A Clustering and Fuzzy Logic Based Intrusion Detection System

Macdonald Mukosera, Dr G Venkata Rami Reddy

Abstract— As intrusion detection systems are vital and critical components in the field of computer and network security and they form a secondary line of defense, that is they come into play after preventive measures like firewalls fail, it is important to keep on researching and continuing to seek ways for improving and enhancing these systems. In this paper we contribute to this field by proposing an intrusion detection system that uses fuzzy logic and clustering techniques. To test this proposed system, we build the knowledge of the intrusion detection system by analyzing the NSL-KDD dataset and clustered the dataset into smaller units allowing us to discover fuzzy rules for the fuzzy inference systems. The proposed system consists of multiple fuzzy inference systems with each FIS handling a particular service under a particular protocol. The experimental work was done in matlab 2014a and the results from the simulations of the proposed system shows that the proposed system have significant improvements in time complexity and detection rate yielding a low false alarm rate

.

Index Terms— Fuzzy inference system (FIS), fuzzy logic, simulation, Intrusion detection system (IDS), anomaly

.

1 INTRODUCTION

INTRUSION detection system is a vital component in network and host security these days and they usually come into play when preventive security measures like firewalls fail to block an intrusion. Intrusion detection [1] is a field that focuses specifically on the detection of computer intrusions. Computer intrusions are activities which are unauthorized or unintended by computer or network administrators. These intrusions may be threat to systems availability and operation or to the integrity and confidentiality of the data housed in the systems.

Intrusion detection uses information gathered from standard computer logs and computer audit trails that are gathered routinely by host computers, communication routers or switches to detect and identify intrusions into a computer system. The diversity on the information that can be used by an intrusion detection system and the different approaches on the techniques of detection of intrusion brings about the differences these systems and two categories of these systems are identified which are misuse or anomaly detection.

1.1 Misuse Detection

Misuse detection or signature based detection techniques approach the intrusion detection problem by attempting to match the events of the traffic under observation with the events of known and pre-stored pattern of events which is characterized as an attack. These events will have been known and categorized as attacks, then collected and stored in the systems knowledge base and can also be referred to as signatures. Upon matching if the traffic under analysis match that of the known attack signatures, that traffic is considered as an attack to the system and appropriate defined measures are taken, if no match is there then the traffic is considered as harmless and legit traffic. These systems can be easily designed to meet a zero false alarm rate on detecting known attacks if enough prior knowledge to the attacks which the system is under threat is known. The major disadvantage of these kind of system is that in a real operational environment, it is impossible to exhaust all attacks and capture them as signatures as new forms of attacks are frequently launched, hence the system is poor in detecting unknown attacks that is any new attack which is not stored in the signature will gain its way to its target in the system.

1.2 Anomaly Detection

Anomaly detection system focuses on building a system with a behavior that is considered as normal. The detection of unknown attacks is best accomplished using anomaly detection where any network access that deviates from the normal behavior is considered as an intrusion. The main challenge here is any change in the network access that is not registered in the system will be considered as an intrusion even in case of change of network behavior due to other reasons which are not attacks. This results in chances of obtaining a high false alarm rate.

In this paper we focus on the category of anomaly detection systems. We aim to reduce the false alarm rate that these system face through the use of a combination of clustering and fuzzy logic techniques in the systems data preprocessing and the rule generation of the IDS engine. Clustering technique will be used to separate the data into smaller clusters such that the data can be analyzed and processed in smaller units, hence there will be a high chance to exhaust and generate fuzzy rules from the data. We are going to use the NSL-KDD dataset for training and testing of the system.

The rest of this paper is organized as follows: Section 1.3 and 1.4 describes the dataset and fuzzy logic concept respectively, section 2 is about the related work, section 3 briefly lists the challenges in IDS, section 4 focuses on the proposed system, section 5 is about the experimental work, section 6 focuses on experimental work, in section 7 we conclude our work, and then acknowledgments and references are the last parts of the paper.

1.3 NSL-KDD Dataset

NSL-KDD [13] is a data set that was produced as a result of KDD CUP dataset [12] refinement. The researchers aimed to solve some of the problems of the KDD'99 data set. The NSL-KDD data set still suffers from some of the problems like redundancy and too much amount of records to be tested in a network evaluation and may not be a noble representative of existing real networks. But however it still can be applied as an effective benchmark data set to help researchers compare and build different intrusion detection methods. The total number of records in the NSL-KDD train and test sets are reasonable and easy to work with.

1.4 Fuzzy Logic

We employ fuzzy logic to build inference systems that models the normal network behavior. The fuzzy rules will be derived from a careful analysis of data from the clusters obtained after clustering the NSL-KDD dataset. Fuzzy logic can be described as [3] a multi valued logic that allows intermediate values to be defined between conventional evaluations like yes/no, true/false. Notions like warm, cold or very cold can be formulated mathematically and processed by computers

The concept of fuzzy logic involves linguistic variables, fuzzy sets and membership functions. A linguistic variable is a variable that can be defined by a fuzzy set, for example linguistic variable src_bytes can be defined by a fuzzy set low, average or high. A fuzzy set includes the possibility of elements belonging partially to one or more set. The input space where all elements are drawn from is referred to as the universe of discourse (UOD)

If Z is the UOD and its elements are denoted by z, then fuzzy set S in Z is defined as the set of ordered pairs

$$S = \{z, U_S(z) \mid z \varepsilon Z\} \quad (1)$$

Where $U_s(z)$is called a membership function of z in S. The membership function is a curve that maps each element in the universe of discourse (Z) to a fuzzy set by a membership value between 0 and 1

The fuzzy rules form the knowledge part and a fuzzy rule takes the form:

If *x1* is '*a1*' AND if *x2* id '*a2*' then *y1* is' *b1*'

Where *a1*, *a2* and *b1* are linguistic values defined by fuzzy sets on the ranges *x1, x2* and *y1* respectively.The If part of the rule is called the antecedent whilst the then part is called the consequent. An example of a fuzzy rule can be:

If *src_bytes* is '*averag*'e **AND** *dst_bytes* is '*low*' AND *count* is' *low*' then intrusion is '*normal*'

In the above rule *average*, *low* and *norma*l are represented by the degree of membership of their respective fuzzy sets

Fuzzy inference system [3] maps a set of inputs to output using fuzzy logic that is the if-then rules described above which are persisted in a rule database. In this paper we will implement the FIS called the Mamdani type. This type of FIS uses sets of fuzzified variables to describe the output variable and finally deffuzification is used to covert back the output to linguistic variables that can be used for conclusion.

2 Related Work

A lot of work has already been done in this field of intrusion detection and here we look at some of the research work done by other researchers. Kapil Wankhade, Sadia Patka and Ravindra Thool [4] focused on a hybrid approach to tackle the intrusion detection system. Their approach is based on the clustering data mining techniques. They propose a method for clustering analysis driven by objectives of improving the detection rate and decreasing the false alarm rate of the IDS. Their proposal was a hybrid data mining approach encompassing feature selection, filtering, clustering, divide and merge and clustering ensemble. Since the main set back of the K means algorithm is coming up with the appropriate number of clusters and choosing the initial clustering centers, they proposed a modification of the k means algorithm and came up with method for calculating the number of the cluster centers and choosing the appropriate initial cluster centers.

Saeed Khazaee and Maryam Sharifi Rad [5] employed fuzzy c means algorithm to improve intrusion detection performance. They performed sampling, normalization-conversion and feature selection. The samples which have inappropriate membership degrees in all of the clusters were removed from the main dataset. This was done as a pre process of the intrusion detection dataset, remaining with refined dataset hence improving performance. After the proposed preprocessing and splitting of appropriate and inappropriate data, Fuzzy-ARTMAP neural network was applied for learning and they came up with a misuse based Intrusion detection system. The Intrusion detection system was evaluated using KDDCUP 99 dataset and showed significant improvement in performance.

Bharanidharan Shanmugam and Norbik Bashah Idris [6] proposed a hybrid model based on improved fuzzy and data mining techniques, which can detect both misuse and anomaly attacks. They reduced the amount of data retained for processing by performing attribute reduction hence improving detection rate. As adding knowledge to the ids is too laborious process, their research aimed on automating the rule generation process. But however this makes work easy but gives higher chances of missing some important rules. They achieved it by using the improved Kuok fuzzy data mining algorithm, which in turn a modified version of APRIORI algorithm, for implementing fuzzy rules, and they constructed if-then rules that reflect common ways of describing network threats.

Baniasadi and others [10] proposed an approach to predict attacker's aims using a combination of fuzzy logic and description logics. In their work they came up with fuzzy description logic for representing a complex model which shows the degree of relation of attacks which enables prediction of attacks and alerts the network administrator hence improving intrusion detection. A review of many types of software archi-

tecture in intrusion detection systems [11] was carried out and some basic components of an IDS were highlighted whilst the importance of the architecture in any software system specifically IDS was strongly emphasized. The main operations of IDS namely Input Data Collection and Preprocessing, Training and Detection were also highlighted in the work.

3 CHALLENGES

- Attack alerts are generated by IDS but reacting and shutting down network connection can end up being to costly which results in the ids system itself introducing more problems and instability to normal system operation.
- It have to deal with large amounts of traffic and high speed traffic up to Gbps.
- The need to continuously adjust to the changing nature of attacks.

4 PROPOSED SYSTEM

In a bid to increase efficiency and accuracy in classification, we propose a system which uses fuzzy logic techniques. The systems components are packet sniffer, Fuzzy inference system controller, Fuzzy inference systems, decision evaluator engine and a database for persisting the intrusions for audit purposes. The training dataset will be reduced dimensionality and then normal records only selected and then separated into appropriate clusters based on the protocol and service. Reducing dimension improves the system performance because there will be few data items to process whilst we aim to maintain a very low misclassification rate. The clusters will resemble different classes of normal traffic.

4.1 Packet Sniffer

.This is a software tool which will be responsible of capturing packets from the internet traffic. The tool will capture the data and prepares it into a format ready for use by the systems components. A wide variety of readymade tools can be found and embedded into the system. There are open source tools available for use like J Pcap or licensed tools are there also. So the packet sniffer is the component that makes the data available to the system for processing.

4.2 Traffic Router

This is a software component with functions that include selecting the required attributes to be processed by the fuzzy inference system. This function will examine the network packet and determine the protocol type and service type to which that packet is destined to. After examining the packet, this function will then forward the protocol to the appropriate fuzzy inference system which is well designed and dedicated to handle traffic of that protocol and service.

4.3 Fuzzy Inference Systems

These form the knowledge of the Intrusion detection system. They consist of fuzzy rules which will be used to determine whether the traffic is an intrusion or legit. We chose this technique because fuzzy logic is flexible and it makes the system easy to manage and extend or add more functionality without starting from scratch again by simply adding more rules or adding another FIS component and also recycling knowledge from experts rather than training the system from scratch.

4.4 Decision Evaluation Engine

Each and every fuzzy inference system after processing the data will forward the data to the decision engine. This module will contain functions to evaluate the decision and alert the system administrator whether if an attack is reported by the system. The function can also be equipped with taking immediate necessary response in case of some attacks. Fig. 1 below summarizes the proposed system and shows how the components are arranged and how the communicate with each other.

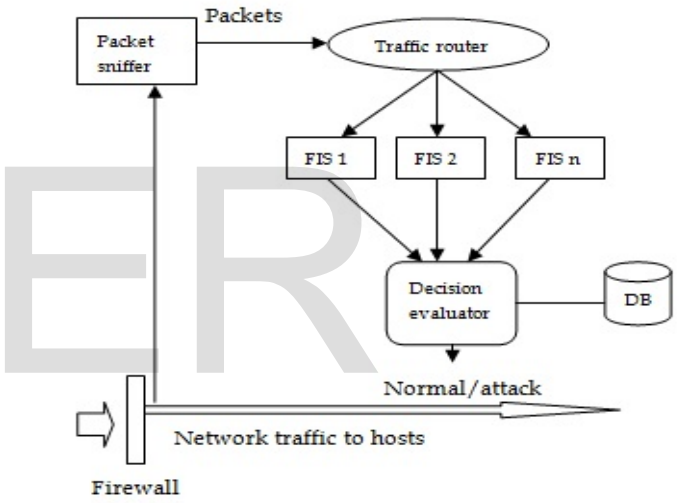

Fig. 1. System architecture,

The fig above shows that the packet sniffer will capture a copy of the traffic to the network and then forwards it to the traffic router. The traffic router will perform its attribute selection and directs the network traffic to the appropriate FIS for processing. All the FISs will communicate to the network administrator through the decision evaluator engine. This function can be able to handle decision evaluation, contingency action and communication tasks. The database will be used for recording history for future enhancements.

5 EXPERIMENTAL WORK

5.1 Dataset Preprocessing

The NSL-KDD dataset was downloaded from [7] where it is available for research and processed by a methodology proposed in article [9]. The dataset had 42 attributes. And was

processed as follows

1. Normal records were selected and all the attack records were left out using data processing software

2. Selection of important features to consider in generating the fuzzy rules using the fuzzy approach to feature reduction technique [8] was done carefully with a goal of minimizing misclassification of the normal or attack records

3. The resultant feature reduced normal records were further categorized into smaller clusters such that from those smaller clusters we could observe patterns that enables easy mining of fuzzy rules, hence forming the knowledge of the IDS

From the process above we came up with 28 clusters where the similarity of cluster elements was based on the protocol and service combination. Of those clusters 17 clusters belonged to tcp protocol, 5 belonged to udp and 6 were icmp. This work enables us to move forward to the generating of fuzzy rules for the fuzzy inference systems of the IDS

5.2 Fuzzy Inference Systems

From the clusters above we developed fuzzy inference systems to process each particular protocol and service combination of the network packets. The fuzzy rules for each cluster were generated by analysis of the data pattern in each cluster. We also analyzed the ranges of the values of the retained attributes and carefully we defined the ranges of low, medium and high then we applied those ranges to the membership functions of the fuzzy inference systems. The membership function we chose was the Gaussian bell shaped because it is smooth and non-zero membership for all values in the universe of discourse. Fig. 2 shows the fuzzy inference system with six inputs and one output.

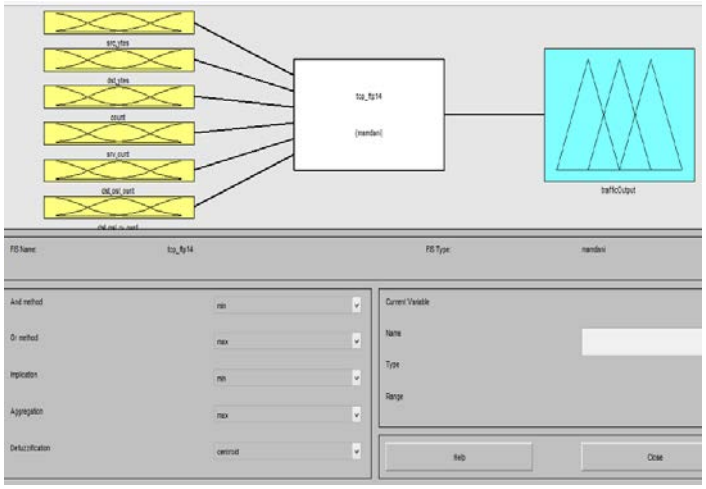

Fig. 2. FIS editor showing six inputs and one output

The fig above is one of the inferences systems and its mamdani type. All the inference systems had 6 inputs namely src_bytes, dst_bytes, count, srv_count, dst_host_count and dst_host_srv_count. The FIS had a single output namely trafficOutput. This will be a value in the range of 0 to 1. The following condition will be used as a basis for whether there was an intrusion or not:

If trafficOutput >= 0.5 then traffic is normal

Else

Traffic is an attack.

The diagram below shows membership functions of one of the input variables that is src_bytes. The range of the src_bytes was divided into low, average, high and highest and the membership functions were as shown in the dagram below. So given an src_bytes input the membership value of the input will be given to the value corresponding point on the membership curve

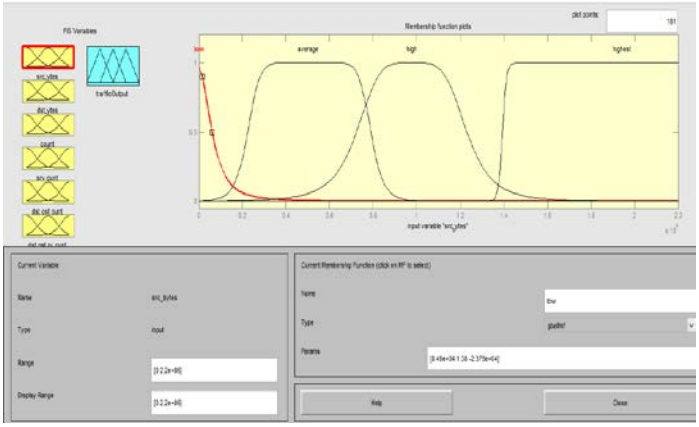

Fig. 3. Membership functions for src_bytes variable

After defining the inputs and functions the fuzzy rules

were entered into the system forming the systems knowledge. Each fuzzy rule had a weight to be used for calculation of the membership to the output function. It is from these rules where the normal or attack decision is going to be made on a data packet.

5.3 Simulation Methodology

Using the simulink library of the matlab system, we performed a simulation of the proposed system. The simulation was performed at unit level that is testing the performance of one FIS at a time and then at system level where all the fuzzy inf
Th
ute
dat
ov
sys
wil
wil
fac
toc

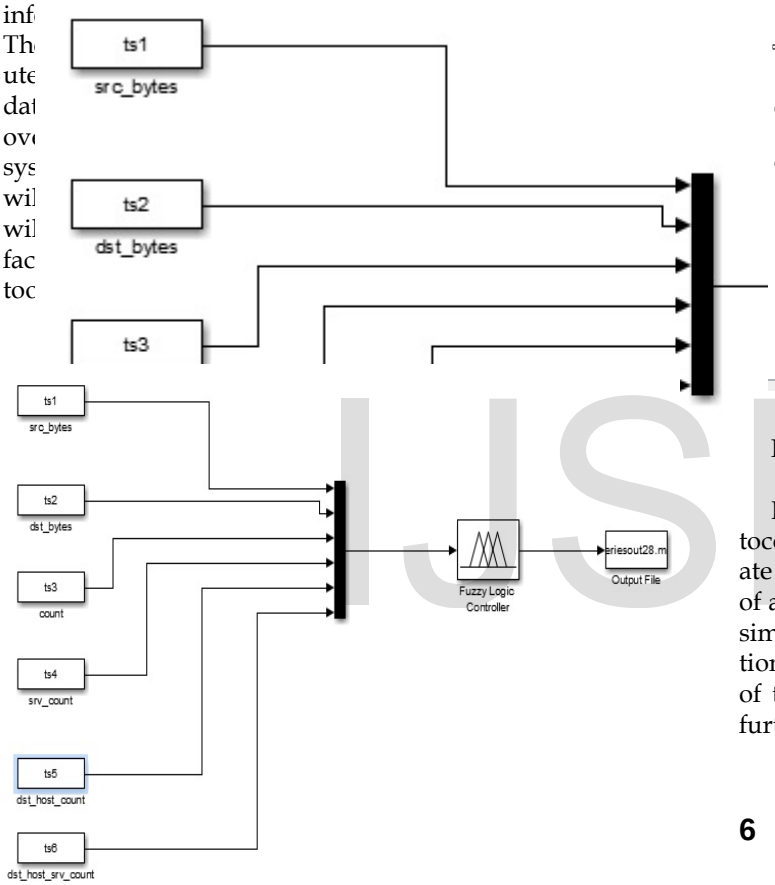

Fig. 4. Simulation and testing of a single FIS

The inputs to the FIS are fetched from the matlab workspace using from workspace blocks from the sources tool box and then using the MUX block from the signal routing tool box the signals are combined into a bus and the fed to the Fuzzy logic controller block from the fuzzy logic toolbox and finally the output from the given sets of inputs is written to the output file. The figure below depicts the simulation of the whole proposed system

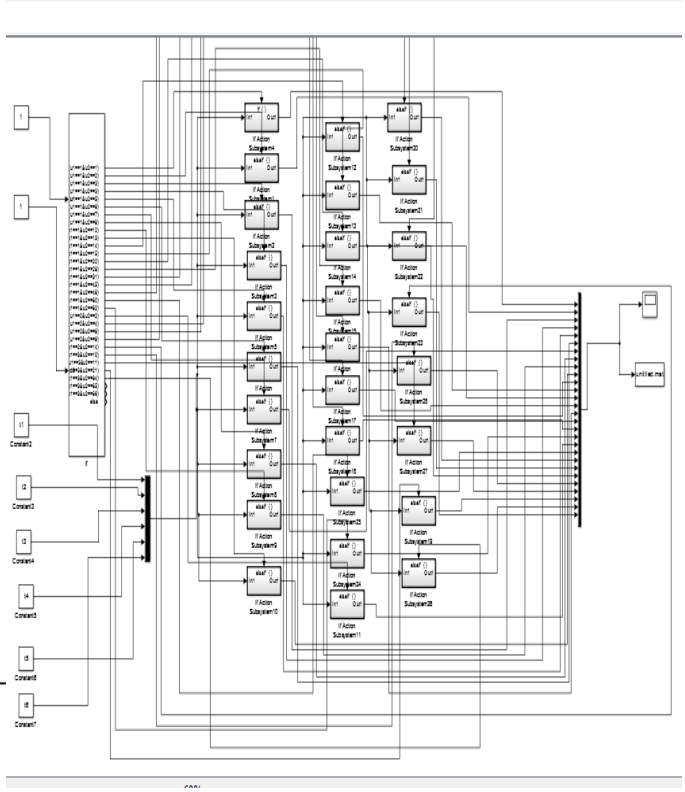

Fig. 5. Simulation and testing of integrated components

In the model above, the IF block was used to check the protocol and service combination and then activate the appropriate subsystem to process that particular system and the output of a block will be directed and recorded to the output file. The simulation was done several times and adjusting the simulation parameters and the ranges in the membership functions of the FISs. The results were recorded in the output file for further analysis.

6 Experimental Results

The simulation models shown above were run several times adjusting system's parameters and continually improving the knowledge base in order to increase accuracy in intrusion detection and reducing the false alarm rate. The results were recorded and several output files were recorded for the model analysis. We performed simulation and testing of each of the 28 FISs we developed and recorded the results then finally we integrated all the multiple FISs into a single system which we also tested several times making adjustments. Whilst we were performing the simulations it was observed that the first simulations were yielding a very low detection rate and a high false alarm rate. But however we kept on adjusting the ranges of fuzzy membership functions and modifying the simulation in each and every yielding better and more accurate results. Below is a screenshot of the integrated system output.

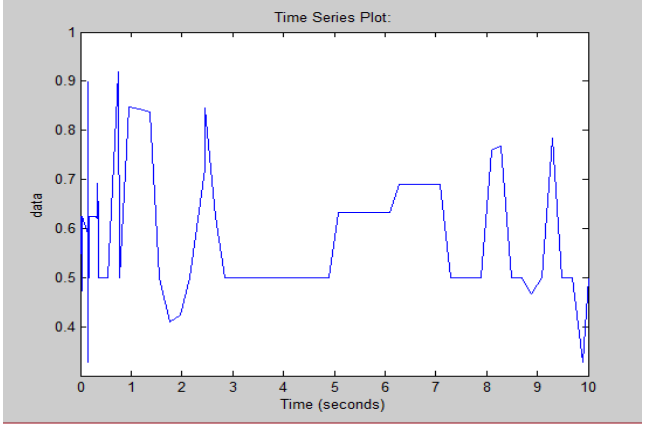

Fig. 6. Testing output

The figure above is a plot of the time series output data from the intrusion detection model at the time whilst the simulation model was run. The axis labeled data is the traffic output value which is a membership value in the range 0 to 1 which is used to determine whether the output is an intrusion or normal traffic. The simulation was run in 10 seconds time. From what is shown in fig. 6 above:

If (data > 0.5) then trafficOutput = normal
Else
trafficOutput = intrusion.

The above equasions specifies that any trafficOutput value obtained as output from the system which is grater than 0.5 is considered as normal traffic whilst any trafficOutput value that is equal to 0.5 or less is regarded as an intrusion.

Using the output from the graph above, we then went on to check with our testing dataset and performed an analysis to evaluate the accuracy and performance of the system. In the analysis process, we looked at particular inputs fed to the system at a particular time whether the input belonged to an intrusion class or was a normal traffic and checked whether it was correctly classified or not and also checking whether there were some inputs reported as attacks whilst they are not in order to be able to predict the detection rate and false alarm rate. The table below summarizes the results over traffic packets. For a 1000 packets, the following were obtained

Total attack records from dataset	376
Total normal records from dataset	624
Total Attacks recorded	379
Total Normal recorded	621
Attacs correctly classified as attacks Tue Positive TP	360
Normal records misclassified as attacks False Positive FP	19
Normal traffic correctly classified as normal True Negative TN	605
Attacks misclassified as normal False Negative FN	16

Detection rate = total number correctly classified as attacks/total records in set

= (360+605)/1000 * 100%
= 96.5%

False alarm rate = number of normal records classified as attacks/ total number of normal patterns

= 19/624 * 100
= 3,044%

7 Conclusions

Our work proposed the architecture of a fuzzy logic based anomaly intrusion detection and used the data preprocessing proposed in [9]. Using matlab and simulink environment we were able to model the system and observe the performance of the system. The results shows that this is very significant contribution into the field of intrusion detection systems as the observed intrusion detection rate of 96.5% and a false alarm rate of 3.044% shows that the system have very good performance and through further tuning of the system this performance can be improved. Our results show that this IDS performs well as other systems proposed in this field.

In the future work of our research we propose to improve the decision engine evaluator function to be able to effectively evaluate and weight a decision and take appropriate action. Also there is need to address issues of working in a distributed environment.

8 Acknowledgment

We would like to thank to Prof. A Govardhan, Director

School of Information technology, for encouraging research Programmes. The authors would like to thank the anonymous reviewers for their valuable comments.

9 REFERENCES

[1] Mohay, George M. *Computer and intrusion forensics*. Artech House, 2003.

[2] Zamani, Mahdi. "Machine Learning Techniques for Intrusion Detection." *arXiv preprint arXiv:1312.2177* (2013).

[3] Bansal, Raj Kumar, Ashok Goel, and Manoj Kumar Sharma. *MATLAB and its Applications in Engineering*. Pearson Education India, 2009.

[4] Wankhade, Kapil, Sadia Patka, and Ravindra Thool. "An efficient approach for Intrusion Detection using data mining methods." *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*. IEEE, 2013.

[5] Khazaee, Saeed, and Maryam Sharifi Rad. "Using fuzzy C-means algorithm for improving intrusion detection performance." *Fuzzy Systems (IFSC), 2013 13th Iranian Conference on*. IEEE, 2013.

[6] Shanmugam, Bharanidharan, and Norbik Bashah Idris. "Improved intrusion detection system using fuzzy logic for detecting anamoly and misuse type of attacks." *Soft Computing and Pattern Recognition, 2009. SOCPAR'09. International Conference of*. IEEE, 2009.

[7] http://nsl.cs.unb.ca/NSL-KDD/, general website.

[8] Das, Anish, and S. Siva Sathya. "A fuzzy approach to feature reduction in KDD intrusion detection dataset." *Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on*. IEEE, 2012.

[9] Macdonald Mukosera, Thabiso Peter Mpofu and Budwell Masaiti **."** Analysis of NSL-KDD dataset for fuzzy based intrusion detection system" International Journal of Science and Research (IJSR), Volume 3 Issue 6, June 2014.

[10] Baniasadi, Zohreh, Arghavan Sanei, and Mohammad Reza Omidi. "A fuzzy description logic model for Intrusion Detection Systems." *Telecommunications (IST), 2010 5th International Symposium on*. IEEE, 2010.

[11] Bahrami, Mehdi, and Mohammad Bahrami. "An overview to Software Architecture in Intrusion Detection System." *arXiv preprint arXiv:1205.4385*(2012).

[12] KDD Cup 1999, Available Online: http://kdd.ics.edu/databases/kddcup99/kddcup99.html, October 2007. [Accessed June 11, 2014].

[13] Tavallaee, Mahbod, et al. "A detailed analysis of the KDD CUP 99 data set."*Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009*.

- ***Macdonald Mukosera*** *is currently pursuing masters of technology degree program in Computer science at JNTU Hyderabad SIT India. He received the B.Tech Hons degree in Computer Science from Harare Institute of Technology Zimbabwe 2010. During 2011-2012, he worked as a Teaching assistant at Harare Institute of Technology in the Software Engineering Department. His research interests include Data mining, information security and cloud computing. E-mail: mmukosera@gmail.com*
- *Dr. G.Venkata Rami Reddy received his M.Tech (CSE) degree from JNT University Hyderabad in 1998. He then received his Ph.D. degree in Computer Science and Engineering from Jawaharlal Nehru Technological University (JNTU) in 2013. He has been working in JNT University since 2000. Presently he is working as an Associate Professor in the Dept of CSE in School of Information Technology, JNT University Hyderabad. He has more than 14 years of experience in teaching and Soft-ware Development. He has presented in more than 15 National and International journals and conferences. His research interests include Image Processing, Pattern Recognition, Network Security, Digital Watermarking, Image retrieval, and computer networks. E-mail: gvr_reddi@yahoo.co.in*