A Layered Framework architecture for intrusion detection using experienced machine learning

Kruthika M.K
Dept. of Computer Science
HKBK College of Engineering
Bangalore, India.

Abstract — **The Internet is a global network connecting millions of computers. More than 100 countries are linked into exchanges of data, news and opinions. Thus the network intrusion detection system is a main thing to protect the data. In this system I have proposed a layered framework integrated with experienced machine learning technique to produce an effective novel intrusion detection system. The proposed work has tested with Knowledge Discovery & Data Mining i.e. KDD 1999 dataset and captured the live data packet using WinPcap Software .The systems are compared with existing approaches of intrusion detection system using SVM and decision tree . The results show that the proposed system has captured more type of attacks compare to all existing algorithms with high detection accuracy and efficiency.**

Keywords* — *KDD 1999 dataset; Experienced machine learning; Layered Framework; WinPcap Software

I. Introduction

An intrusion detection system (IDS) is a device or software application that monitors network or system activities for malicious activities or policy violations and produces reports to a management station. IDS come in a variety of "flavours" and approach the goal of detecting suspicious traffic in different ways. There are network based (NIDS) and host based (HIDS) intrusion detection systems. Network Intrusion Detection Systems are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. It performs an analysis for a passing traffic on the entire subnet. Works in a promiscuous mode, and matches the traffic that is passed on the subnets to the library of knows attacks. Once the attack is identified, or abnormal behavior is sensed, the alert can send to the administrator. Host Intrusion Detection Systems are run on individual hosts or devices on the network. A HIDS monitors the inbound and outbound packets from the device only and will alert the user or administrator of suspicious activity is detected. It takes a snapshot of your existing system files and matches it to the previous snapshot. If the critical system files were modified or deleted, the alert is sent to the administrator to investigate.

In Neural Network Intrusion Detection Systems, IDS designers utilize EML (Experienced Machine Learning) as a pattern recognition technique. During training, the neural network parameters are optimized to associate outputs (each output represents a class of computer network connections, like normal and attack) with corresponding input patterns (every input pattern is represented by a feature vector extracted from the characteristics of the network connection record). When the neural network is used, it identifies the input pattern and tries to output the corresponding class. The most commonly reported applications of neural networks in IDSs are to train the neural net on a sequence of information units, each of which may be an audit record or a sequence of commands. Given the significance of the intrusion detection problem, there have been various initiatives that attempt to quantify the current state of the art. In particular, MIT Lincoln Lab's DARPA intrusion detection, evaluation datasets has been employed to design and test intrusion detection systems. In 1999, recorded network traffic from the DARPA 98 Lincoln Lab dataset [5] was summarized into network connections with 41-features per connection. This formed the KDD 99 intrusion detection benchmark in the International Knowledge Discovery and Data Mining Tools Competition. KDD 99 intrusion detection datasets, which are based on DARPA 98 dataset, provide labelled data for researchers working in the field of intrusion detection and is the only labelled dataset publicly available. Numerous researchers employed the datasets in KDD 99 intrusion detection competition to study the utilization of machine learning for intrusion detection and reported detection rates up to 91% with false positive rates less than 1%. To substantiate the Performance of machine learning based detectors that are trained on KDD 99 training data; we investigate the relevance of each feature in KDD 99 intrusion detection datasets. To this end, information gain is employed to determine the most discriminating features for each class. This paper indicates that normal, Neptune and Smurf classes are highly related to certain features that make their classification easier. Since these three classes make up 98% of the training data, it is very easy for a machine learning algorithm to achieve good results [6].

After the introduction in Section I, Design and methodology are described in Section II. Section III describes the architecture of the proposed systems. Section IV explains the dataset, attack types & features used for classifying connection records. Section V shows the details of the experimental setup and results. Section VI concludes the paper with a discussion of results and scope of future work.

II. Design and Methodology

A. Dataset

KDD Cup 99 dataset which consists of a set of 41features [6] derived from each connection and a label which specifies the connection records as either normal or specific attack type. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment. Generally the attacks fall into four main categories, namely DoS, Probe, R2L and U2R. kdd cup.Data_10_percent.gzis used as training and validation dataset having exactly 494,021 instances with 22 attack types and corrected.gz as test dataset having exactly 311,029 instances. Test data is also labeled as either normal or as one of the attacks belonging to the four attack classes. It is important to note that the test data includes specific attacks which are not present in the training data. This makes the intrusion detection task more realistic [7].

B. WinPcap

In the field of computer network administration, pcap (packet capture) consists of an application programming interface (API) for capturing network traffic. Unix-like systems implement pcap in the libpcap library; Windows uses a port of libpcap known as WinPcap. Proposed IDS uses WinPcap to capture packets travelling over a network and to get a list of network interfaces for possible use with WinPcap. It uses NDIS (Network Driver Interface Specification) to read packets directly from a network adapter and support saving captured packets to a file. After saving captured packets, the proposed model detects the different type of attack and allows normal packets.

C. Layered Framework

Present networks and enterprises follow a layered defense approach to ensure security at different access levels by using a variety of tools such as network surveillance, perimeter access control, firewalls, network, host and application intrusion detection systems, data encryption and others. Given this traditional layered defense approach, only a single system is employed at every layer which is expected to detect attacks at that particular location. However, with the rapid increase in the number and type of attacks, a single system is not effective enough, given the constraints of achieving high attack detection accuracy and high system throughput. Hence, we propose a layered framework for building intrusion detection systems to build a network intrusion detection system which can detect a wide variety of attacks reliably and efficiently when compared to the traditional network intrusion detection systems. [8] Uses a layered framework with number of separately trained and sequentially arranged subsystems in order to decrease the number of false alarms and increase the attack detection coverage. [9] Uses a layered framework to build a network IDS which can detect a wide variety of attacks reliably and efficiently when compared to the traditional network IDS but the accuracy of less occurring attack is not good.

D. Experienced Machine Learning (EML) System

Neural networks are a form of artificial intelligence that uses multiple artificial neurons, networked together to process information. This type of network has the capability to learn from patterns, and extrapolate results from data that has been previously entered into the network's knowledge base. This ability makes neural network applications extremely valuable in intrusion detection. [1] Presents an approach of user behaviour modelling that takes advantage of the properties of neural network algorithms coupled with an expert system. In [2], [4] neural networks have been applied to build keyword-count-based misuse detection systems. Using known pattern/keyword of attack, EML system is trained in such a way that it detects when the similar kind of attack appears in network traffic or data set.

An attempt is made in this paper to build IDS by integrating layered framework with EML system so as to combine the advantages of both the approaches.

Thus, an integrated IDS is proposed which can detect a wide variety of attacks with less false alarm rate and can operate efficiently in high speed network.

III. PROPOSED INTEGRATED IDS ARCHITECTURE

IDS under consideration combine the advantages of both layered framework and EML System. The proposed IDS is used to detect four common types of attacks like Denial of Service(DoS), Probe, Remote to Local(R2L), User to Root (U2R) and normal records also. Thus, IDS is divided into four layers which are used to classify attacks as mentioned in Fig.1. In this architecture, data preprocessor not only collect the data, but also perform the task of data cleaning by extracting features for each layer using Principal Component Analysis (PCA) method. Since the proposed IDS comprises of four layers corresponding to each attack so PCA is applied to individual layers.

Principal Component Analysis (PCA) is a classical statistical method to find patterns in high dimensionality data sets. PCA allows obtaining an ordered list of components that account for the largest amount of the variance of the data in terms of least square errors. The amount of variance captured by the first component is larger than the amount of variance on the second component and so on. We can reduce the dimensionality of the data by neglecting those components with a small contribution to the variance [13].

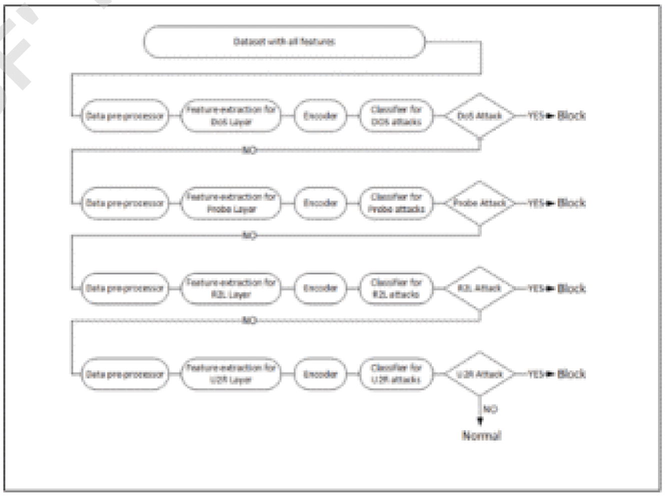

Dataset with all features
Data pre-processor
Feature-extraction for DoS Layer
Encoder
Classifier for DOS attacks
DoS Attack
YES
Block
NO
Data pre-processor
Feature-extraction for Probe Layer
Encoder
Classifier for Probe attacks
Probe Attack
YES
Block
NO
Data pre-processor
Feature-extraction for R2L Layer
Encoder
Classifier for R2L attacks
R2L Attack
YES
Block
NO
Data pre-processor
Feature-extraction for U2R Layer
Encoder
Classifier for U2R attacks
U2R Attack
YES
Block
NO
Normal

Fig 1: Architecture of proposed IDS based on layered framework integrated with neural network

Each layer of IDS consists of three components:

- Data preprocessor

This component is used to collect the data from a desired source. Here as input, KDD cup 99 datasets is used which is publicly available and WinPcap software is used for capturing packets from network traffic and save the captured packet in file.

- Encoder

The encoder is basically used to encode the data into desired format. The attribute given in KDD data set are converted into double data type to make it compatible with EML system.

- Classifier

This component is used to analyze the audit pattern and classify it to detect attacks. Here, Layered framework integrated with back propagation neural network (BPN) with 'trainscg' as training algorithm [12] is used to classify the records as normal or attack. To detect the attack, we use apriori association rules mining in Classifier block. Rules can be viewed simply as an [IfThen Else] structure, such as:

If protocol="TCP" or protocol="UDP" Then AttackType="smurf"

The algorithm is an influential algorithm for mining frequent item sets. It uses level-wise search, where k-item sets (an item set that contains k items) is used to explore (k+1) item set. Simply, a first-item set generated will be used to generate the second-item set, in turn generate the third item set until no more k-item set can be found [8]. In [11], we can find different techniques to count the support and confidence value from the dataset have been used. A number of association rules have been derived for each type of attack and how association rule is best for IDS

The following algorithm is general for any kind of data set. Here F contains the largest frequent item set. Min_supp defines the user define support and Min_conf defines the user defines confidence. RULE contains the desired rules generated from the data set.

Algorithm 1: Apriori Association Rule

```
1. Take the largest frequent item set F with
Min_Supp and Min_Conf value.
2. Generate all possible subsets of F and store
it in SUB.
3. Count SUPP and CONF value for each
elements of SUB.
4. If (SUPP>=Min_Supp &&
CONF>=Min_Conf) then
a. Choose the particular elements of SUB
and store in RULE
b. Generate various rules and store in
RULE.
5. Else reject the particular element of SUB
and go to step 3.
6. Return RULE.
7. End.
```

IV. Dataset, Attack Types & Features Description

KDD dataset consists of 41 features for each connection, which are detailed in Appendix 1 of [6]. Features are grouped into four categories:

Basic Features: Basic features can be derived from packet headers without inspecting the payload. Basic features are the first six features listed in Appendix 1 of [6].

Content Features: Domain knowledge is used to assess the payload of the original TCP packets. This includes features such as the number of failed login attempts;

Time-based Traffic Features: These features are designed to capture properties that mature over a 2 second temporal window. One example of such a feature would be the number of connections to the same host over the 2 second interval;

Host-based Traffic Features: Utilize a historical window estimated over the number of connections – in this case 100 – instead of time. Host based features are therefore designed to assess attacks, which span intervals longer than 2 seconds.

The KDD 99 intrusion detection benchmark consists of three components, which are detailed in Table 1. In the International Knowledge Discovery and Data Mining Tools Competition, only "10% KDD" dataset is employed for the purpose of training. This dataset contains 22 attack types and is a more concise version of the "Whole KDD" dataset. It contains more examples of attacks than normal connections and the attack types are not represented equally. Because of their nature, denial of service attacks account for the majority of the dataset. On the other hand the "Corrected KDD" dataset provides a dataset with different statistical distributions than either "10% KDD" or "Whole KDD" and contains 14 additional attacks.

TABLE 1. BASIC CHARACTERISTICS OF THE KDD 99 INTRUSION DETECTION DATASETS IN TERMS OF NUMBER OF SAMPLES

Dataset	DoS	Probe	U2R	R2L	Normal
10% KDD	391458	4107	52	1126	97277
Corrected KDD	229853	4166	70	16347	60593
Whole KDD	3883370	41102	52	1126	972780

The DARPA 1999 test data consisted of 190 instances of 57 attacks which included 37 Probes, 63 DoS attacks, 53 R2L attacks, 37 U2R/Data attacks with details on attack types given in Table 2.

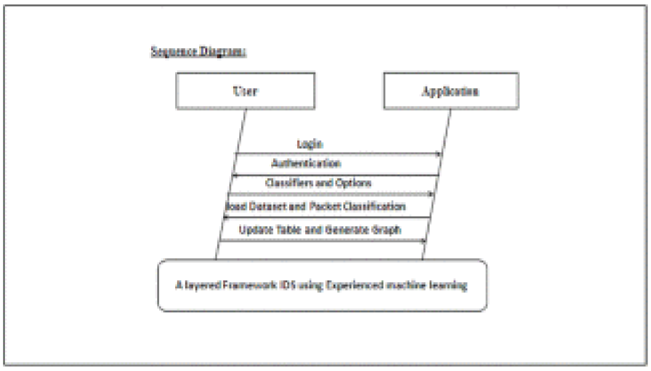
TABLE 2. ATTACKS PRESENT IN DARPA 1999 DATASET

Attack class	Attack type
Probe	portsweep, ipsweep, queso, satan, msscan, ntinfoscan, lsdomain, illegal-sniffer
DoS	apache2, smurf, neptune, dosnuke, land, pod, back, teardrop, tcpreset, syslogd, crashiis, arppoison, mailbomb, selfping, processtable, udpstorm, warezclient
R2L	dict, netcat, sendmail, imap, ncftp, xlock, xsnoop, sshtrojan, framespoof, ppmacro, guest, netbus, snmpget, ftpwrite, httptunnel, phf, named
U2R	sechole, xterm, eject, ps, nukepw, secret, perl, yaga, fdformat, ffbconfig, casesen, ntfsdos, ppmacro, loadmodule, sqlattack

V. EXPERIMENTAL SETUP AND RESULT

A. Experimental Setup

Proposed IDS is simulated to obtain results using Microsoft Visual Studio 2010 and C# language. All experiments are done on 133-MHz Intel Pentium-class processor @ 3.30 GHz having 4 GB of RAM. The operating system used is Windows 7. Simulation is performed using data set described in Section IV.


Sequence Diagram:
User
Application
Login
Authentication
Classifiers and Options
load Dataset and Packet Classification
Update Table and Generate Graph
A layered Framework IDS using Experienced machine learning

In the sequence diagram above the complete sequence of execution of Intrusion detection has been shown. The server starts and verifies the user. As per the choice of the user for classifiers and an algorithm to be applied it processes the dataset and detects the intrusion type by comparing it with the preloaded dataset if Intrusion is made. Then the output is generated and the updates are saved in the database.

User sends the classified data to the testing phase for detecting attack using experienced machine learning and generate graph for analysis as shown in Data Flow Diagram (Fig 2).

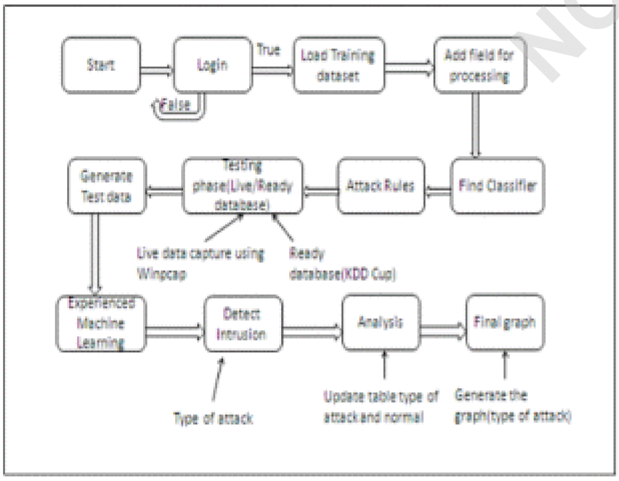
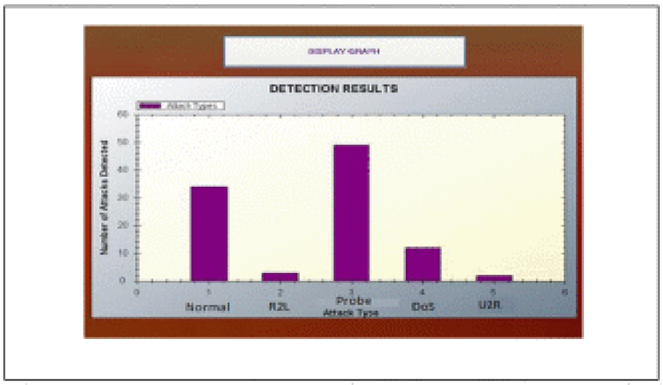


Fig 2: *Data Flow Diagram of proposed IDS based on layered framework integrated with neural network*

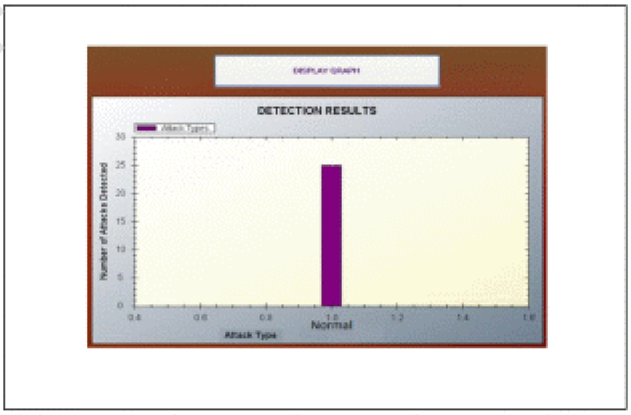
B. Experimental Result

I have used KDD dataset and captured packets from WinPcap as input. Model is tested for 100 packets of KDD data set as input to proposed IDS. Same data set is used as input to already existing IDS which uses SVM technique to detect the attack [14]. 100 Packet Data using EML took 7.53 millisecond to detect the attack type whereas 100 Packet Data using SVM took 10.09 second to detect attack type. Graph depicts 34 packets as Normal attack, 50 packets as Probe attack, 11 packets as DoS attack, 2 packets as U2R attack and 3 packets as R2L attack type.

When I captured 25 packets in real time by accessing websites using WinPcap, 25 Packet Data using EML method took 2.1642 second to detect the attack type and all packet detected as Normal attack type as Antivirus blocks the suspicions attack and allow only Normal packets.

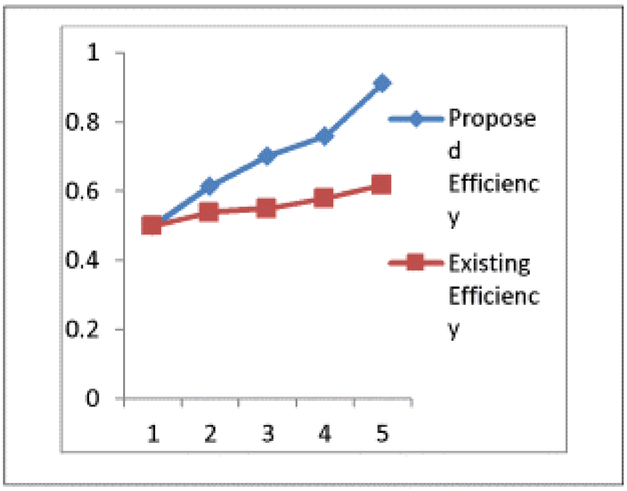


I have used ROC curve to compare the accuracy and efficiency of proposed model and existing model. Proposed model is Layered framework IDS with EML and existing model is IDS with SVM technique.



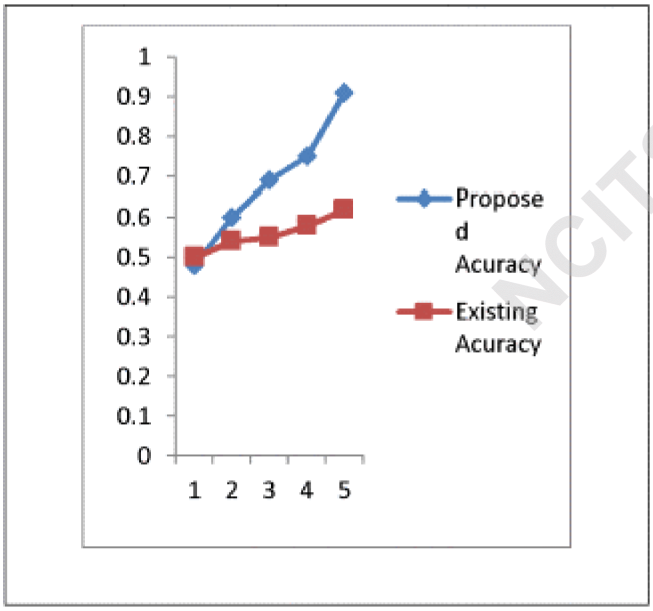
Proposed Model Efficiency	Existing Model Efficiency
0.5	0.5
0.615384615	0.54
0.701923077	0.55
0.759615385	0.58
0.913461538	0.62

Proposed Model Accuracy	Existing Model Accuracy
0.48	0.5
0.6	0.54
0.69	0.55
0.75	0.58
0.91	0.62



VI. Conclusion

In this paper, Intrusion detection system (Fig 1) is designed by integrating layered framework with EML. It is observed that proposed model which considers feature extraction using PCA algorithm reduces training time and increase in the success rate of attack detection. Results of ROC curve suggest that proposed IDSs works effectively and accurately for detecting various attacks in the network.



References

[1] Herv`e Debar, Monique Becke, Didier Siboni, "A Neural Network Component for an Intrusion detection system," Proceedings of the IEEE Symposium on Research in Security and Privacy, pp. 240–250,1992.

[2] J. Ryan, M. Lin, R. Mikkulainen, "Intrusion Detection with Neural Networks," Advances in Neural Information Processing Systems, vol. 10, 1998, MIT Press.

[3] Jai Sundar Balasubramaniyan, Jose Omar Garcia-Fernandez, David Isacoff, Eugene H. Spafford ,Diego Zamboni, "An Architecture for Intrusion Detection Using Autonomous Agents," Proceeding of IEEE 14th Annual Computer Security Applications Conference, pp. 13–24, 1998.

[4] R. Lippmann, R. Cunningham, "Improving Intrusion Detection Performance using Keyword Selection and Neural Networks," RAID Proceedings, West Lafayette, Indiana, Sept 1999.

[5] Richard Lippmann, Joshua W. Haines, David J. Fried, Jonathan Korba, Kumar Das Lincoln Laboratory MIT, "The 1999 DARPA Off-Line Intrusion Detection Evaluation, " 1999

[6] H. Gunes Kayacık, A. Nur Zincir-Heywood, Malcolm I. Heywood "Selecting Features for Intrusion Detection:A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets"

[7] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani,"A Detailed Analysis of the KDD CUP 99 Data Set," Proceedings of the IEEE Symposium on Computational Intelligence in Security and Defence Applications, 2009

[8] Flora S. Tsai, Nanyang Technological University, "Network Intrusion Detection Using Association Rules," International Journal of Recent Trends in Engineering, Vol 2, No. 2, November 2009

[9] Kapil Kumar Gupta, Baikunth Nath, Ramamohanarao Kotagiri, "Layered Approach Using Conditional Random Field for Intrusion Detection," IEEE Transactions on Dependable and Secure Computing, vol. 7(1), pp. 35-49, March,2010.

[10] B.Bhanu Chander, K. Radhika, D. Jamuna," AN APPROACH ON LAYERED FRAMEWORK FOR INTRUSION DETECTION SYSTEM," Asian Journal of Computer Science and Information Technology 2012, AJCSIT

[11] ASIM DAS & S.SIVA SATHYA Pondicherry University, "ASSOCIATION RULE MINING FOR KDD INTRUSION DETECTION DATA SET," International Journal of Computer Science and Informatics ISSN, March , 2012

[12] N.V.N. Indra Kiran, M.Pramiladevi Devi and G.Vijaya Lakshmi, " Training Multilayered Perceptrons for Pattern Recognition: A Comparative Study of Five Training Algorithms," International MultiConference of Engineers and Computer Scientists, March 20011

[13] Gopi K. Kuchimanchi, Vir V. Phoha, Kiran S. Balagani, Shekhar R. Gaddam,"Dimension Reduction Using Feature Extraction Methods for Real-time Misuse Detection Systems," Proceedings of the 2004 IEEE Workshop on Information Assurance and Security, June 2004.

[14] Snehal A. Mulay, P.R. Devale, G.V. Garje," Intrusion Detection System using Support Vector Machine and Decision Tree," International Journal of Computer Applications (0975 – 8887), June 2010