DDOS Attack Detection Based on an Ensemble of Neural Classifier

Madhav Kale and D.M.Choudhari††,

Department of Computer Science and Engineering
KLS Gogte Institute of Technology,
Belgaum-59008, INDIA

Abstract:
This Paper Reviews the Detection of DDoS Attck.This DDoS attacks could be detected using the existing machine learning techniques such as neural classifiers. These classifiers lack generalization capabilities which result in less performance leading to high false positives. This paper evaluates the performance of a comprehensive set of machine learning algorithms for selecting the base classifier using the publicly available KDD Cup dataset. Based on the outcome of the experiments, Resilient Back Propagation (RBP) was chosen as base classifier for our research. The improvement in performance of the RBP classifier is the focus of this paper. Our proposed classification algorithm, RBPBoost, is achieved by combining ensemble of classifier outputs and Neyman Pearson cost minimization strategy, for final classification decision. Publicly available datasets such as KDD Cup, DARPA 1999, DARPA 2000, and CONFICKER were used for the simulation experiments. RBPBoost was trained and tested with DARPA, CONFICKER, and our own lab datasets. Detection accuracy and Cost per sample were the two metrics evaluated to analyze the performance of the RBPBoost classification algorithm. From the simulation results, it is evident that RBPBoost algorithm achieves high detection accuracy (99.4%) with fewer false alarms and outperforms the existing ensemble algorithms. RBPBoost algorithm outperforms the existing algorithms with maximum gain of 6.6% and minimum gain of 0.8%

.Key words:
DDoS Attack, Feed forward Neural Network, KDDCup dataset, Probabilistic Neural Network, Intrusion Detection, Machine Learning Techniques

1. Introduction

Intrusion Detection System is a tool that is being used by many Network Security Officers in order to protect Organization from attacks from different sources. Intrusion Detection Systems (IDSs) have emerged in the computer security area because of the difficulty of ensuring that an information system will be free of security flaws. Computer systems suffer from security vulnerabilities regardless of their purpose, manufacturer or origin, and it is technically difficult as well as economically costly, in terms of both building and maintaining such a system, to ensure that computer systems and networks are not susceptible to attacks. The damages of attacks include loss of intellectual property and liability for compromised customer data hence there is a need for IDS. The block diagram representation of the architecture of Intrusion Detection System is shown in figure 1.

The data collected from the internet is fed into the preprocessing unit where the raw data is formatted to make it compatible with the intrusion detection system under consideration. Subsequently data is classified as normal or attack. Normal data is allowed to pass through while on the other hand attack data is classified to the type of attack, saved in the database (Db) and an alert is raised. There have been lots of traditional rule-based works to the design of IDS [1]. Recently, an increasing amount of research has been conducted on applying artificial neural networks to detect intrusions in the network. It has been shown that network traffic can be efficiently modeled using artificial neural networks. This method proves to be advantageous as it goes through rigorous training, validation and level 1 testing phases before being actually fed to the network for detecting attacks. An artificial neural network consists of a collection of processing elements that are highly interconnected. Given a set of inputs and a set of desired outputs, the transformation from input to output is determined by the weights associated with the interconnections among processing elements. By modifying these interconnections, the network is able to adapt to the desired outputs. IDS may be divided into two categories: misuse detection and anomaly detection

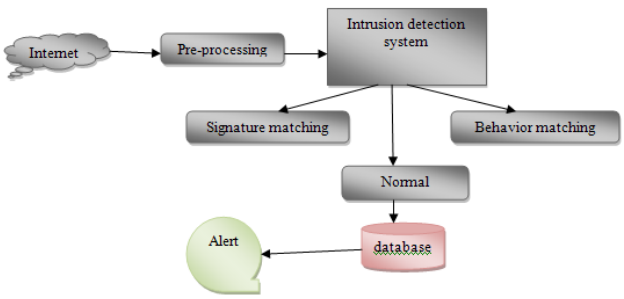

Fig 1: architecture of intrusion detection system

Manuscript received July 5, 2014
Manuscript revised July 20, 2014

We have designed an anomaly based detection system which carefully analyses the network traffic for known attacks for which the system has been trained previously and alerts the system administrators for abnormal packets. For this purpose we have used KDD cup '99 dataset to test the feasibility, efficiency and effectiveness of our model. In our system it becomes necessary to group network traffic together to present it to the neural network. We have used all 41 attributes of the KDD cup '99 dataset to design the IDS. Previous papers have used 30 attributes using neural network [2]. To accomplish this we have used neural network pattern recognition back propagation algorithm which has been shown to be effective in novelty detection [3]. After the introduction in section I, we describe related works in section II. The data set used for simulation is discussed in section III. Section IV describes our IDS architecture. Simulation environment and experimental results are provided in section V. The paper is concluded in section VI with some highlights on future works.

Ease of Use

A distributed denial of service (DDoS) attack [1, 6] is a large-scale, coordinated attack on the availability of services of a victim system or network resources, launched indirectly through many compromised computers on the Internet. The first well-documented DDoS attack appears to have occurred in August 1999, when a DDoS tool called Trinoo was deployed in at least 227 systems, to flood a single University of Minnesota computer, which was knocked down for more than two days1. The first large scale DDoS attack took place on February 20001. On February 7, Yahoo! was the victim of a DDoS attack during which its Internet portal was inaccessible for three hours. On February 8, Amazon, Buy.com, CNN and eBay were all hit by DDoS attacks that caused them to either stop functioning completely or slowed them down significantly

Our proposal is to make intelligent message discard Decisions based on Neural Networks to result in fewer false alarms.

The contributions of this paper include the following:

- Generic architecture of DDoS attack detection and response system for collaborative environment.
- Implementation of RBPBoost algorithm for the classification of network traffic.
- A classification accuracy of dataset
- 55.2% when training and testing of KDD dataset.
- 55.5% when training and testing on the lab dataset.

2. Related Works:

2.1 DDoS attack

DDoS attack is broadly classified into bandwidth depletion and resource depletion attack. In bandwidth depletion attack, attackers flood the victim with large traffic that prevents the legitimate traffic and amplify the attack by sending messages to broadcast IP address. In resource depletion attack, attackers attempt to tie up the critical resources (memory and processor) making the victim unable to process the service. A structural approach for DDoS attack classification is proposed in [27]. The detailed analysis on DDoS attacks and available attack tools [10] show that the DDoS attack has the following characteristics:

- Source and Destination IP address and port numbers of the
- Packets are spoofed and randomly generated.
- Window size, sequence number, and packet length are fixe during the attack.
- Flags in the TCP and UDP protocols are manipulated.
- Roundtrip time is measured from the server response.
- Routing table of a host or gateway is changed.
- DNS transaction IDs (reply packet) are flooded.
- HTTP requests are flooded through port 80.

2.2 Real time feature extraction

Features are statistical characteristics derived from the collected dataset. Selection of real time feature set plays a vital role in online traffic classification. More number of features leads to better accuracy. But, computation of more number of features in real time causes more overhead and time consuming. 248 features are given and 1 feature is used to describe the class (normal or attack). Computation of all the 248 features [13] took approximately two days on a dedicated System Area Network. Out of 248 features, some features such as maximum interpacket arrival time cannot be calculated until the entire flow is completed. Moreover, features based on Fast Fourier Transform values need better signal processing methods to reduce the computation time. So, less number of appropriate statistical features is to be selected for better pattern classification.

Feature extraction [36] is classified into two stages:

1. Feature Construction
2. Feature Selection.

Constructing the features is either integrated into the modeling process or into the preprocessing stage which includes standardization, normalization, etc. Feature Selection is divided into Filter methods and wrapper methods. In filter methods, selection is based on distance and information measures in the feature space. In wrapper methods, selection is based on classifier accuracy. Three statistical features are used in [25]. Nine features are used in [35]. Flow based feature selection has been shown to block legitimate traffic in [35]. Flow based selection gives summary of metadata. By blocking the IP address and port, flow based selection does not permit the legitimate requests. Hence, instead of flow based solution, packet based solution has been used in this paper.

2.3 Machine learning methods

Machine learning is mostly focused on finding relationships in data and analyzing the process for extracting such relations Machine learning paradigms are classified as Supervised Learning (SL), Unsupervised Learning (UL), and Reinforcement Learning (RL). In SL, the algorithm attempts to learn some function with given input vector and actual output. In UL, the algorithm attempts to learn only with given input vector by identifying relationships among data. In RL, the algorithm learns with a single bit of information which indicates to the neuron whether the output is good or bad. Though many evolutionary algorithms exist, neural network algorithms provide a promising alternative in classifying the DDoS attack patterns based on statistical features. Because of its generalization Capability, neural networks are able to work with imprecise and incomplete data. Further, these machine learning techniques can also recognize the patterns not presented during a training phase. Several ML algorithms [25, 35] have been proposed for DDoS attack detection. Most of the ML algorithms applied to DDoS attack detection have not considered minimizing the cost of the errors. These errors lead to more false alarms.

2.4 Ensemble of classifiers – motivation

Single classifier makes error on different training samples. So, by creating an ensemble of classifiers and combining their outputs, the total error can be reduced and the detection accuracy can be increased. There are two main components in all ensemble systems, viz., a strategy to build an ensemble that is as diverse as possible and the combination of outputs of classifier for the accurate classification decisions. Decision boundaries of each classifier have to be uniquely different from others. To achieve this diversity, ensemble of classifiers can be constructed by manipulating training data, feature sets, and injecting randomness. For the construction of the ensemble, the entire dataset is divided into subsets and each classifier is trained with each subset. In order to construct the ensemble by manipulating input feature sets, it is divided into smaller feature subsets and each classifier is trained with the same dataset. Another method to construct the ensemble is by randomly initializing the parameters such as weights, etc., and training with different parameter values at different times. As the number of features selected for training in this paper is less, the ensemble construction by feature set is not suitable. The advantage of constructing an ensemble by manipulating training data is that the generated hypothesis performs fairly well even when there are only small changes in traffic data. So, ensemble construction by manipulating training data was chosen, as it would correctly detect the deviations,

.Classifier combination is divided into two categories:

- Classifier selection, where each classifier is trained to become an expert in some local area of the total feature space.
- Classifier fusion, where all classifiers are trained over the same feature space.

Classifier outputs can be combined by methods such as Majority Voting, Weighted Majority Voting (WMV), etc. In this paper, popular ensemble methods such as Bagging [16], Boosting, and AdaBoost [31] are compared with our proposal, RBPBoost. Our algorithm differs from existing algorithms in two ways, viz., achieving diversity of the classifiers and combining the classifier outputs through WMV and Weighted Product Rule (WPR).

3. Proposed System Design

The system is to develop an Intrusion detection system based on learning technique. Firstly known classes of intrusion like DDOS, Perl attack, Neptune attack signature is formed from standard KDD dataset. This has several string values which are not understood by the classifier. Therefore these values are converted to suitable numbers based on their properties. Database is partitioned into two parts: Training and Testing. Testing involves giving one row from the dataset as input. System classifies the row as Normal or Abnormal.

The same concept is then adopted in a real time environment to detect anomaly in internet access from college data. Router log is used to extract the features. As these features are not reclassified, we use a regression technique rather than classification to find the similarity with any data of earlier dates. Baseon protocol used, we then classify the data as normal or abnormal.

Multiple alert of the same signature leads to misleading inferences for intrusion database. Therefore system should be able to detect and store only those intrusions that are relevant for future detection and those that are

significantly independent signature. Generally such a system work offline where firstly all the intrusions are marked as they appear and then the aggregation system aggregates the data. To validate this concept we use a Mesh network based simulation where we detect intrusion based on BER, Delay and energy consumption

The proposed system design architecture consists of the four main modules that are:

A: data collection module:
B: preprocessing.
C: Classification
D: Response

A receiver process running in promiscuous mode captures all incoming packets and stores in data storage server. The data is stored as set of traffic flows, with each instance being described by a set of features. Each instance is expressed in vector space model (A).

Preprocessing refers to the process of extracting information about packet connections from data and construction of new statistical features. The preprocessing steps are explained as follows:

1. Let 'x' be the input vector of dimension 'n',

Such that x = [x1, x2, x3, . . .xn]. The variables xi of the input vector is the original features.

2. Let 'tx' be a vector of transformed features of dimension 'tn

The statically characteristics features are used to find the statistical properties such as standard deviation and variance.. These values are used as inputs for machine learning algorithms (B).

In this module, Dataset of particular class is split into subsets. Each subset is trained with Ensemble of classifiers and results are combined by WMV [7]. TK is the total number of classifiers chosen using cross-validation. Cross-validation is a popular method of manipulating training data to subdivide the training data into 'k' disjoint subsets and to reconstruct training sets by leaving out some of the subsets. Results of each classification system are further combined by WPR [7]. The efficiency of classification of the classifier is significant in the decision making process. Hence, it is measured by a parameter Q-statistic . For effective decision, the Q-statistic should be zero. The training time depends on the number of times the classifier needs training which in turn depends on the mean square error between iterations reaching global minimum. The training is speeded up by removing the overlapping data and retaining only the training samples adjacent to the decision boundary. This method again consists of training and classifier is the sub stages (C).

Detection system deployed in each site maintains a hash table and updates IP address and port number (attack signature) of the suspicious blacklist nodes. When a site receives the attack signature, it checks if it exists in its hash table. If present, it means that the system is already alerted. If not, attack signature is added to the infected list. The updated attack signature is sent to all collaborating nodes, to prevent any damage that may be caused to the available services (D)

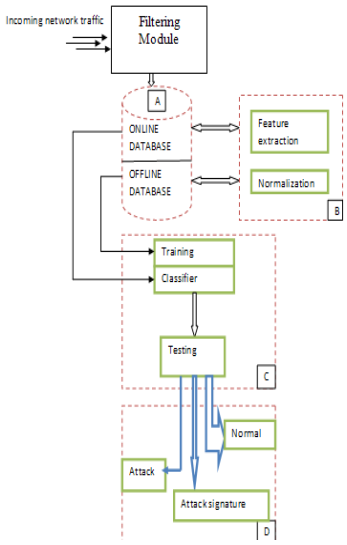

Fig 2: Architecture of DDoS attack detection

4. Proposed System Algorithm

The classification of the preprocessed data is carried out using RBPBoost algorithm. The block diagram shows that how RBPBoost algorithm uses the KDDCup99 dataset. This dataset is divided into two subset dataset and each subsets dataset is tested with this algorithm. Each subset is trained with ensemble of classifiers and results are combined by WMV [13]. TK is the total number of classifiers chosen using cross-validation. Cross-validation is a popular method of manipulating training data to subdivide the training data into 'k' disjoint subsets and to reconstruct training sets by leaving out some of the subsets. Results of each classification system are further combined by WPR [13]. The efficiency of classification of the classifier is significant in the decision making process.

The training of dataset is carried out by using feed forward neural network. But this neural network does not provide good detection accuracy. So in order to increase the detection accuracy we are used the RBP neural an ensemble of classifiers is trained for each individual data subset and the results are combined. A new classifier is added at each iteration. In our algorithm as given in Figure 3.Two classes (Normal and DDoS attack traffic) are considered.

The inputs to the algorithm are as follows:

- Training data comprised of ‘n’ instances with correct output labels.
- Resilient Back Propagation algorithm (RBP) as supervised base classifier.
- Number of classifier network

This dataset trained using artificial neural network and then tested with the RBP neural network to find its detection accuracy.

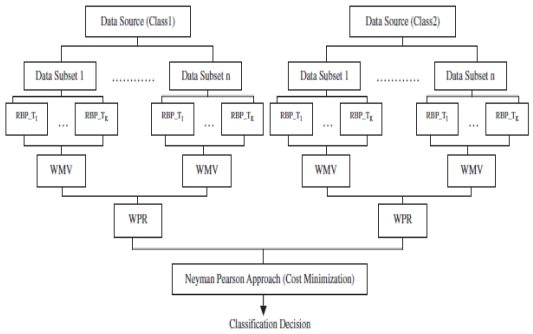

Fig 3: The Block Schematic of RBP Algorithm

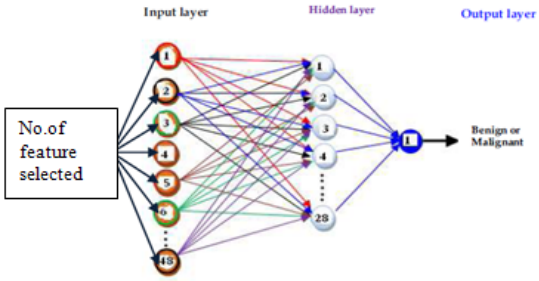

Fig 4: Proposed Architecture of RBP Neural Network

Multilayer networks typically use sigmoid transfer functions in the hidden layers. These functions are often called "squashing" functions, because they compress an infinite input range into a finite output range. Sigmoid functions are characterized by the fact that their slopes must approach zero as the input gets large. This causes a problem when you use steepest descent to train a multilayer network with sigmoid functions, because the gradient can have a very small magnitude and, therefore, cause small changes in the weights and biases, even though the weights and biases are far from their optimal values.

The purpose of the resilient backpropagation (Rprop) training algorithm is to eliminate these harmful effects of the magnitudes of the partial derivatives. Only the sign of the derivative can determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update. The size of the weight change is determined by a separate update value. The update value for each weight and bias is increased by a factor delt_inc whenever the derivative of the performance function with respect to that weight has the same sign for two successive iterations. The update value is decreased by a factor delt_dec whenever the derivative with respect to that weight changes sign from the previous iteration. If the derivative is zero, the update value remains the same. Whenever the weights are oscillating, the weight change is reduced. If the weight continues to change in the same direction for several iterations, the magnitude of the weight change increases.

First, investigate the storage format of the network. RBF networks are stored in objects with head RBFNet. The first component contains the parameters and the second component is a list of rules. Initialize an RBF network with three inputs, two outputs, and five neurons. This is done by initializing a network with matrices of the appropriate size without any data.

Input:
- Training Data ‘DS’ of size ‘N’ with correct labels $y_i \in \Omega = \{y_1, y_2\}$
- Supervised algorithm base classifier
- No. of iterations or classifiers (y)
- Number of Classes (L)

Initialize:
- $\mu = 0.5$ // False Alarm Threshold
- $L = 2$

Training:

Do j = 1... L
1. Choose samples from class ‘j’ and form Data Source DS_j
2. Split DS_j into ‘k’ subsets ($S_1, S_2, \ldots\ldots\ldots, S_k$)

Do m = 1.....k and t = 1.....y
a. Train S_m by supervised algorithm and obtain hypothesis h_t
b. Compute error of h_t: $\varepsilon_t = \sum_i^n [h_t(x_i) \neq y_i]$ (4)
c. If $\varepsilon_t > \mu$, then drop hypothesis and go to step 2.a
Else add the classifier C_t to the Ensemble ‘E_m’.
d. Normalized error (β_t): $\beta_t = \varepsilon_t / 1 - \varepsilon_t$ $0 < \beta_t < 1$ (5)

End
End

Testing:

Given an unlabeled instance ‘X’
A. Evaluate the ensemble ‘E’ of each data subset for particular class on ‘X’
B. Obtain composite hypothesis for each subset by Weighted Majority Voting
C. Each subset’s ensemble decision in combined by Weighted Product Rule
D. Choose the class with more weights.

Fig 5: Proposed RBP Algorithm

As like the above Algorithm, here we are given some specified sample of dataset to the neural network, which takes these samples and classified into its normal and attack class Implementation is carried out using MATLAB Neural Network Toolbox for the purpose DDoS attack detection.

Here we implement three separate modules which are as followings

1. The first module is according to the base paper, which finds the detection accuracy of DDoS detection attack

using different types of neural network with KDDCup 99 dataset input for this one.
2. The second one is the DDOS attack like matching using our college log file.
3. The final one is just shows us that how the intrusion is happened in normal wireless artificial immune system.

5. Simulation Results

To access the effectiveness of the proposed intrusion detection approach, the following simulation was performed. Pentium® Dual-Core CPU E5200 @ 2.50GHz, having 2.98 GB of RAM was used. The operating system used was Microsoft Windows XP Professional with Service Pack 3. Simulation was performed using Matlab 2012a version 7.9.0.529. KDD dataset containing 311030 sample data was used as input to the IDS. Out of this 217720 samples were used for training, 46655 samples were used for validation and the rest 46655 samples were used for level 1 testing.

Figure 6 shows the experimental neural network setup for the proposed IDS. The proposed IDS is capable of handling five classes of attack namely DoS, U2R, Probe, U2L and normal. Data and target value are considered to configure the network's inputs and outputs to match. Configuration is the process of setting network input and output sizes and ranges, input preprocessing settings and output post processing settings, and weight initialization settings to match input and target data. The network is trained for different values of epochs and error goal, where epoch and error goal are training parameter. Typically one epoch of training is defined as a single presentation of all input vectors to the network. The network is then updated according to the results of all those presentations. Training occurs until a maximum number of epochs occur, the performance goal is met, or any other stopping condition of the training function occurs. Figure 6 show the training of feed forward Neural Network

After executing the neural network we got the better detection accuracy as 55.86.this we can show below.

Figure 7 shows the performance of the system taking into account training, validation and level 1 testing data. It shows that the best validation performance was 2.963e-005 at epoch 41

Figure 8 shows the neural network training state plot. It also shows validation check at epoch 41 and highlights that there is no validation failure up to this epoch.

The graphical representation of the visual impression of the distribution of Errors (Targets- Outputs) is shown as the Error Histogram plot for the given data in figure 9. It consists of tabular instances shown as adjacent rectangles erected over discrete bins. It shows that maximum error ~ 0.08117.

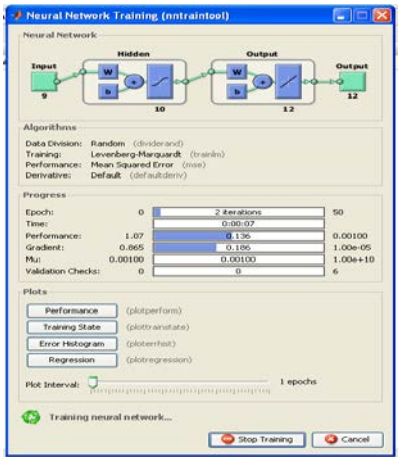

Fig 6:Training Feedforward Neural Network

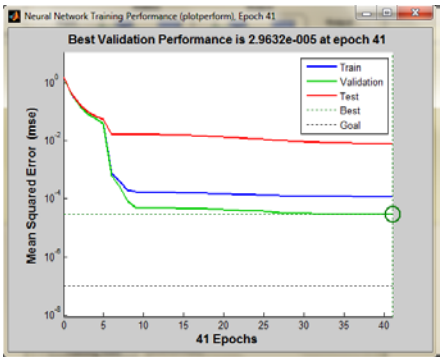

Fig 7: System performance of the during training, validation and testing

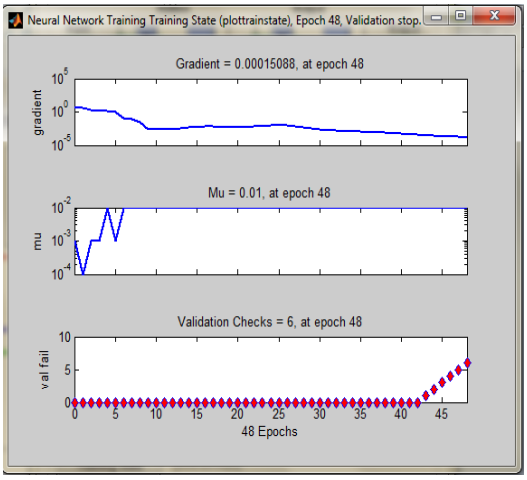

Fig 8: Neural Network Training State plots

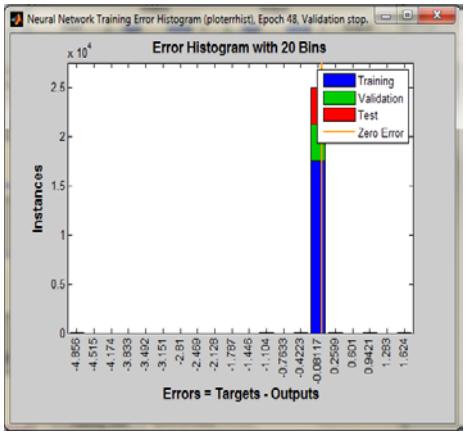

Fig 9: Neural Network Error Histogram plot

The receiver operating characteristics for training, validation and testing phases of the dataset are shown in Figure 10. The ideal value should be close to one. As per the simulation we have got data which are mostly True Positive.

So finally we are got the good detection accuracy. So that we are chosen RBP neural network is the base classifier.

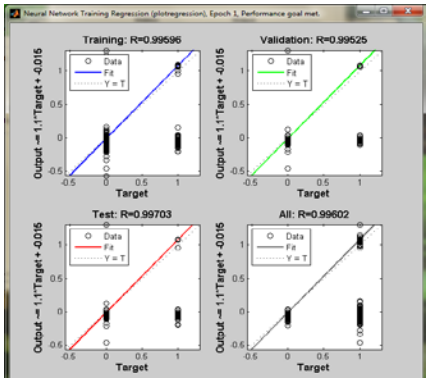

Fig 10: Graph showing True Positive vs. False Positive

Table 1: success rate of algorithms.

Algorithm name with functions	*No .of sample*	*Detection accuracy*
Normal neural network	10000	2.88%
Best validation	10000	2.9623e^(-005)
Mean deviation	10000	0.01
Error rate	10000	0.08115
gradient	10000	0.00015
RBP neural network	10000	55.36%

6. Conclusion and Future Scope

Critical services are often badly affected by DDoS attacks, in spite of the conventional deployment of network attack prevention mechanisms such as Firewall and Intrusion Detection Systems. Some intrusion detection systems detect only attacks with known signatures. Predicting the future attacks is impossible. Hence, the system must be trained and tested in such a way that it learns by observing the aberrant patterns associated with the network traffic and classify the incoming traffic as an attack or normal. The training time depends on the number of times the classifier needs training which in turn depends on the mean square error between iterations reaching global minimum. The training is speeded up by removing the overlapping data and retaining only training samples adjacent to the decision boundary. Also, as the number of input vector is less, the training time is less. Hence, it is evident that RBPBoost algorithm will be suitable for real time environment

Acknowledgement

This space is dedicated to acknowledge all those who have helped in bringing this project to fruition. I am greatly indebted to my guide Asst. Prof. D.M.Choudhari for his unstinted support and valuable suggestions. I am grateful to him not only for the guidance, but also for his unending patience and for keeping my spirits high throughout

References

[1] Abraham Yaar, Adrian Perrig, Dawn Song, FIT: Fast Internet Traceback, IEEEInfocomm, Mar. 2005.
[2] Alex C. Snoeren et al., Hash-Based IP Traceback, ACM Sigcomm, Aug. 2001, pp. 3–14.
[3] Alex C. Snoeren et al., Single-packet IP traceback, IEEE/ACM Transactions on Networking 10 (6) (2002) 721–734.
[4] Amey Shevtekar, Karunakar Anantharam, Nirwan Ansari, Low rate TCP denialof- service attack detection at edge routers, IEEE Communications Letters 9 (4) (2005) 363–365.
[5] Amey Shevtekar, Nirwan Ansari, A router-based technique to mitigate reduction of quality (RoQ) attacks, Computer Networks 52 (5) (2008) 957–970.
[6] Amey Shevtekar, Nirwan Ansari, Is it congestion or a DDoS attack? IEEE Communications Letters 15 (7) (2009) 546–548.
[7] Andrey Belenky, Nirwan Ansari, On IP traceback, IEEE Communications Magazine 41 (7) (2003) 142–153.
[8] Andrey Belenky, Nirwan Ansari, on deterministic packet marking, Computer Networks 51 (10) (2007) 2677–2700.
[9] Arbor Networks, Worldwide Infrastructure Security Report, Volume IV, November 2008.
[10] P. Arun Raj Kumar, S. Selvakumar, A DDoS threat in Collaborative environment “A survey on DDoS attack tools and Traceback mechanisms “, in: Proceedings of IEEE

International Advance Computing Conference (IACC'09), 2009, pp.1275–1280.
[11] P. Arun Raj Kumar, S. Selvakumar, Simulation of Machine Leaning algorithms in MATLAB for the selection of number of neurons, number of clusters, learning rate for Resilient Back Propagation, Support Vector Machines, KMeans, and K-Nearest Neighbor, respectively'', Technical Report, CDBR-SSE Project, Dept. of CSE, NIT, Trichy, No. Tech Report/CSE/CDBR-SSE/2009/01.
[12] T. Ash, Dynamic node creation in backpropagation networks, Connection Science 1 (4) (1989) 365–375.
[13] T. Auld, A.W. Moore, S.F. Gull, Bayesian neural networks for Internet traffic classification, IEEE Transactions on Neural Networks 8 (1) (2007) 223–239.
[14] G. Bafoutsou, G. Mentzas, Review and functional classification of collaborative systems, International Journal of Information Management 22 (4) (2002) 281– 305.
[15] A. Belenky, N. Ansari, Tracing Multiple Attackers with Deterministic Packet Marking (DPM), IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, 2003, pp.49–52.
[16] L. Breiman, Bagging predictors, Machine Learning 24 (2) (1996) 123–140.
[17] CAIDA UCSD Network Telescope Three Days Of Conficker – <dates used>,Paul Hick, Emile Aben, Dan Andersen and kcclaffy. Available from: <http://www.caida.org/data/passive/telescope-3days-conficker_dataset.xml>
[18] Y. Chen, K. Hwang, Collaborative change detection of DDoS attacks on community and ISP networks, in: IEEE International Symposium on Collaboration Technologies and Systems (CTS'06), Las Vegas, NV, 15–17 May, 2006.
[19] Clayton Scott, Robert Nowak, ''A Neyman Pearson Approach to statistical learning'', Technical Report TREE 0407.
[20] CNN, ''DDoS attacks on Yahoo, Buy.com, eBay, Amazon, Datek, E Trade'', CNN Headline News, 2000.
[21] Dawn Song, Adrian Perrig, Advanced and authenticated marking scheme for IP traceback, IEEE INFOCOMM, Apr. 2001, pp. 878–886.
[22] D. Dean, M. Franklin, A. Stubblefield, An algebraic approach to IP Traceback, Network and Distributed System Security Symposium, Feb. 2001, pp. 3–12.
[23] Devi Parikh, Tsuhan Chen, Data fusion and cost minimization for intrusion detection, IEEE Transactions on Information Forensics and Security 3 (3) (2008) 381–389.
[24] Y. Dhanalakshmi, I. Ramesh Babu, Intrusion detection using data mining along fuzzy logic and genetic algorithms, International Journal of Computer Science Security 8 (2) (2008) 27–32
[25] Dimitris Gavrilis, Evangelos Dermatas, Real time detection of distributed denial-of-service attacks using RBF networks and statistical features, Computer Networks 44 (5) (2005) 235–245.
[26] Dongqi Wang, Guiran chang, Xiaoshuo Feng, Rui Guo, ''Research on the detection of Distributed Denial of service attacks based on the characteristics of IP flow'', NPC 2008, LNCS 5245, 2008, pp. 86 93.
[27] C. Douligeris, A. Mitrokotsa, DDoS attacks and defense mechanisms: classification and state-of-the-art, Computer Networks 4 (5) (2004) 643–666.
[28] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973
[29] Y. Feng, Z.-f. Wu, J. Zhong, C.-x. Ye, K.-g. Wu, An enhanced swarm intelligence clustering-based rbf neural network detection classifier. In: Fourth International Conference on Intelligent Computing, Springer, Shanghai, China, 2008, pp. 526–533.
[30] Flashget.Availablefrom :<http://www.flashget.com>.
[31] Y. Freund, R.E. Schapire, Decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139
[32] G. Guo, H. Wang, D. Bell, Y. Bi, K. Greer, Using kNN model for automatic text categorization, Soft Computing – A Fusion of Foundations, Methodologies and Applications 10 (5) (2006) 423–430
[33] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, Upper Saddle River, NJ, 1994.
[34] J. Henry, C. Lee, et al., ICMP Traceback with Cumulative Path, An Efficient Solution for IP Traceback, International Conference on Information and Communications Security, Springer Lecture Notes in Computer Science, Sept. 2003, pp. 124–135.
[35] Hoai-Vu Nguyen, Yongsun Choi, Proactive detection of DDoS attacks using k- NN classifier in an Anti-DDoS Framework, International Journal of Computer System Science and Engineering (2008) 247–252.
[36] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, Lotfi A. Zadeh, Feature Extraction: Foundation and Applications, Physica-Verlag, Springer, 2006. ch.1.
[37] Jianjing Sun1, Han Yang, Jingwen Tian, Fan Wu, ''Intrusion Detection Method Based on Wavelet Neural Network'', IEEE Second International Workshop on Knowledge Discovery and Data Mining, 2009, pp. 851–854.
[38] KDD data set, 1999. Available from: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.
[39] J. Kittler, M. Hatef, R. Duin, J. Matas, On combining classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (3) (1998) 226–239.
[40] Y. Li, Y. Ge, X. Jing, Z. Bo, A new intrusion detection method based on fuzzy hmm, in: Third IEEE Conference on Industrial Electronics and Applications, Singapore, 2008.