JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

DETECTION OF DDoS ATTACKS ON NETWORKS USING AN IMPROVED NOVEL ANN MODEL

[1]Dr.R.BASHEER MOHAMED, [2]M.SHAHEDA BEGUM, [3]S.RUMANA FIDOSE, [4]A.HARIPRIYA, [5]J.K.SUPRIYA

[1]Professor, [2]Assistant Professor, [2]Assistant Professor, [3]Assistant Professor, [4]Assistant Professor, [5]Assistant Professor

[1]Computer Science & Engineering, [2]Computer Science & Engineering, [3]Computer Science & Engineering, [4]Computer Science & Engineering, [5]Computer Science & Engineering

[1]Ashoka Women's Engineering College, Kurnool, AP, India, [2]Ashoka Women's Engineering College, Kurnool, AP, India, [3]Ashoka Women's Engineering College, Kurnool, AP, India, [4]Ashoka Women's Engineering College, Kurnool, AP, India, [5]Ashoka Women's Engineering College, Kurnool, AP, India

Abstract: Attacks on the internet have been a growing threat in recent years, with hackers attempting to hack or illegally tamper with data available across networks. On the other side, the number of research contributions to effectively counter these attacks and develop a strong defense mechanism has increased. In recent years, a slew of intelligent and soft computing-based algorithms and frameworks have been established. These evolution-based algorithms play a critical role in self-adapting the system under assault to the ever-increasing number of different sorts of attacks. The artificial neural network, also known as ANNs, is one of the soft computing algorithms examined in this . They function in the same way that organic neurons do in the human body. The chapter is organized in a systematic manner to provide an understanding of ANN-based network models to counter DDoS attacks, which is the paper's main focus, as well as the architecture and implementation of ANNs, as well as experimental investigations and findings that aid in drawing inferences about ANN-based defense models.

Keywords: Network attacks, distributed denial of service attacks, ANN, training and confusion matrix.

I. INTRODUCTION

DDoS attacks are a type of online communication attack that occurs when a network uses internet services for storage, processing, or utility [2]. By injecting zombie packets, which are infected packets that contaminate good packets as they move along the communication layers, these assaults increase network congestion. In a DDoS attack, the attacker or hacker delivers an array of infected packets, causing flooding [10], which causes the target system to become overburdened in order to fulfil the flooded requests, reducing network bandwidth and increasing computation system overhead. Although present techniques can detect almost all attacks, there are some attacks that cannot be detected and are referred to as zero attacks.

Figure 1 shows a simple illustration of an intrusion detection technique in which data packets are pre-processed and the conditioned input is delivered to the IDS system. The signature and behavior patterns of irregularities in terms of file size, frequency of repetition, bandwidth, and so on are extracted from the packets.

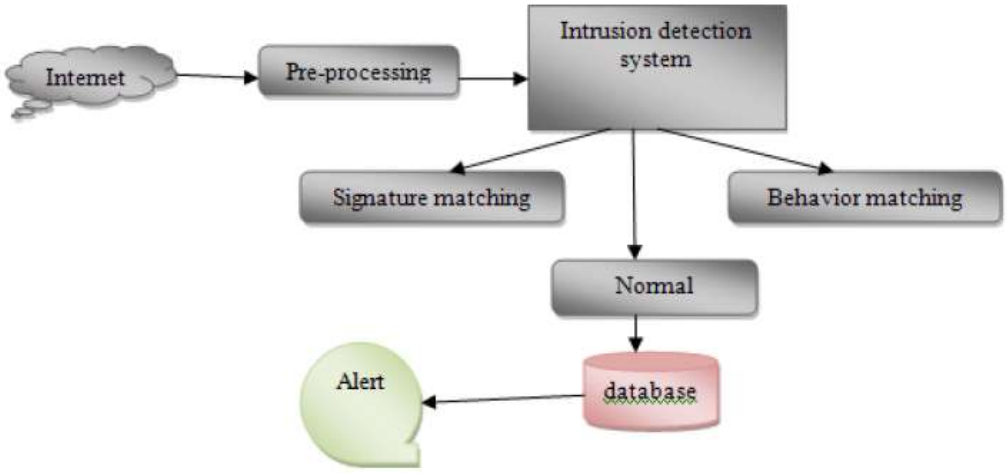

Figure 1 General scheme of intrusion detection mechanism

The given packet of information is identified as normal or infected based on the retrieved features and the adaptation of the IDS system. The main reason for pursuing ANN-based DDoS attack detection in this paper is that they are capable of detecting zero-day assaults based on a certain pattern. These patterns differentiate regular attacks from DDoS attacks. These characteristics could be used to train neural networks to increase detection accuracy. Prior to the actual implementation, it is critical to understand the critical components of the intrusion detection framework, which determine the system's efficiency. The next section delves more into these elements or factors.

II. RELATED WORK

In the literature, rule-based approaches for detecting DDoS attacks on wireless sensor networks have been discovered [4], [8-9], and the process is divided into three steps. With the help of monitor nodes, the first step of information monitoring is carried out in order to filter critical data from the main network. The rule application stage is the second phase, in which predetermined rules are applied to the information that has been filtered by the monitor nodes in the first stage. If any of the information packets fail the rule application test in the final step, a detection alarm is raised. The behavior patterns of the neighboring node in the senor network are extracted in a slight version [12] of this technique.

In this section, a cumulative sum algorithm [15] has been described that continuously examines incoming and outgoing packets of data for any behavioral and pattern changes. The nodes are initially watched and detected whether they originate from genuine or illegitimate nodes, according to a similar three-phase scheme documented in the literature [7]. If they come from valid nodes, the system's typical functions are completed. The features of information packets originating from illegal nodes are extracted, and the rule basis is applied to this data. The above-mentioned set of predefined rules is framed by examining network protocol patterns in a typical communication network. Any variation from the typical pattern flags the incoming packet as unlawful, and subsequent procedures filter it out. Based on an event processing paradigm, another form of the rule-based system can efficiently detect DDoS attacks in IoT networks [4]. The experiments were carried out with SQL as a guide, and the rule codes were saved in the repository. The experimental results show that the memory is used the least, but it has the disadvantage of consuming more system resources at the expense of processing time.

Because of the nature of intelligence and the ability to manage large volumes of incoming and outgoing data while producing a short response time, detection systems have migrated to cluster computing and soft computing. Principal component analysis [6], linear discriminant analysis [12] [15], local binary pattern [10], particle swarm optimization, and greedy search methods are only a few of the soft computing algorithms [14] that have been discovered in the literature. Support vector machine [9] based approaches have also been reported in the literature to be useful for categorization of harmful information packets. For the identification of known threats, support vector machine algorithms have been imposed on mobile agent models. In the literature, two further mobile agents have been identified: the collector agent, which provides input from the wireless sensor network, and the abuse detection agent, which detects known malicious tendencies in the network. Support vector machines were combined with a Gaussian kernel and tested on three different types of data sets, with a 98.7% accuracy reported in the literature.

PCA-based strategies have been reported [11] to reduce data dimensionality, particularly when dealing with large volumes of traffic and a variety of feature-based attacks that occur on the network's mainframe. PCA-based algorithms have been proven to have a high degree of classification due to their capacity to distinguish malicious packets from normal data packets using numerous attribute values, in addition to dimension reduction.

III. PROPOSED WORK

Artificial neural networks, which behave similarly to neurons in our central nervous system, are crucial networks in almost any data processing and computing application. ANNs have high interconnected parts in perfect coordination to achieve an objective function, similar to biological connectivity of neurons. They are used in a variety of applications, such as pattern recognition and classification, detection difficulties, adaptation and control, and so on. Figure 2 depicts a rudimentary artificial neural network system.

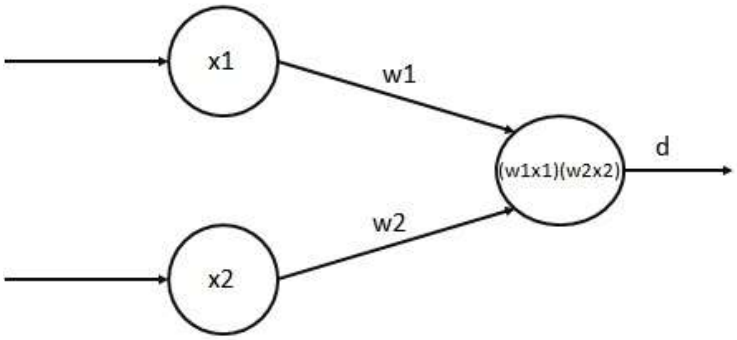

Figure 2 Illustration of a simple neural network model

The diagram above depicts a two-input neural network system with a simple operation. The two inputs x1 and x2, as well as an associated weight function, are fed into the function block shown by the larger circle, which performs a simple product between these two numbers to produce the required output. The neural networks in a feedback system can alter the weights w1 and w2 in subsequent iterations based on the error signal generated by the difference between the obtained and desired outputs. The overall goal of the network illustrated above is to minimize error in the shortest amount of time possible.

The above single layer neural network could be further developed by connecting many more multiple nodes with the goal of changing inputs to desired outputs. Multilayer perceptron models (MLP) are a widely used configuration for intrusion detection systems, in which several inputs of varying patterns are incident on the target system under attack.

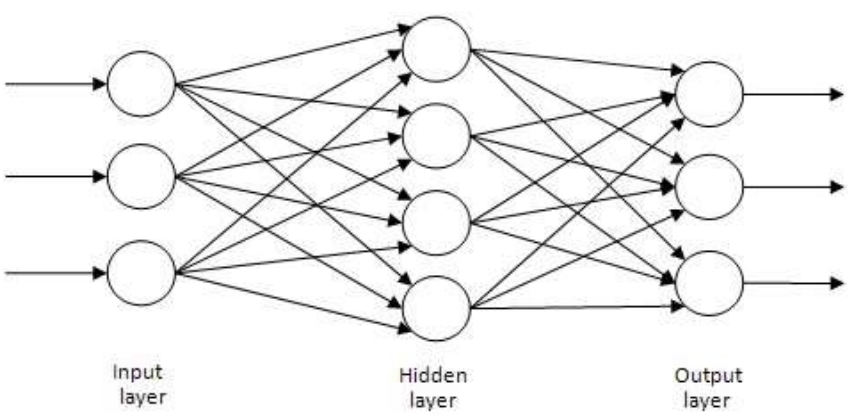

Figure 3 Architecture of Multi-layer perceptron model

The input, hidden, and output layers of the above architecture illustrate a three-layer ANN. In the above problem definition, the number of nodes in the input layer corresponds to the feature vector. The output layer's nodes correspond to the number of sets or classes to which the desired throughput can be assigned. For a brain tumor detection and classification challenge, for example, the output nodes could be two to three, depending on whether the brain tumor is malignant or benign. The number of output layers in the provided problem of attack detection may be limited to two, each of which contains an infected or genuine packet of information.

The input and output layers are linked by the hidden layer, and the updating process starts with some random weights provided to each node. After the first iteration is completed, the weights are adjusted using the weight update equation stated in (1) to minimize the error at the output (2).

$$WI(x, y) + \alpha \quad (1)$$

and

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

where $E\{ e^2[n] \}$ denotes the expectation of mean squared error function, d[n] indicates the desired output and y[n] denotes the obtained output.

This iteration process is guided by a learning algorithm that could include a propagation rule mechanism.

(i) Neural Network Training

Training is the act of causing the implemented neural network to learn feature vector patterns and, as a result, decide on categorizing the given inputs into a class of designated outputs. It is the backbone of neural network efficiency and performance. As shown in figure 4, the overall goal of neural network learning is to lower the loss function using a minimization strategy.

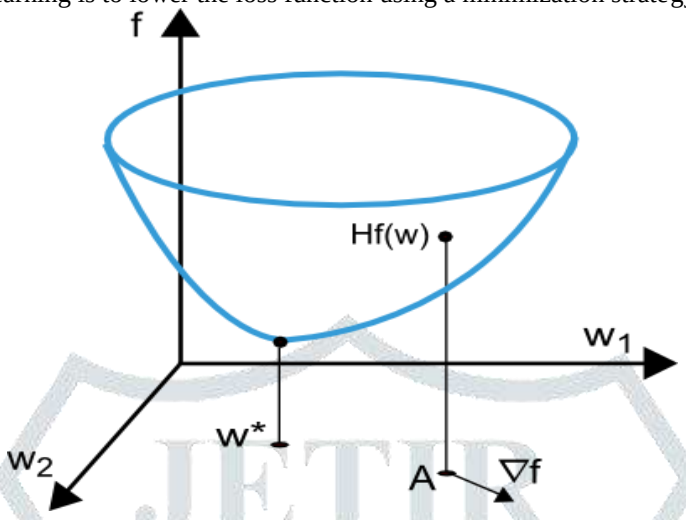

Figure 4 Illustration of overall loss function for given problem

The point w* signifies the minimal value of the given loss function, and the solution to the given issue objective can be obtained by computing the first and second derivatives of this loss function as stated in equations (3 &4).

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

The second derivative could be generalized using the Hessian matrix as

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

The solution to the provided minimization issue could be obtained by mapping these mathematical formulations onto one-dimensional search spaces, as shown in figure 5.

In the figure 5 the minimal function of loss is present in between the point's $\eta_1$and η_2. Golden section and Brent's method is popularly used algorithm for one dimensional loss minimization function. However, most of the real time problem definitions in real time require a multi-dimensional search and minimization strategy including the problem objective of this paper. The next section deals with multi-dimensional optimization techniques and methodologies in detail which greatly aid in implementing the proposed architecture for defense against attacks.

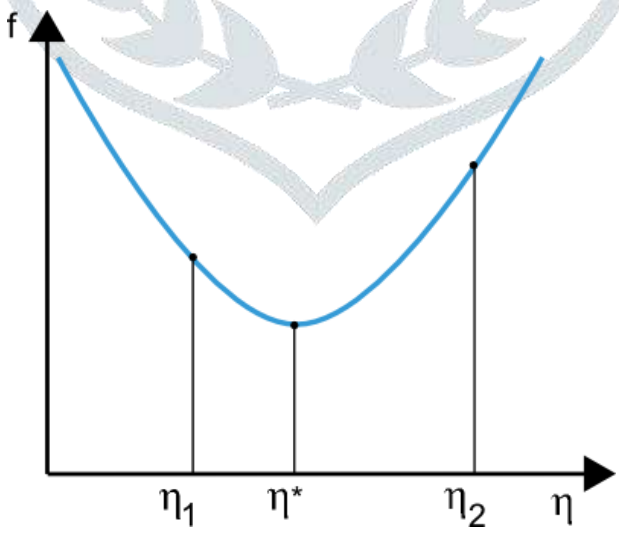

Figure 5 One dimensional mapping of minimum loss function

(ii) Levenberg – Marquardt Method

It is also known as the damped least squares approach and is based on sum of squares iteration. This approach also avoids interfering with the Hessian matrix and its inverse values, instead focusing on a different matrix known as the Jacobian Matrix. A loss function of the form is used to define the Jacobian matrix.

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

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

The weight update equation is given as

$W_{i+1} = W_i - (J^T J + \delta I)^{-1}(2J^T \cdot e_i), \ i = 0,1,\ldots$ (7)

When indicates the damping function in the preceding equation, it becomes a traditional Newton's technique when it equals zero. The word JT J+I signifies the approximation of the Hessian matrix, and I is the identity matrix. To reach to the convergence value, the damping factor value is dropped or increased, and the acceleration towards minima is quite fast with this method, making it suited for training medium-sized neural models. When the size of the Jacobian matrix is doubled, the problem occurs for big networks, affecting cost and complexity.

(iii) Proposed ANN model

This section proposes and implements a three layer ANN feed forward model with modification of principal component analysis (PCA) for dimensionality reduction based on investigations and discoveries connected to ANN implementation methods and their benefits. The purpose of providing a dimensionality reduction is to lower the storage and computation complexity of an LM learning rule for creating Jacobian matrices for large NN models, such as those utilized in this paper. Figure 11 shows the ANN model used in this approach, which is given the reduced dimension input features before training.

The stages of using PCA to reduce the dimension of a high-dimensional data set are listed below.

S1: create a matrix M using the data set.

S2: Normalize the data set using δ-value.

S3: Calculate the decomposition value of the data matrix. $M_x \equiv \frac{1}{x^{n!}} G G^T$

S4: Calculate the variance using the diagonal elements

S5: Sort variances in decreasing order.

S6: Choose the $x$principal components from $M$with largest variances.

S7: Form the transformation matrix M'_xconsisting of those $x$values

S8: Find the reduced projected data set in a new coordinate axis by applying $M'_x$to x.

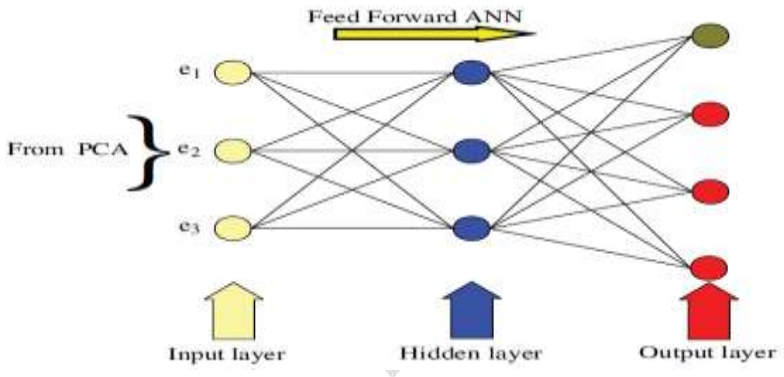

Figure 11 Feed forward ANN architecture for DDoS

The initial centroid value for the input data set with dimensionality M is computed initially, as described in the algorithm above. The variance is then computed for each data point in the dimension. M max is the column with the highest variance, and it is ordered in ascending order of magnitude. Each median is initialized in the cluster centers of the total k subset clusters. The best linear dimension reduction technique is PCA, which is a second-order method that uses the covariance matrix of the variables to calculate the mean-square error. The suggested approach is presented as a pseudo code that accepts the characteristics of DDoS attacks that occur at various times on the network or system under attack, and the output of the trained NN is two-fold, presenting a normal and infected packet to prevent genuine node infection.

The input has x characteristics and m sampled occurrences, and the relevance threshold can be adjusted.

Initialize the matrix

For $i = 1$ to n

begin

randomlyselectaninstanceI

findthenearesthitHandmissJ

for$j = 1$ toN

$$s(i) = S(j) - \frac{|j, I, H^2|}{N} + \frac{diff(j, H^2)}{N}$$

end
Initialize the matrix M = { },
fori = 1 toN
for each subset of S with size j
fdvar(s, m) = 0
return M
the minimum subset satisfies the δ

The dimensionality dimension algorithm provides the data output about the dimension for the input and for a given sample n = [n1, n2, ...n] ∈ φ_n where n is sample size
For u = (n, k)
{
//Initialization
Generation of class label encoding $L_1L_2, L_3 \dots \dots, L_c$
Initialization u(1) = (n, k)
// C is Classification
Set the maximum number of iterations MaxIteration
Set precision ϑ, set counter C = 1
//Dimensionality reduction
While($r < MaxItera \&\& |u(r') - u(r-1)^2| < \varphi_n$)
Do
{
r= r+1
Obtain an approximate solution with gradient iteration method V(r)
}
//print result u = (n, k) }
Import=(l1, l2,……ln) T,ω=Σ lixiyiT
//projection matrix M'
i = 1 {
Calculate the symmetric positive semi definite matrix M, elementsAij = |yiTyj|xiTxj_n, for i = 1,2, … … , n; use the PBB method to solve the optimization problem $\frac{\min lTAl}{2} - lTl$, Constraint conditions 0<= $l < \eta l$,
getl = {l1, l2, … … ln, }T Given l(1) = {l1(1), l2(1), … … ln(1) }T ∈ Rn , λ1 > 0
If l(1)， l(1) replaceρ (l(1))Calculate the projection vector gk = Al(k) − 1 ,If |P(l(k) − gk)² − l(k) |2 < τ
Stop the cycle and jump to the final output statement
Calculate l(k + 1) = P(l(k) − λk ∩ gk)

$$sk = l(k+1) - l(k), \lambda k = \sum skt_{gk}^{T} / (skT(gk + 1T - gk))$$

//long of the step
k = k + 1,
The following is the final output statements
Import： l = (l1, l2, … … ln)T, M′ = Σlixiyi T
}

IV. EXPERIMENTAL ANALYSIS AND FINDINGS

The proposed work flow is presented in figure 12, in which the database is sorted based on the input parameters from the packets of information on their size, bandwidth occupancy, and behavioral pattern.

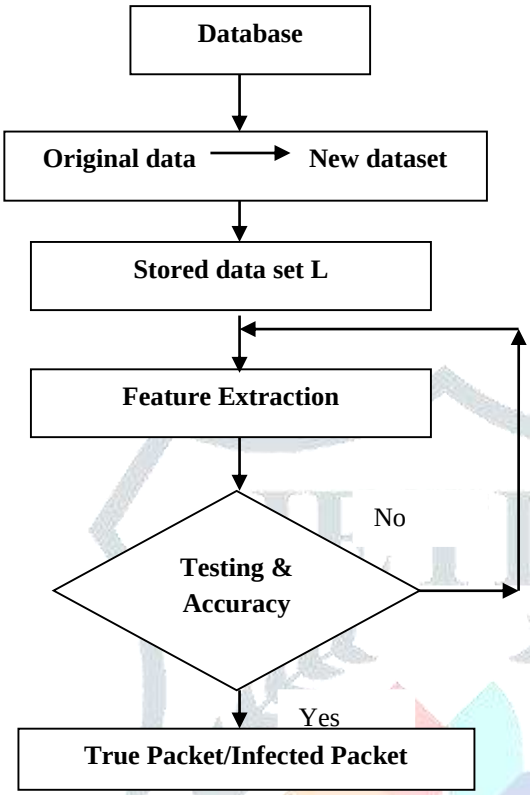

Figure 12 Flow process of proposed DR- ANN

The given feature set is converted to a neural database and trained with the Levenberg-Marquardt method. The suggested system is programmed in Matlab 6.5 and tested on a Celeron processor 1.85 GHz with 2GB RAM running Windows XP. The proposed ANN model was benchmarked using the KDD Cup99 dataset, which was created from the DARPA98 network traffic dataset by combining individual TCP packets into TCP connections. Each TCP connection includes 60 features, each of which has a label that indicates whether the connection is normal or under attack.

The abbreviation in the above flow process stands for dimension reduced ANN model, which is the proposed ANN model for DDoS attack detection in this paper. The proposed network can handle three types of attacks: DDoS, DoS, and Probe, according to the results of the experimental investigations. Figure 13 shows a screenshot of the 'nntool' used in the execution of the LM learning rule and MSE criteria for error convergence.

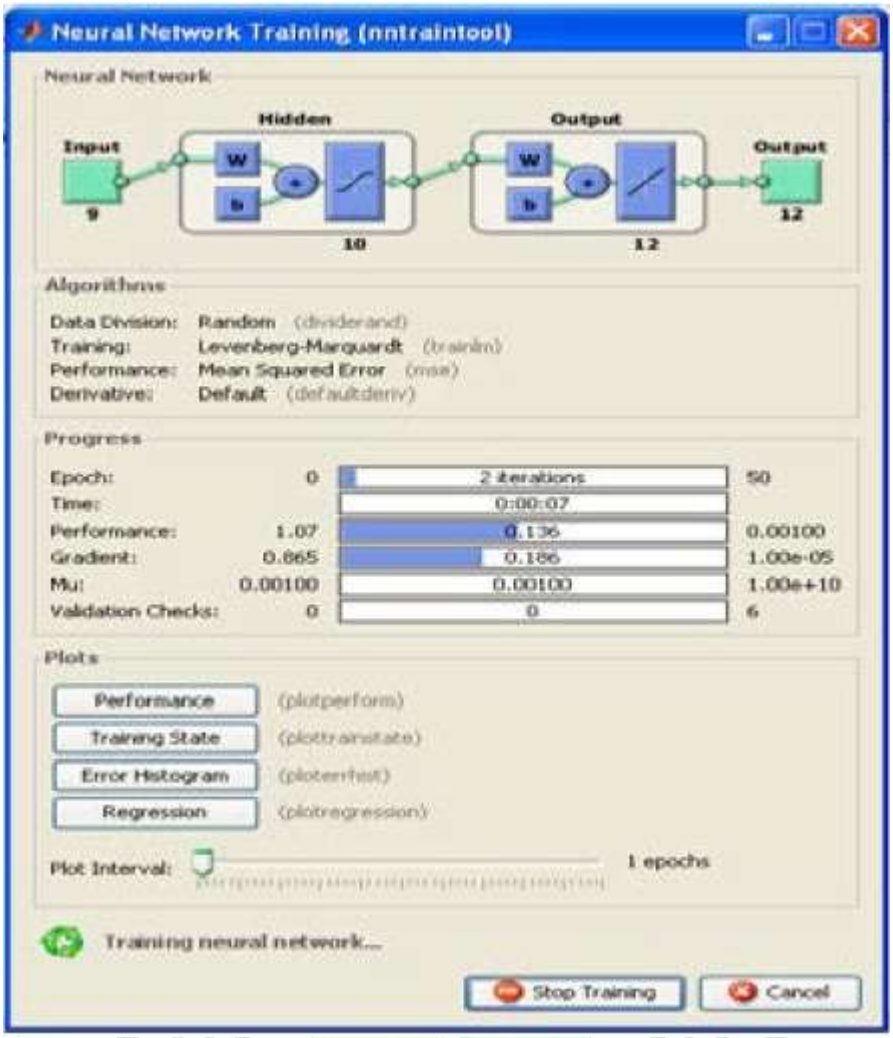

Figure 13 Snapshot of nntrain tool used in the proposed work using LM

The network is trained for various epoch and error goal values, with epoch and error goal serving as training parameters. A single presentation of all input vectors to the network is typically defined as one epoch of training. Following that, the network is updated based on the outcomes of all of the presentations. Training continues until the maximum number of epochs has been reached, the performance target has been reached, or the training function has reached any other stopping condition.
We acquired a superior detection accuracy of 55.86 after running the neural network, and the number of feature set matrix is shown in figure 14.

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

Figure 14 Feature vector matrix for the proposed network

Figure 15 shows the error convergence for the suggested work, and it can be shown that the proposed method is able to produce a faster convergence than existing traditional ANN models. Furthermore, thanks to the combination of PCA with ANN optimization, the data dimension has been considerably reduced.
The regression pattern for the suggested ANN model is shown in Figure 16.

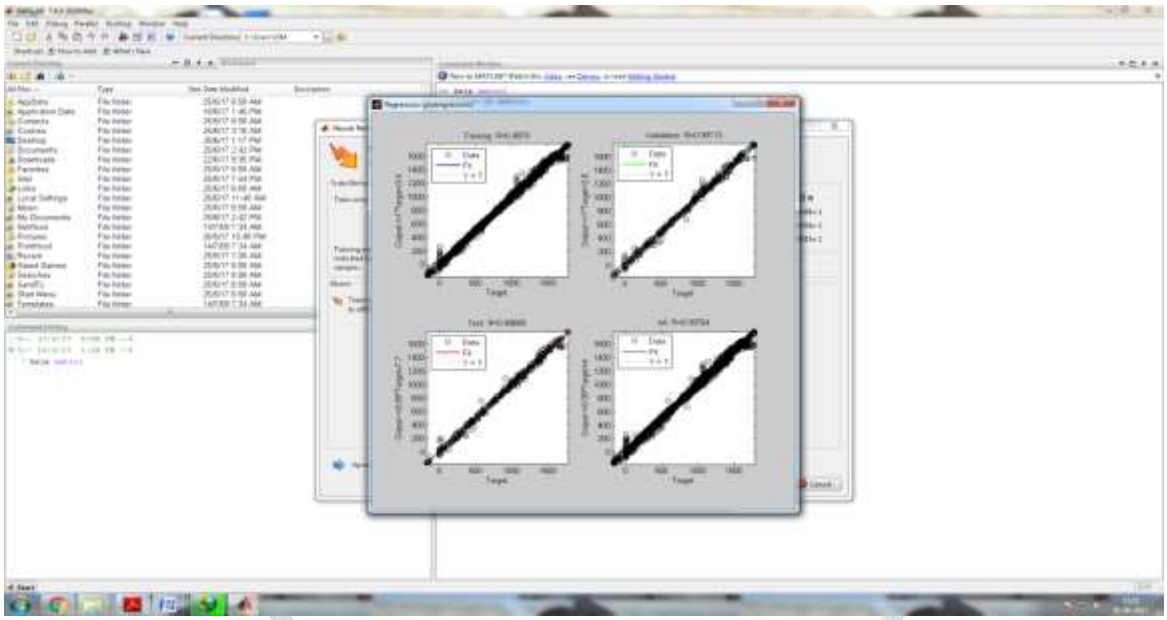

Figure 16 Regression Plot for the proposed network

For the test, validation, and overall phases of the ANN implementation, the regression plot was generated. The ideal error value of 0.9975E-05 is obtained, resulting in a high classification rate. Table 5.1 shows the results of an investigation of the receiver output characteristics.

Table 1 Receiver operating characteristics

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

Figure 17 shows an analysis of the proposed ANN's predictions that are true or erroneous.

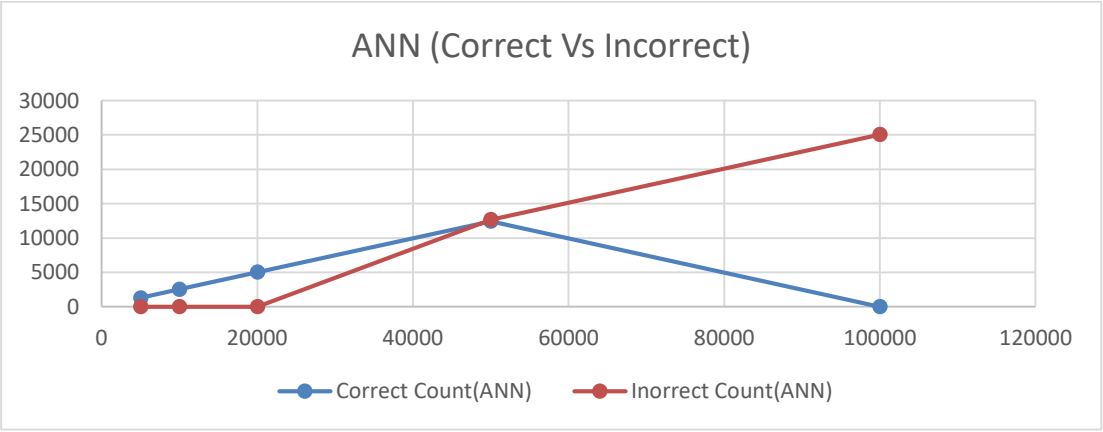

Figure 17 Prediction performance of proposed ANN

V. CONCLUSION

This study effectively dealt with the development of an enhanced ANN model with data dimension reduction feature for constructing an attack detection system, particularly DDoS-type attacks, which have become a serious threat in recent years. The suggested model has been built utilizing constraints and mathematical formulations that have been thoroughly discussed and extended in previous sections of this paper, and it has been tested against a variety of network assaults. The results of the experiments have been tabulated and visually depicted, and they show that when compared to existing ANN algorithms for DDoS attack detection, this method outperforms them. The suggested solution is based on the LM algorithm, which detects anomalous patterns in incoming patterns and classifies them as infected packets that are quarantined to prevent future infections as they travel through the network. Once the infected agents have been removed, the network capacity has been cleared, and the user's original internet speed has been restored, achieving the problem's goal.

VI. REFERENCES

[1]. Ali E.Taki El_Deen,El-Sayed A.El-Badawy and Sameh N.Gobarn (2014), "Digital Image Encryption Based on RSA Algorithm", International Journal of Electronics and Communication Engineering, Vol.9, issue-1, pp. 69-73.

[2]. Arumugam N and C. Venkatesh (2013), "Ant system algorithm based IP traceback method to detect denial of service attack on data network", Australian Journal of Basic and Applied Sciences, Vol. 7, 2013, pp. 732-741.

[3]. Audrey A. Gendreau, Michael Moorman (2016), "Survey of intrusion detection systems towards an end to end secure internet of things", Proceedings of international conference on future internet of things and cloud.

[4]. Bass T (2008), "Intrusion detection systems and multi-sensor data fusion", Communications of the ACM, Vol. 43, No. 4, pp. 99-105.

[5]. Batalla J M, P. Krawiec (2014), "Conception of ID layer performance at the network level for Internet of Things", Springer: Personal and Ubiquitous Computing, Vol. 18, pp. 465-480.

[6]. Bharathi and Sukanesh (2012), "A PCA framework for detection of application layer DDoS attacks", Transactions on information science and applications, Vol. 12, No. 9, pp. 389 – 398.

[7]. Chen Y and K. Hwang (2006), "Collaborative detection and filtering of shrew DDoS attacks using spectral analysis", Journal of Parallel and Distributed Computing, Vol. 66, no. 9, pp. 1137-1151.

[8]. Chen, Min, Wan, Jiafu, Li, Fang (2012), "Machine-to-machine communications: Architecture, standards and applications", KSII Transactions on Internet & Information Systems, Vol. 6, No. 2, pp.480 – 497.

[9]. Choi J, C. Choi, B. Ko, D. Choi, and P. Kim, "Detecting web based DDoS attack using Map Reduce operations in cloud computing environment", Journal of Internet Service Information Security, Vol. 3, No. 3/4, pp. 28–37.

[10]. Dewan Md. Farid, Mohammad Zahidur Rahman (2010), "Anomaly Network Intrusion Detection Based on Improved Self Adaptive Bayesian Algorithm", Journal of Computers, Vol. 5, No. 1, January 2010.

[11]. Jerbi, M. (2009). "Towards efficient geographic routing in urban vehicular networks", Vehicular Technology, IEEE Transactions on, Vol. 58, No. 9, pp. 5048–5059.

[12]. Joo P. Amaral, Lus M. Oliveira, Joel J. P. C. Rodrigues, Guangjie Han, Lei Shu (2014), "Policy and Network-based Intrusion Detection System for IPv6-enabled Wireless Sensor Networks", IEEE Conference on Communications Software, Services and Multimedia Applications.

[13]. Juneja D and N. Arora (2010), "An ant based frameworks for preventing DDoS attack in wireless sensor networks", International Journal of Advancements in technology, Vol. 1, 2010, pp. 34-44.

[14]. Latifur Khan, Mamoun Awad, Bhavani Thuraisingham (2007), "A new intrusion detection system using support vector machines and hierarchical clustering", Journal of VLDB Journal, Vol.16, pp.507-521.

Lee, K., Kim, J., Kwon, K. H., Han, Y., and Kim, S. (2008), "DDoS attack pp. 1659–1665.