ANN Based Scheme to Predict Number of Zombies in a DDoS Attack

Brij Bhooshan Gupta[1,2], Ramesh Chand Joshi[2], and Manoj Misra[2]
(Corresponding author: Brij Bhooshan Gupta)
Department of Computer Science & Engineering, Graphic Era University, Dehradun, India[1]
566/6, Bell Road, Clement Town, Dehradun, Uttarakhand, 248002 India
Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, India[2]
(Email: gupta.brij@gmail.com)
(Received June 3, 2010; revised and accepted Aug. 7 & Nov. 28, 2010)

Abstract

A real time estimation of the number of zombies in DDoS attack scenario is helpful to suppress the effect of attack by choosing predicted number of most suspicious attack sources for either filtering or rate limiting. In this paper, ANN is employed to estimate number of zombies involved in a DDoS attack. The method does not depend on the frequency of attack and hence solves the problem of low detection precision and weak detection stability of ANN which occurs when used for low frequent attack estimation. The sample data used to train the feed forward neural networks is generated using NS-2 network simulator running on Linux platform. Various sizes of feed forward networks are compared for their estimation performance using MSE. The generalization capacity of the trained network is promising and the network is able to predict number of zombies involved in a DDoS attack with very less test error.

Keywords: DDoS attack, entropy, feed forward neural network, intrusion detection, zombies

1 Introduction

Denial of service (DoS) attacks and more particularly the distributed ones (DDoS) are one of the latest threat and pose a grave danger to users, organizations and infrastructures of the Internet. A DDoS attacker attempts to disrupt a target, in most cases a web server, by flooding it with illegitimate packets, usurping its bandwidth and overtaxing it to prevent legitimate inquiries from getting through [6]. Anomaly based DDoS detection systems construct profile of the traffic normally seen in the network, and identify anomalies whenever traffic deviate from normal profile beyond a threshold [7]. This extend of deviation is normally not utilized. Therefore, in this paper, this extend of deviation from detection threshold is used as an input to ANN [2, 3, 13] to predict number of zombies. A real time estimation of the number of zombies in DDoS scenario is helpful to suppress the effect of attack by choosing predicted number of most suspicious attack sources for either filtering or rate limiting. We have assumed that zombies have not spoofed header information of out going packets. Moore et al. [8] have already made a similar kind of attempt, in which they have used backscatter analysis to estimate number of spoofed addresses involved in DDoS attack. This is an offline analysis based on unsolicited responses.

Our objective is to find the relationship between number of zombies involved in a flooding DDoS attack and deviation in sample entropy. In order to predict number of zombies, feed forward neural network is used. Several authors have used ANN in anomaly based DDOS attack detection. In [10], ANN is used to classify a network while under attack. In this implementation, data extracted in a network probing phase is fed to a three layer feed forward neural network and it is trained to output 1 when there is attack and 0 when there is no attack. In [1], feed forward neural network is used to detect different DDOS attacks. Recently [12] have proposed an approach to enhance the detection capacity of ANN. They proposed the use of fuzzy clustering as a preprocessing to the training of ANN. In all the above approaches, ANN is trained using normal and attack traffic data and ANN decides the presence or absence of an attack. In our approach, ANN is used to decide the number of zombies used in a DDOS attack. The method does not depend on the frequency of attack and hence solves the problem of low detection precision and weak detection stability of ANN which occurs when used for low frequent attack estimation. To measure the performance of the proposed approach, we have calculated mean square error (MSE) and test error. Training and test data are generated using simulation. Internet type topologies used for simulation are generated using Transit-Stub model of GT-ITM topology generator [5]. NS-2 network simulator [9] on Linux platform is used as simulation test bed for launching DDoS attacks with var-

ied number of zombies and the data collected are used to train the neural network. In our simulation experiments, attack traffic rate is fixed to 25Mbps in total; therefore, mean attack rate per zombie is varied from 0.25Mbps to 2.5Mbps and total number of zombie machines range between 10 and 100 to generate attack traffic. Varies sizes of feed forward neural networks are compared for their estimation performance. The result obtained is very promising as we are able to predict number of zombies involved in DDoS attack effectively.

The remainder of the paper is organized as follows. Section 2 contains overview of artificial neural network (ANN). Intended analytical model and detection scheme are described in Section 3. Section 4 describes experimental setup and performance analysis in details. Section 5 contains simulation results and discussion. Finally, Section 6 concludes the paper.

2 Artificial Neural Network

An Artificial Neural Network (ANN) [2, 3, 13] is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true for ANNs as well. Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections of new situations of interest and answer "what if" questions. Other advantages include:

1) **Adaptive learning:** An ability to learn how to do tasks based on the data given for training or initial experience.

2) **Self-organization:** An ANN can create its own organization or representation of the information it receives during learning time.

3) **Real time operation:** ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.

4) **Fault tolerance via redundant information coding:** Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

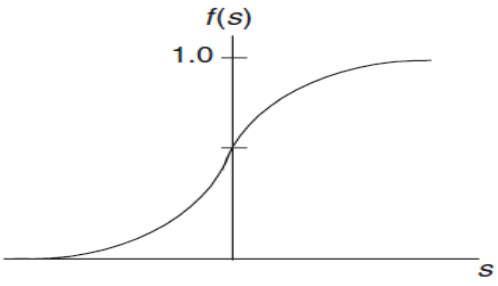

Figure 1: Sigmoid function

An ANN is a layered collection of small processing elements known as neurons and mathematically the output of a single neuron is given as

$$y_i = f_i(w_{ji}x_i + b_j)$$

Where f is the activation function, x is input and b is a bias and w is the weight for each input. The activation function determines the type of neuron and the application where the neuron is to be used. But the sigmoid activation function as shown in Figure 1, is famous for most neural networks and is given by

$$f_i(s) = \frac{1}{1+e^{-s_j}}$$

2.1 Network Architecture

Artificial neural networks are interconnections of individual neurons. There are various network architectures based on the type of connection. A most important type of network is the feed forward neural network shown in Figure 2. It is a three layer neural network: an input layer with three inputs, a hidden layer with four neurons and an output layer with two neurons. The name feed forward is given to this network because signal flows in forward manner without any feedback. This three layer network with enough number of hidden layer neurons and sigmoid activation function has the capacity to learn any nonlinear mapping.

The output y of the neural network is generated by:

$$y_k = f_{ko}(\sum_{j=1}^{N} W_{jk}^h f_k(\sum_{j=1}^{p} W_{ji}^I X_i + b_j) + b_0)$$

Where

- W_{jk}^h is connection weight from hidden layer to output
- W_{ji}^I is connection weight from input to hidden layer
- b_0 is bias of output
- b_j is bias of hidden layer
- f_k is activation function of hidden layer, and
- f_{ko} is activation function of output layer

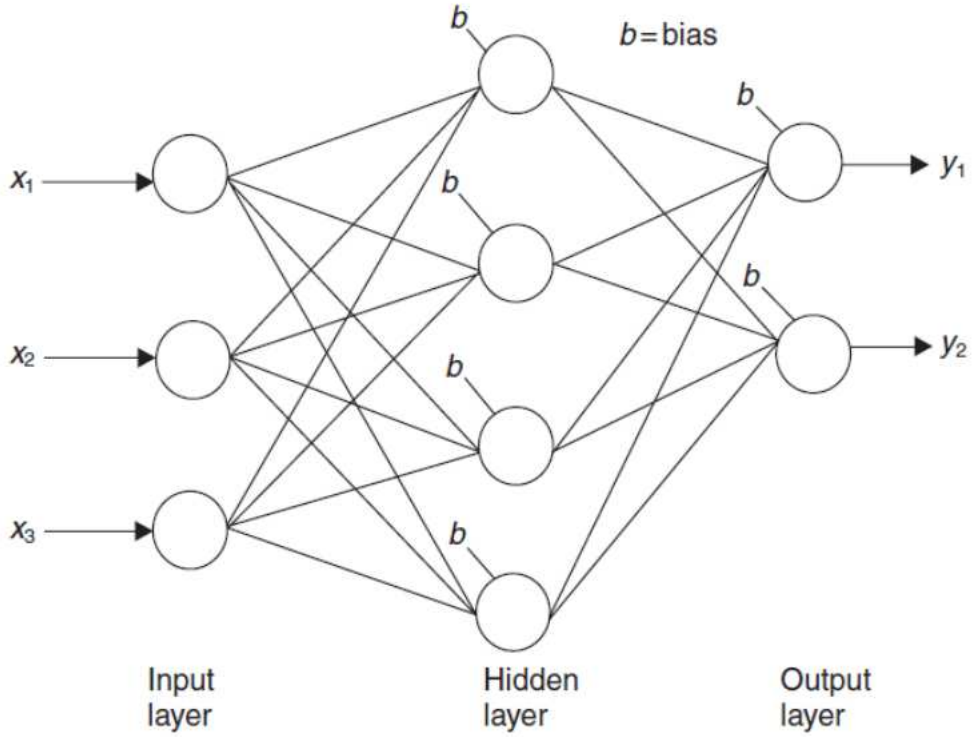

Figure 2: Fully connected, three layers feed forward network

2.2 Learning in Neural Networks

Learning in the context of neural networks is the process of adjusting the connection weights and biases such that for a given input a desired output is achieved. There are two basic training modes.

1) **Supervised learning** - This is a learning paradigm where the neural network is given samples of the input and desired output and the error between the desired output and the actual output of the neural network is used to adjust the connection weights. A famous algorithm of supervised learning is back propagation.

2) **Unsupervised learning** - This does not need any feedback for adjustment of the weights.

2.3 Back Propagation Algorithm

Back propagation is a famous algorithm used to train neural networks. It uses the gradient decent optimization method to train a network. In most cases the sum of squared error is used as objective function.

$$J = \frac{1}{M}\sum_{j=1}^{M}(d_j - y_j)^2$$

where d_j and y_j are the desired and actual network outputs.

Weight update algorithm is given by

$$\begin{aligned} W_{new} &= W_{old} + \Delta W \\ where \Delta W &= -\mu \frac{\vartheta J}{\vartheta w} \end{aligned}$$

2.4 Input and Output

In feed forward neural network, a relationship is developed between number of zombies Y (output) and observed deviation in sample entropy X (input). Here X is equal to $(H_c - H_n)$. Our proposed feed forward neural network based approach utilizes this deviation in sample entropy X to predict number of zombies.

3 Detection of Attacks

3.1 Analytical Model

This section describes an analytical model which is constructed to detect a wide range of flooding attacks. Detecting DDoS attacks involve first knowing normal profile of the system and then to find deviations from this normal profile. Whenever incoming traffic goes out of the normal profile, anomalous system behavior is identified.

Our approach detects flooding DDoS attacks by the constant monitoring of the propagation of abrupt traffic changes inside the ISP network.

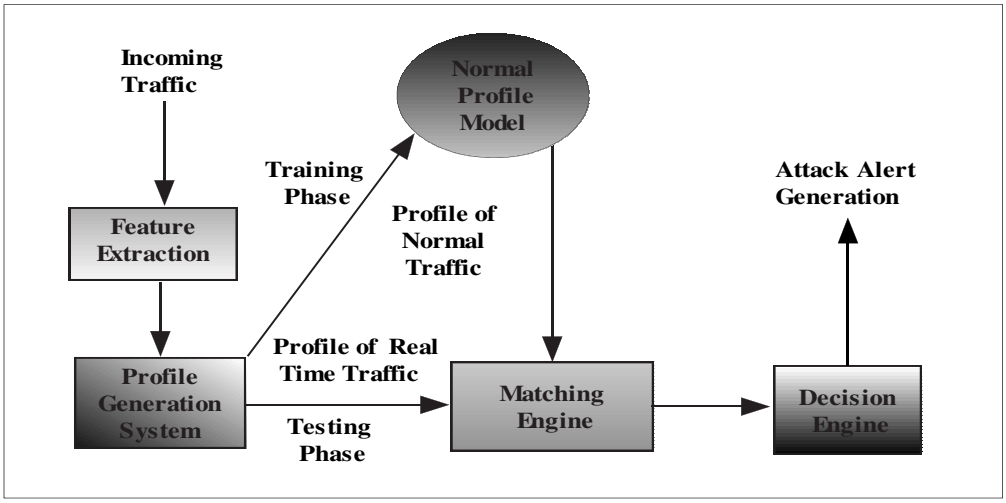

Figure 3: High-level block diagram of DDoS detection system

A high-level block diagram of DDoS detection system is given in Figure 3.

Let M and F be the random vectors compose of m measures m_1, m_2, ..., m_m used for attacks detection and n flows f_1, f_2, ..., f_n containing the incoming traffic to the server, respectively: $M = (m_1, m_2, \ldots, m_m)$, $F = (f_1, f_2, \ldots, f_n)$, where $f_i = (m_1^i, m_2^i, \ldots, m_m^i)$ is i^{th} flow. Consider a random process $\{m_j^i(t),\ t = w\delta,\ \omega \in N\}$, where δ is a constant time interval, N is the set of positive integers, and for each t, $m_j^i(t)$ is a random variables. $1 \leq \omega \leq l$, l is the number of time intervals. Here $m_j^i(t)$ represents the value of m_j in flow i in $\{t-\delta, t\}$ time duration. These relations can be written in matrix form as follows:

$$Z(t) = \begin{pmatrix} m_1^1(t) & m_1^i(t)\ldots & m_1^n \\ m_j^1(t) & m_j^i(t)\ldots & m_j^n \\ \vdots & \vdots & \vdots \\ m_m^1(t) & m_m^i(t)\ldots & m_m^n(t) \end{pmatrix}$$

where, $Z(t)$ contains values of different measures used in $\{t-\delta,\ t\}$. $m_j(t)$ represent total value of j^{th} measure during $\{t-\delta, t\}$ time. $m_j(t)$ can be calculated as follows:

$$m_j(t) = m_j^1(t) + m_j^2(t) + \ldots + m_j^i(t) + \ldots + m_j^n(t)$$

where $1 \leq i \leq n$, n is the number of flows. $1 \leq j \leq m$, m is the number of measures. Normal traffic value of j^{th} measures can be calculated using following equation:

$$m_j^*(t) = \frac{1}{l}\sum_{\omega=1}^{l} m_j(t)$$

where $t = \omega\delta$. Vector A can be used to represent normal traffic measures value: $A = (m_1^*(t), m_j^*(t), \ldots, m_m^*(t))$. To detect the attack, the value of j^{th} traffic measure $m_j(t)$ is calculated in time window δ continuously; whenever there is appreciable deviation from $m_j^*(t)$, anomalous behaviors could be determined. Depending on the measures selected to use or network conditions, following events are defined to determine anomalous system behaviors:

$$\begin{aligned} m_j(t) - m_j^*(t) &> \xi_j^{upper} \\ m_j(t) - m_j^*(t) &< \xi_j^{lower} \end{aligned}$$

where ξ_j^{upper} and ξ_j^{lower} represent value of upper and lower bound of the threshold for j^{th} measure, respectively. ξ_j^{upper} and ξ_j^{lower} can be set as follows:

$$\begin{aligned} \xi_j^{upper} &= \gamma_j^{upper} * \sigma_j \\ \xi_j^{lower} &= \gamma_j^{lower} * \sigma_j \end{aligned}$$

where σ_i represent value of standard deviation for j^{th} measure. r_j^{upper} and r_j^{lower} represent value of tolerance factor to calculate upper and lower bound of the threshold for j^{th} measure, respectively. Effectiveness of an anomaly based detection system highly depends on accuracy of threshold value settings. Inaccurate threshold values cause a large number of false positives and false negatives. Therefore, various simulations are performed using different value of tolerance factors. The choice of tolerance factors varies for different network conditions. Values of tolerance factors also depend on the composition of the normal traffic and the desired degree of the ability to control a DDoS attack. Then, trade-off between detection and false positive rate provides guidelines for selecting value of tolerance factor r_j for j^{th} traffic measure for a particular simulation environment.

3.2 Entropy Based DDoS Detection

Here, we will discuss propose detection system that is part of access router or can belong to separate unit that interact with access router to detect attack traffic. It makes

use of analytical model given in the previous section. Entropy based DDoS scheme [11] is used to construct profile of the traffic normally seen in the network, and identify anomalies whenever traffic goes out of profile. A metric that captures the degree of dispersal or concentration of a distribution is sample entropy. Sample entropy $H(X)$ is

$$H(X) = -\sum_{i=1}^{N} p_i \log_2(p_i)$$

where p_i is n_i/S. Here n_i represent total number of bytes arrivals for a flow i in $\{t-\delta,\ t\}$ and $S = \sum_{i=1}^{N} n_i, i = 1, 2, \ldots, N$. The value of sample entropy lies in the range $0 - \log_2 N$.

To detect the attack, the value of $H_c(X)$ is calculated in time window δ continuously; whenever there is appreciable deviation from $X_n(X)$, various types of DDoS attacks are detected. $H_c(X)$, $X_n(X)$ and gives Entropy at the time of detection of attack and Entropy value for normal profile respectively.

4 Experiment Setup and Performance Analysis

In this section, we evaluate our proposed scheme using simulations. The simulations are carried out using NS2 network simulator. We show that false positives and false negatives triggered by our scheme are very less. This implies that profiles built are reasonably stable and are able to predict number of zombies correctly.

4.1 Simulation Environment

Real-world Internet type topologies generated using Transit-Stub model of GT-ITM topology generator are used to test our proposed scheme, where transit domains are treated as different Internet Service Provider (ISP) networks i.e. Autonomous Systems (AS). For simulations, we use ISP level topology, which contains four transit domains with each domain containing twelve transit nodes i.e. transit routers. All the four transit domains have two peer links at transit nodes with adjacent transit domains. Remaining ten transit nodes are connected to ten stub domain, one stub domain per transit node. Stub domains are used to connect transit domains with customer domains, as each stub domain contains a customer domain with ten legitimate client machines. So total of four hundred legitimate client machines are used to generate background traffic.

Total zombie machines range between 10 and 100 to generate attack traffic. Transit domain four contains the server machine to be attacked by zombie machines. A short scale simulation topology is shown in Figure 4.

Currently, the majority of the DDoS attacks are TCP flooding, so we will consider detection of a wide range of TCP flooding attacks in this section. The legitimate clients are TCP agents that request files of size 1 Mbps with request inter-arrival times drawn from a Poisson distribution. The attackers are modeled by UDP agents. A UDP connection is used instead of a TCP one because in a practical attack flow, the attacker would normally never follow the basic rules of TCP, i.e. waiting for ACK packets before the next window of outstanding packets can be sent, etc. The attack traffic rate is fixed to 25 Mbps in total; therefore, mean attack rate per zombie is varied from 0.25 Mbps to 2.5 Mbps. In our experiments, the monitoring time window was set to 200 ms, as the typical domestic Internet RTT is around 100 ms and the average global Internet RTT is 140 ms [4]. Total false positive alarms are minimum with high detection rate using this value of monitoring window. The simulations are repeated and different attack scenarios are compared by varying total number of zombie machines and at fixed attack strengths.

5 Results and Discussion

5.1 Training Data Generation

Neural network has to be trained by giving sample inputs and corresponding output values and a training algorithm will adjust the connection weight and bias values until a minimum error or other stopping criteria is reached. The training data has to be taken carefully to consider the complete input range. Normalization and other preprocessing of the data improve the training performance.

In our paper, in order to predict number of zombies ($\hat{Y}$) from deviation ($H_c - H_n$) in entropy value, training data samples are generated using simulation experiments in NS-2 network simulator. Simulation experiments are done at the same attack strength 25 Mbps in total and varying number of zombies from 10-100 with increment of 5 zombies i.e. mean attack rate per zombie from 0.25 Mbps-2.5 Mbps. Figure 5 shows entropy variation with 10-100 numbers of zombies at same attack strength in total of 25Mbps. The data obtained is divided into two parts, 78.95% of the data values are used for training and these are shown in Table 1. The remaining data values which are selected randomly are used for testing and are shown in Table 2.

5.2 Network Training

For the prediction of the number of zombies in a DDOS attack, three feed forward neural networks have been tested. The feed forward networks used have different sizes. The size of a network refers to the number of layers and the number of neurons in each layer. There is no direct method of deciding the size of a network for a given problem and one has to use experience or trial error method. In general, when a network is of large size, the complexity of the function that it can approximate will also increase. But as the network size increase, both training time and

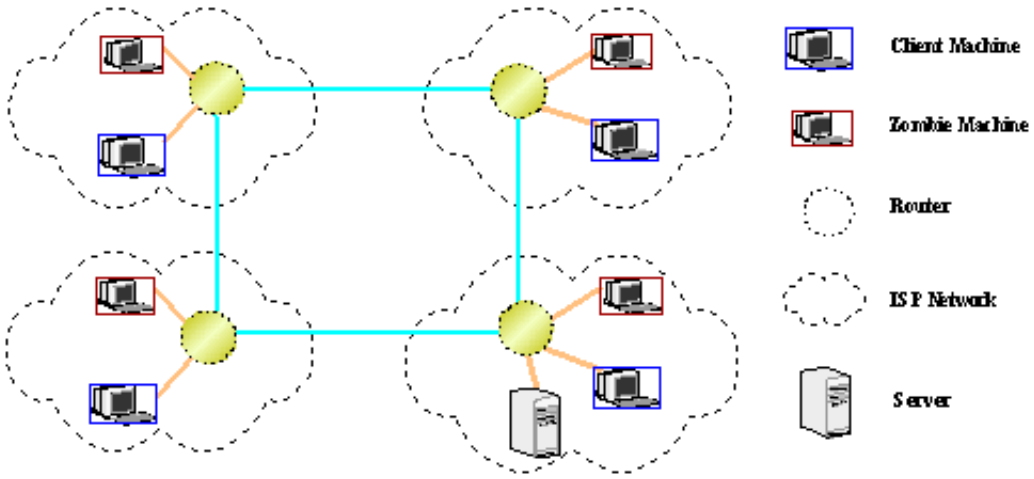

Figure 4: A short scale simulation topology

Table 1: Training data-deviation in entropy with actual number of zombies

Actual Number of Zombies (Y)	**Deviation in Entropy** $(X)(H_c - H_n)$
10	0.045
15	0.046
25	0.050
30	0.068
35	0.087
40	0.099
45	0.111
55	0.130
60	0.139
65	0.148
75	0.163
80	0.170
85	0.176
90	0.182
100	0.192

Table 2: Testing data-deviation in entropy with actual number of zombies

Actual Number of Zombies (Y)	**Deviation in Entropy** $(X)(H_c - H_n)$
20	0.048
50	0.121
70	0.157
95	0.189

Table 3: Training results of various feed forward networks

Network used	Network size	Number of Epochs	MSE in training
2 layer network	5-1	400	6.86
	10-1	400	0.36
	15-1	400	0.0025

its implementation cost increase and hence optimum network size has to be selected for a given problem. For the current problem, two layer feed forward networks with 5, 10 and 15 neurons are selected. The training algorithm used is the Levenberg-Marquardt back propagation algorithm of MATLAB's neural network toolbox. The training results are given in Table 3. Figure 6 shows the training performance of this two layer network.

5.3 Network Testing

Table 4 shows the result of the testing of the networks using the test data values given in Table 2.

From the result of Table 3, we can see that the MSE in training decreases linearly as the network size increase. This is as expected. But in table 4, we can see that in spite of the smaller MSE in training and the increase in network size, the test result for the feed forward network having 15 hidden layer neurons is greater than the networks having 5 and 10 neurons. One reason for this is, for a good network performance, the ration of number of tunable parameters to that of training data size has to be very small and here network size has increased but training data size is the same. For the last network, the number of tunable parameters is 31 and ration is 1.63. And because of this, over fitting has occurred and the generalization performance of the last network is poor though it has good training performance. The training performance is measured using the mean square error (MSE). MSE is the difference between the target and the neural network's

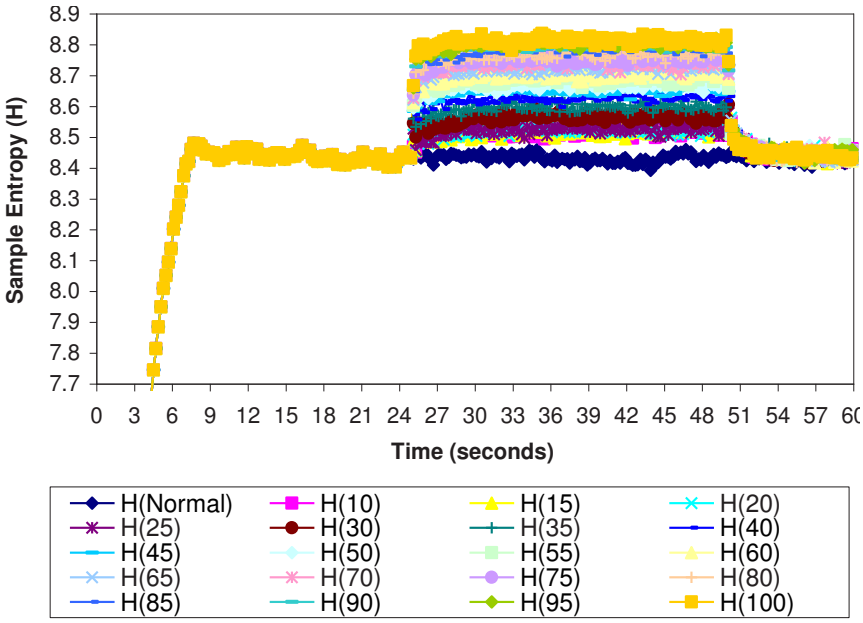

Figure 5: Entropy variation with varied number of zombies

Table 4: Test results of various feed forward networks

Network used	Network size	MSE in training
2 layer network	5-1	2.91
	10-1	2.59
	15-1	3.14

actual output. So, the best MSE is the closest to 0. If MSE is 0, this indicates neural network's output is equal to the target which is the best situation.

Number of zombies of the individual networks can be compared with actual number of zombies for each test data values of table 2 and the results are given in Figure 7, 8, and 9.

To represent false positive i.e. falsely predicted normal clients as zombies and false negative i.e. zombies are identified as normal client, we plot test error. Positive cycle of test error curve represents false positive, while negative cycle represents false negative. The test error of the individual networks is calculated for each test data values of table 2 and the result is given in Table 5, 6, and 7. The results show that the prediction capacity of the neural networks is very close to the actual number of zombies and hence neural networks have the potential to be used to predict number of zombies in real DDOS attack scenarios.

6 Conclusion and Future Work

The potential of feed forward neural network for predicting number of zombies involved in a flooding DDoS attack is investigated. The deviation $(H_c(X) - X_n(X))$ in sample entropy is used as an input and MSE is used as the performance measure. Two layer feed forward networks of size 5, 10 and 15 have shown maximum mean square error (MSE) of 2.91, 2.59 and 3.14 respectively in predicting the number of zombies. Therefore, total number of predicted zombies using feed forward neural network is very close to actual number of zombies. However, simulation results are promising as we are able to predict number of

Table 5: Summary of test error for feed forward neural network for network size 5-1

(X) Entropy Variation	(Y) Number of Zombies	test error
0.048	20	-1.79
0.121	50	1.31
0.157	70	2.59
0.189	95	-0.07

Table 6: Summary of test error for feed forward neural network for network size 10-1

(X) Entropy Variation	(Y) Number of Zombies	test error
0.048	20	1.33
0.121	50	2.88
0.157	70	0.40
0.189	95	0.36

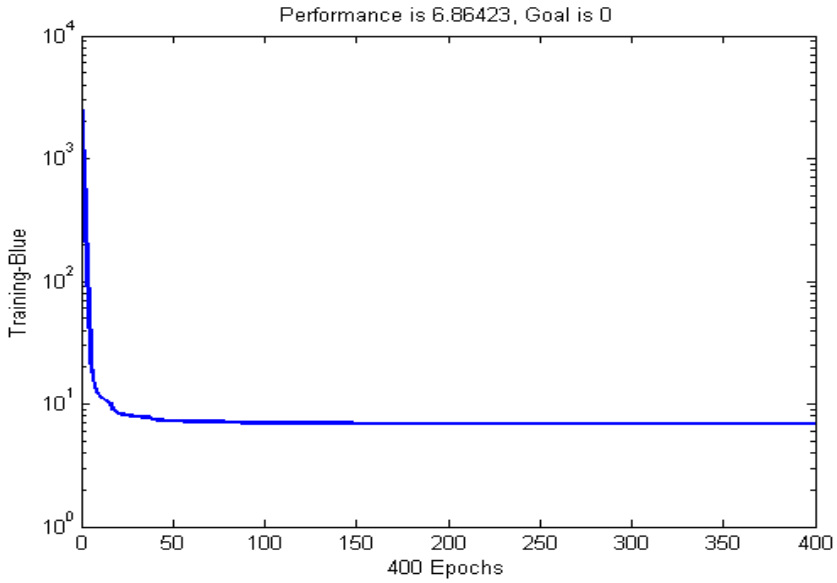

Figure 6: Training performance of two layers feed forward network (5-1)

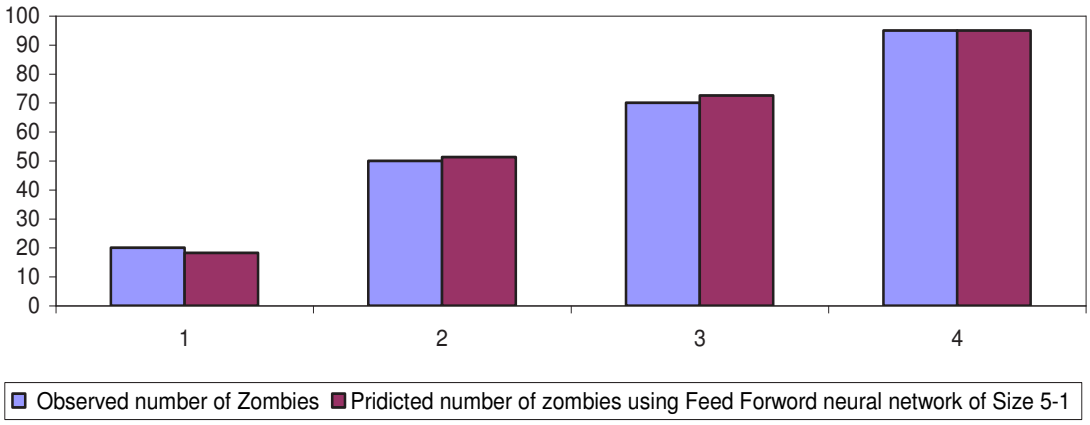

Figure 7: Comparison between actual number of zombies and predicted number of zombies using feed forward neural network of size 5-1

Table 7: Summary of test error for feed forward neural network for network size 15-1

(X) Entropy Variation	(Y) Number of Zombies	Residual error
0.048	20	1.88
0.121	50	2.20
0.157	70	0.85
0.189	95	1.86

zombies efficiently, experimental study using a real time test bed can strongly validate our claim.

Acknowledgements

The authors gratefully acknowledge the financial support of the Ministry of Human Resource Development (MHRD), Government of India for partial work reported in the paper.

References

[1] I. Ahmad, A. B. Abdullah, and A. S. Alghamdi, "Application of artificial neural network in detection of dos attacks," *Proceedings of International Conference on Security of Information and Networks (SIN 2009)*, pp. 229-234, North Cyprus, Turkey, Oct. 6-10, 2009.

[2] R. Burns and S. Burns, *Advanced Control Engineering*, Butterworth Heinemann, 2001.

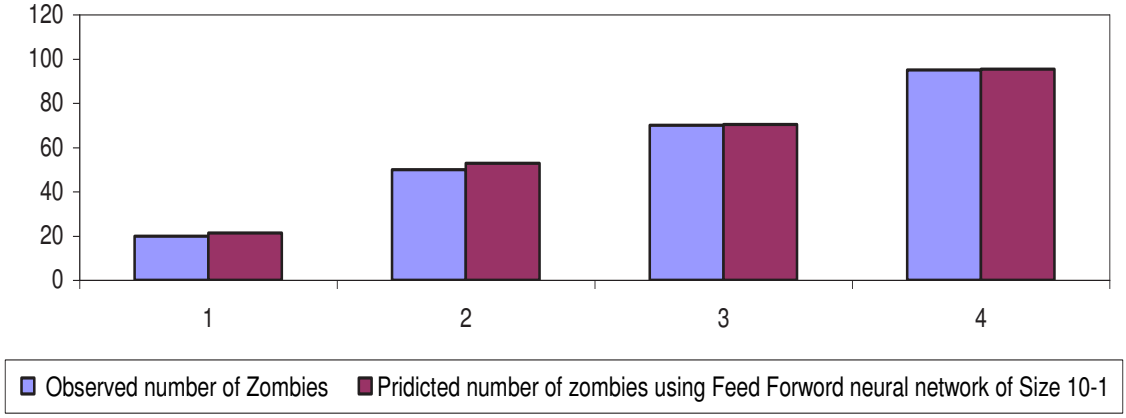

Figure 8: Comparison between actual number of zombies and predicted number of zombies using Feed forward neural network of size 10-1

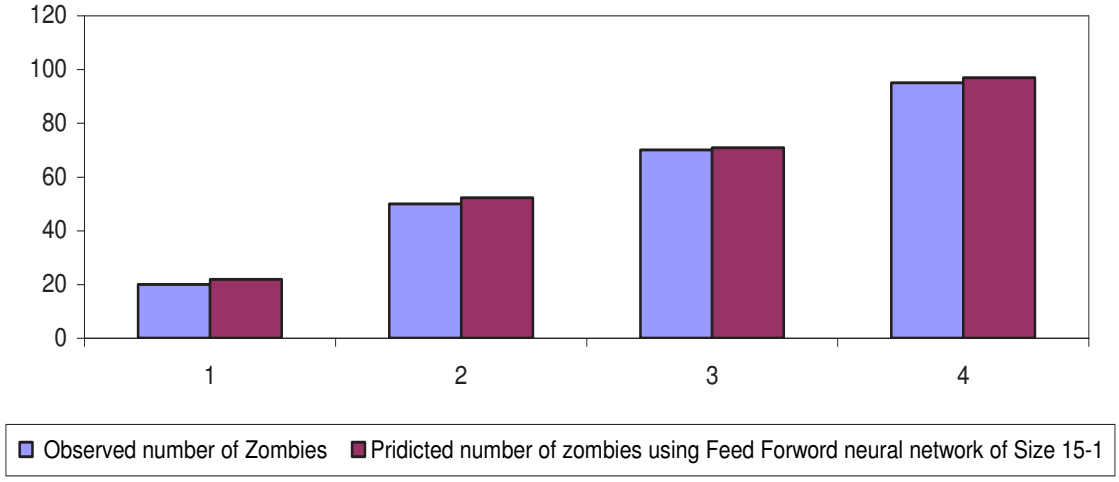

Figure 9: Comparison between actual number of zombies and predicted number of zombies using Feed forward neural network of size 15-1

[3] U. E. Dayhoff and J. M. DeLeo, "Artificial neural networks," *American Cancer Society,* vol. 91, no. S8, pp. 1615-1635, 2001.

[4] B. Gibson, *TCP Limitations on File Transfer Performance Hamper the Global Internet,* White paper, Sep. 2006. (http://www.niwotnetworks.com/gbx/ TCPLimitsFastFileTransfer.htm)

[5] GT-ITM, *Traffic Generator Documentation and Tool*, 2010. (http://www.cc.gatech.edu/fac/ EllenLegura/graphs.html)

[6] B. B. Gupta, M. Misra, and R. C. Joshi, "An ISP level solution to combat ddos attacks using combined statistical based approach," *International Journal of Information Assurance and Security (JIAS)*, vol. 3, no. 2, pp. 102-110, 2008.

[7] B. B. Gupta, R. C. Joshi, and M. Misra, "Defending against distributed denial of service attacks: Issues and challenges," *Information Security Journal: A Global Perspective*, vol. 18, no. 5, pp. 224-247, 2009.

[8] D. Moore, C. Shannon, D. J. Brown, G. Voelker, and S. Savage, "Inferring Internet denial-of-service activity," *ACM Transactions on Computer Systems*, vol. 24, no. 2, pp. 115-139, 2006.

[9] NS Documentation, 2010. (http://www.isi.edu/ nsnam/ns)

[10] S. Seufert and D. O'Brien, "Machine learning for automatic defense against distributed denial of service attacks," *Proceedings of IEEE International Conference on Communications, ICC'07*, pp. 1217-1222, June 24-28, 2007.

[11] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communication Review*, vol. 5, pp. 3-55, 2001.

[12] G. Wang, J. Hao, J. Ma, and L. Huang, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering," *Expert Systems with Applications*, vol. 37, pp. 6225V6232, 2010.

[13] B. Yegnanarayana, *Artificial Neural Networks*, Prentice-Hall, New Delhi, 1999.

B. B. Gupta received the bachelor's degree in Information Technology in 2005 from Rajasthan University, India. He is currently a PhD student in the Department of Electronics and Computer Engineering at Indian Institute of Technology, Roorkee, India. He has published over 20 research papers at international journals/conferences. He has been awarded Canadian Commonwealth Scholarship (CCSP) and Government of Canada Award (GCA), 2009. His research interests include Intrusion detection, Network security, Cryptography, Data mining and mobile computing.

Manoj Misra received the bachelor's degree in Electrical Engineering in 1983 from HBTI Kanpur, India. He received his master's and PhD degree in Computer Engineering in 1986 and 1997 from University of Roorkee, India and Newcastle upon Tyne, UK, respectively. He is currently a Professor at Indian Institute of Technology Roorkee. He has guided several PhD theses, M.E./M.Tech. Dissertations and completed various projects. His areas of interest include Mobile computing, Distributed computing and Performance Evaluation.

R. C. Joshi received the bachelor's degree in Electrical Engineering from Allahabad University, India in 1967. He received his master's and PhD degree in Electronics and Computer Engineering from University of Roorkee, India in 1970 and 1980, respectively. He is currently a Professor at Indian Institute of Technology Roorkee, India. He has a vast teaching experience exceeding 38 years at graduate and postgraduate levels at IIT Roorkee. He has guided over 150 M.Tech and 25 PhD dissertations. He has published over 100 research papers at national and international journals and presented many in Europe, USA and Australia. He has been awarded Gold Medal by Institute of Engineers for best paper. He has chaired many national and international conferences and workshops. Presently, he is actively involved in research in the field of Database management system, Data mining, Bioinformatics, Information security, Reconfigurable systems and Mobile computing.