Network Intrusion Detection System Using Fuzzy Logic

Mohammed Moshiul Hoque[1], Kazi Abdul Mohit[1], Muhammad Ahsan Ullah[2], Meher-Ul-Karim[3]
[1]Department of Computer Science & Engineering, Chittagong University of Engineering & Technology (CUET)
[2]Department of Electrical & Electronic Engineering, Chittagong University of Engineering & Technology (CUET)
[3]Department of Computer Science & Engineering, Chittagong Polytechnic Institute (CPI)
e-mail: moshiulh@yahoo.com, kazi_mohite@yahoo.com, ahsan_cuet@yahoo.com, meher_k@yahoo.com

Abstract: *Computer and network security plays an important role in modern communication. An efficient network is free from malicious activities. Today's network administrators have to pay most of their efforts to make the network secured. This paper presents a network intrusion detection system using fuzzy logic approach. Throughout this paper, we have developed an intrusion detection system that uses fuzzy rules to differentiate normal and abnormal activities. For building fuzzy rules, we use membership function depending on normal activities in network environment. A rule is matched with some degree of membership value. All data are combined and defuzzyfied. We have tested detection system with several datasets with 92% accuracy.*

Key Words: Fuzzy logic, Fuzzy matching, Fuzzy inference, Intrusion detection, Fuzzy Rule and Defuzzyfication.

1. INTRODUCTION

Intrusion detection attempts to detect computer attacks by examining data records observed by process on the same network. The main task is intrusion detection systems are defense of a computer system by detecting an attack and possibly repelling it. Detecting hostile attacks depends on the number and type of appropriate actions. Data generated by intrusion detection systems is carefully examined for detection of possible intrusions [1]. Once an intrusion has been detected, IDS issues alerts notifying administrators of this fact. Commercial tools typically do not provide an enterprise level view of alarms generated by multiple sensor vendors. Most commercial Network Intrusion Detection (NID) system use a form of intrusion Detection called "misuse Detection". These systems are usually only effective when prior knowledge of the detailed characteristics about known attack signatures is available and they fail to detect many cyber attacks because they lack intelligent technique to make correct decisions in detecting distributed unknown attacks [2, 3]. One way build intelligent decision-making systems to use in intrusion detection relies on learning the typical user/application behavior from a set normal data [4]. In the past, association rules and data mining technique were suggested to implement ADS [5, 6]. A prototype system has been presented by combining two detection systems, but working independently [7]. Artificial immune systems have been applied successfully in anomaly based computer network intrusion [8, 9]. However, there are some problems that have prevented this approach from being applied extensively. The main objective of this paper is to propose a NIDS module that can detect unknown attacks based on fuzzy rule. Specify how raw network Data packets of multiple levels are collected from the wire and feature are extracted for matched with fuzzy rule. Fuzzy systems have several important characteristics that suit intrusion detection very well.

2. PROPOSED NETWORK INTRUSION DETECTION SYSTEM ARCHITECTURE

The overall proposed architecture for network intrusion detection as presented in Fig. 1.

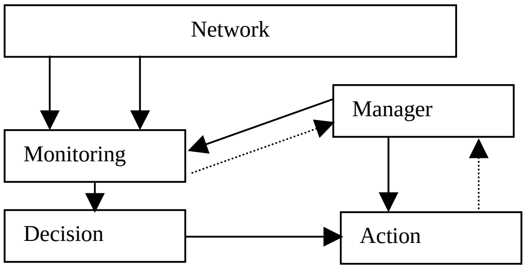


Fig. 1: Proposed NIDS model.

For performing operation effectively this Architecture use multi-agent where different agent performs different functions. In the Fig. 1, monitoring agents are collected packet from network environment, extract packet header and find out what data is required for judging intrusion or normal. Prepare data and supplied the data for taking decision. Decision agent makes decision according collected packet information. Decision is also involved according to the underlying security policies for a given network. The taken decision is supplied to the action agent. Action agent takes action according through the decision. Different software or hardware can be connected with it.

2.1 Monitoring Agent

These agents are vigilant and petrol the network modes, specifically, these agents collected Data and Extract Packet header and supply the Packet information into decision agent. The information which monitoring agent collects is given in Table 1.

Table 1: Collected Packet data from different Level

NETWORK LEVEL	PROCESS LEVEL
LOCAL-SENT-BYTES	PROCESSES
LOCAL-RECEIVED-BYTES	PROCESSES-ROOT
LOCAL-SENT-PACKETS	PROCESSES-USER
LOCAL-RECEIVED-PACKETS	PROCESSES-BLOCKED
REMOTE-SENT-BYTES	PROCESSES-RUNNING
USER LEVEL	**SYSTEM LEVEL**
LOGINS	USED-PHYSICAL-RAM
FAILED-LOGINS	USED-SWAP-RAM
REMOTE-LOGINS	-

2.2 Manager, Decision and Action Agent

Manager agents serve as message carriers or negotiators in order to maintain a liaison among other agents.

Decision agents involved in making decisions or performing specific tasks according to the underlying security policies. Action agents are performed specific action depending decision from decision agents.

2.3 Fuzzy Rule Generation Steps

The steps for generating Fuzzy decision engine are mention in the following:

Step 1: Data is collected from Network or any built-in dataset.

Step 2: Different field of dataset is described in this step.

Step 3: Membership function is generated according to the collected information.

Step 4: Fuzzy Rule is generated for taking decision.

Step 5: Fuzzy Rule is matching for a given input value.

Step 6: Calculate the rule conclusions based on previous degree.

Step 7: defuzzyfication convert the final combined fuzzy conclusions into a crisp one.

Step 8: Take final Decision.

2.3.1 Data Collection and Description

For NIDS we have to collect Data from the network log files where along with packet header data there must be an extra field whether the access was normal or other. We have data collected from KDD-99 Cup Knowledge Discovery having the desired extra field. It has 42 fields and about 10 millions of records. Some field in database that we have used is given in Table 2.

Table 2: Fields in the Dummy Data

Feature Name	Description
duration	length of the connection
protocol_type	type of the protocol, e.g. tcp, udp, etc.
service	network service on the destination, e.g., http, telnet, etc
src_bytes	number of data bytes from source to destination
dst_bytes	number of data bytes from destination to source
num_failed_logins	number of failed login attempts
num_root	number of ``root" accesses
num_file_creations	mber of file creation operations
num_shells	number of shell prompts
is_host_login	1 if the login belongs to the ``hot" list; 0 otherwise
serror_rate	% of connections that have ``SYN" errors

The features described in Table 2 are separated into three class attributes. 1 to 9 known as first class attributes. 10 to 22 known as second class attribute 23 to 31 is known as third class attribute.

2.2.2 Data Selection

For Fuzzy rule generation we select 200 Data from Normal Dataset. After that we synthesize the Data Depending on 5 basic fields of Data set and collect minimum and maximum value from them. The field's are: *src_bytes, dst_bytes, count, srv_count, num_failed_logins,* and *duration*. Temporal statistical features are based on a set of connections with the same reference features, while content of features based on each connection are: *serror_rate* and *rerror_rate.*

2.2.3 Membership Function Generation

The collected minimum and maximum value of each field are used for generate membership function for each field. Minimum value of Src_bytes is 0 and maximum is 1032 bytes or above as well as for dst_bytes. The minimum is 0 and maximum is 500 or above both for count and service count. By using minimum and maximum value we determine range. Tables 3, and 4, describes range of membership function for sent bytes and count respectively. Failed logins can be normal when its value is between 0 to10. But when its value is greater then 200 it is not normal condition.

Table 3: Membership function Range for Send Bytes

Universe of discourse (X)	Degree of Membership(μ)
Src_bytes>=0 to Src_bytes<=18	LOW
Src_bytes>18 to Src_bytes<=100	MEDIUM-LOW
Src_bytes>100 to Src_bytes<=150	MEDIUM
Src_bytes>150 to Src_bytes<=1000	MEDIUM-HIGH
Src_bytes>1000 to Src_bytes<=1032	HIGH
Src_bytes>1032	VERY-HIGH

Table 4: Membership function Range for connection count

Universe of discourse (X)	Degree of Membership (μ)
Count_data >= 0 to count_data < =1	LOW
Count_data > 1 to Count_data <=50	MEDIUM
Count_data > 50 to Count_data <=500	MEDIUM-HIGH
Count_data >500	HIGH

The membership function for the different criterion has been presented in Fig. 2, 3, 4, 5 and 6 respectively.

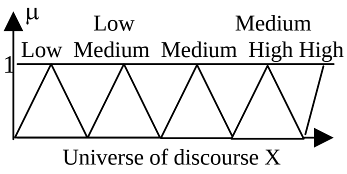


Fig. 2: Membership function for sent bytes.

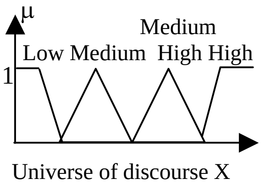


Fig. 3: Membership function for count.

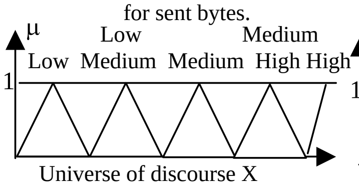


Fig.4: Membership function for received bytes.

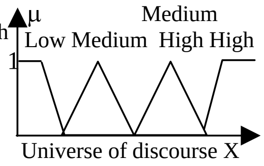


Fig.5: Membership function for service count.

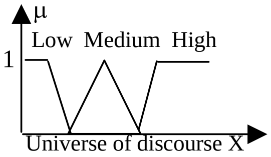


Fig. 6: Membership function for number of failed logins.

2.2.4 Fuzzy Rule Generation

Snmpgetattack is one type of Remote to Local (R2L) attack. Attacker does not have an account on the victim machine, hence tries to gain access. *Ipsweep* and *warezmaster* attack is a probe, where attacker tries to gain information about the target host. *Smurfing* is a Daniel of service (DoS) attack. Attacker tries to prevent legitimate users from using a service. Some fuzzy rules for NIDS have been presented in Table 5.

Table5: Fuzzy Rules for NIDS

NORMAL	**SNMPGETATTACK**
if Count is low THEN Normal	if Count is MEDIUM THEN SNMPGETATTACK
if Count is MEDIUM_HIGH THEN NORMAL	if SRV_Count is Medium THEN Snmpgetattack
SMURF	**IPSWEEP**
if Count is HIGH THEN SMURF	If sent_bytes is LOW THEN IPSWEEP
if SRV_Count is HIGH THEN SMURF	
WAREZMASTER	
If Duration_count is Medium_high then WAREZmaster	

2.2.5 Fuzzy Matching

We calculate degree to which the input data match with the condition of the fuzzy rules. If count is 2 then observed degree is 0.5. When service count is 2 then observed degree is 0.5. If sent bytes are 105 then observed degree is 0.4 and received bytes is 146 then observed degree is 0.4. Number of failed logins is 641 so degree of match is 0.65. Rules for SNMPGETATTACK and NORMAL data are:

Rule 1: If Count is Medium Then SNMPGETATTACK
Rule 2: If Service Count is Medium Then SNMPGETATTACK
Rule 3: If Sent Bytes is Medium Then NORMAL
Rule 4: If Received Bytes is Medium Then NORMAL
Rule 5: If Service Is SNMP and Num_of_failed_Logins is high then SNMPGETATTACK

Fig. 7, 8, 9, 10 and 11 illustrate the Fuzzy matching for rule 1, 2, 3, 4 and 5 respectively.

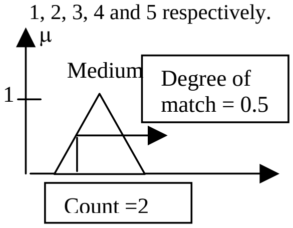


Fig. 7: matching for rule 1

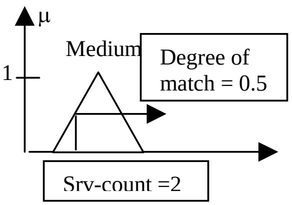


Fig. 8: matching for rule 2

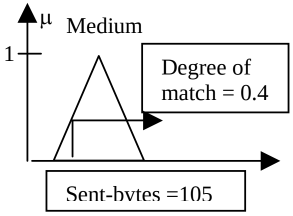


Fig. 9: matching for rule 3

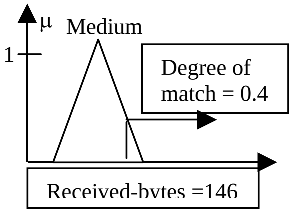


Fig. 10: matching for rule 4

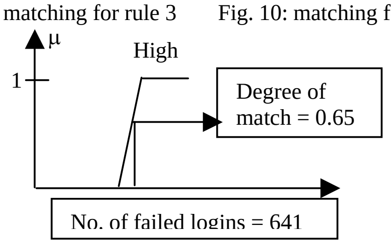


Fig. 11 Fuzzy matching for rule 5

2.2.6 Fuzzy Inference

According to first rule the confidence for committed SNMPGETATTACK is 0.5, confidence for second rule of SNMPGETATTACK is 0.5, third rule confidence for NORMAL is 0.4, and forth rule confidence for NORMAL is 0.4.and fifth rule confidence for SNMPGETATTACK is 0.65. We used clipping method for fuzzy inference. The clipping method cuts the membership function whose value is higher then the matching degree. Fig. 12 illustrates Fuzzy inference of rules.

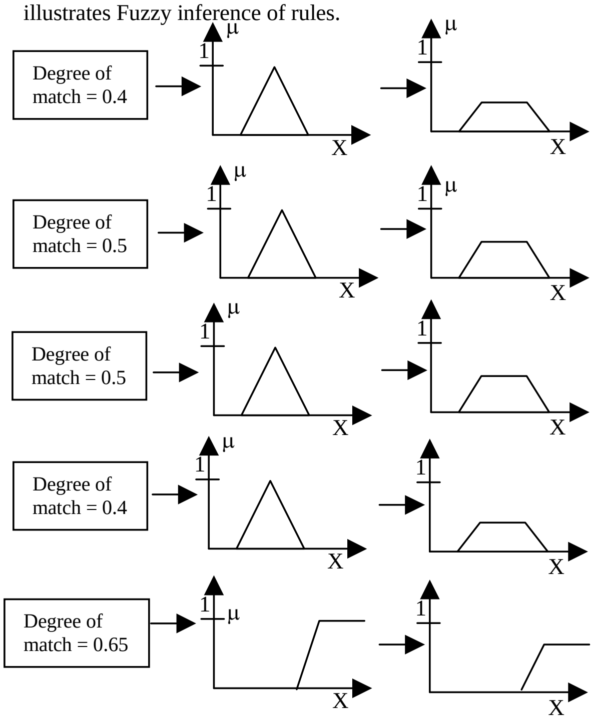


Fig. 12: Fuzzy Inference

2.2.7 Defuzzyfication

For a fuzzy system whose final output needs to be a crisp form, a forth step is needed to convert the final combined fuzzy conclusion into a crisp one. This step is called defuzzification. We used mean of maximum (MOM) defuzzyfication calculates the average of all variables values with maximum membership degree. Fig. 13 shows defuzzyfication of 5 rules

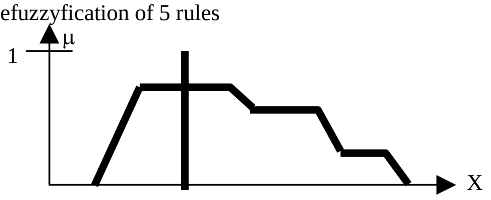


Fig. 13: Defuzzyfication of all rules

2.2.8 Make Decision

According to Defuzzyfied value we determine a range for depending on communication type.

If condition is between 0.3 to 0.4 or 0.7 then NORMAL
If condition is between 0.5 to 0.69 then SNMPGETATTACK
If condition is between 0.71 to 0.8 then SMURF
If condition is between 0.1 to 0.2 then IPSWeep
If condition is between 0.9 to 1 then Warezmaster

We get 0.65 as a result from defuzzyfication. So SNMPGETATTACK committed.

3. Experimental Result and Discussion

For testing we establish a simulated attack environment by using corrected dataset of KDD-99.

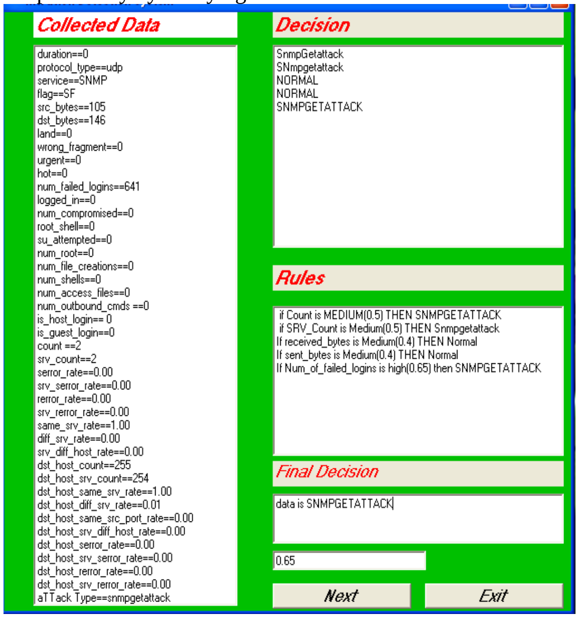
3.1 Performance Details

Dataset contains, DoS attack data: 2, 29853, Probe attack data: 4166, U2R data: 70, R2L data: 16347 and normal data: 60593. We have taken 100 packets randomly and observed that the accuracy of this NIDS program is about 92 %. Table 6 illustrate success rate for individual attack types depending on number of detection and failure of detection.

Table 6: Performance Data

Attack Type	No. of Attacks	Detected	Undetected	Success Rate
Snmpget attack	40	35	5	87.5%
IPSweep	15	15	0	100 %
Smurf	20	20	0	100%
Warezmaster	20	17	3	85%
Normal	5	5	0	100 %

Fig. 14 presents a snapshot of SNMPGETATTACK and shows that as the Snmpgetattack starts, the system senses the possibility by verifying rules in its.



Collected Data

```
duration==0
protocol_type==udp
service==SNMP
flag==SF
src_bytes==105
dst_bytes==146
land==0
wrong_fragment==0
urgent==0
hot==0
num_failed_logins==641
logged_in==0
num_compromised==0
root_shell==0
su_attempted==0
num_root==0
num_file_creations==0
num_shells==0
num_access_files==0
num_outbound_cmds ==0
is_host_login== 0
is_guest_login==0
count ==2
srv_count==2
serror_rate==0.00
srv_serror_rate==0.00
rerror_rate==0.00
srv_rerror_rate==0.00
same_srv_rate==1.00
diff_srv_rate==0.00
srv_diff_host_rate==0.00
dst_host_count==255
dst_host_srv_count==254
dst_host_same_srv_rate==1.00
dst_host_diff_srv_rate==0.01
dst_host_same_src_port_rate==0.00
dst_host_srv_diff_host_rate==0.00
dst_host_serror_rate==0.00
dst_host_srv_serror_rate==0.00
dst_host_rerror_rate==0.00
dst_host_srv_rerror_rate==0.00
aTTack Type==snmpgetattack
```

Decision

```
SnmpGetattack
SNmpgetattack
NORMAL
NORMAL
SNMPGETATTACK
```

Rules

```
 if Count is MEDIUM(0.5) THEN SNMPGETATTACK
 if SRV_Count is Medium(0.5) THEN Snmpgetattack
If received_bytes is Medium(0.4) THEN Normal
If sent_bytes is Medium(0.4) THEN Normal
If Num_of_failed_logins is high(0.65) then SNMPGETATTACK
```

Final Decision

```
data is SNMPGETATTACK
```

```
0.65
```

Fig. 14: NIDS for SNMPGETATTACK

Summary all connection types have been given in Table 7.

Table 7: Summary of all Connection types

Attack type	Reasons	Range
SNMPGETATTACK	Higher Number of Failed Logins	0.65
SMURF	Higher Connection count Higher Service Count	0.8
WAREZMASTER	Higher duration of Connection	0.9
IPSWEEP	Low Sent bytes	0.2
NORMAL	Medium Sent Bytes Medium received Bytes Low Connection count	0.4

	Low service count	

5. Conclusion

The Internet and LANs are expanding at an amazing rate in recent years. While we are benefiting from the convenience that the new technology has brought us, computer systems are exposed to increasing security threats that originate externally or internally. Despite, different protection mechanisms, it is nearly impossible to have a completely secured system. Therefore, intrusion detection is becoming an increasingly important technology that monitors network traffic and identifies network intrusions such as anomalous network behaviors, unauthorized network access, and malicious attacks to computer systems. In this paper, we have implemented the NIDS using fuzzy logic. Though the dataset for the simulation of attack was taken fully different part of the main dataset that hasn't taken part in learning process, the result is very encouraging. The accuracy is always above 90% for different test datasets. Therefore, if proposed NIDS will apply in real network it will be more efficient than other model. Data mining technique can be used co-operatively with fuzzy rule for increasing the accuracy very gratefully. System can be expanded with the data feeds to add host-based anomaly detection.

References

[1] R. Agrawal, T. Imielinski, A. Swami, "Mining Associations between Sets of Items in Massive Databases". *Proc. of ACM-SIGMOD 1993 Int'l Conf. on Management of Data*, Wash. D.C., pp. 207-216, 1993.

[2] D. Dasgupta, F. Gonzalez, "An Intelligence Decision Support System for Intrusion Detection and Response", *Proc. of International Workshop on Mathematical Methods, Models and Architecture for Computer Networks Security (MMM-ACNS)*, May 21-23, 2001.

[3] T. Lane, "Machine Learning Techniques for the Computer Security", Ph. D thesis, Purdue University, 2003.

[4] E. Eskin, "Anomaly detection over noisy data using learned probability distributions", *Proc. 17th International Conf. on Machine Learning*, pp. 255-262, 2005.

[5] D. Barbara, J. Couto, S. Jajodia, L .Popyack, N.Wu "ADAM: Detecting Intrusions by Data Mining". *IEEE Workshop on Information Assurance and Security*, 2001.

[6] S. Bridges, R. M. Vaughn, "Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Detection", *Proc. of the 23rd National Information Systems Security Conference*, Baltimore, MD, 2000.

[7] DETER and EMIST Projects "Cyber Defense Technology, Networking and Evaluation" http://www.isi.edu/deter/docs/acmpaper.pdf. pp. 58-61, 2004

[8] S. Hofmer, S. Forrest, "Architecture for an artificial Immune System", *Evolutionary Computation*, vol. VIII, pp. 443-473, 2000, 4.

[9] D. Dasgupta, "Artificial Immune Systems and their Applications", New York, Springer-Verlag, 1999.