Enhanced method of ANN based model for detection of DDoS attacks on multimedia internet of things

Article *in* Multimedia Tools and Applications · August 2022

DOI: 10.1007/s11042-021-10640-6

Sathiyamoorthi Velayutham
Sona College of Technology
51 PUBLICATIONS **396** CITATIONS

SEE PROFILE

Pushpita Chatterjee
Tennessee State University
10 PUBLICATIONS **173** CITATIONS

SEE PROFILE

Noor Zaman
Taylor's University
420 PUBLICATIONS **4,576** CITATIONS

SEE PROFILE

Ashish Kr Luhach
The Papua New Guinea University of Technology
121 PUBLICATIONS **1,188** CITATIONS

SEE PROFILE

Multimedia Tools and Applications
https://doi.org/10.1007/s11042-021-10640-6

1175 : IOT MULTIMEDIA APPLICATIONS AND SERVICES

Enhanced method of ANN based model for detection of DDoS attacks on multimedia internet of things

R. Gopi[1] • V. Sathiyamoorthi[2] • S. Selvakumar[1] • Ramasamy Manikandan[3] • Pushpita Chatterjee[4] • N. Z. Jhanjhi[5] • Ashish Kumar Luhach[6]

Received: 29 June 2020 / Revised: 4 January 2021 / Accepted: 4 February 2021
Published online: 24 February 2021

Abstract

Due to the huge flow of data and complications in mutable characteristics of the data, Distributed Denial of Services attacks existed in the Multimedia Internet of Things. Attacks over the IoT have become an increasing menace in recent time, which tries to hack or illegally tamper the streaming data available over the networks. On the other hand, there has been an increase in volume in research contributions to effectively counter these attacks and implement a strong defense mechanism. There have been numerous algorithms and frameworks implemented in recent times that are intelligent and soft computing-based. These evolution-based algorithms play a vital role in self-adapting the system under attack towards increasing and new types of attacks which are increasing day by day. One such area of soft computing algorithms investigated in this paper is the Artificial Neural Network or popularly known as ANNs. It works analogously to the biological neurons in the human body. In this paper, we systematically explain the ANN-based network model to counteract the DDoS attacks in the Multimedia Internet of Things, architecture, and implementation of ANNs, the experimental investigations and findings which help in drawing an inference of ANN-based defense models.

Keywords ANN · DDoS attacks · Multimedia IoT · Network attacks · Training

1 Introduction

DDoS attacks are a special class of attacks prevalent in online communication among networks utilizing internet services for storage, processing, and utility [1]. These attacks increase the

✉ Pushpita Chatterjee
puspitachatterjee@tdtu.edu.vn

✉ N. Z. Jhanjhi
noorzaman.jhanjhi@taylors.edu.my

Extended author information available on the last page of the article

congestion of the network by introducing zombie packets which are infected packets that contaminate good packets as they progress down the communication layers.

DDoS attacks make security threats to the modern Internet, especially targets the Internet of Things (IoT) environment as the IoT devices have limited memory, processing power and security measures to prevent DoS attacks. DDoS attacks are severe attacks against IoT connected devices and worsen the performance of the network. It reduces the network and computing resources such as CPU, memory or network bandwidth while transferring multimedia streaming data since streaming data are continuous, time-sensitive in nature.

In a DDoS attack, the attacker or hacker sends an array of infected packets that cause flooding [9] due to which the target system gets occupied to serve the flooded requests which cause severe degradation in the network bandwidth and increases the computation system overhead. Although the existing techniques can be applied for detecting most of the attacks, newly created attacks cannot be detected without identifying their signatures which are called zero-day attacks. An intrusion detection system (IDS) is the most reliable network solution for identifying attacks. There are two types of IDSs. The first one is misuse detection based IDS which detects attacks using known signatures, and the second one is the anomaly detection mechanism which detects abnormal attacks based on normal use patterns [14]. IDS can identify anomalous activities by continuously observing the network. IDS have a vital role in security, fault tolerance and reliability of sensor networks. IDS can ensure secure routing of data packets over WSN [2].

A simple illustration of an intrusion detection scheme is depicted in Fig. 1 where the packets of information are pre-processed, and the conditioned input is given to the IDS system.

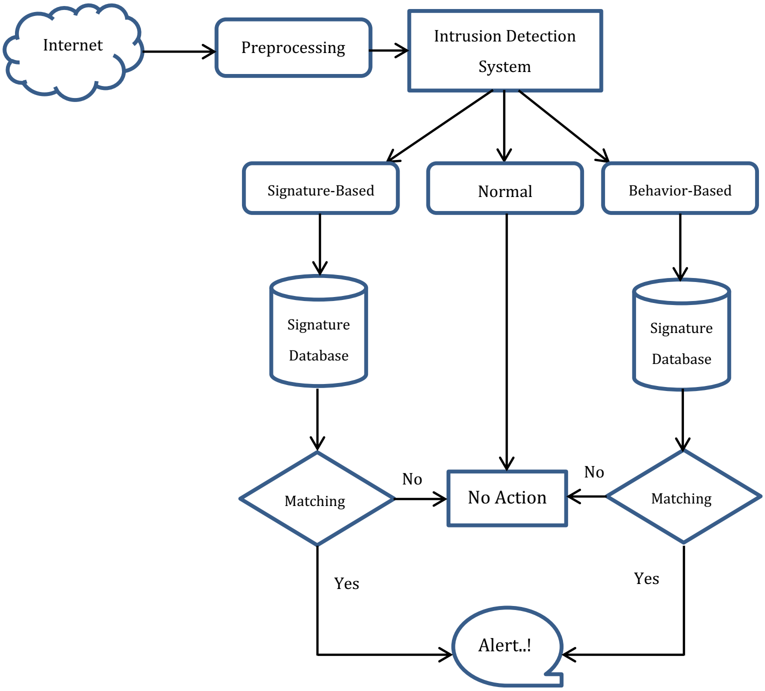


Fig. 1 General scheme of the intrusion detection mechanism

The signature and behavior patterns are extracted from the packets regarding abnormalities in terms of file size, frequency of repetition, bandwidth, etc., Based on the features extracted and the adaptation of the IDS system, the given packet of information is classified as normal or an infected packet.

The essential motivation for going for ANN-based DDoS attack detection in this article is that they are capable of detecting these zero-day attacks using a specific pattern. These patterns distinguish the normal attacks from these DDoS attacks. These features could be given as a vector set to train the neural networks to improve the detection accuracy. Before the actual implementation, it is quite necessary to know the essential components in the intrusion detection framework, which define the efficiency of the implemented system. These components or parameters are elaborated in the following sections. This paper is organized as follows: In Section 2, we have reviewed existing literature related to ANN technology and DDoS attack detection. Section 3 discusses the architectural framework of the proposed system. Section 4 gives the experimental results of the proposed system, and Section 5 gives the conclusion followed by the references.

2 Literature review

A number of research works have already been contributed towards the construction of the Intrusion Detection System and detecting the DoS attacks. Rule-based methods have been found in the literature [3, 7, 8] for the detection of DDoS attack detection wireless sensor networks, and the process occurs in three phases. In the first phase, monitoring of information to filter the vital data from the mainstream network is carried out with the help of monitor nodes. The second phase is known as the rule application stage, where predefined rules are applied to the information that has been filtered in the first stage by the monitor nodes. In the last phase, a detection alarm is raised if any of the information packets fail the rule application test. A detail survey on sevaral attacks and countermeasures have been discussed in [11]. A minor alternative [12] of this method is proposed by extracting the behavior patterns of the neighboring node in the sensor network.

Ferrag et al. [6] have proposed a hierarchical model with three classifiers to suitably categorize each attack, with a high detection rate and low false alarm rate. Manso et al. [17] have proposed a Software-Defined IDS which sensitively destroy the attacks at the beginning, and guarantees the regular functioning of the network. This approach includes an IDS that detects many DDoS attacks and immediately notifies a Software-Defined Networking (SDN) controller.

A cumulative sum algorithm [15] has been reported in the literature which continuously monitors the incoming and outgoing packets of information for any behavioral and pattern changes. A similar three-phase scheme has been reported in the literature [5] where the nodes are first monitored and detected whether they originate from legitimate or illegitimate nodes. If they originate from legitimate nodes, the normal tasks of the system are carried out. Information packets originating from illegitimate nodes are extracted for their features, and the rule base is applied to this information. The set of predefined rules, as mentioned above, are framed by observing the network protocol patterns in a normal communication network. Any deviation from the normal pattern classifies the incoming packet as illegitimate and filtered off in succeeding processes. Another variant of the rule-based method is applicable for effectively detection DDoS attacks in IoT networks [3] based on an event processing model. The experimentations have been done using SQL as a reference, and the rule codes are stored in the repository. The experimental results indicate the least utilization memory but suffer a drawback of utilized more system resources at the cost of minimal processing time.

There has been a migration of detection schemes towards cluster computing and soft computing due to the nature of intelligence and capability to handle huge volumes of incoming as well as outgoing data and at the same time to produce a quick response time. A wide range of soft computing algorithms [13] like principal component analysis [4], linear discriminant analysis [12, 15], local binary pattern [9], particle swarm optimization and greedy search algorithms are found in the existing works. A hybrid multi-objective approach (NSGAII) is proposed by Golrang et al. [10] to detect attacks in a network. They have altered the NSGAII method preserving the diversity control in this approach.

Utilization of support vector machine [6] based methods has also been found in the literature, which is useful for the classification of malicious packets of information. Support vector machine strategies have been imposed on mobile agent models for the detection of known attacks. Two other mobile agents are identified in the literature namely the collector agent who gives feedback from the wireless sensor network, and the other agent is known as a misuse detection agent who detects known malicious patterns in the network. Support vector machines have been integrated with Gaussian kernel and experimented with three types of data sets with a 98.7% accuracy being reported in the literature.

PCA based techniques have been reported to bring about a reduction in data dimensionality, especially when dealing with a huge volume of traffic with a wide range of feature-based attacks incidents on the mainstream of the network. PCA based techniques have been found to exhibit not only a dimension reduction feature but also exhibit a high degree of classification attributed to their ability to differentiate the malicious packets from normal data packets using multiple attribute values. A deep learning-based scheme called as Deep Belief Network (DBN) process for IDS has been proposed by Manimurugan et al. [16]. They have used the CICIDS 2017 dataset for the performance analysis of the proposed IDS model.

In [18] Napiah et al. have proposed an IDS called as the Compression Header Analyzer Intrusion Detection System (CHA-IDS) to analyze the data in 6LoWPAN compressed header to diminish the separate and blended routing attacks. The proposed method decides the important characteristics required to detects the intrusion by using correlation-based feature selection algorithms.

3 Architectural framework of the proposed system

3.1 Artificial neural networks

Artificial neural networks are essential networks in almost any data processing and computing applications and are function analogous by the neurons in our central nervous system. Similar to the biological connectivity of neurons, ANNs are also featured with high interconnected

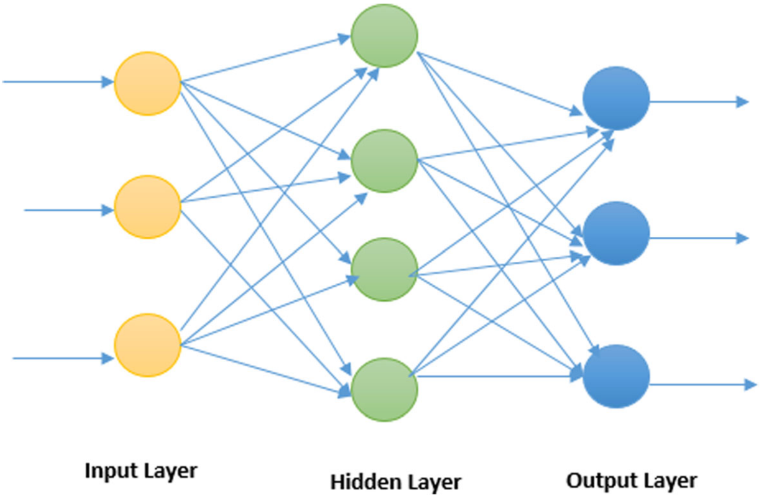


Fig. 2 A simple neural network model

elements in perfect coordination to meet an objective function. They serve several applications which include pattern recognition and classification, detection problems, adaptation and control applications, etc., A simple scheme of the artificial neural network is shown in Fig. 2.

In Fig. 2, a three-input neural network scheme is illustrated whose operation is quite straightforward in approach. The three inputs ×1, ×2 and ×3 along with an associated weight function are given as inputs to the function block denoted by the bigger circle in the hidden layer where a simple product between these two quantities is performed to obtain the desired output. In a feedback system, the neural networks are able to adjust the weights in successive iterations based on the error signal generated from the difference of obtained and desired outputs. The overall objective of the network shown above lies in minimizing the error in the least time possible.

The neural network could be extended further by interconnecting other multiple nodes with the objective function of transforming inputs to desired outputs. A most widely used configuration for intrusion detection systems is the Multilayer Perceptron Model (MLP) as multiple inputs from the different patterns are incident on the target system under attack.

Figure 3 depicts a three-layer ANN, namely input, hidden, and output layer. The number of nodes in the input layer corresponds to the feature vector in the given problem definition. The number of nodes in the output layer is equivalent to the number of sets or classes to which the desired throughput may be designated. For example, the output nodes may be two to three for brain tumor detection, and classification problem where the output needs to know as either the brain tumor is malignant or benign. In the given problem of attack detection, the number of output layer may be limited to two comprising of an infected or valid packet of information.

The input and output layers are connected through the hidden layer and the procedure of the upgrade starts with a few indiscriminate weights that have been allocated to every node. On completion of the first iteration, the weights are updated according to a weight update equation defined in (1) to minimize the error at the output as defined in (2).

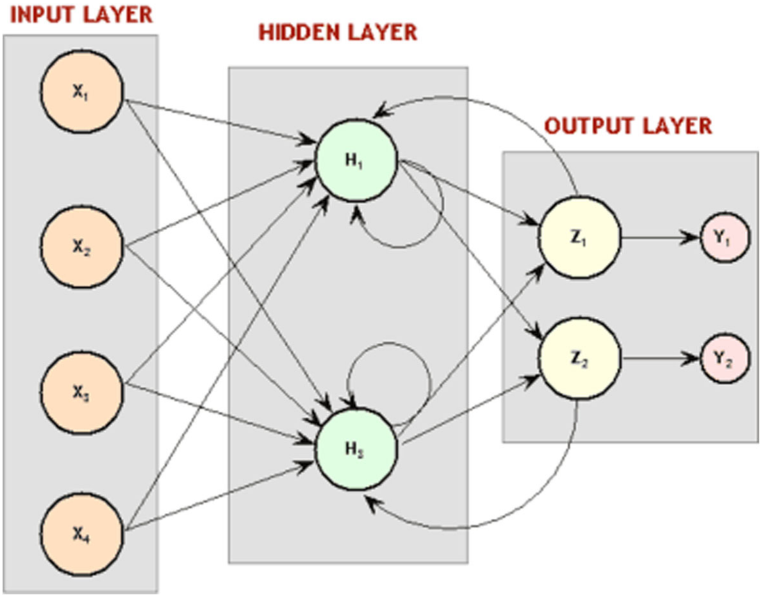


Fig. 3 Architecture of multi-layer perceptron model

$$WI(x, y) + \alpha \tag{1}$$

and

$$E\ \{e2\ [n]\} = E\ \{(d\ [n]-y\ [n])\ 2\} \tag{2}$$

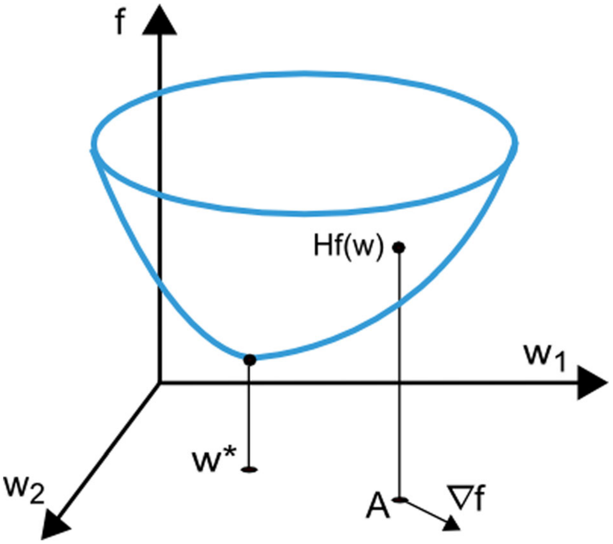


Fig. 4 Design of a general loss function for a given problem

where E{e2 [n]} denotes the expectation of mean squared error function, d[n] indicates the desired output, and y[n] denotes the obtained output. This iteration process follows a learning algorithm that may follow a propagation rule mechanism.

3.2 Training the neural network

The practice of making the implemented neural network to learn the feature vector patterns and thereby decide upon categorizing the given inputs into a class of expected outputs is known as training and forms the backbone of neural network efficiency and working. The overall objective of neural network learning follows a minimization rule to reduce the loss function. This is exposed in Fig. 4.

We can see in Fig. 4 that the point w* represents the least value of the given loss function and we can attain the solution to the given problem by obtaining the I and II derivatives of this failure function as demonstrated in Eqs. (3) & (4)

$$\nabla_i f(w) = df/dw_i \ (i = 1, \ldots, n) \tag{3}$$

Where i = 1……. n.

The second derivative could be generalized using the Hessian matrix as

$$H_{i,j} f(w) = d^2 f/dw_i \cdot dw_j \ (i, j = 1, \ldots, n) \tag{4}$$

Where i, j = 1…. n.

These mathematical formulations could be mapped onto one-dimensional search spaces to get the result of the specified minimization problem. This is illustrated in

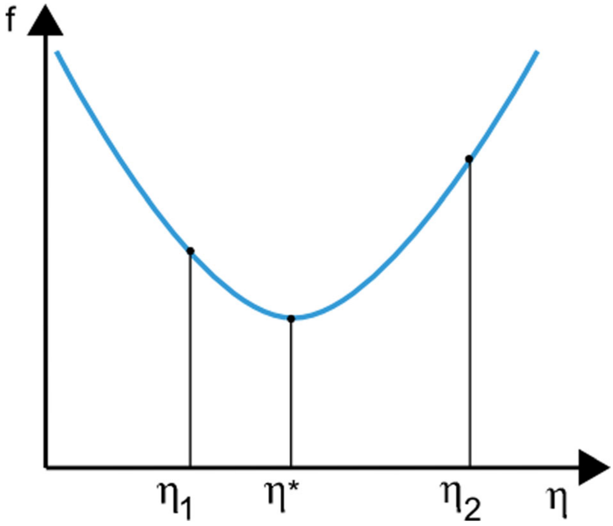


Fig. 5 One dimensional mapping of the minimum loss function

Fig. 5. In this figure, the minimal function of loss exists between the η1 and η2. We have two widely used methods called Golden section and Brent's method for a one-dimensional loss minimization function [19]. Nevertheless, it is necessary that a multi-dimensional search and minimization strategy for most of the real-time problem, including the problem objective of this article. The next section explains the multi-dimensional optimization methods and procedures which significantly assist in employing the proposed architecture for defending network attacks.

3.3 Levenberg – Marquardt (LM) method

It is based on the sum of squares iteration and is alternately referred to as the damped least-squares method [20]. This algorithm also does not have any interference with the Hessian matrix and its inverse values but rather works on another matrix known as the Jacobian Matrix. The Jacobian matrix is defined using a loss function of the form

$$f = \sum {e_k}^2, k = 0, 1, 2, \ldots..l \tag{5}$$

$$J_{i,j} f(q) = \frac{de_i}{dq_j} \tag{6}$$

The weight update equation is given as

$$\mathrm{W}_{\mathrm{i}+1} = \mathrm{W}_{\mathrm{i}} - \left(J^T \mathrm{J} + \delta \mathrm{I}\right)^{-1} \left(2J^T \cdot \mathrm{e}_{\mathrm{i}}\right), \quad \mathrm{i} = 0, 1, \tag{7}$$

Where i = 0,1, …

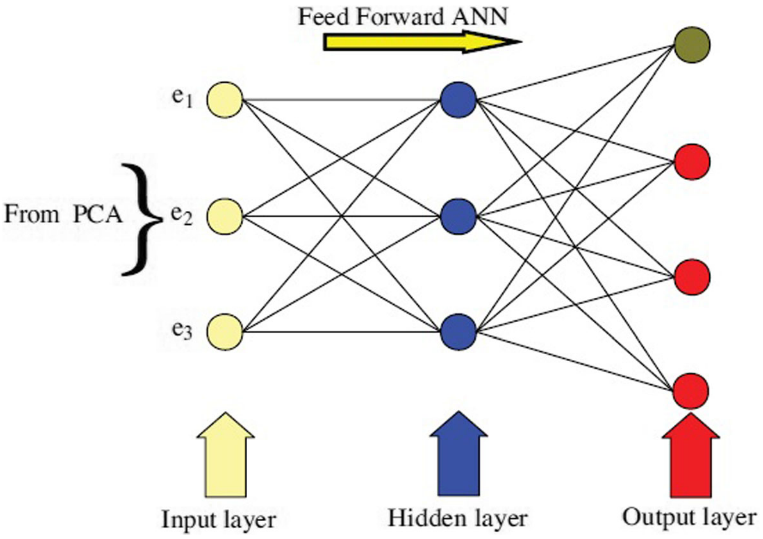


Fig. 6 Feedforward ANN architecture for DDoS

In the above equation, f denotes the damping function, and it becomes a conventional Newton's method when it becomes a zero. I is the identity matrix, and the term denotes the approximation of the Hessian matrix. The damping factor value is decreased or increased to get to the convergence value and acceleration towards minima is very fast in this method, making it suitable for training of medium size neural models. The problem arises for large-sized networks when the size of the Jacobian matrix gets doubled, affecting cost and complexity.

3.4 Proposed ANN model

In accordance with the inquiries and conclusions associated with the ANN implementation types and their facts, a three-layer ANN feed-forward model has been projected and implemented in this article with modification of principal component analysis (PCA) for dimensionality reduction. The reason behind providing a dimensionality reduction is to reduce the storage and computation complexity of an LM learning rule when deriving the Jacobian matrix for large Neural Network (NN) models as used in the proposed article. Figure 6 illustrates the ANN model utilized in this implementation, where the decreased dimensional input characteristics are provided to the ANN model prior to training.

The following subsequent stages are involved in applying the above said PCA technique to decrease the facet of high-level dimensions in the data set.

S1: generate a matrix M by means of the data set.
S2: Regularise the data set by means of-value.
S3: Compute the decomposition value of the data matrix.
S4: Compute the discrepancy utilizing the diagonal components
S5: Sort discrepancies in reducing order.
S6: Select the principal elements from with biggest discrepancies.
S7: Form the alteration matrix consisting of those values
S8: Observe the diminished anticipated data set in a new coordinate axis by applying to.

The first step as indicated in the algorithm above involves the computation of the preliminary value of centroid as the input data set with dimensionality subsequently; the discrepancy can be calculated for every single data in dimension. The column with maximum variance M_{max} represents the column with maximal discrepancy which can be sorted in the descending order of magnitude. In the entire k subset groups, every single median will be initiated by the Cluster Centers (CCs). In the linear factor reduction method, PCA is the best second-order technique, in which the mean-square error is calculated upon the covariance matrix of the variables. The proposed algorithm is given as a pseudo-code that accepts the features of the incident of the DDoS attack at various time instants on the network or system under attack, and the yield of trained NN is twofold representing normal and infected packets to avoid more infection of true nodes. The input is adjustable for the relevance threshold, and m number of examples sampled and x characteristics.

```
For u= (n, k)
{
//Initialization
        Creation of class label encoding L1, L2,…LC
Initialization u (1) = (n, k)
// C is Classification
        Set the maximum number of iteration MaxIteration
        Set precisionθ, set counter C = 1
// Dimensionality reduction
        While (r<Maxiteration | u (r') – u(r-1)2 |φ)
Do
{       r=r+1
Obtain an approximate solution with gradient iteration method V(r)
}
// print result
u = (n, k)
}
Import = (I1, I2,.….In)T. ω = ΣlixiyiT
// projection Matrix M'
I=1 {
Calculated the symmetric positive semi definite matrix, elements Aij = |yiTyj|xiTxjn|; for i = 1,
2,.….n:
Use the PBB method to solve the optimization problem $\frac{min\,|TA|}{2}$ - |T|, Constraint conditions
0<=l< ηl,
getl = {l1,l2,..ln} T     Given l(1) = {l1(1), l2(1), …ln(1)}T є Rn , λ1 >0
If l(1) , l(1) replace (l(1)) calculate   the projection   vector gk = Al(k -1, if |P(l(k)-gk)2 – l(k) | 2
< r
Stop the cycle and jump to the final output statement
Calculate l(k+1) = P(l(k) – λk ∩ gk)
                sk = l(k+1) – l(k), λ(k) = $\sum_{gk}^{T} skt/(skt(gk + lT - gk))$
// long of the step
k=k+1
The following is the final output statements
Import: l= (l1, l2,..ln)T, M' = ΣlixiyiT
}
```

We have taken a specific model with size 'n' as the input for the dimensionality procedure which produces the output data for n = [n1, n2, ...n]. The proposed workflow is depicted in Fig. 7 in which the input values from the package of information about the behavioral pattern, usage of bandwidth, and a database will be used to sort the size.

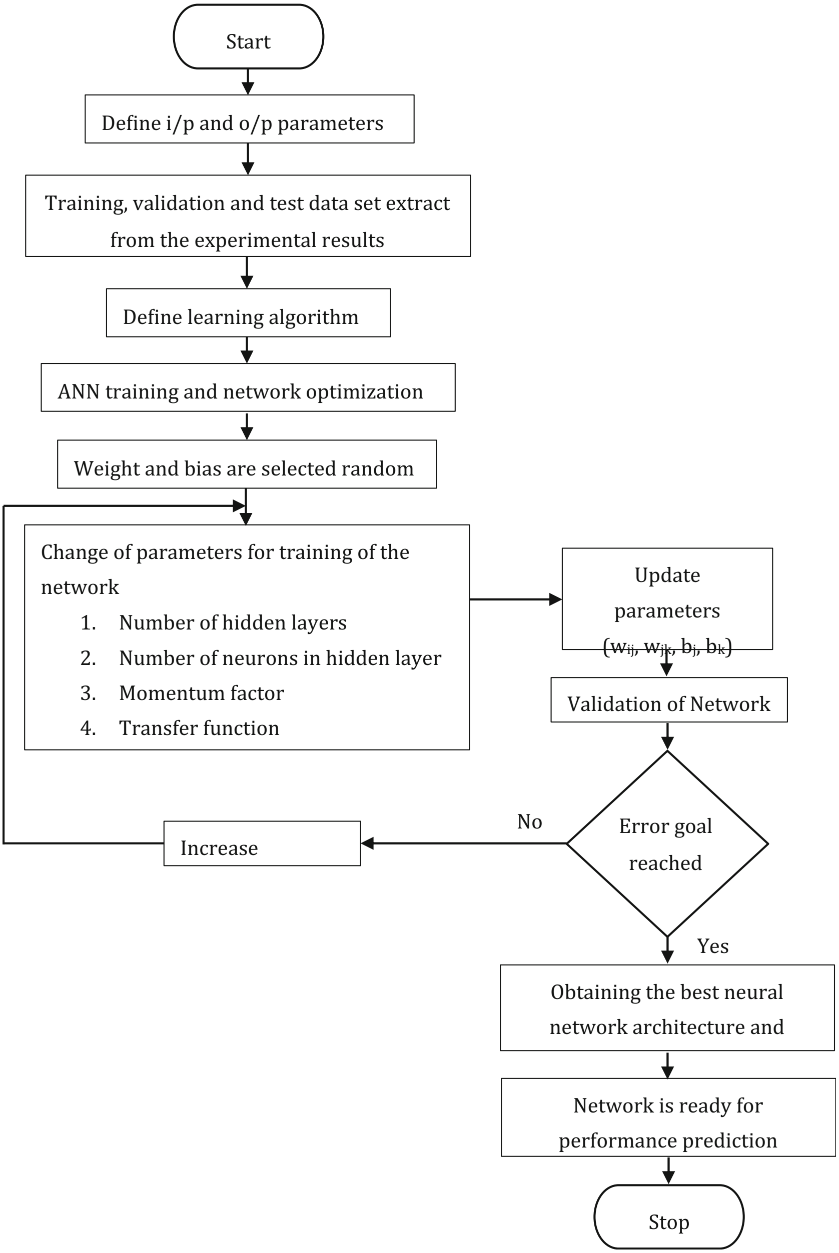
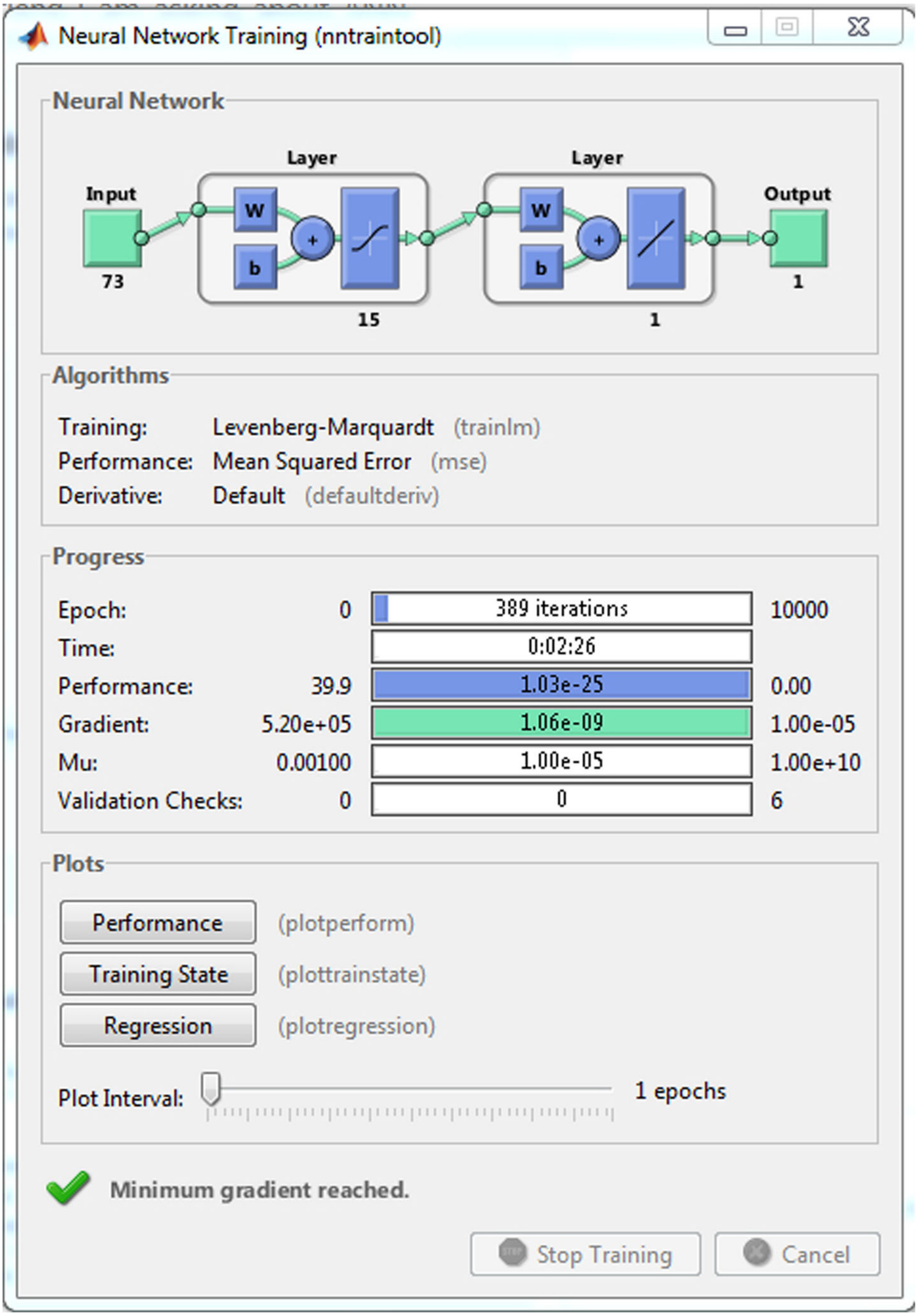

Start
Define i/p and o/p parameters
Training, validation and test data set extract from the experimental results
Define learning algorithm
ANN training and network optimization
Weight and bias are selected random
Change of parameters for training of the network
1. Number of hidden layers
2. Number of neurons in hidden layer
3. Momentum factor
4. Transfer function
Update parameters
$(w_{ij}, w_{jk}, b_j, b_k)$
Validation of Network
Error goal reached
No
Increase
Yes
Obtaining the best neural network architecture and
Network is ready for performance prediction
Stop

Fig. 7 Flow process of the proposed ANN training process



Neural Network Training (nntraintool)

Neural Network

Algorithms

Training: Levenberg-Marquardt (trainlm)

Performance: Mean Squared Error (mse)

Derivative: Default (defaultderiv)

Progress

```
Epoch:                   0   389 iterations   10000
Time:                        0:02:26
Performance:          39.9   1.03e-25         0.00
Gradient:         5.20e+05   1.06e-09         1.00e-05
Mu:                0.00100   1.00e-05         1.00e+10
Validation Checks:       0   0                6
```

Plots

Performance (plotperform)

Training State (plottrainstate)

Regression (plotregression)

Plot Interval: 1 epochs

✓ Minimum gradient reached.

Fig. 8 Screenshot of nntraintool utilized for projected work using LM

Table 1 Distribution of testing data

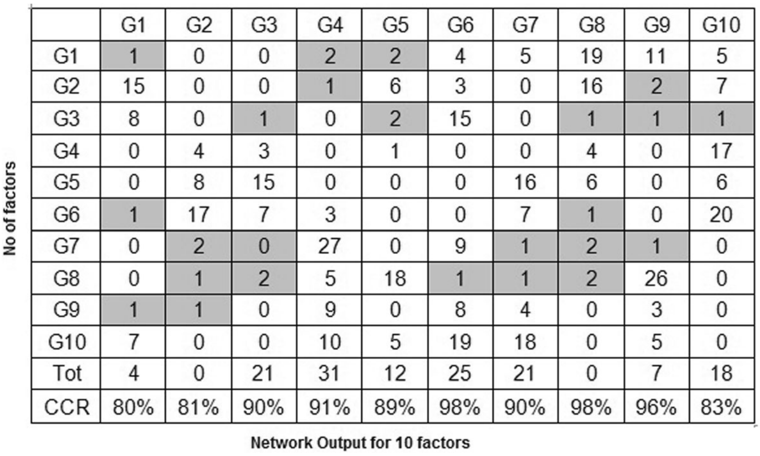
Types of attacks	Number of samples
Normal	6633
UDP Flood	6629
TCP SYN	6739

4 Experiment results

The specified characteristic set will be converted to the neural database and utilize Leven berg- Marquardt training procedure to train the same. The projected structure has been fully examined on a Celeron 1.85 GHz processor with 2 GB RAM on Windows XP Operating System and programmed in Matlab 6.5. The KDD Cup99 dataset is resulting from the DARPA98 network traffic dataset by accumulating each end every TCP packet into TCP connections has been used for benchmarking the proposed ANN model. Every TCP connection has 60 characteristics with a tag that determines the condition of a connection such as either normal or a particular assault type.

In the above flow process, the abbreviation denotes dimension reduced ANN model, which is the proposed model of ANN for DDoS attack detection in this article. The planned network can process the three classes of attacks like DDoS, DoS, and Probe which are found from the experimental investigations. The snapshot of the 'nntool' utilized in the implementation utilizing MSE principles and LM learning rules for error convergence is represented in Fig. 8.

We have used the dataset with 60,000 connections, out of which we have 20,000 normal connections, 20,000 TCP SYN flood attack connections and 20,000 UDP flood attack connections. The proposed IDS uses ANN with a Particle swarm optimization (PSO) design. The PSO is used to train the ANN in order to configure the anomaly classifier and select the optimal weights and bias. In our experiment, the anomaly classifier is trained with 40,000



No of factors

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
G1	1	0	0	2	2	4	5	19	11	5
G2	15	0	0	1	6	3	0	16	2	7
G3	8	0	1	0	2	15	0	1	1	1
G4	0	4	3	0	1	0	0	4	0	17
G5	0	8	15	0	0	0	16	6	0	6
G6	1	17	7	3	0	0	7	1	0	20
G7	0	2	0	27	0	9	1	2	1	0
G8	0	1	2	5	18	1	1	2	26	0
G9	1	1	0	9	0	8	4	0	3	0
G10	7	0	0	10	5	19	18	0	5	0
Tot	4	0	21	31	12	25	21	0	7	18
CCR	80%	81%	90%	91%	89%	98%	90%	98%	96%	83%

Network Output for 10 factors

Fig. 9 Feature vector matrix for the proposed network

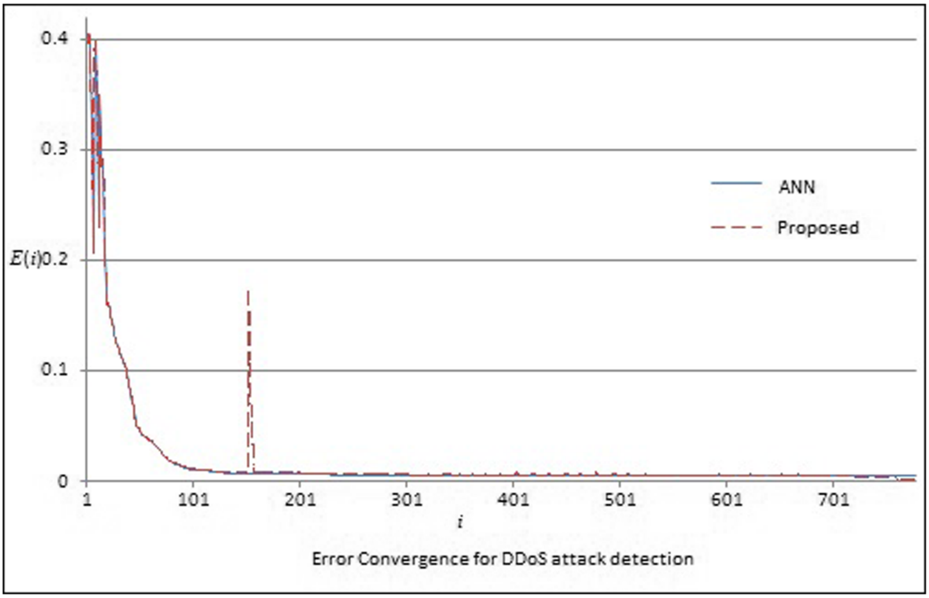


Fig. 10 Graph showing convergence of error of the projected model

connections and tested with 20,000 connections; the distribution of testing data is depicted in Table 1.

The network is trained for dissimilar values of error goal and epochs, where error goal and epochs are training factors. Usually, one epoch of training has been described as an individual presentation of every input vectors to the network. The network is then modernized as per the outcome of all such presentations. Training takes place until the highest possible counts of epoch happen, the target of the performance is fulfilled, or any other blocking situation of the training event happens.

We have received the superior detection precision as 55.86 and the number of feature set matrix after implementing the neural network as shown in Fig. 9.

Figure 10 demonstrates the error convergence for the planned work, and it could be seen that the projected technique has the ability to attain a quicker convergence in comparison to the current traditional ANN models. Moreover, the data dimension has been greatly reduced due to the integration of PCA with ANN optimization.

We have tested the proposed approach with the Feedforward neural network parameters as given in Table 2.

Table 2 Neural network parameter

Feedforward neural network parameters	
Number of layers	3 (input, one hidden layer, output)
Number of neurons in input layer	13 attributes
Number of neurons in hidden layer	8
Number of neurons in output layer	3
Maximum iteration	1000
Activation function	Sigmoid

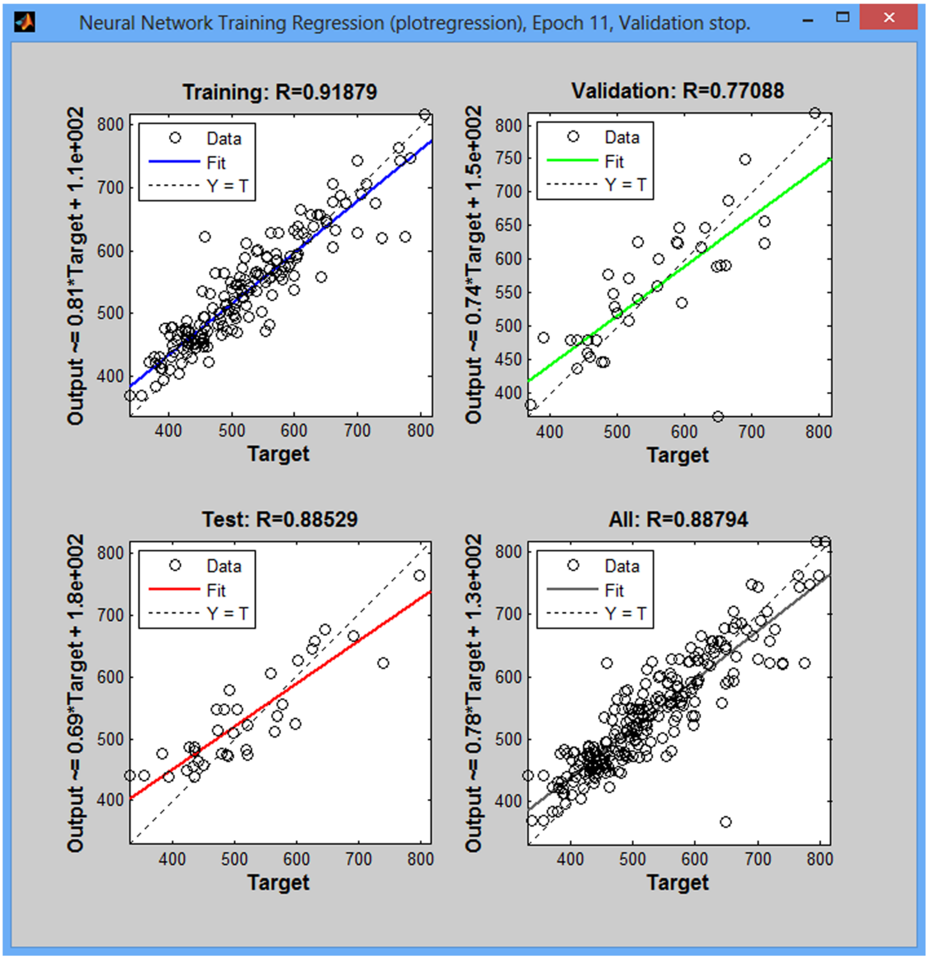


Fig. 11 Graph showing regression plot of the planned network

The performance of the anomaly classifier can be evaluated as follows:

- True Positive (TP): The number of positive records that are correctly classified. The rate of true positive or sensitivity can be computed by

$$\mathrm{TPR} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN})\mathrm{TPR} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN}) \tag{8}$$

Table 3 Receiver operational features

Parameters	ANN	Proposed ANN
Elapsed Time	0.92 s	0.71 s
Packet delivery ratio	0.74	0.51
Receiver throughput	812 kbps	770 kbps
Transmission throughput	790 kbps	646 kbps

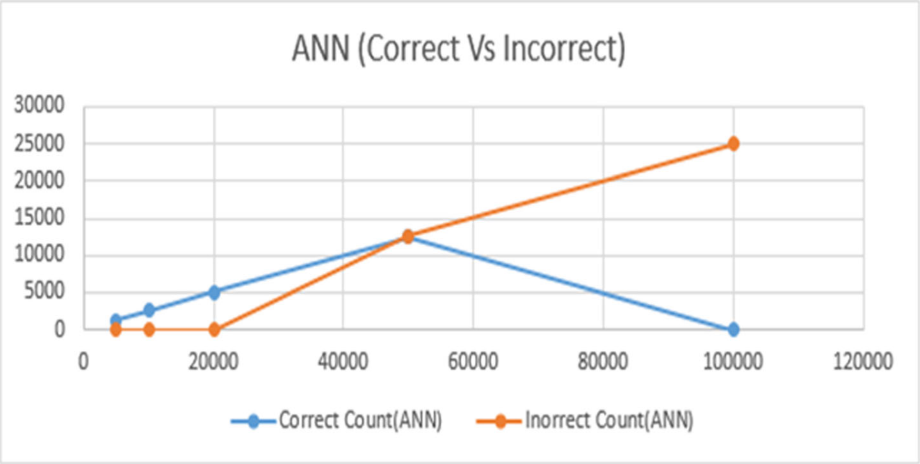


Fig. 12 Prediction performance of proposed ANN

- True Negative (TN): The number of negative records that are correctly classified. The rate of True Negative or specificity can be calculated by

$$\mathrm{TNR} = \mathrm{TN}/(\mathrm{FP} + \mathrm{TN}) \tag{9}$$

- False Positive (FP): The number of records that are incorrectly identified as a positive, although they are in fact negative. It can be obtained by

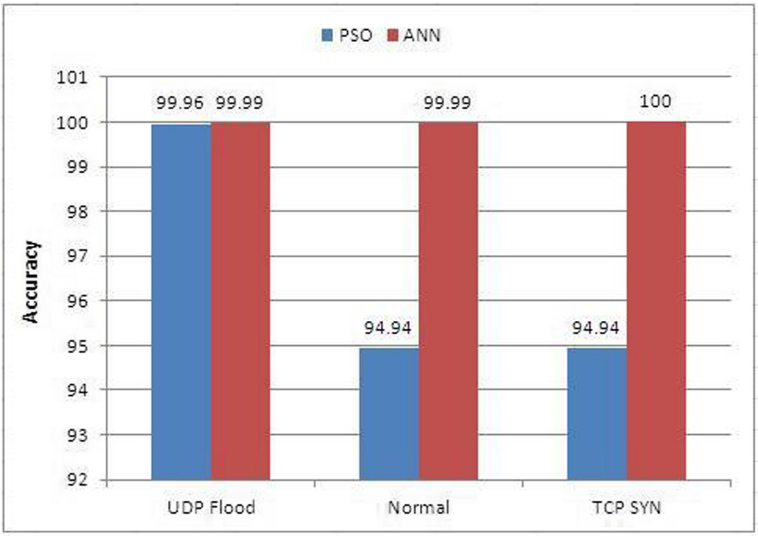


Fig. 13 Detection rate for ANN and PSO

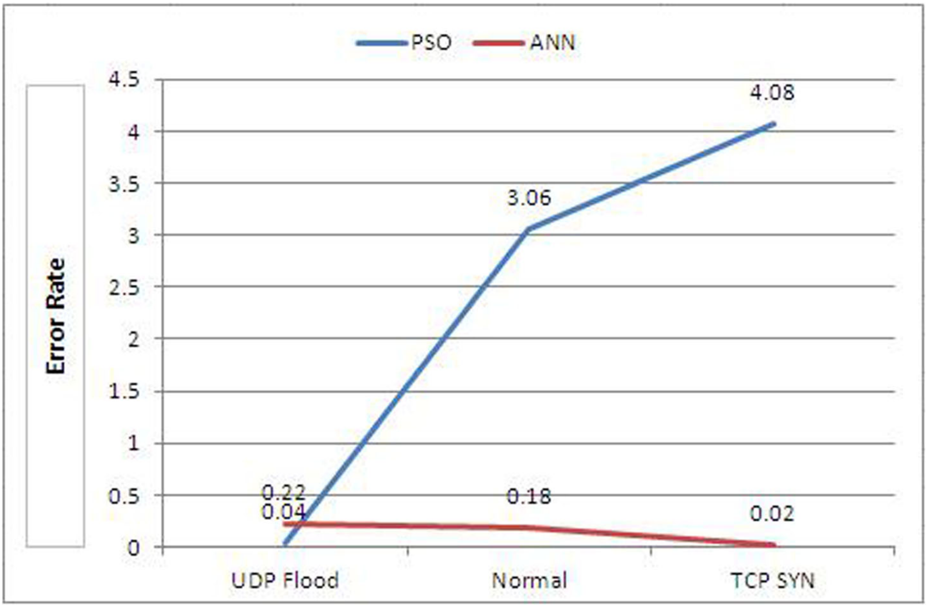


Fig. 14 Error rate for ANN PSO

$$\mathrm{FPR} = \mathrm{FP}/(\mathrm{FP} + \mathrm{TN})\mathrm{FPR} = \mathrm{FP}/(\mathrm{FP} + \mathrm{TN}) \tag{10}$$

- False Negative (FN): The number of records that are incorrectly identified as a negative, although they are in fact positive. It can be computed by

$$\mathrm{FNR} = 1{-}\mathrm{TPRFNR} = 1{-}\mathrm{TPR} \tag{11}$$

Figure 11 portrays the regression plan for the projected ANN model. This plot is attained for the test, authentication, and total stages of the ANN implementation.

The regression plot is attained for the test, authentication, and total stages of the implementation. The optimum error value of 0.9975E-05 is attained that provides a great percentage of classification. An analysis of the receiver output characteristics has been tabularized in Table 3.

An analysis of the predictions of the proposed ANN which are correct/incorrect is given in Fig. 12.

We have compared the proposed ANN model with the PSO model. The performances of these two approaches are compared by using the accuracy of detection, which is presented in Fig. 13. Figure 14 shows the comparison of the error rate for the two approaches. From these graphs, it is proved that the ANN-based model has maximum accuracy and the minimum false alarm rate when compared to PSO based approach.

5 Conclusions

This paper has effectively dealt with the implementation of an improved ANN model with data dimension reduction feature for implementing an attack detection system, especially DDoS types of attacks which prove to be a menace in recent times. The projected model has been carried out utilizing mathematical, and constraints formulations were dealing with and elaborated in the earlier sections and tested against a wide variety of network incidents. Investigational observations have been tabularized and visually characterized, and the findings specify an excellent performance in comparison to the current ANN methods for DDoS attack detection. The projected technique is trained by the LM algorithm, which detects the abnormal patterns of the incoming patterns and entitles them into infected packets which are isolated from triggering further infections while they progress along with the network. When such infected operatives are freed, the network bandwidth has been blank, and the initial rate of internet usage will be restored to the user, therefore, meeting the issue target.

References

1. Arivudainambi D, Varun Kumar KA, Chakkaravarthy SS (2019) LION IDS: a meta-heuristics approach to detect DDoS attacks against software-defined networks. Neural Comput Applic 31(5):1491–1501
2. Chhaya L, Sharma P, Bhagwatikar G, Kumar A (2017) Wireless sensor network based smart grid communications: cyber attacks, intrusion detection system and topology control. Electronics 6(1):5
3. Colom JF, Gil D, Mora H, Volckaert B, Jimeno AM (2018) Scheduling framework for distributed intrusion detection systems over heterogeneous network architectures. J Netw Comput Appl 108:76–86
4. Duraipandian M, Palanisamy C (2015) Analysis of a combined parameter-based multi-objective model for performance improvement in wireless networks. Wirel Pers Commun 83(4):2425–2437
5. Farris I et al (2018) A survey on emerging SDN and NFV security mechanisms for IoT systems. IEEE Commun Surv Tutor 21(1):812–837
6. Ferrag MA, Maglaras L, Ahmim A, Derdour M, Janicke H (2020) RDTIDS: rules and decision tree-based intrusion detection system for internet-of-things networks. Futur Internet 12(3):44
7. Fu Y, et al. (2017) An automata based intrusion detection method for internet of things. Mob Inf Syst 2017
8. Gandhi UD, Kumar PM, Varatharajan R, Manogaran G, Sundarasekar R, Kadu S (2018) HIoTPOT: surveillance on IoT devices against recent threats. Wirel Pers Commun 103(2):1179–1194
9. Gendreau AA, Moorman M (2016) Survey of intrusion detection systems towards an end to end secure internet of things. In: 2016 IEEE 4th international conference on future internet of things and cloud (FiCloud), pp. 84–90. IEEE
10. Golrang A, Golrang AM, Yayilgan SY, Elezaj O (2020) A novel hybrid IDS based on modified NSGAII-ANN and random forest. Electronics 9(4):577
11. Hassija V, Chamola V, Saxena V, Jain D, Goyal P, Sikdar B (2019) A survey on IoT security: application areas, security threats, and solution architectures. IEEE Access 7:82721–82743
12. Hussain F, Hussain R, Hassan SA, & Hossain E (2020) Machine learning in IoT security: current solutions and future challenges. IEEE Commun Surv Tutor
13. Khan R, Kumar P, Jayakody DNK, Liyanage M (2019) A survey on security and privacy of 5G technologies: potential solutions, recent advancements and future directions. IEEE Commun Surv Tutor
14. Kim J, Kim J, Kim H, Shim M, Choi E (2020) CNN-based network intrusion detection against denial-of-service attacks. Electronics 9(6):916
15. Ksasy MS et al (2018) A new advanced cryptographic algorithm system for binary codes by means of mathematical equation. ICIC Exp Lett 12(2):117–125
16. Manimurugan S, Al-Mutairi S, Aborokbah MM, Chilamkurti N, Ganesan S, Patan R (2020) Effective attack detection in internet of medical things smart environment using a deep belief neural network. IEEE Access 8:77396–77404
17. Manso P, Moura J, Serrão C (2019) SDN-based intrusion detection system for early detection and mitigation of DDoS attacks. Information 10(3):106

18. Napiah MN, Idris MYIB, Ramli R, Ahmedy I (2018) Compression header analyzer intrusion detection system (cha-ids) for 6lowpan communication protocol. IEEE Access 6:16623–16638
19. Pejic D, Arsic M (2019) Minimization and maximization of functions: golden-section search in one dimension. In: Exploring the Data Flow Supercomputing Paradigm (pp. 55–90). Springer: Cham
20. Suratgar AA, Tavakoli MB, Hoseinabadi A (2005) Modified Levenberg-Marquardt method for neural networks training. World Acad Sci Eng Technol 6(1):46–48

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

R. Gopi[1] • **V. Sathiyamoorthi**[2] • **S. Selvakumar**[1] • **Ramasamy Manikandan**[3] • **Pushpita Chatterjee**[4] • **N. Z. Jhanjhi**[5] • **Ashish Kumar Luhach**[6]

[1] Department of Information Technology, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamilnadu, India

[2] Department of CSE, Sona College of Technology, Salem, India

[3] School of Computing Science and Engineering, VIT Bhopal University, Bhopal, India

[4] Future Networking Research Group, Ton Duc Thang University, Ho Chi Minh City, VA, Vietnam

[5] School of Computer Science and Engineering SCE, Taylor's University, Subang Jaya, Malaysia

[6] Department of Electrical & Communication Engineering, The PNG University of Technology, Lae, Papua New Guinea