017.

[43] M. Madhiarasan and S. Deepa, "A novel method to select hidden neurons in ELMAN neural network for wind speed prediction application", *WSEAS Transactions on Power Systems*, Vol. 13, pp. 13-30, 2018.

[44] L. Xiao, X. Chen, and X. Zhang, "A joint optimization of momentum item and Levenberg-Marquardt algorithm to level up the BPNN's generalization ability", *Mathematical Problems in Engineering*, Vol. 2014, 2014.

[45] K. G. Sheela and S. N. Deepa, "Review on methods to fix number of hidden neurons in neural networks", *Mathematical Problems in Engineering*, Vol. 2013, 2013.