ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2014.0097

REAL-TIME INTELLIGENT MULTILAYER ATTACK CLASSIFICATION SYSTEM

T. Subbhulakshmi[1], S. G. Keerthiga[2] and R. Dharini[3]
[1]*Department of Information Technology, Sethu Institute of Technology, India*
E-mail: subbulakshmitce@yahoo.com
[2, 3]*Department of Computer Science and Engineering, Thiagarajar College of Engineering, India*
E-mail: [2]keerthiga.sg@gmail.com, [3]dharinirengarajan322@gmail.com

Abstract

Intrusion Detection Systems (IDS) takes the lion's share of the current security infrastructure. Detection of intrusions is vital for initiating the defensive procedures. Intrusion detection was done by statistical and distance based methods. A threshold value is used in these methods to indicate the level of normalcy. When the network traffic crosses the level of normalcy then above which it is flagged as anomalous. When there are occurrences of new intrusion events which are increasingly a key part of system security, the statistical techniques cannot detect them. To overcome this issue, learning techniques are used which helps in identifying new intrusion activities in a computer system. The objective of the proposed system designed in this paper is to classify the intrusions using an Intelligent Multi Layered Attack Classification System (IMLACS) which helps in detecting and classifying the intrusions with improved classification accuracy. The intelligent multi layered approach contains three intelligent layers. The first layer involves Binary Support Vector Machine classification for detecting the normal and attack. The second layer involves neural network classification to classify the attacks into classes of attacks. The third layer involves fuzzy inference system to classify the attacks into various subclasses. The proposed IMLACS can be able to detect an intrusion behavior of the networks since the system contains a three intelligent layer classification and better set of rules. Feature selection is also used to improve the time of detection. The experimental results show that the IMLACS achieves the Classification Rate of 97.31%.

Keywords:

Distributed Denial Service of Attacks, Intrusion Detection System, Support Vector Machine, Neural Networks, Fuzzy Inference System

1. INTRODUCTION

Firewall and filtering techniques are used for securing the network. Relying on a firewall system alone is not sufficient to prevent a corporate network from all types of network attacks. This is because a firewall cannot defend the network against intrusion attempts on open ports required for network services. Hence, an Intrusion Detection System (IDS) is usually installed to complement the firewall. An IDS collects information from a network or computer system, and analyzes the information for symptoms of system breaches. IDS inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system. Intrusion detection is the act of detecting actions that attempt to compromise the confidentiality, integrity or availability of a resource. Intrusion detection does not, in general, include prevention of intrusions. An IDS is essentially a burglar alarm system for your network. It enables you to monitor your network for intrusive activity. When intrusive activity occurs, your IDS generate an alarm to let you know that your network is possibly under attack. Like regular burglar alarms, however, your IDS can generate "false positives" or "false alarms". A false positive occurs when your IDS generates an alarm from normal user activity. If your IDS generate too many false positives, then you will lose confidence in the capability of your IDS to protect your network. If you have a burglar alarm that continually goes off incorrectly, the police will become conditioned to the fact that your establishment is prone to false alarms. During an actual break-in, the police may not respond as quickly, thinking that the alarm is just another false alarm. Therefore, it is crucial that you configure your IDS to minimize the number of false positives that it generates. IDS may also experience false negatives. In this situation, an attack occurs against your network and your IDS fails to alarm even though it is designed to detect such an attack. Your IDS should almost never generate false negatives. In fact, it is preferable for your IDS to actually generate more false positives rather than generating any false negatives.

1.1 PROBLEM STATEMENT

Let S be the server which is to be attacked, $A = \{A_1, A_2, A_3, \ldots, A_n\}$ be the attacking sources and $L = \{L_1, L_2, L_3, \ldots, L_n\}$ be the legitimate users. Any of the legitimate users, say L_i requires the information from server *S*. During the normal access the attacking sources $A = \{A_1, A_2, \ldots A_n\}$ unnecessarily requests the server *S* after establishing the connection with the server. These attacking sources overload the server *S* by flooding packets. When the legitimate users say L_i request information from server *S*, the service cannot be provided or the process is slowed down. An IDS D is required to classify request from legitimate users L_i and attacking sources *A*i as $D_L = \{L_1, L_2, \ldots, L_n\}$ and $D_A = \{A_1, A_2, \ldots, A_n\}$.

1.2 PROBLEM DESCRIPTION

This paper focuses on capturing the packets which are transferred through the network and extracting the attributes from the packet. Using this attributes, the derived attributes are computed and written within the file. This file with class label is used as training data for support vector machine which is a binary classifier. The output of SVM gives the records which are detected as attack and this is given to neural networks where training and testing is done. The output of neural networks is

DOI: 10.21917/ijsc.2014.0097

given to FIS. Based on the rules generated in the Fuzzy Inference system, the type of attack is detected.

2. LITERATURE SURVEY

One of the successful approaches based on data-mining framework used RIPPER rules which have been presented by Lee et al. [14] Association rules and Frequent Episodes algorithms have been used to derive correlations between features and represent the sequentially of audit records, respectively. They considered four types of attacks which are DoS, Probe, U2R and R2L using the DARPA dataset that KDD99 dataset is based on. Most of their anomaly detection rates were lower than the detection rates with known/trained data.

Off-line intrusion detection approaches have also been proposed using the KDD99 as input dataset such as Katos et al. [12] where analysis of data and clustering evaluation were investigated, and Chen et al. [5] where anomaly score of a packet was computed based on the deviation from the normal behavior.

2.1 SVM FOR IDS

Snehal A. Mulay, P. R. Devale, G.V. Garje [26] proposed the combination of SVM and the decision tree approaches for preparing decision making models. They proposed that the integrating different models give better performance than the individual learning or decision-making models. Integration reduces the limitations of individual model. Rung-Ching Chen, Kai-Fan Cheng and Chia-Fen Hsieh [24] used RST (Rough Set Theory) and SVM (Support Vector Machine) to detect intrusions. First, RST is used to preprocess the data and reduce the dimensions. Next, the features were selected by RST will be sent to SVM model to learn and test respectively.

2.2 NN FOR IDS

Ryan et al. [25] described an off-line anomaly detection system (NNID) which utilized a back-propagation MLP neural network. The MLP was trained to identify users' profile and at the end of each log session, the MLP evaluated the users' commands for possible intrusions. Cannady [10] used a three layer neural network for offline classification of connection records in normal and misuse classes. The system designed in this study was intended to work as a standalone system (not as a preliminary classifier whose result may be used in a rule-based system).

Cunningham and Lippmann [13] used ANNs in misuse detection. They used an MLP to detect Unix-host attacks by searching for attack specific keywords in the network traffic. InSeon [9] in 2002 tried to integrate a smart detection engine into a firewall and detecting unusual structures in data packets uses a classical feed- forward multi-layer perceptron network: a back propagation neural network and time delay neural network to program-based anomaly detection. Moradi and Zulkernine [18] used a Multi-Layer Perceptron (MLP) artificial neural network in off- line mode to classify normal network activity, Satan (Probe) attacks and Neptune attacks using the KDD99 dataset.

2.3 FIS FOR IDS

Agarwal and Joshi proposed a framework for learning a rule-based model (PN rule) to make classifier models on a dataset that has widely different class distributions in training data. Fuzzy Sets is another technique often used for IDS. This technique generally falls into two categories, fuzzy misuse detection [1] and fuzzy anomaly detection [19]. Abraham and Jain [1] used three types of fuzzy rules to compare with linear generic programming (LPG), Decision Tree, and Support Vector Machines (SVM). Their result showed that one of their fuzzy rules gave the best detection rate using the 41 features of the DARPA 1998 dataset. Liao et al. used fuzzy logic and an expert system with the DARPA 2000 dataset and achieved more than 91.5% detection rate over all attack types, while reducing complexity of traditional techniques for ranking fuzzy numbers.

2.4 HYBRID METHODS FOR IDS

Hybrid neural-net-based IDS is a category of neural-net-based.IDS encompasses systems that combine supervised and unsupervised neural nets. Jirapummin [13] proposed employing hybrid neural network for both visualizing intrusions using Kohenen's SOM and classifying intrusions using a Resilient Propagation neural network (RPROP). The artificial neural network approach is one of the most popular techniques for the design of IDS. Jirapummin et al. [11] proposed a hybrid neural network using a combination of Self-Organizing Map (SOM) and Resilient Back- Propagation Neural Network (BPNN). To evaluate their approach, they used an available well known preprocessed dataset which is KDD99.The KDD99 dataset is a network packet dataset consisting of normal network activity as well as many network attack types. The dataset is based on the DARPA98 dataset from MIT Lincoln laboratory, which provides answer class (labeled data) for evaluation of intrusion detection.

Pan et al. [21] designed a hybrid system by using a BPNN and a C4.5 Decision Tree considering the KDD99 dataset. The results showed that using only a BPNN without C4.5 Decision Tree. Ngamwitthayanon et al. [19] designed a multi-state IDS system to classify normal data and each attack type using the KDD99 dataset. Their results showed a higher detection rate in each classification category than when only a single state was used to classify all categories.

2.5 UNIQUENESS OF THE APPROACH

Our proposed system uses layered approach for intrusion detection. The input is the real time dataset which is got from network. The layered approach involves 3 layers namely layer 1 is the binary classification of support vector machine, layer 2 is the classification by neural networks and layer 3 involves classification of further attack types by fuzzy inference system. By using this layered approach the detection rate is increased.

- This paper provides the IDS dataset in three formats for all attacks which can be used by all the other researchers for DDoS attack data related experiments
- Real time dataset is used as input for the classification algorithms. Since real time dataset is used, the possibility of extending the paper for online Multi level ACS is much easier
- The layered approach used in this paper helps in increasing the detection rate and the precision of the experiments

- The paper combines both rule based and learning based classifiers and hence the classification is precise and also adaptable

3. INTELLIGENT MULTILAYER ATTACK CLASSIFICATION SYSTEM

This IMLACS architecture involves feature extraction from wire shark, selecting required features for attacks, processing the required features using the java library used for network operations such as packet transfer, packet manipulation, extracting packet header information etc. The file which is generated as a result of processing in *'jpcap'* is given as input to IMLACS. The modules in the system design are, explained in the further sections.

3.1 TRAFFIC GENERATION

The DDOS attack traffic is generated in the network using "packit" network packet injection tool. In the network, first the number of test nodes will be selected based on the attack scenario. In the case of DOS attack, only one source node and one sink node will be selected and in the case of DDOS attacks, two or more source nodes and one sink node will be selected.

The user has to upload the source and sink programs to the server and specify the corresponding source and sink nodes in which the programs has to be executed. The user should also specify the time interval in which the programs should be executed. During the execution of program the specified source programs will generate traffic towards sink nodes. This traffic along the sinks will be recorded using the '*tshark*' traffic/ protocol analyzer and dumped into a file at the sink nodes. The dumped traffic contains both the generated traffic and the normal traffic. Two outputs are collected from traffic generation.

1. Alerts in *.csv file from snort
2. Packet header details in *.pcap from tshark

Format of collected Alerts

The dump file when given as input to the snort, alerts will be generated. The alerts will be recorded in Comma Separated Value (CSV) files for further processing.

Format of packet header attributes

Table.1. *.pcap file format description

NAME	FEATURE
Destination address	Address to which the packet is forwarded
Destination port	Port number of the application to which the packet is sent
Hardware destination address	Mac address of the destination system
Hardware source address	Mac address of the source system
Number	Provides numbering to the packets delivered
Information	Content about the packet
Network destination address	Network address of the destination system
Network source address	Network address of the source system
Packet length	Total number of bytes in a packet
Protocol	Type of the protocol
Relative time	The time took for the packet to arrive
Source port	Port number of the application in source

The alerts generated from snort consist of 6 tuples("msg", "proto", "srcip", "srcport", "dstip","dstport").

The details about the contents in *.pcap file and the explanation for each attribute is shown in Table.1. For example, source address indicates the address of the originator.

3.2 FEATURE EXTRACTION

One of the difficulties for detecting attack is to extract and select the most important attributes that represent attack behaviors to clearly distinguish them from normal activities. Since the programs for extracting attributes from raw network traffic are not available, program is written (in Java) to extract the attributes from the packet. It is a network intrusion data set. From the Dataset, the derived attributes are computed.

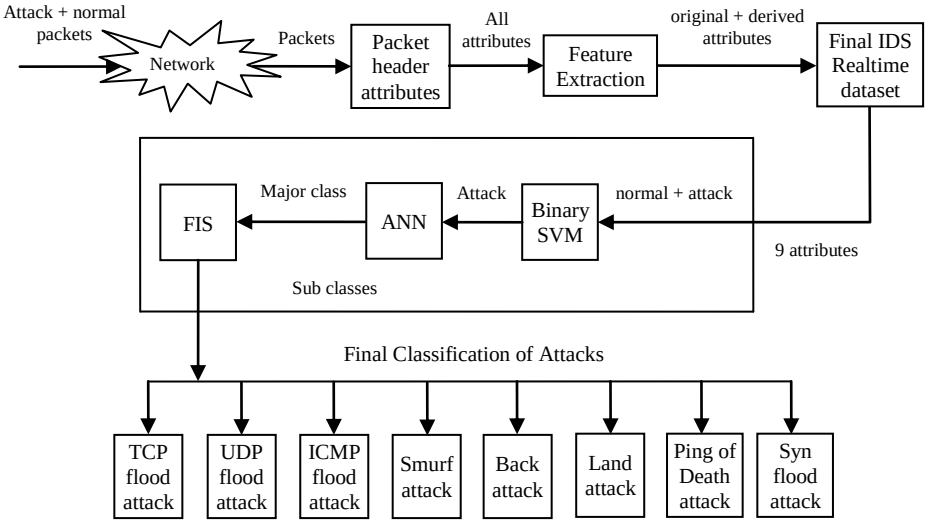

Fig.1. IMLACS Architecture

DOI: 10.21917/ijsc.2014.0097

The packet header details are displayed using wire shark and some of the attributes are extracted. The '*jpcap'* library of java is used to access the attributes from the packet header. For example, source address is retrieved by *src_ip*, destination address by *dst_ip*, source port by *src_port*, destination port by *dst_port*.

A very important decision is the selection of feature that would be used in attack detection. The features of network traffic should be in a suitable form in order to be easily processed and representative of network activity in order to be able to distinguish normal and abnormal activity. It is important that the selected features increase the contrast between normal and abnormal activity concerning attacks. The analysis about different types of attacks and the attributes necessary for that attack are done. The rules for all attacks and attributes required for that rules are identified by learning. The attributes given below are taken for processing and some of the attributes are obtained from pre-processing step that is described below. The most important features for the detection of DDOS attacks include

- Source Address – IP address of the source machine which sends packet
- Destination Address – IP address of the destination machine to which the packet is forwarded
- Source port – Port number of the application in source
- Destination port – Port number of the application to which the packet is sent
- Count – Total number of packets for the same source and destination
- Syn Count – Total number of packets with syn flag enabled
- Echo Request Count –Total number of packets with echo request flag enabled
- Protocol – Type of the protocol
- Source Bytes – Displays how many bytes have been transferred in packet

These features [22] have been used in our approach considering the fact that they are the most important for the detection of attacks.

3.3 FINAL RTIDS DATASET CREATION

In data extraction process some of the basic attributes are derived from packet header. In preprocessing the derived attributes necessary for attack detection have to be retrieved. Java program using *'jpcap'* library is used for this purpose. The process is to find all the attributes related to particular source and particular destination for some specified interval. For example, within 1 minute between a particular source say 10.0.0.123 and particular destination 192.34.14.2 the number of packets sent , protocol used, how many packets are sent with syn flag enabled, how many packets are sent with ICMP echo request, what is the source bytes or packet length.

If source address and destination address, source port and destination port are equal then separate count is made for that condition. Flag variables can be created for special cases like if the destination address is a broadcast address and if the source bytes are greater than maximum allowed bytes (65535). Calculating how many packets are enabled with echo request count and maintaining the count details. Similarly, the count details for packets enabled with syn flags are calculated.

The calculation of how many packets are of TCP protocol for particular source and destination and how many packets are of ICMP and also for UDP have been manipulated.

The file size of the packets and the number of attack and normal packets for each protocol is shown in Table.2. The file size is represented in MB. The table indicates the details for 10 bytes and 100 bytes of packet size.

Table.2. Packets and file size

Type of Attack	File size (MB)		Number of Packets	
	Packet size		Packet size	
	10 bytes	100 bytes	10 bytes	100 bytes
UDP	455	499	6281618	4753502
TCP	809	410	11163047	5658335
ICMP	319	274	2990933	2571618
SMURF	432	24.3	4049226	227917

3.4 IMLACS

3.4.1 Layer 1 Binary SVM Classification:

SVM is learning machines that plot the training vectors in high dimensional feature space, labeling each vector by its class. SVMs classify data by determining a set of support vectors, which are members of the set of training inputs that outline a hyper plane in the feature space . SVMs provide a generic mechanism to fit the surface of the hyper plane to the data through the use of a kernel function. The user may provide a function (e.g., linear, polynomial, or sigmoid) to the SVMs during the training process, which selects support vectors along the surface of this function. The SVM is a learning machine that plots the training vectors in high dimensional feature space, labeling each vector by its class. SVMs classify data by determining a set of support vectors, which are members of the set of training inputs that outline a hyper plane in the feature space. SVMs provide a generic mechanism to fit the surface of the hyper plane to the data through the use of a kernel function. The user may provide a function (e.g., linear, polynomial, or sigmoid) to the SVMs during the training process, which selects support vectors along the surface of this function. The number of free parameters used in the SVMs depends on the margin that separates the data points but not on the number of input features.

The preliminary step is to label the records in preprocessed file. Two class labels are used namely 1 for attack records and –1 for normal records. This labeled file is used for training the SVM. A set of records is taken as testing data. After the completion of training the testing data is given as input. The training data consist of all the possibilities of attack and normal record types so that classification can be done. The output is written in two files one containing only attack records and the other contains normal records.

3.4.2 Layer 2 Neural Networks Classification:

The neural network consists of interconnected neurons. By modifying the connections between these nodes the network is

able to adapt to the desired outputs. The neuron computes the weighted sum of the inputs it gets from the other neurons and gives an output as a single number to another neuron that performs the same task. The result of the transformation is determined by the characteristics of the neurons and the weights associated with the interconnections among them. The neurons in a neural network are organized into layers. The layers are divided into an input layer, hidden layer and output layer. The inputs to the input layer are packet header information. This layer does not play any significant role to the computing of the result. It only feeds information into the neural network. The hidden layers have no external connections. They only have connections with other layers in the network. The interconnection between the hidden layers continues until some condition is satisfied. The outputs from the output layer are returned to the environment.

3.4.3 Layer 3 FIS Classification:

The result of neural network is given to FIS which classifies the attack based on the rules specified. The output is evaluated using the evalfis function in FIS. The triangular membership function is used for specifying the ranges. For each attribute the ranges are specified. The output of the neural network is used along with other attributes for evaluation. The result of FIS is written to a file along with other attributes.

4. IMPLEMENTATION DETAILS

4.1 PREPROCESSING

The implementation involves preprocessing and classification. In preprocessing, the input file is created from the packet captured in the network. The dataset consist of packet with size 100 bytes of TCP, 100 bytes of UDP, 200 bytes of ICMP packets. The *'jpcap'* library is installed for processing the packet header. The packet header attributes are retrieved and stored in a separate file. The data is separated into records by connection between two ip addresses. The attributes used are,

Table.3. Packet header attributes

• Source address • Source port • Count • Echo request count • Flag for same IP address • Flag for broadcast address	• Destination address • Destination port • Syn count • Protocol • Flag for same port • Source bytes

The details about the attributes of packet header are shown in Table.3. It shows the attributes which are used as input. There are 9 attributes and 3 flag variables which is required for further processing in classification. The output of this java program is written to a text file.

The details of file size and number of records created for the packets shown in Table.4. The file size is represented in KB. The tabulation is done for 10 bytes of dataset and 100 bytes of dataset separately. From this the dataset for 100 bytes is given as input to our system.

Table.4. Records and file size

Type of Packet	File size (KB)		Number of Records		Total Records
	10 bytes	100 bytes	10 bytes	100 bytes	
UDP	27	48	526	1058	1584
TCP	15	15	277	273	550
ICMP	47	35	1036	781	1817
SMURF	1	1	15	15	30
Total Records			1854	2127	3981

4.2 SVM IMPLEMENTATION

The preprocessed file is modified to include the class label for attack. An integer value of 1 is used for attack and -1 is used for normal. Each record in preprocessed file is labeled before it is used for training a total of 19280354 packets are used for testing.

Table.5. Testing dataset

Number of normal packets	Number of attack packets	Total packets
68873	19211481	19280354

The overall preprocessed dataset used for the classification is given in Table.5. The total number of normal packets and attack packets used as the input for the layer 1 is mentioned. The number of attack packets includes the packets for all the eight types of attacks.

Table.6. Data set used for SVM

	Attack records	Normal records	Total records
Training set	300	400	700
Testing set	1040	1198	1238
Total set	1340	1598	1938

The data set used as the input of SVM is shown in Table.6. The total number of attack records is found to be 1340. The total number of normal records is found to be 1598. The total number of records in training and testing set is also shown. It is found that the testing records are taken at least three times more than the training records.

4.3 NEURAL NETWORKS IMPLEMENTATION

The neural network consists of input layer, hidden layers and output layer. Four hidden layers are in between the input and output layer. Seven neurons are in the input layer and hidden layers. Each node in input layer communicates with every node in hidden layer. Weights are assigned to every connection in between input to hidden and hidden to output layer. Output layer has only one neuron. The seven neurons in input and hidden layer are,

Table.7. NN input attributes

• Count • Echo request count • Flag for same IP address • Class label	• Syn count • Protocol • Flag for same port • Source bytes

DOI: 10.21917/ijsc.2014.0097

The attributes used in is shown in Table.7. From the table it is known that every record has eight fields, first seven fields are input to the input layer and eighth field is class label. It represents attack type.

Table.8. NN implementation details

Number of input neurons	9
Number of output neurons	1
Number of hidden neurons	9
Number of hidden layers	4
Number of iterations	100

The implementation details for layer 2 neural network classification are shown in Table.8. It shows the number of input attributes used, number of output, number of hidden layers used, number of iterations involved for training the dataset and number of neurons used in hidden layers.

Table.9. Class label for type of attack

Attack Type	Classlabel
TCPFlood attack	0.4
UDPFlood attack	0.5
ICMPFlood attack	0.6

The class label specified for each type of attack is shown in Table.9. The table indicates the target value used for different attack types. A user defined value can be used for target value. According to the table a value of 0.4 represents TCP flood attack.

Table.10. Dataset used for NN

Number of training records	781
Number of testing records	1040
Total records	1821

The dataset used for neural network classification layer is given in Table.10. The table shows that the number of training records is found to be 781 and the number of testing records is found to be 1040.

Table.11. Output of NN

Attack type	Number of Packets	Number of records
TCP flood	10694478	267
UDP flood	4884087	289
ICMP flood	3632916	465
Total	19211481	1021

The output of neural networks is shown in Table.11. The table indicates the number of packets and number of records created for the packets used as input for each type of attack. The number of records for TCP flood attack is found to be 267, the number of records for UDP flood attack is found to be 289 and the number of records for ICMP flood attack is 465.

4.4 FIS IMPLEMENTATION

The output of neural network is stored in a file along with other attributes. This file is given as input to FIS where the rules are written. Triangular membership function is used and based on rules the attack detection is done.

Table.12. Input attributes for FIS

• Count	• Syn count
• Echo request count	• Protocol
• Flag for same IP address	• Flag for same port
• Output of neural network	• Source bytes

The attributes used as input to FIS classification layer is given in Table.12. The table indicates the input attributes used for FIS. Along with other attributes the output of neural network is used as input to FIS.

Table.13. Values of membership functions

Variable	Membership Function	Minimum Value	Maximum Value
Count	Triangular	-1	5000000
Protocol	Triangular	0	20
Syn count	Triangular	-1	300000
Echo Request	Triangular	-1	200000
Source Bytes	Triangular	-1	100
Port	Triangular	0	10
IP	Triangular	0	15
Target	Triangular	0	0.9
Attack type	Triangular	0	9

The triangular membership function is used in FIS and the range of values for each attribute is shown in Table.13. For example, count may be given as the total range of -1 to 5000000. In that the programmer needs to decide which ranges are to given as low and which range as medium and range values for high, say for count 0 to 350 as low, 351 to 5000000 as high. Similarly the range is set for all attributes and processing is done.

The dataset for FIS classification layer is shown in Table14. The table gives details about the type of attacks, the number of packets for each attack and the number of records used for FIS. The number of records for TCP flood attack is found to be 255, the number of packets for TCP flood is found to be 5516499. Similarly the number of records and packets are given for 8 attack types.

Table.14. Dataset for FIS

Attack type	Number of Packets	Number of records
TCP flood	5516499	255
UDP flood	4629931	273
ICMP flood	2760620	408
Syn flood	5516002	260
Smurf	527745	29
Ping of death	332751	32

Back attack	859716	24
Land attack	4584219	24
Total	24727483	1305

The rules in FIS for classifying different types of attack are,

SAMPLE RULES

1. If (protocol is TCP) and (syn is low) and (port is unequal) and (IP is unequal) and (target is TCP) then (attack1 is TCPflood)
2. If (protocol is TCP) and (syn is high) and (port is unequal) and (IP is unequal) and (target is TCP) then (attack1 is synflood)
3. If (protocol is TCP) and (syn is low) and (port is unequal) and (IP is equal) and (target is TCP) then (attack1 is Backattack)
4. If (protocol is TCP) and (syn is high) and (port is unequal) and (IP is equal) and (target is TCP) then (attack1 is Backattack)
5. If (protocol is TCP) and (syn is low) and (port is equal) and (IP is equal) and (target is TCP) then (attack1 is Landattack)

4.5 ISSUES IN IMPLEMENTATION

Issue1: The main issue in detecting the attacks is due to spoofing, where all type of attacks employ IP spoofing while generating the attack.

Possible solution: MAC address can be used along with IP address and other attributes to detect the spoofing.

Issue2: In real time detection of attacks, the extraction of packet header attributes requires more processing time and the CPU resources may be exhausting.

Possible solution: The program for extracting packet header can be parallelized to reduce the execution time.

Issue3: SVM and Neural networks require more training to classify the type of attack.

Possible solution: The training record should be given for all possible combinations for attack detection in both SVM and neural networks.

Issue4: The weight should be adjusted and updated in such a way that it can fit for all type of packets and suits in training and testing dataset.

Possible solution: The attributes can be represented in binary format and the weight update can be done to fit in training and testing dataset.

5. RESULTS

Our experimental data consist of both attack and normal packets which are merged and displayed using wire shark. A total of 19280354 packets are used for this experiment out of which normal and attack packets are classified. The result comprised of number of packets detected for each attack, percentage of attack detection for each attack and graphical representation of attack detection. The results are taken and shown for SVM, neural networks and FIS separately.

The detection rate of SVM is shown in Table.15. It indicates the total number of records used for normal and attack type, number of records detected in each category and the detection rate for each category. A total of 2284 records are given as input where 2238 records are detected into two categories namely normal which has 1198 records detected and attack which has 1040 detected records.

Table.15. Detection rate of SVM

	Total Number of Records	**Number of records detected**	**Detection rate (%)**
Normal records	1219	1198	98.27
Attack records	1065	1040	97.65
Overall detection rate	2284	2238	97.98

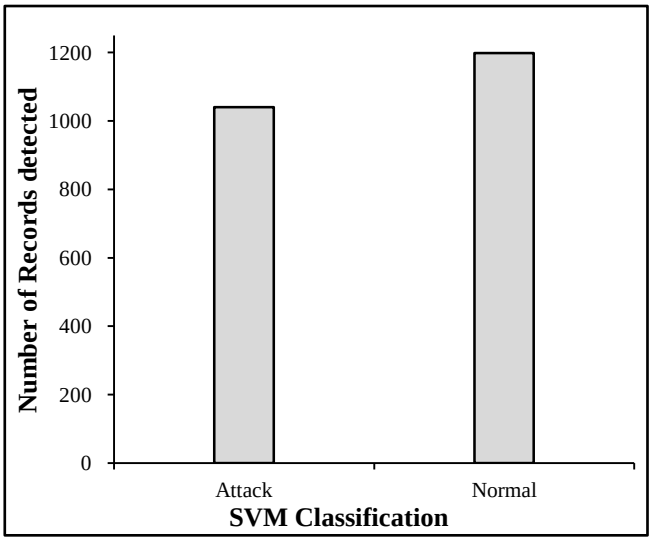

Fig.2. SVM Classification

The graphical representation of SVM classification is shown in Fig.2. The classification is done in terms of number of records. Around 1000 records are classified as attack and around 1200 records are classified as normal.

The detection rate of neural networks classification is shown in Table.16. The detection rate is defined as the ratio of number of attack detected for each type to the total number of actual generated attacks for that type. Using this detection rate the performance of neural networks is measured (i.e.) how far the neural networks can detect the attack for the given testing data set. The detection rate of ICMP flood attack is 98.93, which is maximum and the detection rate of TCP flood attack is 97.09, which is minimum.

Table.16. Detection rate of NN

Attack types	**Total Number of Records**	**Number of records detected**	**Detection rate (%)**
TCP flood	275	267	97.09
UDP flood	295	289	97.96

DOI: 10.21917/ijsc.2014.0097

ICMP flood	470	465	98.93
Overall detection rate	1040	1021	98.17

The graphical representation of neural network classification is shown in Fig.16. It indicates the attack detected and the number of records occurred for each attack. This figure is derived from the tabulation results shown above. From the graph it is known that ICMP flood attack top the list in number of attack records.

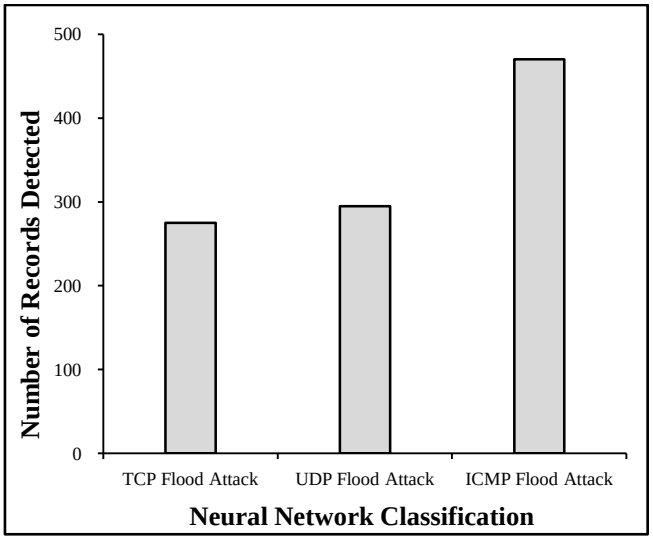

Fig.3. Neural network classification

The detection rate of FIS classification is shown in Table.17. It indicates the detection rate of different attack types, total number of records given for each attack and total number of records detected for each attack. The packets are converted into records by separating the packets between source and destination. The overall detection rate is 97.31 and the detection rate of TCP flood attack is 98.43 which is maximum and the detection rate of land attack is 92.30 which is minimum.

Table.17. Detection rate of FIS

Attacks	Total records	Attack detected (records)	Detection Rate (%)
Land attack	24	22	91.66
Back attack	24	23	95.83
Syn flood attack	260	249	95.76
Smurf attack	29	27	93.10
UDP flood attack	273	267	97.80
TCP flood attack	255	251	98.43
ICMP flood attack	408	401	98.28
Ping of Death	32	30	93.75
Overall Detection Rate			97.31

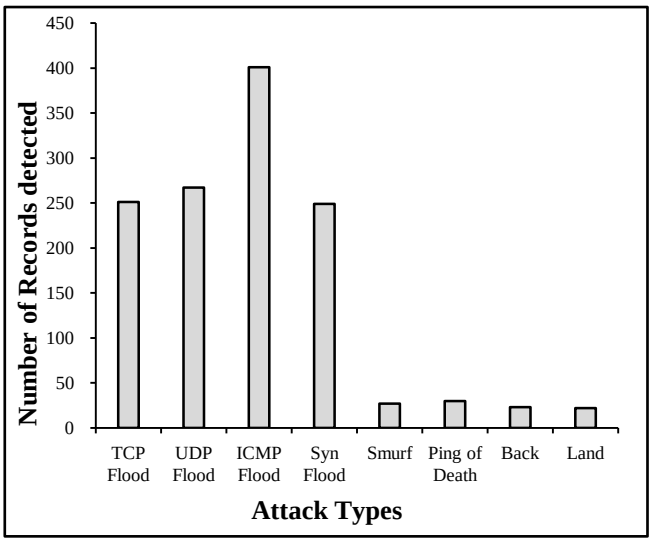

Fig.4. FIS classification

The comparison of FIS and layered approach SVM-NN-FIS is shown in Fig.5. The graph indicates that the detection rate of layered approach is 97.31% and the detection rate of FIS is 95.82%. This shows that layered approach has high detection rate than the existing system FIS.

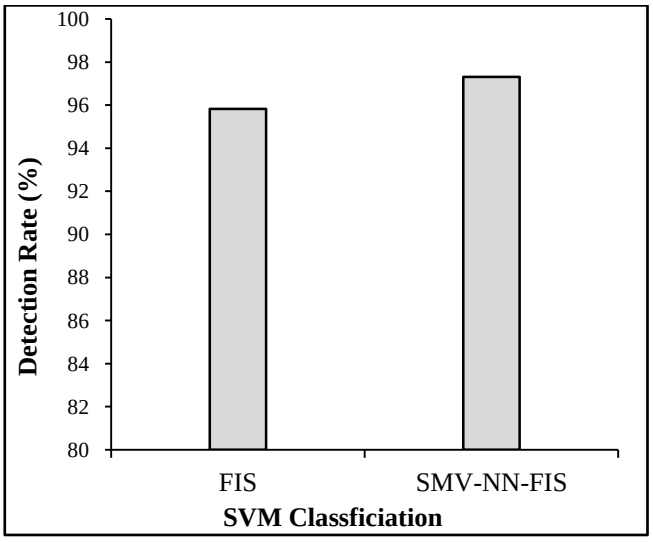

Fig.5. Comparison with existing system

5.1 OUTCOMES

1. A total of 19280354 packets are used as input out of which normal and attack packets are classified. The .csv file and .pcap file is shown in appendix and can be further used for research.
2. The detection rate of SVM is 97.18%, the detection rate of neural networks is 98.17% and the overall detection rate of three layered approach from FIS is 97.31%
3. The real time dataset collected from the network helps in tuning the performance of the classification algorithms to move towards online detection of attacks.
4. The rule based algorithm FIS enhances the precision of the micro classification of attacks and the learning

algorithms SVM and NN enhances the detection of any latest attacks in the DDoS category.

6. CONCLUSION AND FUTURE WORK

The paper real time attack detection based on machine learning techniques and rule based approach involves preprocessing and classification where in preprocessing, the creation of real time dataset is done and in classification there are three layers namely SVM as layer 1 where binary classification is done, neural network classification as layer 2 where major classes of attack type are detected and FIS classification as layer 3 where subclasses of attack type are detected. This layered approach of classification improves the performance. The detection rate of this layered approach is 97.31% and this has high detection rate than the existing system FIS. This work can be extended to different misuse detection type of attacks and it can be combined with some other classifiers to improve the performance.

REFERENCES

[1] Ajit Abraham and Ravi Jain, "Soft computing models for network intrusion detection systems", *Classification and Clustering for Knowledge Discovery, Computational Intelligence*, Vol. 4, pp. 191–207, 2005.

[2] Adel Nadjaran Toosi and Mohsen Kahani, "A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers", *Communication Communications*, Vol. 30, No. 10, pp. 2201 – 2212, 2007.

[3] Aikaterini Mitrokotsa and Christos Douligeris, "Detecting Denial of Service Attacks Using Emergent Self-Organizing Maps", *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology*, pp. 375-380, 2005.

[4] Aida O. Ali, Ahmed I. Saleh and Tamer R. Badawy, "Intelligent Adaptive Intrusion Detection Systems Using Neural Networks", *International Journal of Video & Image Processing and Network Security*, Vol. 10, No. 1, pp. 1-12, 2010.

[5] C. M. Chen, Y. L. Chen and H. C. Lin, "An efficient network intrusion detection", *Computer Communications*, Vol. 33, No. 4, pp. 477-484, 2010.

[6] http://www.mathworks.in/products/neural-network/description4.html.

[7] http://www.mathworks.in/products/fuzzy-logic/description4.html.

[8] Huwaida Tagelsir Elshoush and Izzeldin Mohamed Osman, "Alert correlation in collaborative intelligent intrusion detection systems – A survey", *Applied Soft Computing*, Vol. 11, No. 7, pp. 4349-4365, 2011.

[9] InSeon Y and Ulrich U.N, "An intelligent firewall to detect novel attacks – An integrated approach based on anomaly detection against virus attacks", *SOFtware SEMinar (SOFSEM) – Student Research Forum*, pp. 59–64, 2002.

[10] James Cannady, "Artificial neural networks for misuse detection", *Proceedings of the National Information Systems Security Conference*, pp. 443-456, 1998.

[11] C. Jirapummin, N. Wattanapongsakorn and J. Kanthamanon, "Hybrid neural networks for intrusion detection system", *Proceedings of the International Technical Conference on Circuits/Systems, Computers and Communications*, pp. 928–931, 2002.

[12] V. Katos, "Network intrusion detection: evaluating cluster, discriminant, and logit analysis", *Information Sciences: an International Journal*, Vol. 177, No. 15, pp. 3060–3073, 2007.

[13] R. P. Lippmann and R. K. Cunningham, "Improving Intrusion Detection performance using Keyword selection and Neural Networks", *Computer Networks: The International Journal of Computer and Telecommunications Networking – Special Issue on recent advances in intrusion detection systems*, Vol. 34, No. 4, pp. 597-603, 2000.

[14] J. H. Lee, J. H. Lee, S. G. Sohn, J. H. Ryu and T. M. Chung, "Effective value of decision tree with KDD 99 intrusion detection datasets for intrusion detection system", *Proceedings of the 10th International Conference on Advanced Communication Technology*, pp. 1170-1175, 2008.

[15] N. Liao, S. Tian and T. Wang, "Network forensics based on fuzzy logic and expert system", *Computer Communications*, Vol. 32, No. 17, pp. 1881–1892, 2009.

[16] W. Lee, S. Stolfo and K. Mok, "Mining in a data-flow environment: experience in network intrusion detection", *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 114-124, 1999.

[17] M. Tavallaee, E. 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 Defense Applications*, pp. 1-6, 2009.

[18] M. Moradi and M. Zulkernine, "A neural network based system for intrusion detection and classification of attacks", *Proceedings of the IEEE International Conference on Advances in Intelligent Systems Theory and Applications*, pp. 148–153, 2004.

[19] N. Ngamwitthayanon, N. Wattanapongsakorn and C. Charnsripinyo "Multi-stage network-based intrusion detection system using back propagation neural networks", *Asian International Workshop on Advanced Reliability Modeling*, pp. 609-616, 2008.

[20] N. Ngamwitthayanon, N. Wattanapongsakorn and D.W. Coit, "Investigation of fuzzy adaptive resonance theory in network anomaly intrusion detection", *Proceedings of the International Symposium on Neural Networks: Advances in Neural Networks*, pp. 208-217, 2009.

[21] Z. Pan, S. Chen, G. Hu and D. Zhang, "Hybrid neural network and C4.5 for misuse detection", *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 2463–2467, 2003.

[22] Phurivit Sangkatsanee, Naruemon Wattanapongsakorn and Chalermpol Charnsripinyo, "Practical real-time intrusion detection using machine learning approaches", *Computer Communications*, Vol. 34, No. 18, pp. 2227-2235, 2011.

[23] S. Pukkawanna, V. Visoottiviseth and P. Pongpaibool, "Lightweight detection of DoS attacks", *Proceedings of the IEEE International Conference on Networks*, pp. 77-82, 2007.

[24] Rung-Ching Chen, Kai-Fan Cheng and Chia-Fen Hsieh, "Using Roughset and support vector machine for Intrusion detection system", *International Journal of Network Security and its Applications*, Vol. 1, No. 1, pp. 1-13, 2009.

[25] J. Ryan, M. Lin and R. Miikkulainen, "Intrusion Detection with Neural Networks", *Advances in Neural Information Processing Systems*, pp. 943-949, 1998.

[26] Snehal A. Mulay, P. R. Devale and G.V. Garje, "Intrusion Detection System using Support Vector Machine and Decision Tree", *International Journal of Computer Applications*, Vol. 3, No. 3, pp. 40-43, 2010.

[27] T. Subbulakshmi, S. Mercy Shalinie, C. Suneel Reddy and A. Ramamoorthi, "Detection and Classification of DDoS attacks using Fuzzy Inference System", *Communications in Computer and Information Science*, pp. 242-252, 2010.

[28] T. Subbulakshmi and S. Mercy Shalinie, "Detection and Classification of Intrusions using Machine Learning Algorithms", *European Journal of Scientific Research*, Vol. 47, No. 3, pp. 334-346, 2010.

[29] T. Subbulakshmi and S. Mercy Shalinie, "Misuse Detection Systems using Support Vector Machines", *Proceedings of International Conference on Information Technology and Business Intelligence*, pp. 1-6, 2009.

[30] C. H. Tsang, S. Kwong and H. Wang, "Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection", *Pattern Recognition*, Vol. 40, No. 9, pp. 2373–2391, 2007.

[31] Xuan Dau Hoang, Jiankun Hu and Peter Bertok, "A program-based anomaly intrusion detection scheme using multiple detection engines and fuzzy inference", *Journal of Network and Computer Applications*, Vol. 32, No. 6, pp. 1219-1228, 2009.