OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

A Technique by using Neuro-Fuzzy Inference System for Intrusion Detection and Forensics

Abhishek choudhary[1], Swati Sharma[2], Pooja Gupta[3]
[1]*M .Tech (C.S) (Assistant Professor, Bhagwant University, Ajmer, India)*
[2]*M .Tech (C.S) (Scholar, Bhagwant University, Ajmer, India)*
[3]*M .Tech (C.S) (Scholar, Bhagwant University, Ajmer, India)*
Department of Computer Sciences and Engineering Bhagwant University, Ajmer, India

ABSTRACT—*This paper proposes a technique uses decision tree for dataset and to find the basic parameters for creating the membership functions of fuzzy inference system for Intrusion Detection and Forensics. Approach of generating rules using clustering methods is limited to the problems of clustering techniques. To trait to solve this problem, several solutions have been proposed using various Techniques. One such Technique is proposed to be applied here, for an analysis to Intrusion Detection and Forensics. . Fuzzy Inference approach and decision algorithms are investigated in this work. Decision tree is used to identify the parameters to create the fuzzy inference system. Fuzzy inference system used as an input and the final ANFIS structure is generated for intrusion detection and forensics. The experiments and evaluations of the proposed method were done with NSL-KDD intrusion detection dataset.*

Keywords—*Intrusion detection, Forensics, Technique decision tree, NSL-KDD intrusion detection dataset, decision tree algorithm, Fuzzy inference system, ANFIS approach.*

I. Introduction

An Intrusion Detection System [1] [2] is a computer program that attempts to perform ID By either misuse or anomaly detection, or a combination of techniques. Intrusion detection is the process of identifying network activity that can lead to the compromise of a security policy. IDS's are classified in many different ways, including active and passive, network-based and host-based, and knowledge-based and behaviour-based. Intrusion Detection System is a most essential part of the security infrastructure for the network connected to the internet, because of the numerous way to comprise the stability and security of the network .Intrusion Detection System [1] [2] can be used to monitor and analysis of network for unauthorized activity.

Forensic science [3] is based on a methodology composed by a group of stages, being the analysis one of them. Analysis is responsible to determine when a data constitutes evidence; and as a consequence it can be presented to a court. When the amount of data in a network is small, its analysis is relatively simple, but when it is huge the data analysis becomes a challenge for the forensics expert

Decision tree [4] as a predictive model which maps observations about an item to conclusions about the item's target value. Decision tree is used to data sheet to find suitable parameters. This technique is applied to generate the rules for fuzzy expert system. This approach proves to be better than the traditional approach of generating rules for fuzzy expert system.

Neuro-Fuzzy Inference System [5] combines the neural network with the fuzzy system, which Implements the fuzzy inference system under the framework of neural network. Neuro-Fuzzy Inference System has the ability to construct models solely based on the target system sample data.

The objective of this paper is to generate fuzzy rules for fuzzy expert system using a technique called decision tree [4] for suitable parameters as the input of fuzzy interference System. The proposed system has been testing using the NSL-KDD dataset [6] to find out the best suitable parameters. The NSL-KDD data set suggested to solve some of the inherent problems of the KDDCUP'99 [7] data set. KDDCUP'99 is the mostly widely used data set for Anomaly detection. But Tavallaee et al conducted a statistical analysis on this data set and found two important issues that greatly affected the performance of evaluated systems, and results in a very poor evaluation of anomaly detection approaches. To solve these issues, they proposed a new data set, NSL-KDD, which consists of selected records of the complete KDD data set The NSL-KDD dataset which consist of selected records of the complete KDD dataset.

The subsequent parts of this paper are organized as follows: At first, in section 2, the NSL-KDD dataset on which experiments are conducted briefly reviewed. Next, at section 3 and 4, the proposed system is

explained and experimental results are discussed respectively. Finally, section 5 makes some concluding remarks and proposes further areas for research.

II. NSL-KDD Dataset

The NSL-KDD data includes 41 features and 5classes that are normal and 4 types of attacks: Dos, Probe, R2L, and U2R. Denial of Service Attack (DoS) is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine. Probing Attack is an attempt to gather information about a network of computers for the apparent purpose of circumventing

Its security controls. User to Root Attack (U2R) is a class of exploit in which the attacker starts out with access to a normal user account on the system (perhaps gained by sniffing passwords, a dictionary attack, or social engineering) and is able to exploit some vulnerability to gain root access to the system. Remote to Local Attack (R2L) occurs [8] when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine.41 attributes are consisted three features: Basic features, Content features, and Traffic features. Table 1 shows Features name and type of features.

TABLE I: SHOWS FEATURES NAME AND TYPE OF THE FEATURES

Type	***Features***
Nominal	Protocol type(2), Service(3), Flag(4)
Binary	Land(7), logged in(12), root shell(14), su_attempted(15), is_host_login(21),, is_guest_login(22)
Numeric	Duration(1), src_bytes(5), dst_bytes(6), wrong fragment(8), urgent(9), hot(10), num_failed_logins(11), num_compromised(13), num_root(16), num_file_creations(17), num_shells(18), num_access_files(19), num_outbound_cmds(20), count(23) srv_count(24), serror_rate(25), srv_serror_rate(26), rerror_rate(27), srv_rerror_rate(28), same_srv_rate(29) diff_srv_rate(30), srv_diff_host_rate(31), dst_host_count(32), dst_host_srv_count(33), dst_host_same_srv_rate(34), dst_host_diff_srv_rate(35), dst_host_same_src_port_rate(36), dst_host_srv_diff_host_rate(37), dst_host_serror_rate(38), dst_host_srv_serror_rate(39), dst_host_rerror_rate(40), dst_host_srv_rerror_rate(41)

III. Proposed System

The proposed system is discussed in details in this section. First, the proposed frame work is explained. Then, data selected from KDD for training the system, are introduced. Afterward, layers of proposed framework are presented in more details.

- ***proposed framework***

The integration of neural and fuzzy systems leads to a symbiotic relationship in which fuzzy systems provide a powerful framework for expert knowledge representation, while neural networks provide learning capabilities and exceptional suitability for computationally efficient hardware implementations. In our approach

we are proposing a combination of these two standalone techniques as a t to efficiently detect the anomalous behavior. The proposed framework is as follows:

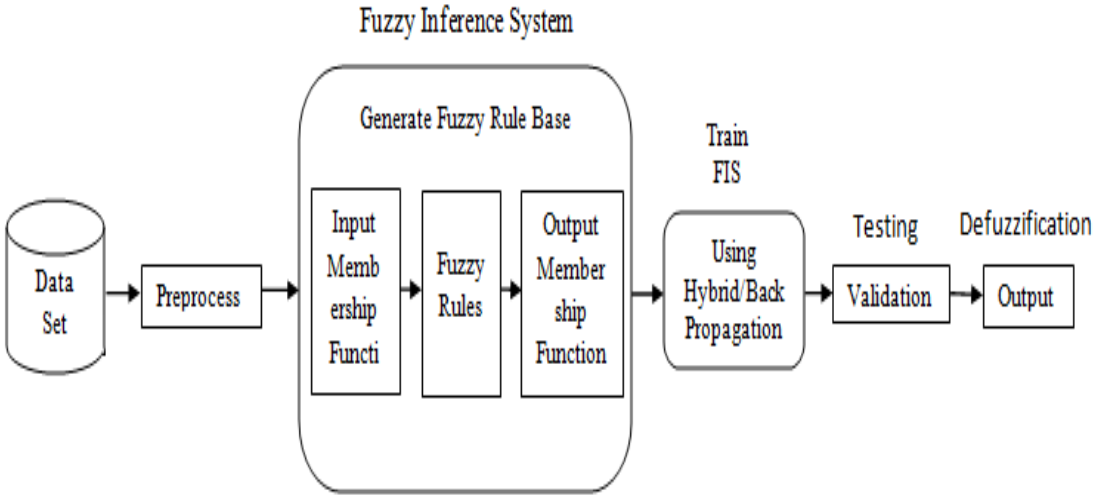

Figure 1: Proposed work structure

NSL-KDD dataset is first loaded to Neuro-Fuzzy tool. Some of the attributes are removed and then all the attributes are normalized in the range of 0-1 so that the fuzzy decisions can be made. After that rules for fuzzy expert system are generated using association rule mining and decision trees. We used WEKA tool to generate decision trees for dataset and to find the basic parameters for creating the membership functions of fuzzy inference system. Based on these rules we have created the fuzzy inference system that is used as an input to neuro-fuzzy system and the final NFIS structure is generated. Neuro-Fuzzy system results in accuracy in detecting the probe attack.

3.2 Data Sources

NSL-KDD consists of selected records of the complete KDD data set and is publicly available for researchers. In each connection there are 41 attributes describing different features of the connection and a label assigned to each either as an attack type or as normal. Table II shows all the 41 attributes present in the NSL-KDD dataset with their description as well.

TABLE II: LIST OF NSL-KDD DATASET FEATURES WITH THEIR DESCRIPTION

No	***Feature***	***Description***
1	Duration	Duration of the connection
2	protocol_type	Connection protocol (e.g. TCP,UDP, ICMP)
3	service	Destination service
4	flag	Status flag of the connection
5	src_byte	Bytes sent from source to destination
6	dst_byte	Bytes sent from destination to source
7	land	1 if connection is from/to the same host/port; 0 otherwise
8	wrong_fragment	Number of wrong fragments
9	urgent	Number of urgent packets
10	hot	Number of "hot" indicators
11	num_failed_login	Number of failed logins
12	logged in	1 if successfully logged in; 0 otherwise
13	num_compromised	Number of "compromised" conditions

14	root_shell	1 if root shell is obtained; 0 otherwise
15	su_attempted	1 if "su root" command attempted; 0 otherwise
16	num_root	Number of "root" accesses
17	num_file_creation	Number of file creation operations
18	num_shells	Number of shell prompts
19	num_access_file	Number of operations on access control files
20	num_outbound cmds	Number of outbound commands in a ftp session
21	is_host_login	1 if login belongs to the "hot" list; 0 otherwise
22	is_gust_login	1 if the login is the "guest" login; 0 otherwise
23	count	Number of connections to the same host as the current connection in the past 2 seconds
24	srv_count	Number of connections to the same service as the current connection in the past two seconds
25	serror_rate	% of connections that have "SYN" errors
26	srv_serror_rate	% of connections that have "SYN" errors
27	rerror_rate	% of connections that have REJ errors
28	srv_rerror_rate	% of connections that have REJ errors
29	same_srv_rate	% of connections to the same service
30	diff_srv_rate	% of connections to different services
31	srv_diff_host_rate	% of connections to different hosts
32	dst_host_count	Count of connections having the same destination host
33	dst_host_srv_count	Count of connections having the same destination host and using the same service
34	dst_host_same_srv_rate	% of connections having the same destination host and using the same service
35	dst_host_diff_srv_rate	% of different services on the current host
36	dst_host_same_src_port_rate	% of connections to the current host having the same src port
37	dst_host_srv_diff_host_rate	% of connections to the same service coming from different hosts
38	dst_host_serror_rate	% of connections to the current host that have an S0 error
39	dst_host_srv_serro_rate	% of connections to the current host and specified service that have an SO error

40	dst_host_rerror_rate	% of connections to the current host that have an RST error
41	dst_host_srv_rerror_rate	% of connections to the current host and specified service that have an RST error

The training dataset is made up of 21 different attacks out of the 37 present in the test dataset. The known attack types are those present in the training dataset while the novel attacks are the additional attacks in the test dataset i.e. not available in the training datasets. The attack types are grouped into four categories: DoS, Probe, U2R and R2L. The training dataset consisted of 25193 instances among which 13449 are normal and 11744 are attack type whereas the testing dataset consist of 2152 normal and 9698 attack types. Table III represents the attacks presented in training dataset which is 20% of the complete NSL-KDD intrusion detection dataset.

TABLE III: ATTACKS IN TRAINIG DATASET

Training Dataset (20 %)	**Attack – Type (21)**
DoS	Back, Land, Neptune, Pod, Smurf, teardrop
Probe	Satan, Ipsweep, Nmap, Portsweep,
R2L	Guess_Password, Ftp_write,Imap, Phf,Multihop, Warezmaster, Warezclient, Spy
U2R	Buffer_overflow, Loadmodule, Rootkit

Fig 2: Number of Instances in training dataset

TABLE IV: ATTACKS IN TESTIING DATASET

***Testing Dataset* (20 %)**	***Attack- Type (37)***
DoS	Back, Land, Neptune, Pod, Smurf, teardrop, Mailbomb, Processtable, Udpstorm, Apache2, Worm,
Probe	Satan, Ipsweep, Nmap, Portsweep, Mscan, Saint
R2L	Guess_Password, Ftp_write, Imap, Phf, Multihop, Warezmaster, Xlock, xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named
U2R	Buffer_overflow, Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps

Fig 3: Number of Instances in Testing Dataset

Each record in NSL-KDD dataset is a network linking record. Each link consists of 41 attribute properties containing 3 symbolic variables (protocol type, service and flag). In order not to affect the clustering result, the attribute values need to be pre-treated. Firstly, these three symbolic attributes are removed and then all the remaining numerical attributes (39 attributes) are normalized in the range of [0 1] so that the attributes having higher values do not dominate over attributes with low values. The standard deviation transform is shown as follows:

(1)

The normalized transform is as follows:

(2)

Table II consists of all 42 attributes of the NSL-KDD dataset. Three of the symbolic data attributes (protocol type, service and flag) are removed as non-contributing.

- ***Decision tree algorithm***

We used WEKA 3.7 a machine learning tool [10], to generate decision trees for dataset and to find the basic parameters for creating the membership functions of fuzzy inference system. Several decision tree based techniques are applied to the NSL-KDD dataset to find out the best Suitable parameters. Figure 4 shows a simple output of J48 decision tree algorithm.

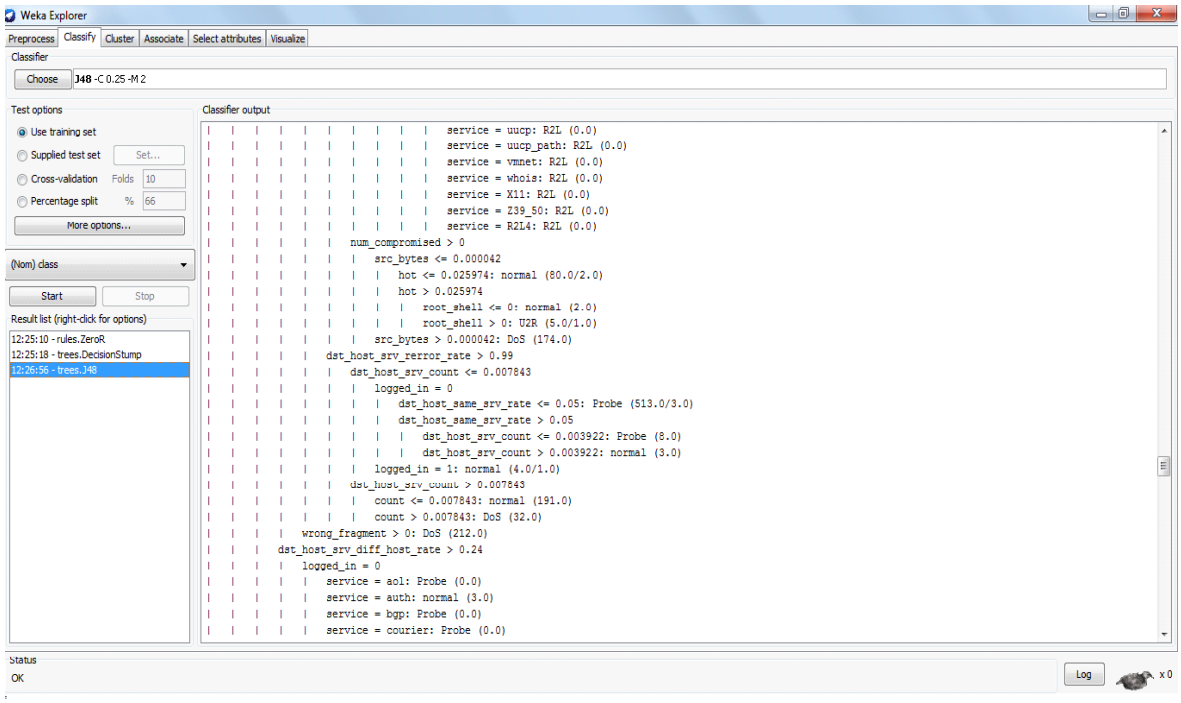

Figure 4. Shows a simple output of J48 decision tree algorithm

Here, we are combining both the approaches of association rule mining [for selecting the best attributes together and decision tree for identifying the best parameters together to create the rules for fuzzy expert system. Decision Trees in WEKA tool is used to identify the required parameters. The minimum and maximum is set to the range 0-1. Each attribute is distributed over a range of low, mid (medium) and high. Range of parameters is shown in table V.

TABLE V: MEMBERSHIP PARAMETERS FOR PROBE ATTACKS

Attributes	***Min***	***Low***	***Mid***	***High***	***Max***
source_bytes	0	0.2	0.34	0.7	1
logged_in	0	-	-	-	1
serror_rate	0	0.295	0.453	0.89	1
srv_serror_rate	0	0.294	0.454	0.81	1
rerror_rate	0	0.131	0.335	0.682	1
same_srv_rate	0	0.126	0.44	0.643	1
diff_srv_rate	0	0.069	0.483	0.732	1
dst_host_srv_rate	0	0	0.426	0.436	1
dst_host_serror_rate	0	0.296	0.452	0.691	1

- ***Fuzzy Inference System***

Fuzzy Inference Systems (FIS) have the ability to make use of knowledge expressed in the form of linguistic rules, thus they offer the possibility of implementing the expert human knowledge and experience. Neuro-Fuzzy Inference System (NFIS) models have become one of the major areas of interest as it combines the beneficial features of both neural networks and fuzzy systems. Based on these rules we have created the fuzzy inference system that is used as an input to neuro-fuzzy system. Figure 5 shows the generated FIS

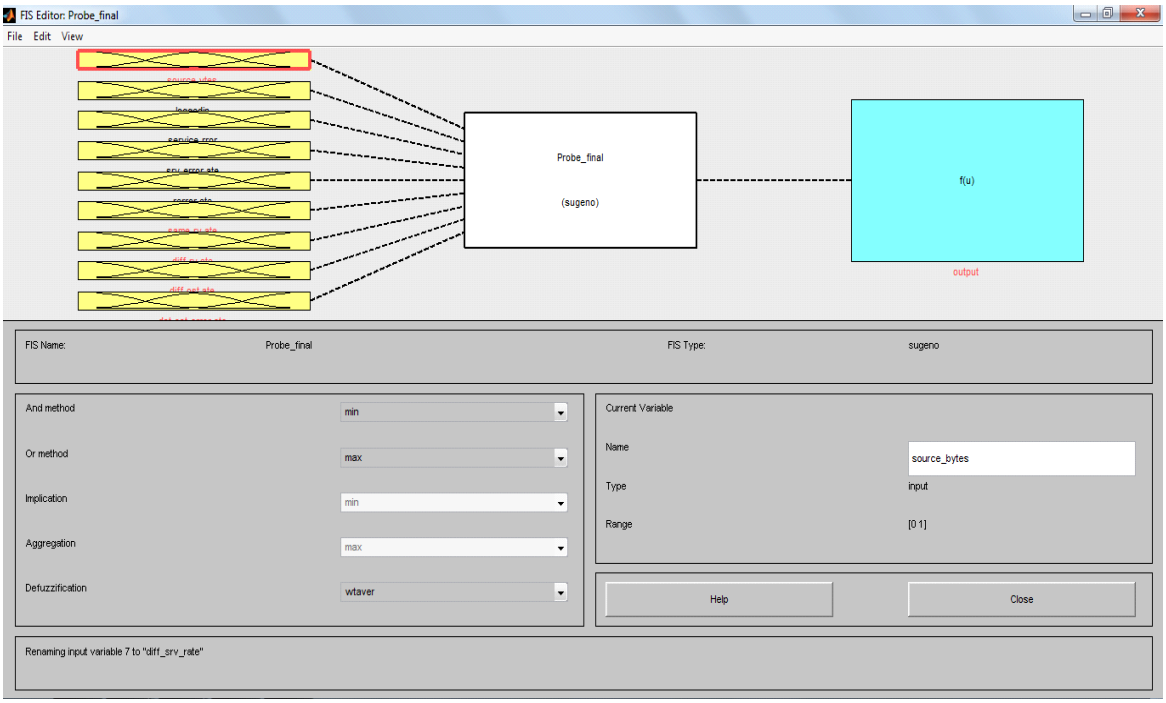

Figure 5: Fuzzy inference system

Membership functions are generated using trapezoidal functions and one Gaussian function is used for Boolean type of attribute (i.e. logged in). Nine inputs having three membership functions except logged in (i.e. having 2 membership functions) generated for the probe attack. Figure 6 shows the generated fuzzy membership functions

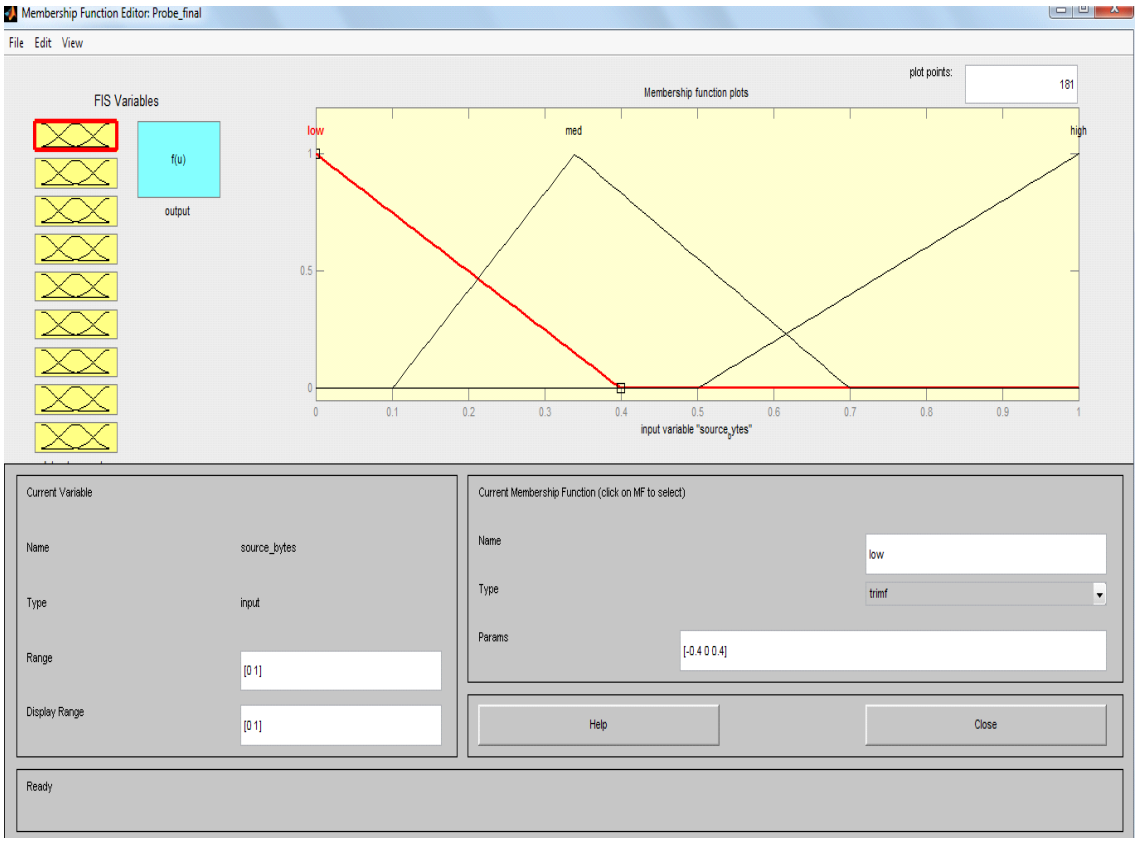

Figure 6: Membership functions for Probe Attack

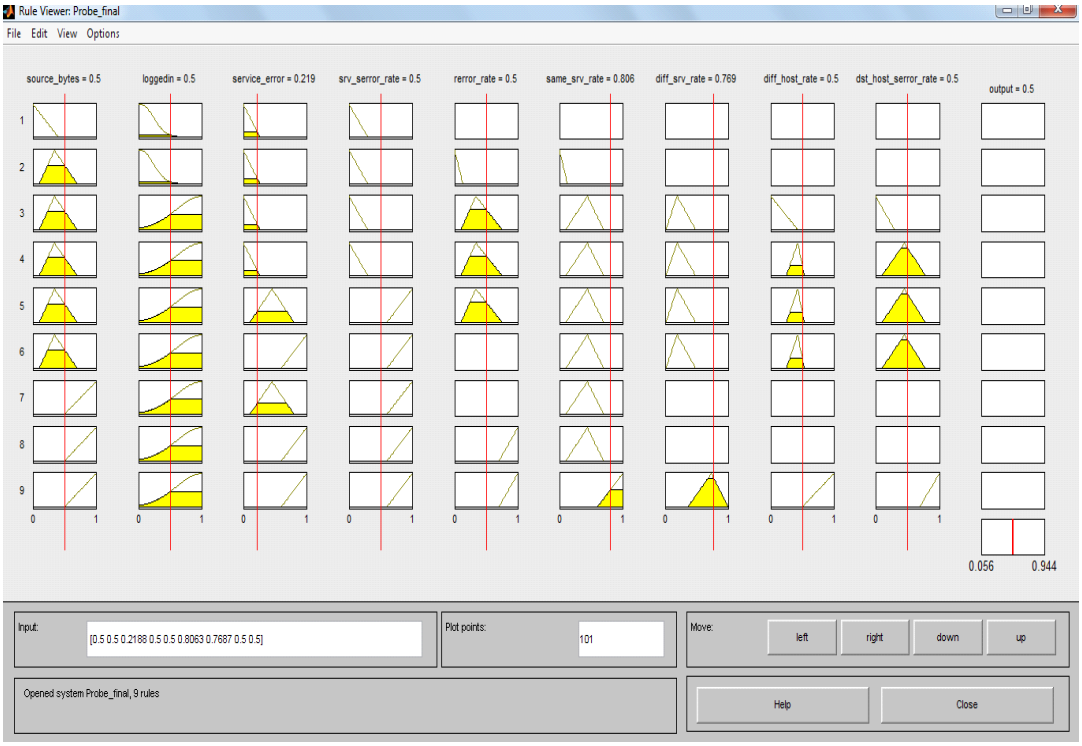

Figure 7: Fuzzy inference system output

Figure 7 shows the output of the fuzzy inference system. This system can be adjusted according to the values of the input attributes. Fuzzy Inference system shows an accuracy of 94.4 %. The last attribute shows the defuzzified output of the fuzzy system. Rules generated for fuzzy inference system is shown in figure 8. These rules are used to makes decision for neuro-fuzzy system. The final outcome is the production of these rules.

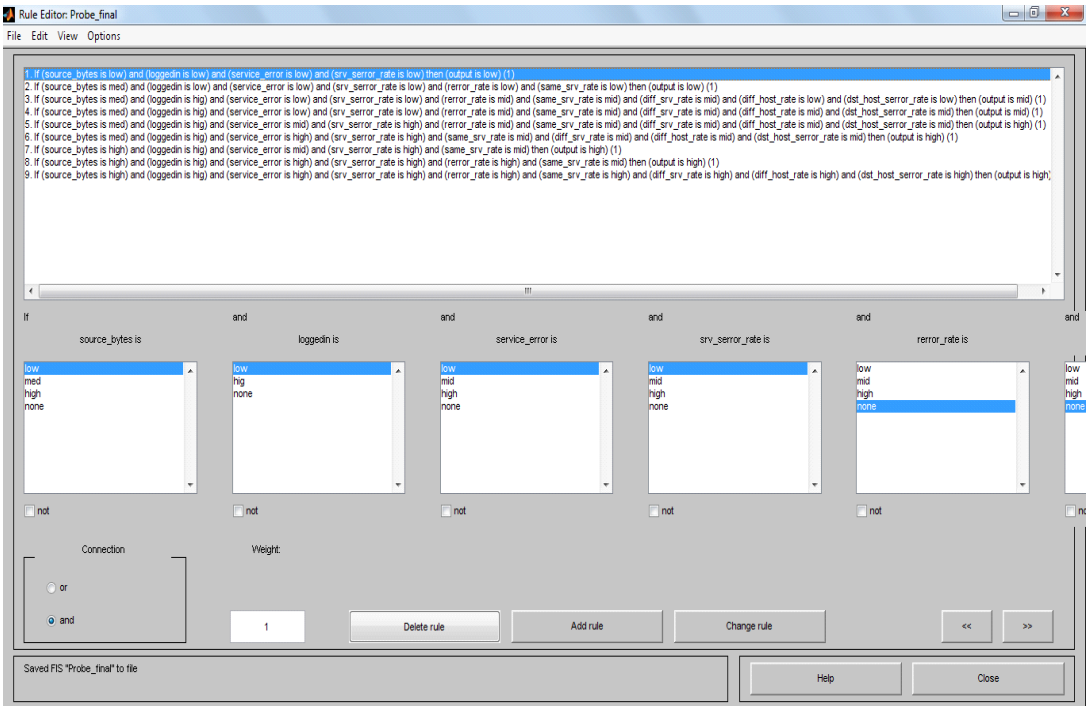

Figure 8: Rules in fuzzy inference system

Now this fuzzy inference system is loaded to neuro-fuzzy toolbox as an input and the final ANFIS. Structure is generated as shown in figure 9 below.

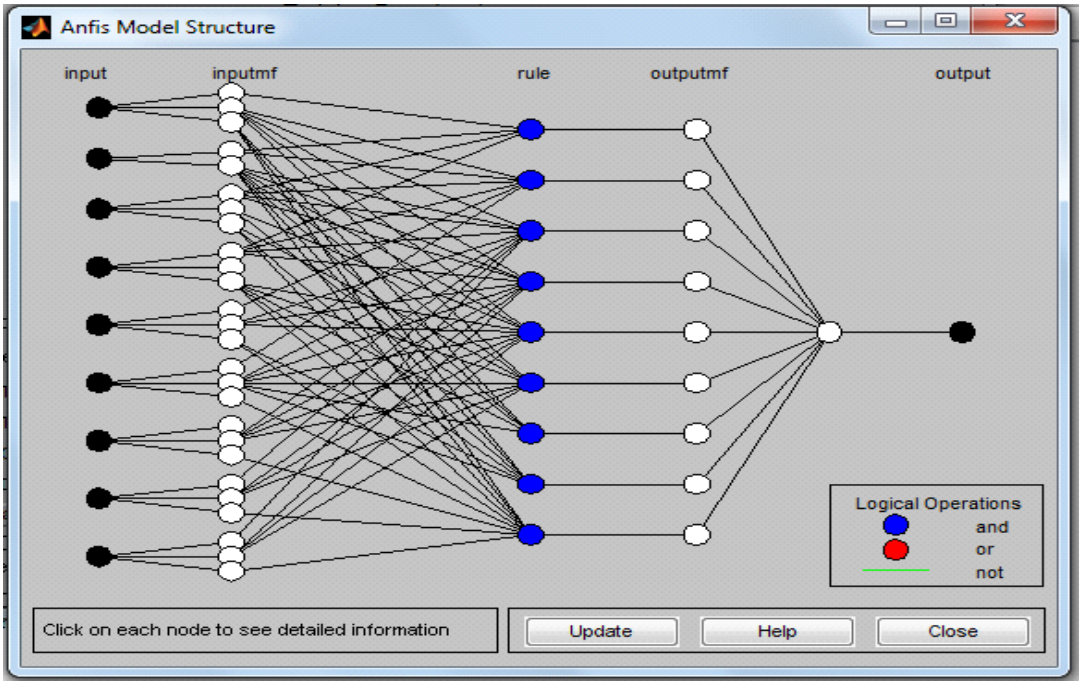

Figure 9: ANFIS structure

IV. Results

ANFIS tool output is evaluated on the basis of accuracy in predicting the attacks clearly. Results below show the outcome of neuro-fuzzy approach [11].

The output of neuro-fuzzy toolbox [12]. Is the weighted neurons that are defuzzified using centroid defuzzification method Neuro-Fuzzy system results in 99.68 % accuracy in detecting the probe attack. A slight comparison to previous literature is given as follows:

TABLE VI: COMPARISON RESULTS

Approach in [paper]	***Probe Detection rate (%)***
[84]	73.20
[85]	100
[86]	72.8
[87]	99
ANFIS Approach	99.68

V. Conclusion

In this paper, a technique uses decision tree for dataset and to find the basic parameters for creating the membership functions of fuzzy inference system for Intrusion Detection and Forensics was successfully demonstrated its usefulness on the training and testing subset NSL-KDD dataset. The NFIS network was used as a Systems. NFIS is Capable of producing fuzzy rules without the aid of human experts. A fuzzy decision-making engine was developed to make the system more powerful for attack detection, using the fuzzy inference approach. At last, this paper proposed a technique to use Decision tree algorithms to optimize the decision-making engine. The decision tree classifier will be evaluated with the NSL-KDD dataset to detect attacks on four attack categories: Dos, Probe, R2L, and U2R.Experimentation results showed the proposed method is effective in detecting intrusions in computer networks. Our future work will focus investigating how to improve the performance of neuro-fuzzy logic to meet the requirements of computer security and forensics. Also, the work will be continued to Developing an automotive system that can generate rule for its own without any interaction of human experts would be a major challenge to researchers.

REFERENCES

[1] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," *Network, IEEE*, vol. 8, no. 3, pp. 26-41, 1994.

[2] R. Barber, "The Evolution of Intrusion Detection Systems-The Next Step," *Computer & Security*, vol. 20, no. 2, pp. 132-145, 2001

[3] G. Palmer, "A road map for digital forensic research," *First digital forensic research workshop (DFRWS'01)*, pp. 27-30, 2001.

[4] Z. Liu and D. Feng, "Incremental Fuzzy Decision Tree-Based Network Forensic System," in *Computational Intelligence and Security*. vol. 3802: Springer Berlin Heidelberg, 2005, pp. 995-1002.

[5] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *Systems, Man and Cybernetics, IEEE Transactions*, vol. 23, no. 3, pp. 665-685, 1993.

[6] M. Sabhnani and G. Serpen, "Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context. [Available At] http://www.eecs.utoledo.edu/~serpen/.

[7] KDD CUP 1999 Intrusion detection dataset: http://kdd.ios.uci.edu/databases/kddcup99/kddcup99.html

[8] P. G. Jeya, M. Ravichandrajan, and C. S. Ravichandran, "Efficient Classifier for R2L and U2R Attacks," *International Journal 6 Computer Science Applications*, vol. 45, no. 21, pp. 28-32, 201

[9] A. Al-Hmouz, S. Jun, R. Al-Hmouz, and Y. Jun, "Modeling and Simulation of an Adaptive Neuro-Fuzzy Inference System (ANFIS) for Mobile Learning," *IEEE Transactions on Learning Technologies*, vol. 5, no. 3, pp. 226-237, 2012.

[10] "WEKA – Data Mining Machine Learning Software," *[Available Online]*. *http://www.cs.waikato.ac.nz/ml/weka/*.

[11] P. Vuorimaa, T. Jukarainen, and E. Karpanoja, "A neuro-fuzzy system for chemical agent detection," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 4, pp. 415-424, 1995.

[12] Fuzzy Logic Toolbox, [Available At] http://www.mathworks.in/help/fuzzy/index.html