www.ijsrnsc.org

Available online at www.ijsrnsc.org

IJSRNSC

Volume-6, Issue-5, October 2018
Research Paper

Int. J. Sc. Res. in
Network Security
and Communication

E-ISSN:2321-3256

An Expert Forensic Investigation System for Detecting Malicious Attacks and Identifying Attackers in Cloud Environment

P. Santra

Criminal Investigation Department, West Bengal, Kolkata, India

Corresponding Author: santra.mic.palash@gmail.com, Tel.: +91-9038091620

Received: 26/Aug/2018, Accepted: 14/Oct/2018, Published: 31/Oct/2018

Abstract— In recent years' complex and high level computations is done in cloud environment to achieve better performance with low cost. Different large and medium organizations are moving towards cloud computing due to its several trending features. This leads to a drastic increase in cloud services. However, shared on demand characteristic of cloud increases the vulnerability of several security threats. Several security mechanisms and intrusion identification techniques are proposed in the recent years to ensure a better quality of services. But ensuring a complete flawless system is very difficult. So, forensic science or investigation helps in identifying the adversary and collecting proper evidence against the intruder. No traditional digital and network forensic methods are applicable in cloud computing due to its different architectural features compared to a client-server network. A generic forensic model is proposed in this paper for cloud environment. Focus is given on the identification phase of the forensic system because a proper identification of the intruder leads to better forensic evidence generation. A strong fuzzy based expert forensic model "Fuzzy Expert System for Network Log Analysis" and "Expert System for Management Log Analysis" is projected which analyses network and management logs from cloud server for identifying the intruder. A "Forensic Investigation Report" is prepared to serve as a forensic report that will help to smoothly continue the forensic investigation as well as serve as evidence. The proposed model is also simulated in a private cloud environment showing improved accuracy up to ~5.6% over known forensic systems.

Keywords—Cloud, Forensic, Intrusion, Learning, Network, Association, Attacks.

I. INTRODUCTION

With such substantial development in the field of information and communication technology it is perceived that computing will become the 5th utility service after water, gas, electricity and telephony [1-3]. Among the various computing paradigms cloud computing has gained considerable attention recently. This is because cloud computing enables ubiquitous, on-demand access to various shared pool of resources which are provisioned and released with least management effort [4]. The distinguishable features of cloud computing that makes it advantageous over other computing paradigms are - low cost, scalability, elasticity, and location independence [2] [3]. This enables the small and medium sized companies to migrate its infrastructure to various Cloud Service Providers (CSPs) which offers flexible pay-as-you-demand services. According to a report made by Market Research Media, usage of cloud computing is expected to grow at a rate of 30% compound annual growth reaching $270 billion by 2020 [5].

Due to such rapid development in the field of cloud computing and its fast acceptance among the users makes it prone to various malicious attacks [6-11]. Adversary either harm the cloud infrastructure or launch various attacks from the Virtual Machines (VMs) to damage other systems. So providing security to user's data located in the cloud server is of primary concern to motivate users to migrate their data and infrastructure to the cloud environment [12-14]. A recent survey conducted by IDCI shows 74% of the IT executives hesitates to move in the cloud environment due to various security threats [15]. So the primary focus for a Cloud Service Provider (CSP) is to provide security to user's private data. The multi-tenant cloud architecture and virtualization of resources helps cloud utilize resources in optimal manner. However, due to this characteristic feature it becomes difficult to prevent and investigate attacks in cloud environment. To address the security issues the concept of cloud forensics was introduced in 2011 [16] [17]. There are several established techniques and tools available in the field of digital forensics. However, none of the methods are directly applicable in cloud environment [18] [19]. Digital forensics deals with standalone systems or small interrelated network. So, the various assumptions that could be easily made in digital forensic for evidence collection is not applicable in the cloud environment [17] [19]. Digital forensic techniques lack virtualization, multi-tenancy, distributed system and huge data records that need to be considered during cloud forensic analysis [20].Thus cloud forensics is new research area which is combination of digital forensic and network forensics with the flavor of multi-tenant, distributed cloud architecture.

Researchers aim towards developing a strong cloud forensic system to prevent and identify the attacker. The key steps for any forensic investigation are - Identification, Collection, Preservation, Analysis and Presentation [21] [22]. For correct identification proper investigation of the incident need to be done in the cloud environment. In this paper we aim to build a strong cloud forensic architecture and have proposed an expert system for identification of attacked area in the cloud environment. This has real life significance to enforce law against cybercrimes.

II. MOTIVATION

Investigating a crime scene by identifying the types of attacks and attackers is very crucial. In cloud environment one way of identifying the invader and his crime is analyzing cloud log records [21-23]. Cloud log has the record of all the activities taking place in the cloud environment serving as one of the important evidences for forensic analysis [17] [20] [22]. However, the investigators as well as the users have little or no control over the cloud logs. So for log record collection the investigator has to rely completely on CSP. To obtain the log records investigators issues summon to the CSP. Investigator in that case need to dimly accept the log record provided by the CSP since there is no means of verifying the legitimacy of the provided records. At the same time investigator may have to face some glitches in acquiring the network log from the CSP. Even if he receives the log record, identifying the intruder and the nature of attack from such huge data record is very challenging. To overcome these challenges, we are motivated towards developing an expert system for log record collection, monitoring and analysis. This automated system helps in identifying the exact attacked area and the type of attack in the vast distributed cloud system. The automated system provides a Forensic Investigation Report (FIR) with the attack and attacker's detail. This report helps the investigator to issue a cyber-warrant against the attacker and can smoothly continue with the investigation. The contributions of this work are as follows –

A. An innovative cloud forensic methodology is proposed by identifying the techniques that can be taken up for each of the forensic phases.

B. To the best of our thoughts first time a dynamic expert system for cloud forensics is developed for identifying various types of network attacks, the attacker and the zone of attack in the cloud environment.
   1. A separate Fuzzy Expert System for Network Log Analysis (FESNA) is developed that analyses the network log module to recognize the attack type and the area of attack in cloud network.
   2. An Expert System Management for Log Analysis (ESMA) is developed for monitoring and analyzing the management log information for identifying the details of the attacker.

C. The fuzzy system is capable of identifying known, as well as unknown anomalous attack pattern.

D. The expert system provides a Forensic Investigation Report (FIR) that helps the investigator to issue warrants against the anticipated attacker and take immediate action to stop further damage. This FIR also helps the forensic expert to collect proper evidences to ascertain the crime in court of law.

E. Further the proposed expert system is tested in our private cloud infrastructure. The result shows-
   1. An improved true positive rate compared to some existing forensic method
   2. Reduced false positive rate
   3. Improved accuracy
   4. Improved precision

Further this work is arranged as follows. In section 3 some of the related works are discussed. Section 4 presents the proposed cloud forensic architecture. In section 5 some of the significant cloud attacks are discussed. Section 6 demonstrates the working principle of the training and testing engine. The detailed structure of the forensic investigation report is shown in section 7. Result and Conclusion are listed in section 8 and 9 respectively.

III. RELATED WORK

Contribution of this paper focuses on the network log based intrusion monitoring system for cloud forensic. In that case, several recent work on network and cloud forensic are analyzed here to get an idea of improvement. In the area of network forensic, intrusion identification is the most important security aspect. In the recent research papers of network forensic as well as digital forensic, log based approach is mainly focused because of its relevant feature for intrusion detection. In [22-24], necessity of log records for forensic analysis has been discussed with the solution of achieving it. Authors suggested here to collect log data from local log module which is synchronized with main server. Log file collection from API monitoring is also suggested as a trusted source of log records. Cloud management logs are also taken into consideration in [23-25]. Log records or information from different management modules such as validation engine, scheduler, hypervisor interface, load distributor are treated as important attributes for intrusion monitoring. Source and type of log records are described in [22-25], but analysis of such records to make a decision about intrusion is also a challenge.

There are very limited research papers on concrete forensic analysis for cloud environment. However, some of the intrusion detection techniques in network forensics can be implemented with the flavor of cloud. To implement it in the cloud server, several advancements need to be done over traditional digital forensic schemes. In [26] [27], fuzzy based network traffic analysis scheme is discussed that can be implemented for cloud environment with some upgradation. Here audit data is used for network analysis and to build a fuzzy system for attack prevention. Network parameters with their fuzzy values are used to generate rules for detecting the anomaly attack. To make it more reliable, an entropy parameter on IP address and Hurst parameter is introduced in [28]. This technique is built for effective DDoS defense system. Genetic Algorithm and Misuse detection based mechanism are proposed in [29] [30] for an effective intrusion defense system. Although a forensic data analysis system for intrusion detection is also proposed in [31], this technique is implemented using fuzzy c-means clustering of audit data. According to the literature survey, fuzzy system is widely used for intrusion detection and defense mechanism. So, for forensic investigation in cloud environment fuzzy technique will be very useful and a forensic model grounded on fuzzy algorithm is proposed in this paper.

IV. PROPOSED CLOUD FORENSIC ARCHITECTURE

In this section, complete methodology of the proposed forensic framework is discussed. There are 5 key steps in our proposed architecture through which the complete investigation is done. The process flow of this methodology is described in figure 1.

A. Identification:

This is the very first step of forensic analysis where occurrence of malicious activity in cloud environment is reported and identified. On receiving the report of illegal activity from the client or CSP the forensic analyzer investigates the crime scene to identify the area of attack, type of attack and the attacker. Identification of the attack requires analyzing and differentiating the normal activity and illegal activity that interfered the security of the system. Analyzing cloud network traffic is very challenging because large amount of network data need to be gathered and examined. So it is advantageous to use an expert system for identification purpose. In this context, two new module FESNA and ESMA are proposed in our work which helps in identifying the zone of attack, attacker and type of attack. An FIR is generated based on the predicted result obtained from the expert system. This report is important for the next steps of forensic investigation which is discussed in section 7. Detail description of FESNA and ESMA have been discussed in Section 6.2 and Section 6.3 respectively.

B. Collection:

Cloud consists of high capacity of on demand resources. Data from these resources need to be captured to collect strong forensic evidences. FIR may not be enough to proof a crime in the court of law. In that case, evidences from acquired resources need to be presented. In this step a File System Dump (FSD) technique can be used to extract the required data from the cloud distributed file system.

C. Pre Analysis Preservation:

Preservation is one of the important tasks in cloud forensic to make evidence tamper free. According to the FIR, FSD extracts the needed information for further analysis. However, the dumped file system is not very secure. Outsiders as well as cloud service provider can tamper the said file system. A secure data preservation scheme is required to maintain its security. HASH or Incremental HASH algorithm can be used to encrypt the file system and enhance its security.

D. Analysis:

Detection of criminal activity has been done in the step of identification. Still further analysis may be needed on the dumped file system. Different FSD techniques generally extract full volume of distributed resources from a particular site. The volume of extracted data may be in terms of TeraByte and GigaByte. This huge volume of data cannot be presented in court of law as evidence. So reduction and selection of particular segment of file system is desired. Generally, this investigation is done manually by expert investigator.

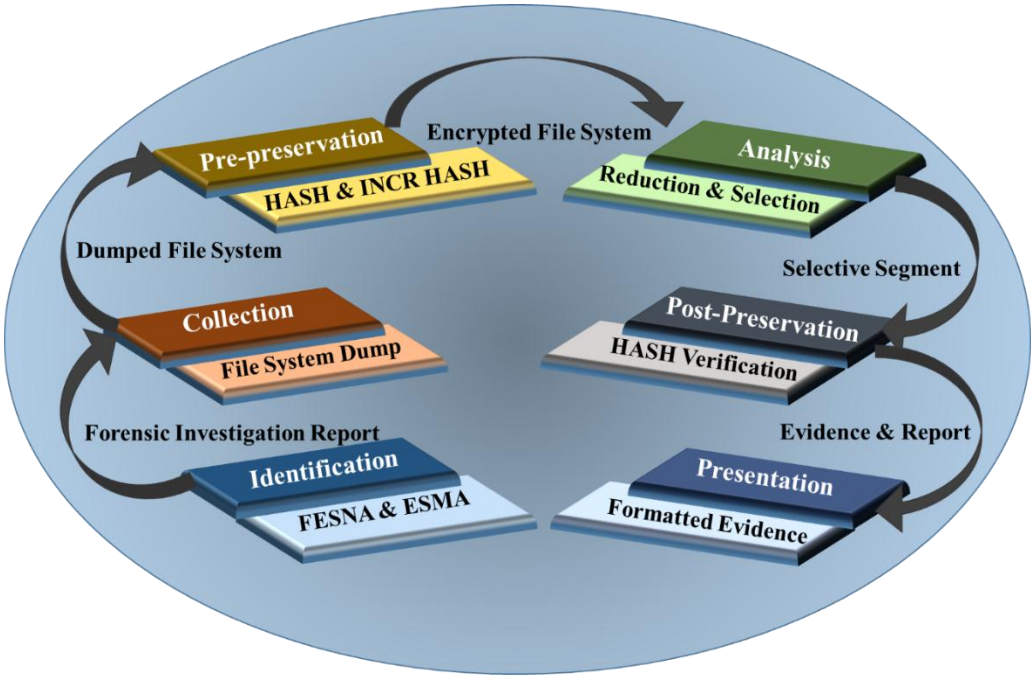


Figure 1. Proposed Forensic Mythology for Cloud

E. Post Analysis Preservation:

After the analysis part, analyzed data again need to be preserved for validation of tamper free evidence. During analysis, investigator may have tampered the evidences. To make it reliable, a verification with previously preserved data is also desired. This verification process can be accomplished using HASH value checking.

F. Presentation:

This is the final step of forensic investigation. On the basis of generated evidences, a report is been prepared to explain the complete criminal activity in the court of law.

Identification process is the primary and very important step in forensic investigation because next level of tasks depends on the accuracy of intrusion identification, detection and the generated FIR as a substantiation. A crime cannot be proved in court of law without proper evidence. Gathering evidences from a cloud system is very challenging for any forensic analyzer because of the multi-tenancy in cloud systems and Service Level Agreement (SLA) owned by the CSPs. In such a scenario identification of the exact crime scene and confirming crime has taken place is a sheer necessity. Much accurate intrusion detection system leads to more reliable analysis of the gathered evidence. In this paper, we are focusing on the identification phase consisting of – a) FESNA which helps in identifying the attacked network, b) ESMA which identifies the attacker as well as attacked area and c) FIR which will help the forensic analyzer issue a warrant against the intruder and collect evidences against him to prove his crime.

V. Cloud Attacks & Fuzzy Systems

Recently most of the companies are moving to the cloud environment to store their private data and at the same time avail various services provided by the service provider. So, workload on the cloud system is increasing gradually. In that case, cloud system need to enhance their security to protect user’s private data from attackers and increase their reliability. On account of extensive surveys [9-11], it has been observed that any network is susceptible to four main types of attacks.

A. Denial of Service (DoS):

Cloud system is more penetrable to Denial of Service attack since several users’ uses cloud services and resources simultaneously. When system workload increases the number of resources that can be allocated to any user gets restricted. This enables the attacker to harm the service availability by flooding the system with malicious requests. The victim resources become too busy in serving the malicious request blindly thus damaging the systems service availability. DoS may take place in a distributed manner in cloud environment referring to it as Distributed Denial of Service (DDoS). Back, apache, smurf are the examples of DoS attack.

B. Remote to User Attacks (R2L):

In this type of attack, malicious user sends spy packets to the victim system remotely to find the vulnerability of the system.

It supplies sensitive information to the intruder. In cloud environment malware packets are injected [32] to the different VMs for spying about delicate information of client. Xnsnoop and guest are some well-known R2L attacks.

C. User to Root Attacks (U2R):

This attack is based on authenticating a user to a cloud system. There are several ways of authenticating a valid user in a cloud system based on various information available to him. The authentication process is not very secured in cloud environment. Since user information is stored in distributed manner, data breaching becomes easier making it an easy target to the intruders. Promoting guest user to its maximum range maliciously is a way of cracking the authenticity. Also by using the super user privilege attackers can harm the victim system. Perl, xterm are the examples of U2R.

D. Probing:

In the process of probing, attackers inspect the victim systems about their vulnerabilities such as open port through which the system can be harmed. Mscan, nmap are widely used probing technique. One of the ways of probing in cloud environment is Side channel attack where a malicious VM is created and placed by the side of authenticated VM [33] to make its activity malicious.

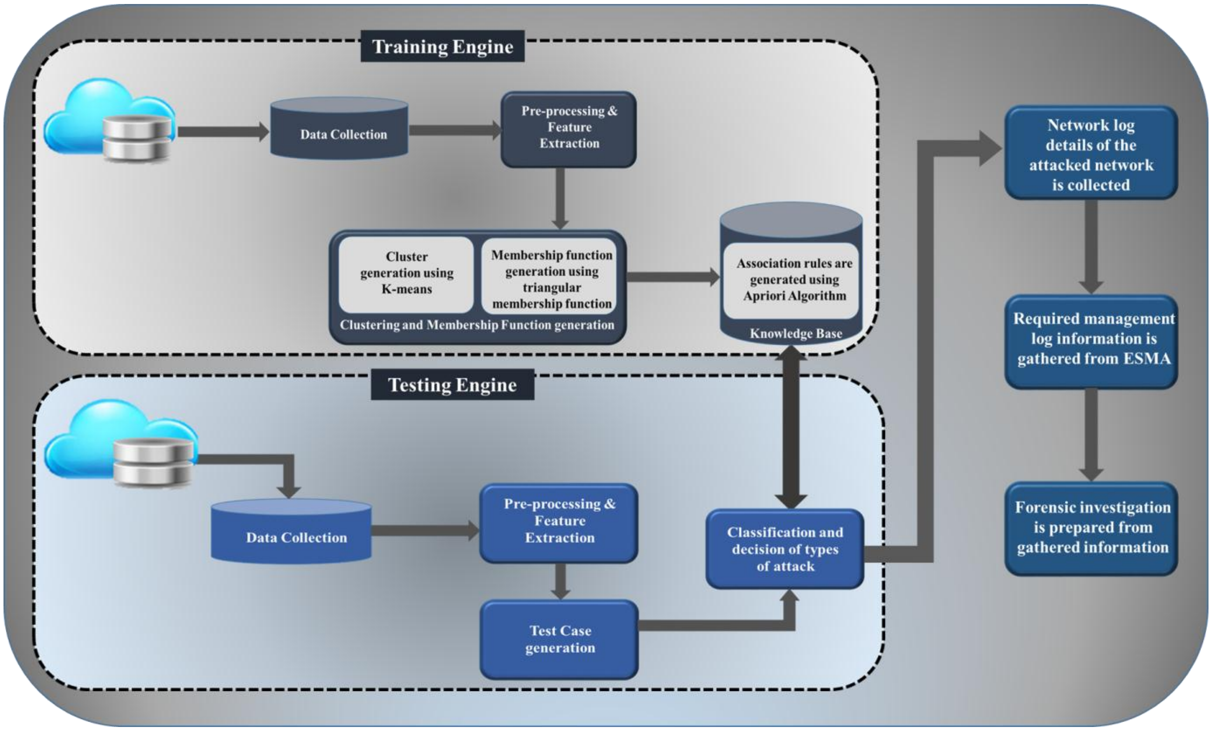

Training Engine
Data Collection
Pre-processing & Feature Extraction
Cluster generation using K-means
Membership function generation using triangular membership function
Clustering and Membership Function generation
Association rules are generated using Apriori Algorithm
Knowledge Base
Testing Engine
Data Collection
Pre-processing & Feature Extraction
Test Case generation
Classification and decision of types of attack
Network log details of the attacked network is collected
Required management log information is gathered from ESMA
Forensic investigation is prepared from gathered information

Figure 2. Training & Testing Engine of FESMA

To identify the various types of attacks in a cloud environment intrusion monitoring system is used. This identifies unwanted incident of a network, which creates internal threat or malicious movement in a system [34-36]. This unwanted incident leads the system to an unstable state. In cloud computing the nature of the attacks are different from normal network attacks. So, to maintain client's private data in cloud environment an effective Malicious Pattern Detection System is desired which will help to build an effective cloud forensic analysis. Here fuzzy theory based attack detection technique is used for identifying the major known attacks in the cloud environment. There are several advantages of choosing the fuzzy theory. Data distribution of every attribute in a cloud transaction is continuous in nature instead of fixed or crisp value. So, fuzzy system is more exact and correct than other machine learning techniques for pattern recognition [27- 29]. The continuous data are clustered to generate rule set for detecting the attack. The rules are based on normal IF, THEN logic. In this case, the rules are simple linguistic terms, which are user friendly and easy to implement. Since cloud is vulnerable to diverse types of attacks, when a new attack pattern appears in the system new rules can be easily added in a fuzzy system without retraining the entire learning engine.

VI. PROPOSED TRAINING & TESTING ENGINE

In the identification phase two modules FESNA and ESMA are proposed which helps in identifying malicious events and hosts by extensively analyzing cloud log records. FESNA inspects network logs whereas ESMA investigates cloud management information for detecting spiteful intruders. Network logs consist of several critical parameters such as “duration”, “service protocol”, “port number” [25][26][28][34] whose signatory value helps in predicting the known major attacks like DoS, R2L, U2R, probing in cloud environment. Cloud management logs consist of scheduling, memory and virtualization based information which helps in detecting the contaminated Virtual Machines.

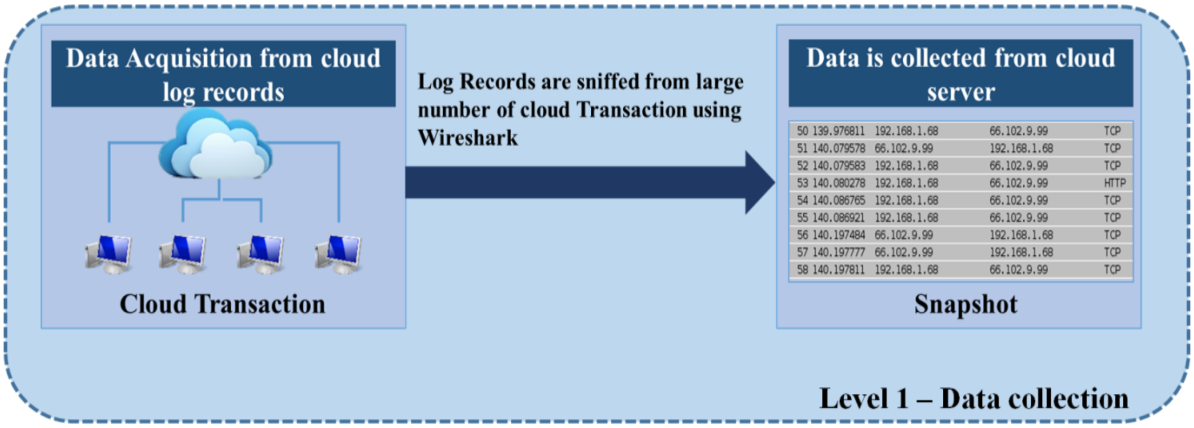
A. Relevance of Log in Cloud Forensic

All the services of cloud (SaaS, PaaS, IaaS) generally use distributed or virtual storage based file system instead of traditional long lived data storage to meet its characteristic features of high scalability and availability. So physical storage device analysis is not effective for multi-tenant cloud architecture where the same high capacity storage devices are simultaneously used by several users in the same time frame. In that case metadata analysis is very crucial for cloud forensics. Various metadata [23] [37] and its acceptance for cloud forensic is discussed in table 1.

Table 1. Meta-Data Analysis

Type of Meta-Data	**Relevance in Cloud Forensics**
Network Log	• Data extracted using network sniffer tools provide detail transaction history of cloud. • Router or gateway dump files provide protocol or service based transaction record.
Cloud Management Log	• IaaS based computational and resource management history makes a clear idea about intrusion. • Scheduling information, waiting queue and size of virtual memory helps to make inference on illegal access.
Hypervisor Log	• As a monitoring tool hypervisor monitors all the running virtual machine (VM) files and directories. • Application Programming Interface (API) monitoring using hypervisor helps to make decision on malicious activity.

Here three types of log files are discussed in relevance to cloud forensics. The log records serve as digital finger prints for forensic analysis [12] [17] [18]. Sniffer extracted network logs are very useful in network forensics which is one of the intrinsic part of the cloud forensic. This type of log data helps in detecting the attacked network. With this information the relevant resources maintained under the attacked network is searched for. Management log, in a cloud server maintains a management module where numerous tasks assigned to several internal resources are retained. There are several parameters of management logs which help in classifying normal and malicious task. VM hypervisor also provides important information for malicious activity on the basis of VM file type, maximum and minimum size of file, access and modification time. Kernel logging activities are also monitored using hypervisor logs. Aim of this work is to build a forensic architecture on the basis of network and management logs which are extracted by popular sniffers.


Data Acquisition from cloud log records
Cloud Transaction
Log Records are sniffed from large number of cloud Transaction using Wireshark
Data is collected from cloud server
Snapshot
Level 1 – Data collection

```
50 139.976811  192.168.1.68    66.102.9.99     TCP
51 140.079578  66.102.9.99     192.168.1.68    TCP
52 140.079583  192.168.1.68    66.102.9.99     TCP
53 140.080278  192.168.1.68    66.102.9.99     HTTP
54 140.086765  192.168.1.68    66.102.9.99     TCP
55 140.086921  192.168.1.68    66.102.9.99     TCP
56 140.197484  66.102.9.99     192.168.1.68    TCP
57 140.197777  66.102.9.99     192.168.1.68    TCP
58 140.197811  192.168.1.68    66.102.9.99     TCP
```

Figure 3. Level 1. Data Collection

VII. PROPOSED FESNA MODEL

FESNA is designed to be a specialized network analyzer where simple dumped network logs are analyzed and malicious user details are provided for forensic analysis. The proposed model is made to behave as a supervised classifier which has the capability of classifying normal and malicious log records. Supervised classifier needs a learning module [27] [35] [36]. Here the learning module is built using a combination of Fuzzy Membership function and A-priori algorithm. Then the testing engine is built to test the cloud log records to identify attacks. The FESNA model is made up of five layers namely – a) Data Collection, b) Preprocessing and feature selection, c) Membership function generation, d) Training of FESNA using automated rule generation, e) Testing FESNA model using cloud log record. The detail work flow of the entire learning and testing engine is described in section 6.2 to 6.3.

A. Level 1 - Data Collection:

Cloud computing is emerging as the fifth utility in recent world [1] [2] which results in huge number of cloud users. The number of service requests a cloud server needs to handle is numerous, resulting in large number of cloud transactions. A dedicated server for log storage is used for managing the history of various transactions [38]. When a malicious activity occurs in the cloud environment the incident reporter [39] reports it to the forensic investigator. The transaction logs during the intruded time period is collected. The dedicated server for network log management provides the log in a timely manner. The collection of such log records is done using "Wireshark" based system, a widely used network log dumping solution for Linux and Windows platform [40] [41].

B. Level 2 - Preprocessing of Dumped log records and Feature Extraction from processed Data model:

1. Preprocessing of Raw Records to Build New Data Model:

These collected logs are generally uncategorized, incomplete and inconsistent in nature. It needs to be preprocessed by which the raw log records become eligible for learning and testing purpose. Redundancy is discarded from the training dataset and it is smoothed by substituting missing value and noise cleaning. Mathematical formulation of both substituting missing value and noise removal is shown in the equation (1) and (2) respectively. Understandable and analyzable training data set is produced as result of preprocessing. In the equation (1), x_i is the missing value and m is the total number of distribution in that particular attribute. Whereas x in equation (2) determines the noise value in dataset and β is a noise threshold above which attribute value are treated as noise and need to be smoothed. β depends on the data distribution of the dataset.

$$f(x_i) = \begin{cases} \frac{\sum_{i-n/10}^{i+n/10} x_i}{2\left(\frac{n}{10}\right)+1}, & x_i = missing/abnormal\ value \\ x_i, & x_i \neq missing/abnormal\ value \end{cases} \quad (1)$$

$$f(x_i) = \begin{cases} x_i - \ mean(x_1, x_2, x_3, x_4, \ldots\ldots x_m), & (x_i - mean(x_1, x_2, x_3, x_4, \ldots\ldots x_m)) > \beta \\ x_i, & (x_i - mean(x_1, x_2, x_3, x_4, \ldots\ldots x_m)) \leq \beta \end{cases} \quad (2)$$

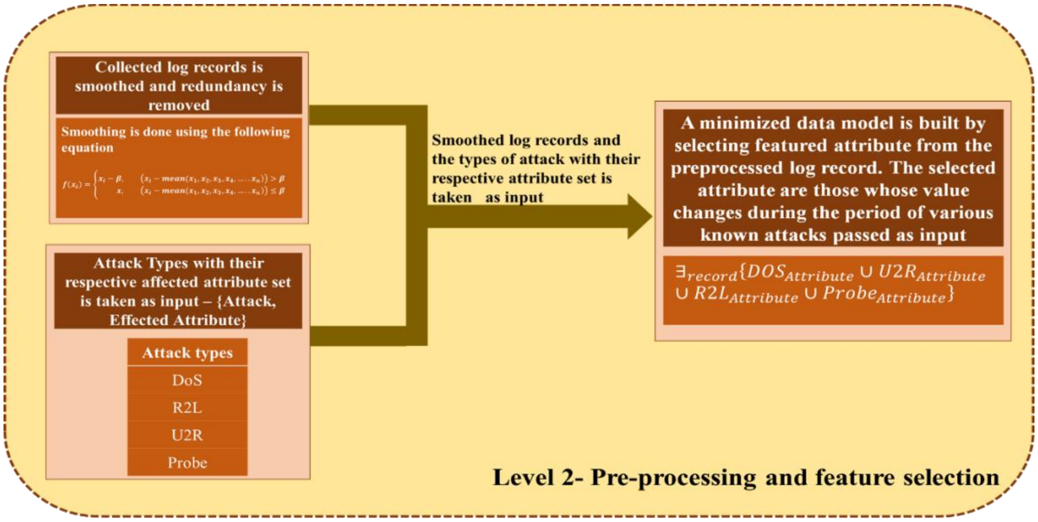

Collected log records is smoothed and redundancy is removed
Smoothing is done using the following equation
$f(x_i) = \begin{cases} x_i - \beta, & (x_i - mean(x_1, x_2, x_3, x_4, \ldots\ldots x_n)) > \beta \\ x, & (x_i - mean(x_1, x_2, x_3, x_4, \ldots\ldots x_n)) \leq \beta \end{cases}$
Attack Types with their respective affected attribute set is taken as input – {Attack, Effected Attribute}
Attack types
DoS
R2L
U2R
Probe
Smoothed log records and the types of attack with their respective attribute set is taken as input
A minimized data model is built by selecting featured attribute from the preprocessed log record. The selected attribute are those whose value changes during the period of various known attacks passed as input
$\exists_{record}\{DOS_{Attribute} \cup U2R_{Attribute} \cup R2L_{Attribute} \cup Probe_{Attribute}\}$
Level 2- Pre-processing and feature selection

Figure 4. Level 2. Preprocessing & Feature Selection

2. Feature extraction from the preprocessed Data Model:

Cloud log records consist of large number of attribute sets. Analyzing every attribute of the processed data set may not be equally effective for detection of different types of attack. Cloud System is generally prone to the following network attacks DoS, R2L, U2R and Probe. The attributes that are majorly affected for the said attack need to be extracted from the preprocessed data set to generate various fuzzy rules. Generating fuzzy rules with such huge number of attributes will result in a complex logical reasoning for attack detection. This will affect the detection accuracy as well as Quality of Service (QoS) in cloud environment. So important features need to be selected from the pre-processed data model to improve the efficiency of the system. Detail attack wise attribute set is given in Table 2 according to several attribute selection strategies [25] [26] [28] [34] [42] [43]. Feature extraction process is referred as in figure 4.

Table 2. Extracted Features and corresponding attack

Sl. No.	Attribute Name	DoS	R2L	U2R	Probe
1	duration	✓	✓		✓
2	protocol_type	✓	✓		✓
3	service		✓		✓
4	flag	✓	✓		✓
5	src_bytes	✓	✓		✓
10	hot		✓	✓	
11	num_failed_logins		✓		
12	logged_in		✓		
13	num_compromised		✓	✓	
14	root_shell			✓	
16	num_root			✓	
17	num_file_creations		✓	✓	
18	num_shells		✓	✓	
19	num_access_files		✓	✓	
22	is_guest_login		✓		
23	count	✓			
34	dst_host_same_srv_rate	✓			
38	dst_host_serror_rate	✓			
39	dst_host_srv_serror_rate	✓			

C. Level 3 - Clustering and Membership function generation:

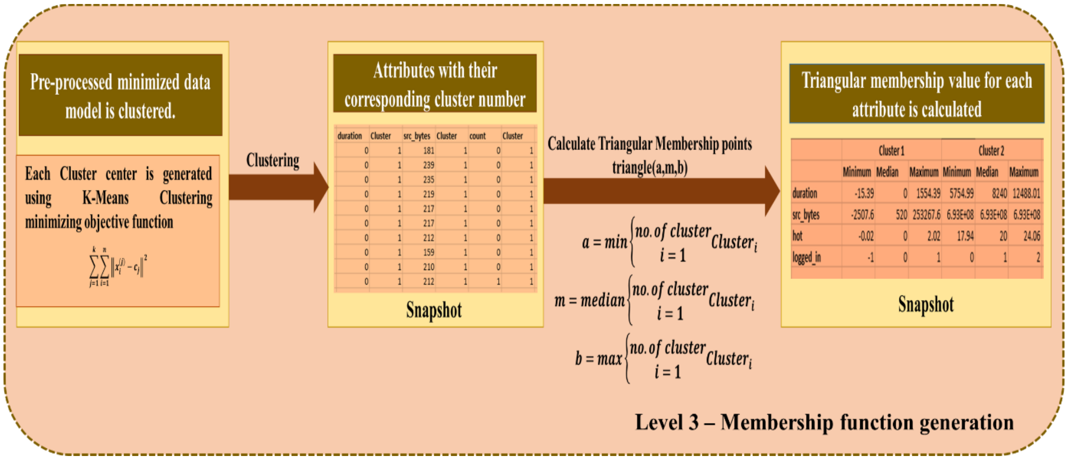
1. Clustering of attribute values

The extracted attributes in the pre-processed data set has continuous range of values. Discretization of these continuous value need to be done to generate individual cluster. These clusters will serve as the antecedent of various fuzzy rules to be applied in FESNA. For example, an attribute X has a continuous distribution {0.001 – 0.009} this distribution need to be discretized for generating every cluster. So X is broken into 3 clusters- a) Cluster 1- {0.001-0.003}, b) Cluster 2- {0.004-0.006}, c) Cluster 3- {0.007-0.009}.In the preprocessed dataset each attribute belongs to a number of states based on its value. Cluster for every attribute state need to be generated. In our work, we have used traditional k-means algorithm [44] for clustering the data set to distinguish state of each attribute individually. K-means is the simplest unsupervised learning technique with minimum time complexity [45] that can solve most of the simple clustering problems. Clustering the data set with minimum computational time is desired. Since k-means algorithm needs almost half execution time compared to Fuzzy c-means algorithm [46] so we chose k-means algorithm for clustering purpose. In the very initial stage k-means algorithm needs to determine the value of k. For our purpose the value of k is assigned to 5 as the attribute values are broadly classified into 5 major classes- a)Very Low, b)Low, c)Medium, d)High, e)Very High. The cluster members can be calculated by minimizing equation (3), where $\left\|x_i^{(j)} - c_j\right\|$ is the euclidian distance of each data item $x_i^{(j)}$ from its center. k is the total number of cluster centers and n is the total number of data points. According to the cluster center each attribute range is classified into different clusters as shown in Table 3.

$$arg\,\min_i \sum_{j=1}^{k} \sum_{i=1}^{n} \left\|x_i^{(j)} - c_j\right\|^2 \qquad (3)$$

Table 3. Cluster wise data distribution

Attribute Name	Range	Cluster1 (Very Low)	Cluster2 (Low)	Cluster3 (Medium)	Cluster4 (High)	Cluster5 (Very High)
duration	0-58658	0-1554	1501-5841	5755-12488	12331-24872	25091-58658
src_bytes	0-5135720	0-253272	454497-717822	2100423-2504014	3131425-393375641	5131381-5135720
hot	0-30.02	0-2.02	2.94-9.06	9.93-17.07	17.94-24.06	27.98-30.02
num_failed_logins	0-5	0-1	1-2	1-3	2-4	3-5
logged_in	0-2	0-1	0-1	0-1	0-2	0-2
num_compromised	0-885	0-22	38-103	238-282	766-768	883-885
root_shell	0-2	0-1	0-1	0-1	0-1	0-2
num_root	0-995	0-3	4-55	118-120	268-307	855-995
num_file_creations	0-28	0-1	1-3	4-8	9-16	20-28
num_shells	0-3	0-1	0-1	0-1	0-2	1-3
num_access_files	0-8	0-1	0-2	2-3	3-5	5-8
is_guest_login	0-2	0-1	0-1	0-1	0-1	0-2
count	0-512	0-66	65-172	171-246	245-390	389-512
dst_host_same_srv_rate	0-1	0-1	0-1	0-1	0-1	0-1
dst_host_serror_rate	0-1	0-0.05	0.0586-0.2014	0.2078-0.4322	0.4366-0.7834	0.7879-1
dst_host_srv_serror_rate	0-1	0-0.0404	0.0480-0.2520	0.2872-0.5728	0.5974-0.8626	0.8687-1


Pre-processed minimized data model is clustered.
Clustering
Attributes with their corresponding cluster number
Snapshot
Calculate Triangular Membership points triangle(a,m,b)
Triangular membership value for each attribute is calculated
Snapshot
Level 3 – Membership function generation

Each Cluster center is generated using K-Means Clustering minimizing objective function

$$\sum_{j=1}^{k}\sum_{i=1}^{n}\left\|x_i^{(j)}-c_j\right\|^2$$

duration	Cluster	src_bytes	Cluster	count	Cluster
0	1	181	1	0	1
0	1	239	1	0	1
0	1	235	1	0	1
0	1	219	1	0	1
0	1	217	1	0	1
0	1	217	1	0	1
0	1	212	1	0	1
0	1	159	1	0	1
0	1	210	1	0	1
0	1	212	1	1	1

$$a = min\left\{\begin{matrix} no.of\ cluster \\ i=1 \end{matrix}\right. Cluster_i$$

$$m = median\left\{\begin{matrix} no.of\ cluster \\ i=1 \end{matrix}\right. Cluster_i$$

$$b = max\left\{\begin{matrix} no.of\ cluster \\ i=1 \end{matrix}\right. Cluster_i$$

	Cluster 1			Cluster 2		
	Minimum	Median	Maximum	Minimum	Median	Maximum
duration	-15.39	0	1554.39	5754.99	8240	12488.01
src_bytes	-2507.6	520	253267.6	6.93E+08	6.93E+08	6.93E+08
hot	-0.02	0	2.02	17.94	20	24.06
logged_in	-1	0	1	0	1	2

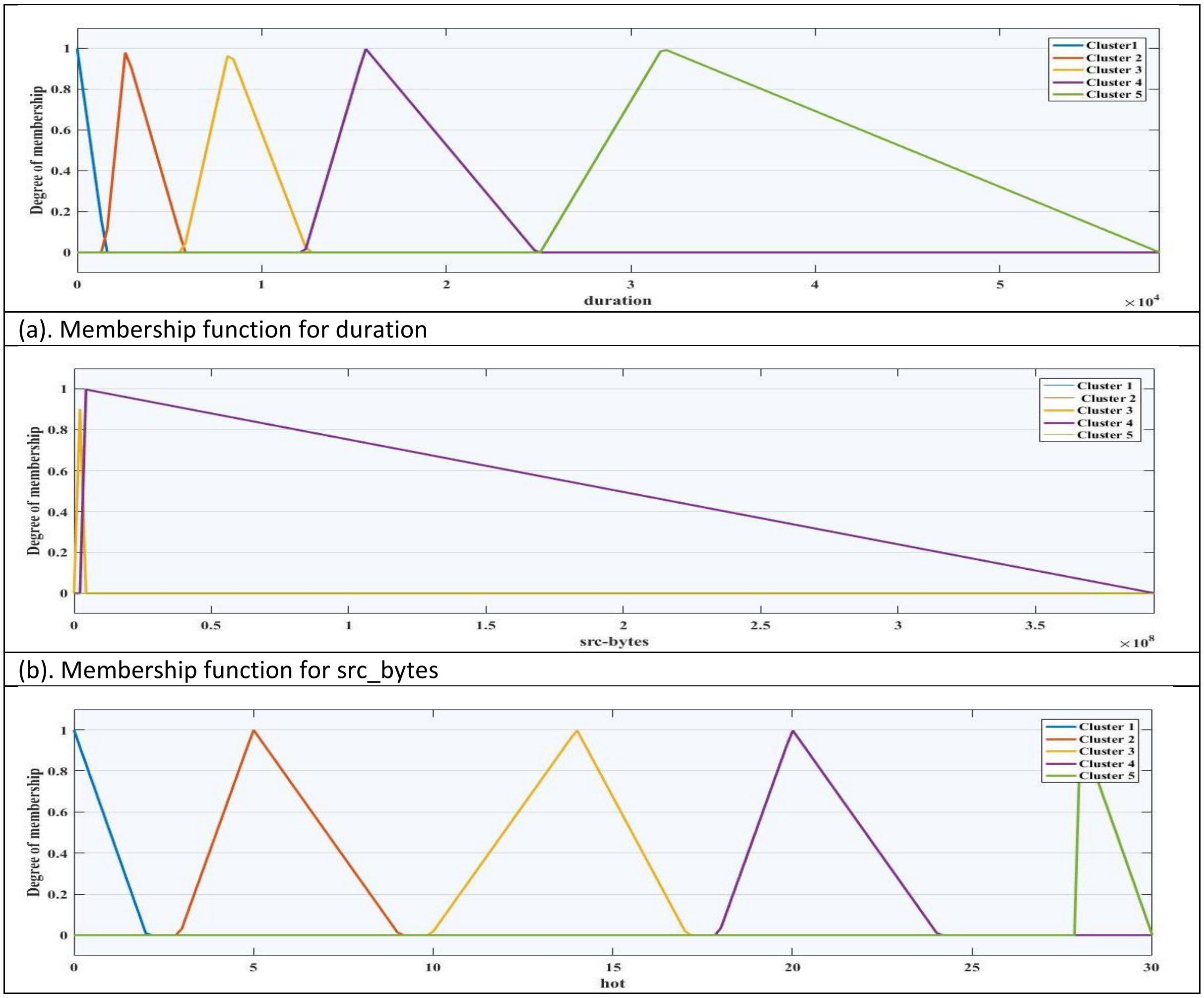
Figure 5. Level 3. Membership Function Generation

2. Generate Membership Function

Fuzzy inference engine works on fuzzy membership function and its associated rules [26-29]. So, before defining the rule set a proper membership function need to be designed for each attribute. There are several types of membership functions from which triangular, trapezoidal and Gaussian are widely used for generating inference engine. In our work triangular membership function is used. Each cluster of an attribute is converted to its corresponding triangular membership function by computing Min, Median & Max (MMM) of the cluster values. According to the theory of descriptive statistics five number summary is capable of providing information about the entire data set. Five number summary states that if the entire data set is sampled in five percentiles it will give information about the entire data set The five percentiles are – a) The smallest value of the sample, b) lower quartile which is the center for lower range of data set, c) the middle value, d) upper quartile referred to as the center of the upper range of dataset and e) the largest value of the dataset [47]. For our case each cluster represents a data set. So by using MMM process all the five sample of the cluster can be computed easily. Therefore, each triangular membership function is capable of giving information about the entire cluster. The MMM is calculated using equation (4). Triangular membership function has the three parameters$a$,b and m where a and b are referred to as the two end points and m is the middle point. The triangular membership functions are generated using Matlab 2014b. The membership function for each attribute is shown in Figure 6. The proposed FESNA model is trained with the generated membership functions.

For each Cluster

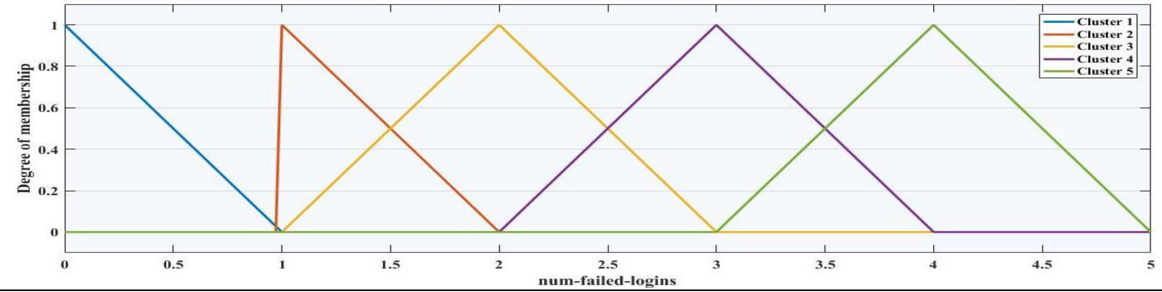
$$\begin{aligned} a &= min(Element_1, Element_1, Element_1, \dots, Element_n) \\ m &= median(Element_1, Element_1, Element_1, \dots, Element_n) \\ b &= max(Element_1, Element_1, Element_1, \dots, Element_n) \end{aligned} \quad (4)$$



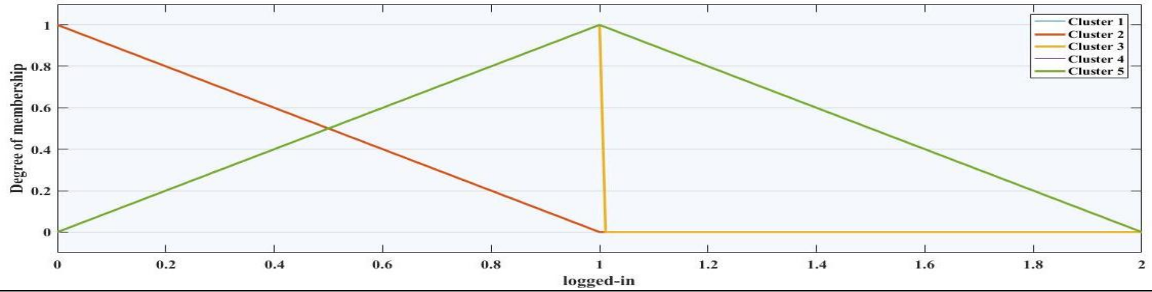
(a). Membership function for duration

(b). Membership function for src_bytes

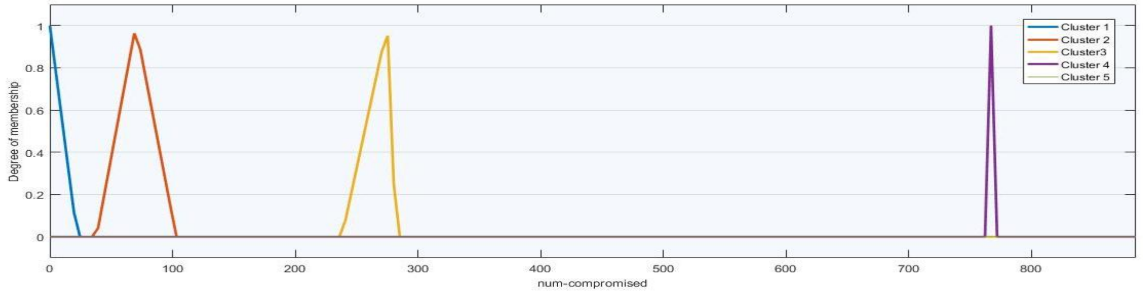
(c). Membership function for hot



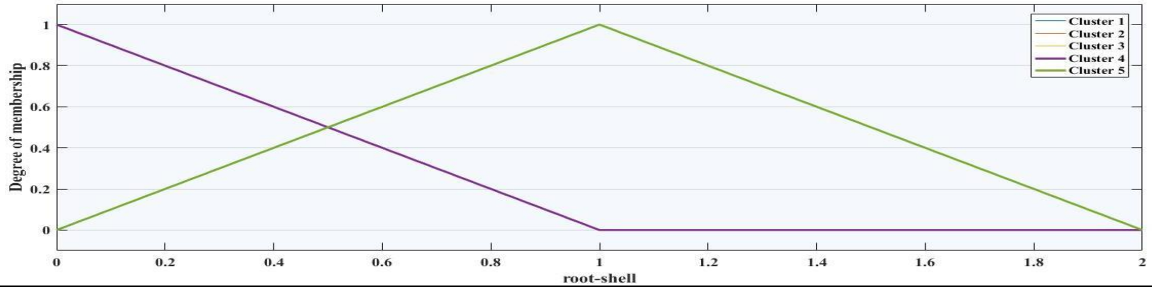
(d). Membership function for num_failed_logins



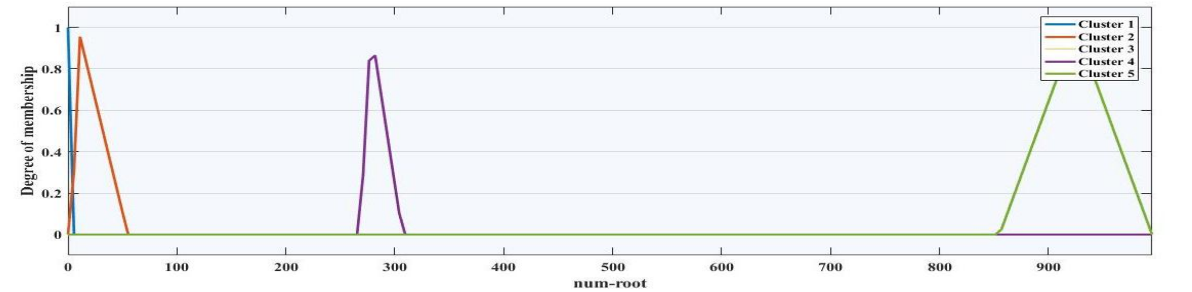
(e). Membership function for logged_in



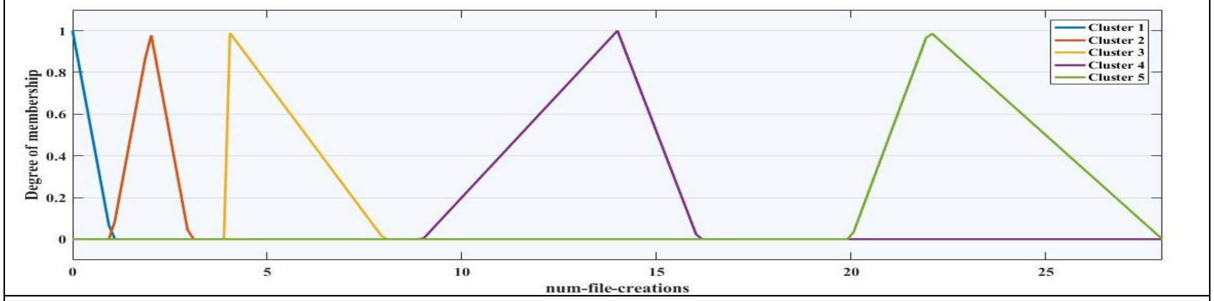
(f). Membership function for num_compromised



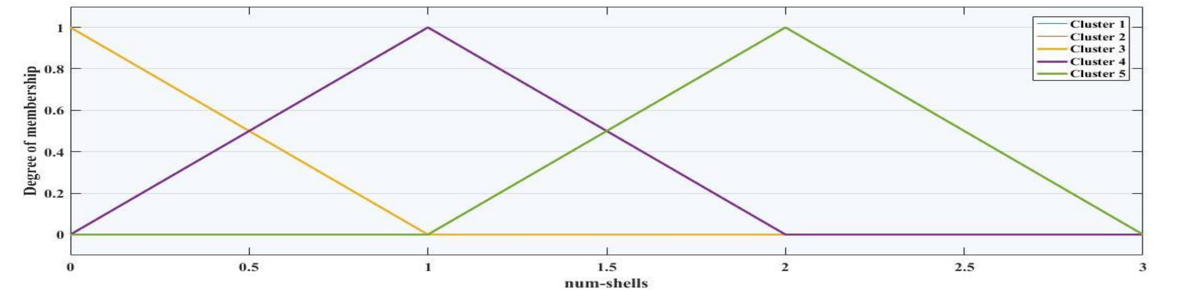
(g). Membership function for root_shell



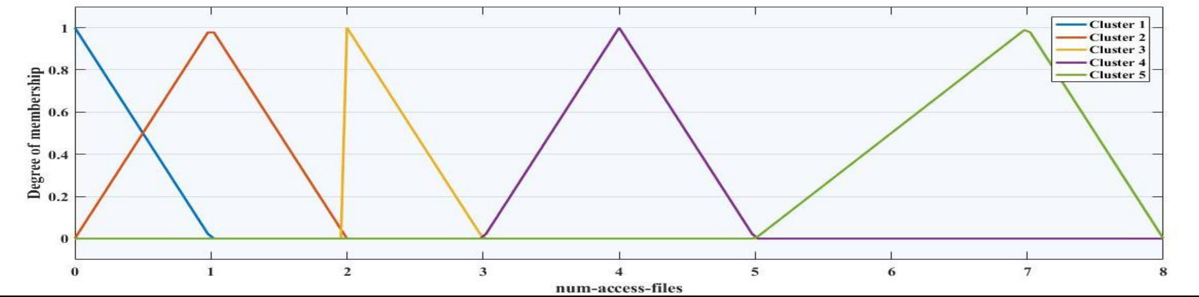
(h). Membership function for num_root



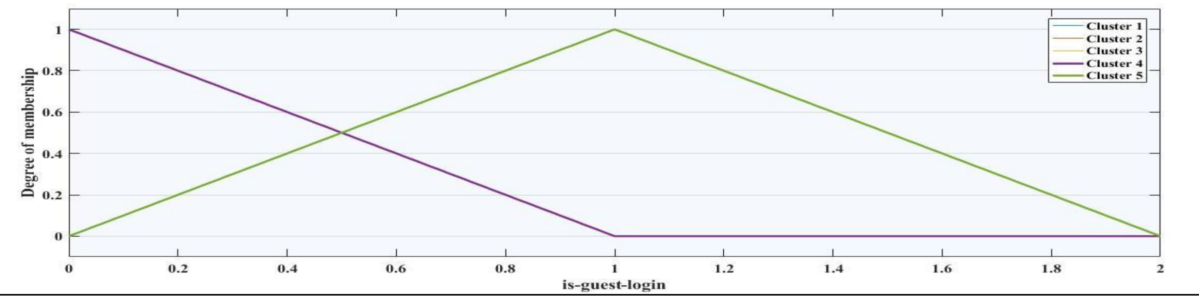
(i). Membership function for num_file_creation

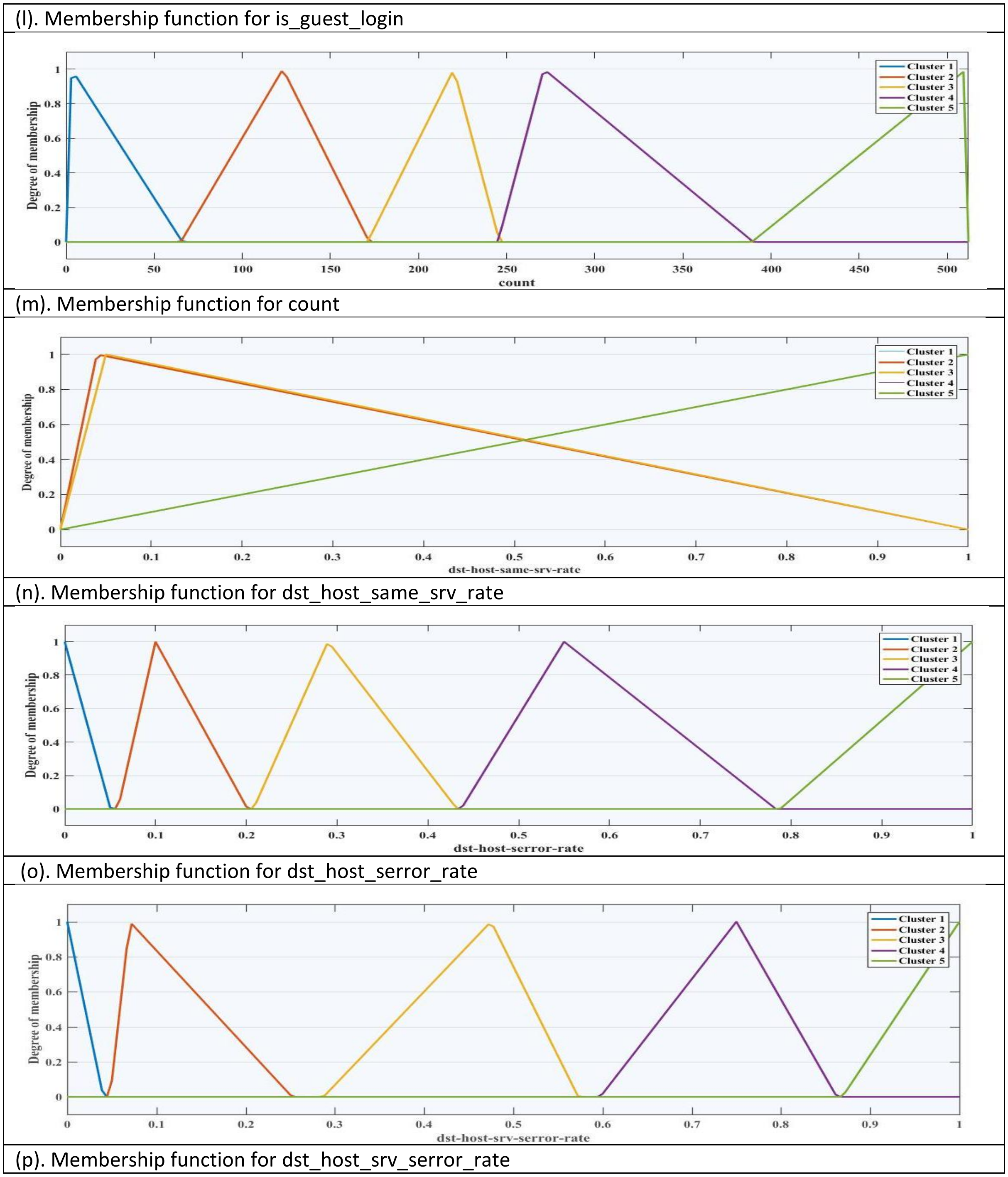


(j). Membership function for num_shells



(k). Membership function for num_access_files





(l). Membership function for is_guest_login

(m). Membership function for count

(n). Membership function for dst_host_same_srv_rate

(o). Membership function for dst_host_serror_rate

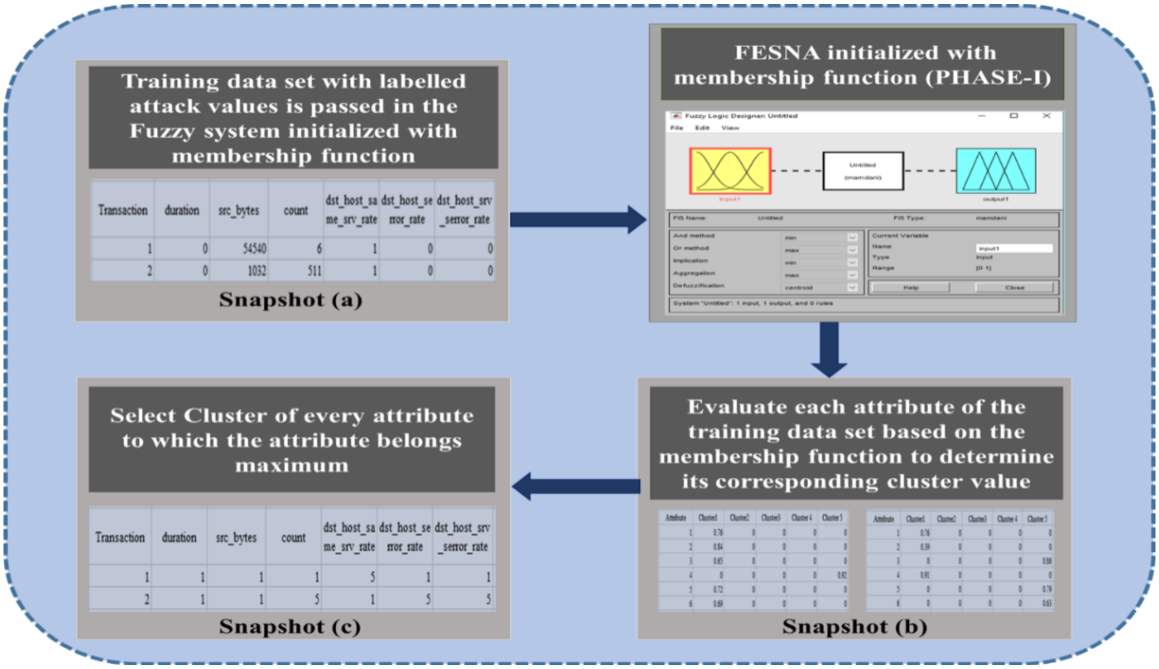
(p). Membership function for dst_host_srv_serror_rate

Figure 6 (a-p). Membership Function Sample

D. *Level 4 - Training of FESNA using automated rule generation*

1. Pattern Generation

To make an effective intrusion monitoring system a specific pattern for every attack needs to be identified. In our proposed FESNA model every known attack is considered for building the knowledge base of the fuzzy system. Knowledge base of fuzzy system is nothing but a set of pattern or rule. So in this phase the fuzzy system, FESNA is initialized with the various triangular membership functions of the identified attribute values obtained in the previous step. After that, known attack records are processed through the membership value initialized FESNA system. As a result, each data item is then converted to its corresponding fuzzy values depending upon the membership function. For example, attribute X has 3 membership functions (Cluster 1, Cluster 2, and Cluster 3). X has its item value to be 56000 when passed through the system it generates corresponding fuzzy values (Cluster 1: 0, Cluster 2: 0.78, Cluster 3: 0.23). From the above fuzzy values the system will choose only maximum value with its corresponding membership function.



Training data set with labelled attack values is passed in the Fuzzy system initialized with membership function

Transaction	duration	src_bytes	count	dst_host_same_srv_rate	dst_host_serror_rate	dst_host_srv_serror_rate
1	0	54540	6	1	0	0
2	0	1032	511	1	0	0

Snapshot (a)

FESNA initialized with membership function (PHASE-I)

Fuzzy Logic Designer: Untitled

FIS Name: Untitled FIS Type: mamdani

And method: min
Or method: max
Implication: min
Aggregation: max
Defuzzification: centroid

Current Variable
Name: input1
Type: input
Range: [0 1]

System "Untitled": 1 input, 1 output, and 0 rules

Evaluate each attribute of the training data set based on the membership function to determine its corresponding cluster value

Attribute	Cluster1	Cluster2	Cluster3	Cluster 4	Cluster 5
1	0.76	0	0	0	0
2	0.84	0	0	0	0
3	0.65	0	0	0	0
4	0	0	0	0	0.92
5	0.72	0	0	0	0
6	0.69	0	0	0	0

Attribute	Cluster1	Cluster2	Cluster3	Cluster 4	Cluster 5
1	0.76	0	0	0	0
2	0.39	0	0	0	0
3	0	0	0	0	0.86
4	0.91	0	0	0	0
5	0	0	0	0	0.79
6	0	0	0	0	0.63

Snapshot (b)

Select Cluster of every attribute to which the attribute belongs maximum

Transaction	duration	src_bytes	count	dst_host_same_srv_rate	dst_host_serror_rate	dst_host_srv_serror_rate
1	1	1	1	5	1	1
2	1	1	5	1	5	5

Snapshot (c)

Figure 7. Pattern Generation

In Figure 7, training data set is passed through the FESNA (PHASE-I) where the expert system is initialized only with the membership function generated in the previous step. Transaction 1 and 2 as shown in the snapshot (a) has six different attribute values. Each of the attributes of every transaction is passed through FESNA (PHASE I) and mapped with its corresponding membership function. Each attribute consists of five different membership function (Cluster 1, Cluster 2, Cluster3, Cluster4, Cluster5). So for every attribute in the transaction five different membership values are obtained as shown in the snapshot (b). From the generated list for every transaction the cluster number with maximum membership value for each attribute is chosen as shown in snapshot (c). In this manner each of the transactions is analyzed to generate a pattern.

2. Association Rule Generation Using Apriori Algorithm

Data mining algorithms are generally used for extracting the required information from a large dataset. If rules for the proposed Fuzzy system is built using all the generated patterns from the processed data model it will not result in an efficient knowledge base. Determining effective rules for a fuzzy system is necessary to avoid biased decision is an important requirement. In this context, an effective data mining algorithm is needed to build a proficient knowledge base with appropriate rules for fuzzy based cloud forensic analysis system. Well-known data mining technique apriori algorithm [48] is used for extracting the effective rules.
Apriori algorithm is a mining technique for association rule mining. Using bottom-up approach, it finds the frequent patterns in a dataset. It uses breath first search and hash tree structure for rule mining [48]. Apriori algorithm finds out effective association rules with best combination of different attributes from a large dataset. Working principle of the algorithm is driven by two parameters support and confidence value [49]. It measures the number of occurrences of certain parameter in a set of transactions. A minimum threshold value (Minimum support to qualify) for this support identifies the best rules.

As cloud system deals with large number of network connections hence huge number of network logs need to be analyzed. Different types of attacks in cloud follows different attack pattern. In that case, association rule mining is needed to identify effective rules. In this phase, aprioi data mining algorithm is used for identifying the effective rules with which the knowledge base can be trained. Usually for network forensic, association rules are manually inserted on the basis of investigation. However, cloud system deals with large number of log records. So, automatic creation and insertion of association rules in the fuzzy system is desired.

The two parameters, support and confidence of apriori algorithm works as follows.

Support: Support of rule A→B is the proportion of transaction where the full transaction set contains A→B where, A and B are attributes in the dataset. Frequent repetition of such attributes can be determined using support value [50] [51]. Calculation of support value works as in equation (5).

$$support(A \rightarrow B) = \frac{A \cup B}{N} \qquad (5)$$

Where, $(A \rightarrow B)$ represents the number of transaction containing all the item in the rule and N represents the total number of transaction. The parameter support is used to measure how frequently a set of items is occurred.

Confidence: Confidence of rule $(A \rightarrow B)$ is the proportion of transaction where A occurs on the occurrence of B. The confidence values are calculated using the equation (6).

$$confidence(A \rightarrow B) = \frac{support(A \rightarrow B)}{support(A)} \qquad (6)$$

It is a very important to measure the effectiveness of a rule in the system. Antecedent and consequent of a rule can be calculated using support and confidence value [50] [51]. The most effective rules are selected on the basis of threshold β_1 and β_2 which is minimum support and confidence value respectively referred in equation (8).

$$\begin{aligned} &Rule(A \rightarrow B) = Selected, \quad if\ Support, Confidence \geq \beta_1, \beta_2 \\ &Rule(A \rightarrow B) = Not\ Selected,\ if\ Support, Confidence < \beta_1, \beta_2 \end{aligned} \qquad (7)$$

Determining a proper threshold value β_1 *and* β_2 is very important for a system. Since too low threshold value will pass a large number of useless pattern and also too high value will cause loss of useful pattern. Statistical average of support and confidence value referred in equation (8) helps in determining the threshold β_1 *and* β_2 where T is the total number of transactions [50] [51].

$$\beta_1 = \frac{\sum support(A \rightarrow B)}{T},\ \beta_2 = \frac{\sum confidence(A \rightarrow B)}{T} \qquad (8)$$

Table 4 shows a sample rule generation for the proposed FESNA system with their corresponding Support and Confidence value. For this data model value of $\beta_1$and β_2 is set to 0.6 and 0.45 respectively. So rule (1,4) qualifies to the rule set of the training engine whereas rule (2,3) is not introduced in the rule set.

The association rules forms the knowledge base of the automated FESNA system. It is capable of detecting malicious incidents of exact as well as anomalous signature. The knowledge base can be updated dynamically when a forensic investigator has identified a new pattern of attack.

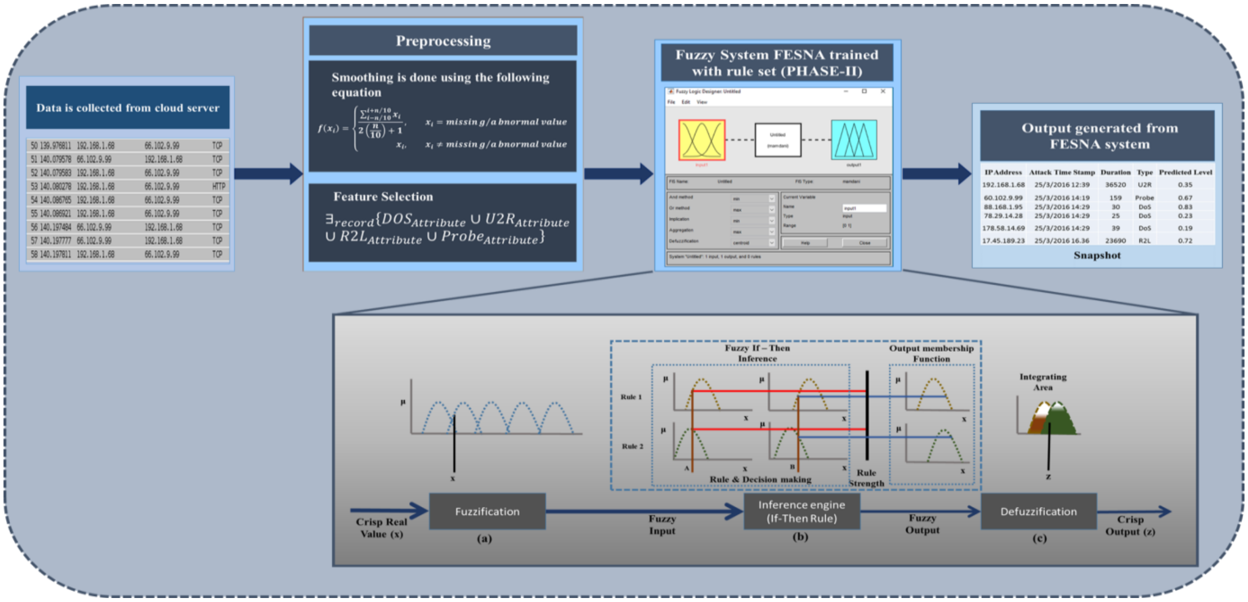
E. Level 5 - Testing Engine of FESNA

In the testing part of the system, above-mentioned proposed model FESNA is tested with different test cases generated from our cloud record. The test matrix of log record has been prepared followed by two initial steps- a) data collection and b) preprocessing. Each attribute of the log record is calculated as per its membership function and fuzzy rule base mapping to determine an attack. The preprocessed test cases are passed through the trained FESNA knowledge base and the attack identification is done based on the fuzzy knowledge base if-then rule. The classification of the test cases are done according to classical aggregation and implication theory. The rules of knowledge base consist of two basic logical operations AND, OR on fuzzy linguistic term. In the application of fuzzy theory, operator AND is denoted as minimum of membership values whereas OR determines maximum of membership values.

The preprocessed data set is passed into the Fuzzy Knowledge Base. The real value of every attribute is fuzzified by mapping it to its corresponding fuzzy membership value. This generates a fuzzy value for each of the corresponding attributes which is then passed into the FESNA knowledge base. The fuzzy inputs are combined together based on the fuzzy If-Then-Else rules to generate the strength of a rule. The output membership function is plotted as a surface area based on the generated rule strength. Integrating the various output membership functions surface areas center of mass or centroid is calculated to get the proper defuzzified crisp value. This crisp value determines the attack type with its level of harm. The complete testing phase works as in the figure 8.

Table 4. Rule Selection

Sl. no	Rules	Support (antecedent and consequent)	Support (antecedent)	Confidence
1	If (duration=very low) AND (src_bytes=very low) then attack=dos	0.4578	0.9352	0.489521
2	If (duration=very low) AND (dst_host_same_srv_rate=very high) then attack=dos	0.2613	0.5989	0.4363
3	If (src_bytes=low) AND (dst_host_same_srv_rate=high) then attack=dos	0.4829	0.6358	0.759516
4	If (src_bytes=low) AND (dst_host_same_srv_rate=high) AND (dst_host_serror_rate=very high)then attack=dos	0.3119	0.6819	0.457398


Data is collected from cloud server
Preprocessing
Smoothing is done using the following equation
Feature Selection
Fuzzy System FESNA trained with rule set (PHASE-II)
Fuzzy Logic Designer: Untitled
File
Edit
View
Untitled
(mamdani)
input1
output1
Help
Close
Output generated from FESNA system
Snapshot
Fuzzy If – Then Inference
Output membership Function
Integrating Area
Rule 1
Rule 2
μ
x
A
B
z
Rule & Decision making
Rule Strength
Crisp Real Value (x)
Fuzzification
(a)
Fuzzy Input
Inference engine (If-Then Rule)
(b)
Fuzzy Output
Defuzzification
(c)
Crisp Output (z)

50	139.976811	192.168.1.68	66.102.9.99	TCP
51	140.079578	66.102.9.99	192.168.1.68	TCP
52	140.079583	192.168.1.68	66.102.9.99	TCP
53	140.080278	192.168.1.68	66.102.9.99	HTTP
54	140.086765	192.168.1.68	66.102.9.99	TCP
55	140.086921	192.168.1.68	66.102.9.99	TCP
56	140.197484	66.102.9.99	192.168.1.68	TCP
57	140.197777	66.102.9.99	192.168.1.68	TCP
58	140.197811	192.168.1.68	66.102.9.99	TCP

$$f(x_i) = \begin{cases} \dfrac{\sum_{i-n/10}^{i+n/10} x_i}{2\left(\frac{n}{10}\right)+1}, & x_i = missing\,g/a\,bnormal\ value \\ x_i, & x_i \neq missing\,g/a\,bnormal\ value \end{cases}$$

$$\exists_{record}\{DOS_{Attribute} \cup U2R_{Attribute} \cup R2L_{Attribute} \cup Probe_{Attribute}\}$$

FIS Name: Untitled FIS Type: mamdani

And method	min	Current Variable	
Or method	max	Name	input1
Implication	min	Type	input
Aggregation	max	Range	[0 1]
Defuzzification	centroid		

System "Untitled": 1 input, 1 output, and 0 rules

IP Address	Attack Time Stamp	Duration	Type	Predicted Level
192.168.1.68	25/3/2016 12:39	36520	U2R	0.35
60.102.9.99	25/3/2016 14:19	159	Probe	0.67
88.168.1.95	25/3/2016 14:29	30	DoS	0.83
78.29.14.28	25/3/2016 14:29	25	DoS	0.23
178.58.14.69	25/3/2016 14:29	39	DoS	0.19
17.45.189.23	25/3/2016 16.36	23690	R2L	0.72

Figure 8. Testing Engine of FESNA

VIII. Proposed ESMA Model

In the identification phase of the proposed forensic model, ESMA is proposed to detect contaminated resources of cloud environment by analyzing management log information. The management log is maintained by a dedicated server in the cloud system. It holds the log record of the schedulers and hypervisors present in the cloud server. According to the task request and VM activity, this log server is updated at each time interval. So the various management log information such as VM_Detail, User_Detail, API_Detail and Process_Detail are available for analysis. In this module of ESMA, this information is analyzed to detect the contaminated VM, user, process and API. VM_Detail holds the information about the associated VM including VM Identification, VM IP, Average memory usage and Average Virtual memory usage. User_Detail holds relevant information about requesting user including User Identification, User Name, User Email and their subscription. Application Program Interface (API) is one of the critical elements for cloud services since it provides gateway or interface for various cloud based services. API_Detail holds information of associated API including API identification, API name, functionality and process served by that particular API. Information processed in the cloud server is held by Process_Detail including Process Identification, Process Name and Created VM. ESMA takes identified malicious network details – Infected_IP, AttackTimeStamp, AttackDuration, AttackType as input identified by FESNA.

The processing of ESMA is based on pure logical reasoning with help of the output from FESNA as referred in table 6.

Table 5. ESMA Log Structure

ESMA LOG ENTITIES	PARAMETER	
IP V4/V6 address	IP	
Duration	Access_Duration	
Timestamp	Access_Timestamp	
Virtual Machine Detail	VM_Identification	
	VM_IP	
	Average_Memory_Usage	
	Average_Virtual_Memory_Usage	
User Detail	User_Detail	User_Identification
		User_Name
		User_Email
		User_Subscription
API Detail	API_Detail	API_Identification
		API_Name
		Function
		Served_Process
Process Detail	Process_Detail	Process_Identification
		Process_Name
		Created_By_VM

Here malicious VM is identified using total information gain in packet transfer from malicious VM to victim in case of DDoS and Probing. Since R2L and U2R attack is based on unregistered process and user, it only concentrates on the above two entities. Generally, total information gain is calculated using entropy of that particular attribute. In the following algorithm, malicious VM is recognized on the basis of entropy of VM packet transfer. In situation of DDoS and Probing, attacker has been identified using threshold of mean entropy of packet transfer for all of the running VMs in the system at that time. This algorithm also collects all the unregistered processes which were running at that time instance. This operation generally occurs when R2L attack is encountered. In the situation of U2R attack authentication breaching is identified and the intruder user information are captured. In case of all of the attack involved remote process and API are also captured to enhance further forensic analysis.

IX. Forensic Investigation Report Generation

Forensic investigation report (FIR) is generally a data packet of descriptor, which describes the result of the proposed automated system FESNA and ESMA. This auto-generated descriptor has the ability to proof the malicious activity and the intruder primarily. Further steps can be taken based on this descriptor. Header part determines the investigation's Meta data such as Evidence identification and size, Issuing Data Time. Digital Signature Detail contains the digital signature of the investigator making it more reliable to the end user as well as to the court of law. Details part signifies details of analyzed malicious activity along with intruder. It also implies related information about VM, User subscription, API and process running in the system.

Evidence generation is an important part of forensic analysis because without any type of evidence, crime cannot be established in court of law. So a well-organized easily understandable evidence is desired. In this proposed cloud forensic model, evidence generation is done in the analysis phase where investigator collects needed information from dumped VM file system. This evidence can be produced in court of law against any cyber as well as criminal cases. The collection of the information is dependent on the nature of complains or cases. However, investigator need to take permission from court or customer for such data collection according to previously signed Service Level Agreement (SLA) [52] [53]. So an investigation warrant is needed for such data collection and investigation purpose. The FIR will provide such warrant based on automatic identification of malicious activity. FIR can also be produced as low level and intermediate evidence for emergency purpose. Customer account suspension and blocking can be done during this emergency period according to SLA. Figure 9 describes the complete format of FIR.

Table 6. Processing of ESMA

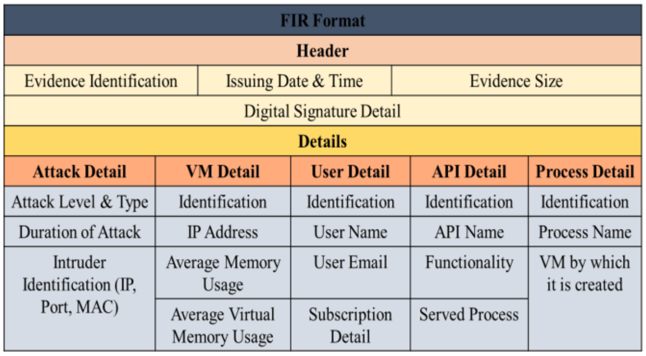
Input Considerations = {Infected_IP : The detected IP by FESNA as a possible attacker
AttackTimestamp: The timestamp value when the attack took place
AttackDuration: The duration of the attack period
AttackType: The type of attack predicted by FESNA}
Output Considerations = {MaliciousVMInfo, UserInfo, MaliciousProcessDetail, APIInfo}

Procedure:

```
Step 1 : Begin
Step 2 : Repeat
Step 3 : For each j in ESMA log  do
Step 4 : Read ESMAlog(j);
Step 5:     If (FESNAout.Infected_IP==ESMAlog(j).IP) AND
            (FESNAout.AttackDuration==ESMAlog(j).Access_Duration) AND
            (FESNAout.Attack_Timestamp==ESMAlog(j).Access_Timestamp) do
    /* DOS attack checked and corresponding malicious VM, the API used and user info are collected*/
Step 6:         If (FESNAout.AttackType==DoS)
Step 7:             For each i=1 to No_of_Associated_VM do
                        If (Entopy(VMDetail.PacketTransfer)>Mean(Entropy for all VM)) then
          /* Entropy is calculated using $E = -\sum_{i=1}^{n} P(pt) log P(pt)$* where $pt$ is packet transfer/
Step 8:                     MaliciousVMInfo= MaliciousVMInfo U VMInfo(i);
Step 9:                     APIInfo=APIInfo U API_Detail(i);
                        End If
                    End For
Step 10:        UserInfo=User_Detail(j); // Users associated with the Malicious VM
  /* R2L attack checked and corresponding malicious VM location ,process, the API used are collected with
                                          their user info*/
Step 11:        Else If(FESNAout.AttackType==R2L)
Step 12:            For each i=1 to No_of_Associated_Process do
Step 13:                MaliciousProcessDetail= MaliciousProcessDetail U UnregisteredProcessInfo(i);
Step 14:                API_Info=API_Info U API_Detail(i);
Step 15:                MaliciousVMInfo= MaliciousVMInfo U
VMInfo(Location(UnregisteredProcessInfo(i)));
      done
Step 16:        UserInfo=User_Detail(j);
    /* U2R attack checked and corresponding user details with its associated VM and API information is
                                           collected*/
Step 17:        Else If(FESNAout.AttackType==U2R)
Step 18:            UserInfo=User_Detail(j);
Step 19:            MaliciousVMInfo= MaliciousVMInfo AssociatedVM(AllocatedToUser(UserInfo));
Step 20:            APIInfo=ResquestedAPICalled(UserInfo);
  /* Probe attack checked and corresponding affected VMs neighbor VMs information along with API used
                                  and user detail is collected */
Step 21:        Else If(FESNAout.AttackType==Probe)
Step 22:            For each i=1 to NoOfAssociatedVM do
Step 23:                If(|(neighbor(VMInfo(i)).CreationTimestamp - VMInfo(i).CreationTimestamp)|>
VeryHigh AND Entopy(VMDetail.PacketTransfer)> Mean(Entropy for all VM))
Step 24:                    MaliciousVMInfo = MaliciousVMInfo U neighbor(VMInfo(i));
Step 25:                    APIInfo= APIInfo U API_Detail(i);
                        End if
                    done
Step 26:            UserInfo=User_Detail(j);
              \End if
Step 27:  End if
Step 28: Until all ESMA log is read
Step 29: END
```

X. Result and Discussion

The proposed cloud forensic expert system is simulated here. A private cloud environment is built for experimental purpose using OpenStack software platform [54] [55]. OpenStack is an open source software that helps in creating a private cloud environment. It offers a user friendly graphical user interface by which users can create new network node referred to as VM with varying configurations. Each VM serves as a dedicated cloud resource that has the capability of providing Platform and Infrastructure as a service to its users.



FIR Format				
Header				
Evidence Identification	Issuing Date & Time	Evidence Size		
Digital Signature Detail				
Details				
Attack Detail	VM Detail	User Detail	API Detail	Process Detail
Attack Level & Type	Identification	Identification	Identification	Identification
Duration of Attack	IP Address	User Name	API Name	Process Name
Intruder Identification (IP, Port, MAC)	Average Memory Usage	User Email	Functionality	VM by which it is created
	Average Virtual Memory Usage	Subscription Detail	Served Process	

Figure 9. Structure of FIR

Our private cloud architecture is built using 15 number of systems with the following configuration - 64bit Xeon Processor with 32 GB RAM, 2TB HDD, 1GB NIC. The test environment is set up by first installing Linux server Ubuntu 12.10 (IP: 10.32.14.232) with configuration 64bit Xeon Processor with 12 GB RAM, 500GB HDD, 1GB NIC and setting up the OpenStack, acting as a controller node of the private cloud network. The 15 different systems are connected to our OpenStack server following the star topology where each system serve as computer node. The Linux server is also responsible for maintaining cloud network log and cloud management log. The log records are periodically updated using shadow paging. Requests come to the server from various clients which are served on First Come First Serve basis. On arrival of client request the task is allocated by creating VMs on the computer node. Wireshark, a popular sniffing tool is used for collecting network log. The extracted network log record is fed into the FESNA system that identifies the attacked network area and the type of attack. The output of FESNA and management log is fed to the ESMA system that generates the FIR. The data set used for training purpose is a prepared preprocessed sampled data set. Cloud log is obtained from the private cloud deployed in our University lab. The data set is prepared by extensively analyzing the log record by a forensic expert for over a year. Then the sampled training set is prepared by combining the analyzed data set with KDD 1999 data set, which is a popular benchmark for evaluating various anomalies in network environment. This makes the training engine of the FESNA model more efficient and robust. The time period when the network log was sniffed for experimental purpose there were 20 VMs running to serve client request. Table 7 shows the configuration of the created VMs at that time period.

Table 7. Created VM Configuration

VM No	Core of Processor	RAM	HDD	Operating System
VM 1 – VM 4	2 Core	2 GB	200 GB	Cent OS 6.0
VM 5- VM 8	1 Core	1GB	50 GB	Ubuntu 10.10
VM 9 – VM12	4 Core	4 GB	120 GB	Windows 7 64bit SP1
VM 13 – VM 20	2 Core	2 GB	80 GB	Windows 7 32bit

A. Performance Evaluation

The performance evaluation of the proposed expert system is done based on certain quality metrics. The correctness of supervised machine learning algorithm is estimsted based on the generated confusion matrix. The coloumn of this matrix consists of cases belonging to the predicted class. Whereas, the row represents the cases belonging to the actual class. Figure 10 shows a sample confusion matrix where Condition Positive refers to the number of positive cases where as Condition Negative refers to the number of negative cases. True positive referes to correct predicted value when the predicted and expected condition is positive. True negative refers to correct rejection when the predicted and expected condition is negative. False Positive referes to false alarm when the predicted condition is positive but actual condition is negative. False negative refers to the missed condition when Predicted condition is negative where as actual condition is positive. Based on the confusion matrix the proposed expert system is evaluated and True positive rate, False positive rate, True negative rate, False negative rate, accuracy and precision value is measured as given in equation (9-14). This quality metrics helps in measuring the sensetivity (true positive rate) and specificity (true negative rate) of the system, a well known statistical measure for evaluating performance of such classification algorithm.

True Condition	Total Population	Predicted Condition	
		Predicted Condition Positive	Predicted Condition Negative
	Condition Positive	True Positive	False Negative
	Condition Negative	False Positive	True Negative

Figure 10. Confusion Matrix

1. True Positive Rate (TPR)

It is the proportion of predicted positive condition to total expected positive condition where TPR = True Positive Rate, TP = True Positive, FN = False Negative. Figure 11 shows a comparison of true positive rate.

$$TPR = \frac{TP}{(TP+FN)} \quad (9)$$

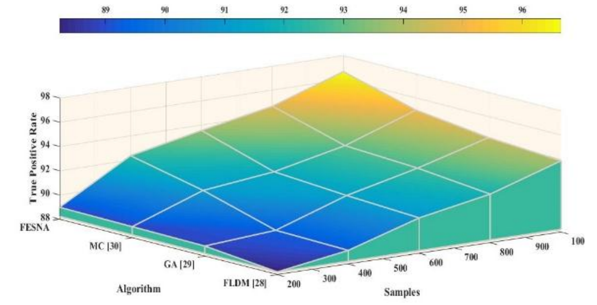
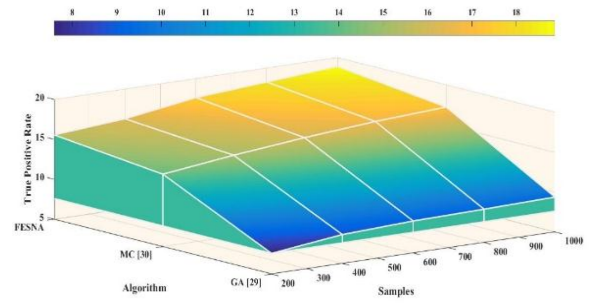
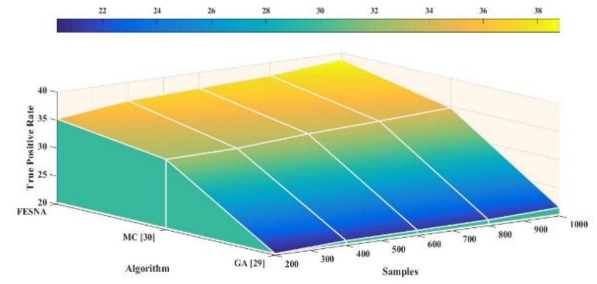
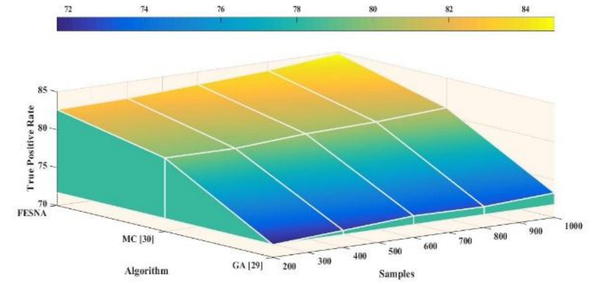
True positive rate (TPR)	
	
True Positive Rate of DoS Attack	True Positive Rate of R2L Attack
Improved by ~1.6%	Improved by ~1.7%
	
True Positive Rate U2R Attack	True Positive Rate of Probe Attack
Improved by ~4.2%	Improved by ~2.5%

Figure 11. True Positive Rate

2. False Positive Rate (FPR)

Proportion of wrongly predicted positive condition to total expected positive condition is determined by false positive rate where FPR = False Positive Rate, FP = False Positive, TN = True Negative.

$$FPR = \frac{FP}{(FP+TN)} \quad (10)$$

Figure 12 shows a comparison of false positive rate.

3. True Negative Rate (TNR)

Proportion of correctly predicted negative condition to total expected negative condition is determined by true negative rate where TNR = True Negative Rate, FP = False Positive, TN = True Negative.

$$TNR = \frac{TN}{(FP+TN)} \quad (11)$$

Figure 13 shows a comparison of true negative rate.

4. False Negative Rate (FNR)

Proportion of wrongly predicted negative condition to total expected negative condition is determined by false negative rate where FNR = False Negative Rate, FN = False Negative, TP = True Positive.

$$FNR = \frac{FN}{(TP+FN)} \qquad (12)$$

Figure 14 shows a comparison of false negative rate.

5. Accuracy

It is the overall accuracy of the system that interprets how well an algorithm predicts the classification. System accuracy is calculated by above mentioned quality metric.

$$Acuuracy = \frac{TP+TN}{(TP+FN+TN+FP)} \qquad (13)$$

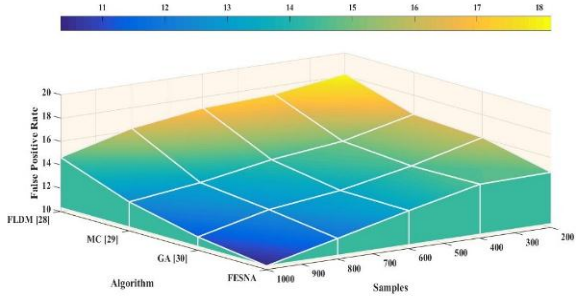
Figure 15 shows a comparison of Accuracy of known forensic schemes with FESNA.

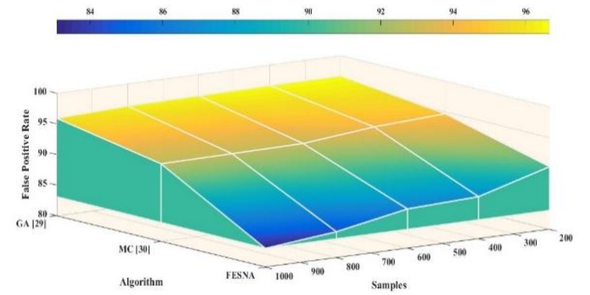
6. Precision

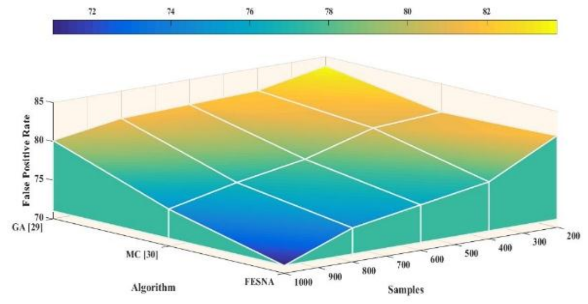
It is referred as positive predictive value which is proportion of true positive and total positive condition.

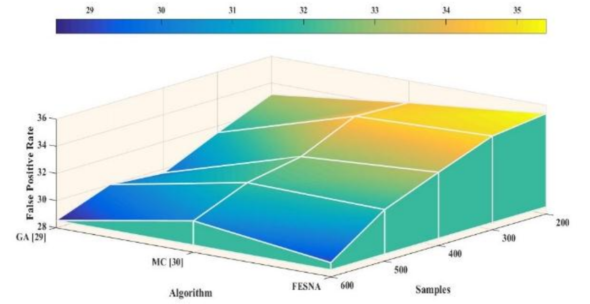
$$Precision = \frac{TP}{(TP+FP)} \qquad (14)$$

Figure 16 shows a comparison of precision of known forensic schemes with FESNA.

<table>
<tr><th colspan="2">False Positive Rate (FPR)</th></tr>
<tr><td>


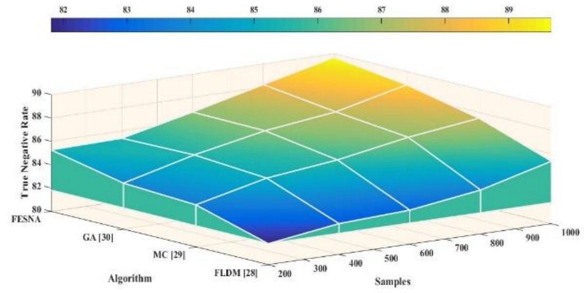
</td><td>


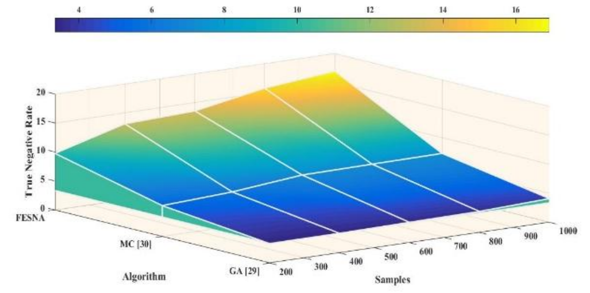
</td></tr>
<tr><td>False Positive Rate of DoS Attack</td><td>False Positive Rate of R2L Attack</td></tr>
<tr><td>Reduced by ~2%</td><td>Reduced by ~9%</td></tr>
<tr><td>


</td><td>


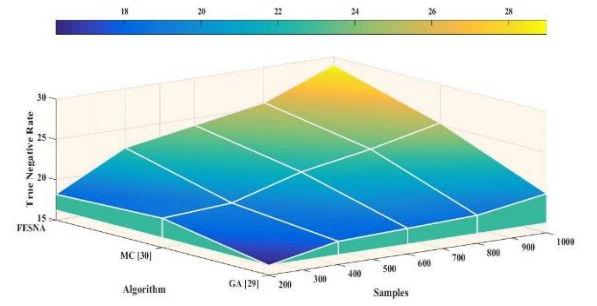
</td></tr>
<tr><td>False Positive Rate of U2R Attack</td><td>False Positive Rate of Probe Attack</td></tr>
<tr><td>Reduced by ~3%</td><td>Reduced by ~1.5%</td></tr>
</table>

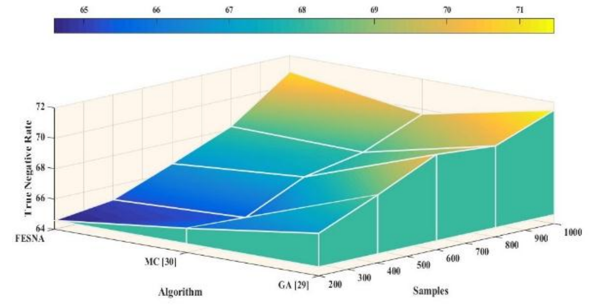
Figure 12. False Positive Rate

<table>
<tr><th colspan="2">True Negative Rate (TNR)</th></tr>
<tr><td>


</td><td>


</td></tr>
<tr><td>True Negative Rate of DoS Attack</td><td>True Negative Rate of R2L Attack</td></tr>
<tr><td>Improved by ~2.3%</td><td>Improved by ~9.25%</td></tr>
</table>

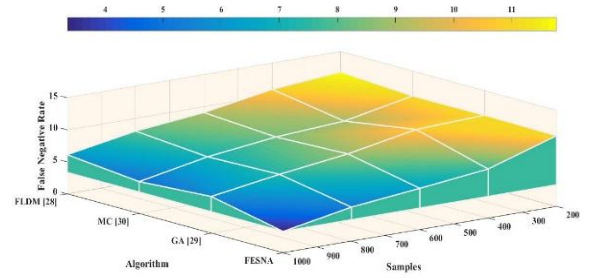


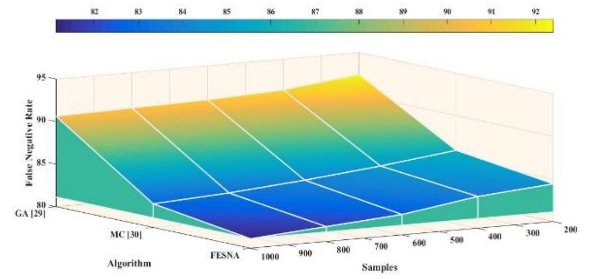


True Negative Rate of U2R Attack	True Negative Rate of Probe Attack
Improved by ~3.4%	Improved by ~1.2%

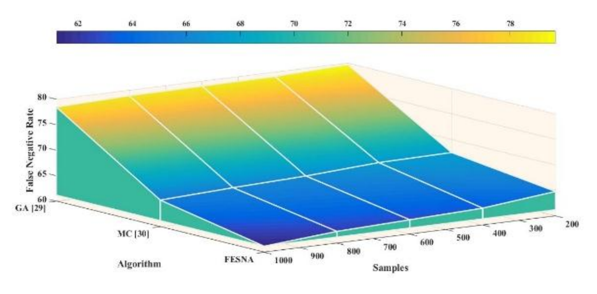
Figure 13. True Negative Rate

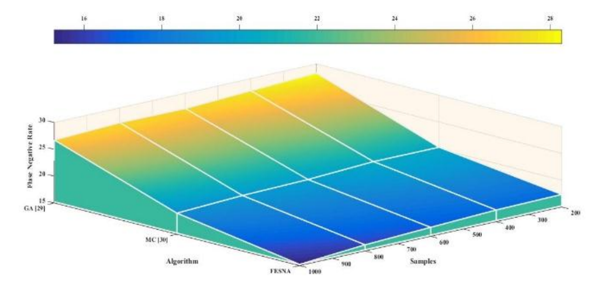
False Negative Rate (FNR)





False Negative Rate of DoS Attack	False Negative Rate of R2L Attack
Reduced by ~1.4%	Reduced by ~1.3%

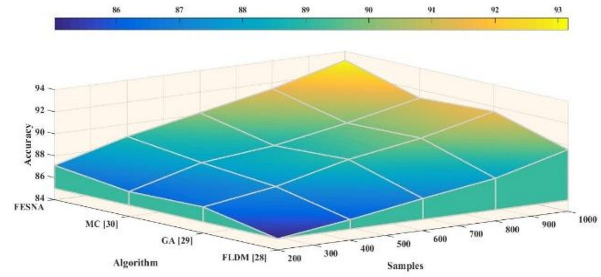


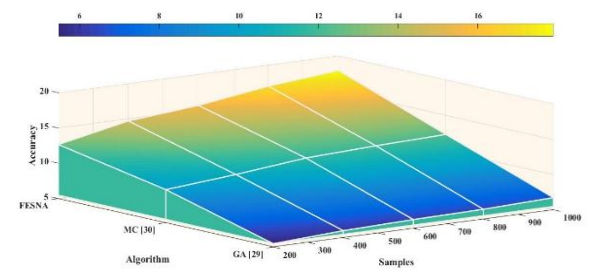


False Negative Rate of U2R Attack	False Negative Rate of Probe Attack
Reduced by ~4.2%	Reduced by ~3.6%

Figure 14. False Negative Rate

Accuracy





Accuracy of DoS Attack	Accuracy of R2L Attack
Improved by ~1.7%	Improved by ~5.6%

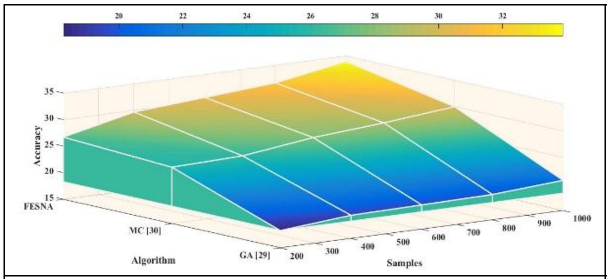
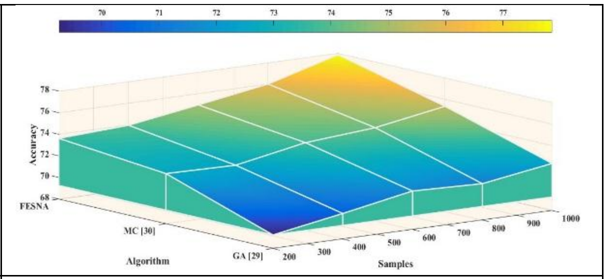
	
Accuracy of U2R Attack	Accuracy of Probe Attack
Improved by ~4%	Improved by ~2.5%

Figure 15. Accuracy

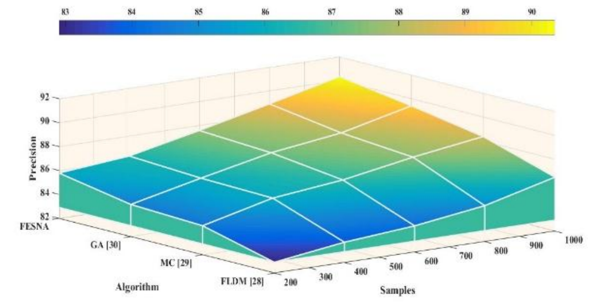
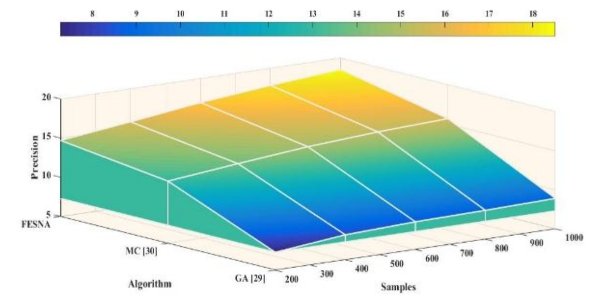
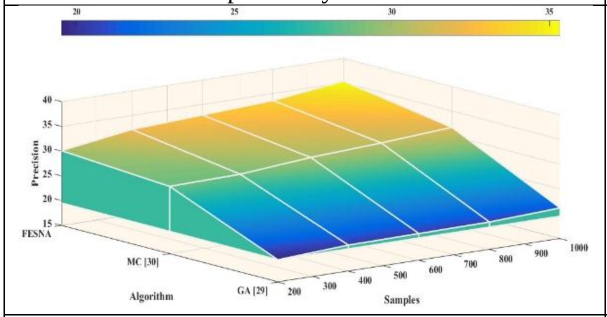
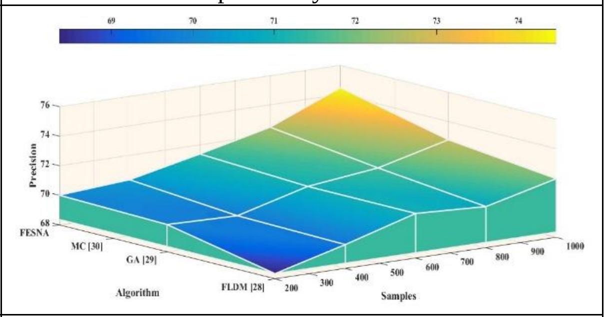
Precision	
	
Precision of DoS Attack	Precision of R2L Attack
Improved by ~2.1%	Improved by ~3.15%
	
Precision of U2R Attack	Precision of Probe Attack
Improved by ~4.3%	Improved by ~2.4%

Figure 16. Precision

B. Comparative Analysis

Table 8 shows a comparative analysis and novelty of the proposed automated forensic identification system with some well-known existing techniques. It is capable of identifying all the four known network attacks as well the fuzzy knowledge base is capable of dynamic rule generation. This makes the system capable of identifying anomalous attacks.

XI. CONCLUSION

Security is an essential feature which is required for an enhanced utility service. Cloud is one of the popular utility framework in recent years and it is prone to various malicious activities and attacks. Different types of attacks affect user's data as well as cloud performance. Forensic investigation is necessary to detect and collect evidences of any criminal activity. It also helps to improve the security framework in the cloud environment to prevent future attacks. In this paper, a strong forensic architecture has been proposed to monitor different types of attacks and collect evidences for malicious activity in cloud environment. The process of identification, an important part of any forensic system is proposed in this paper. A novel automated approach combining two modules FESNA and ESMA is developed and simulated for log analysis. It helps to predict and identify intrusion along with its intruder by analyzing cloud network and management log. As a result of the identification phase a forensic identification report FIR is also generated to drive the further forensic analysis smoothly. It helps the law enforcement to issue suspension and analysis warrant for detail evidence collection. The expert system is tested in our University cloud environment and the results show an improvement in accuracy upto ~5.6% and improved precision upto ~4.3% of detecting the type of attack compared to other existing network and cloud forensics system. If CSP include this expert system in their cloud environment it will help them improve their reliability and deliver good quality of service to their esteemed customers.

Table 8. Comparison with different well-known technique

Properties	**Existing Approach**			**Proposed Approach**
	FLDM [28]	**GA [29]**	**MC [30]**	
Scenario	Defense system is developed for DDoS in cloud environment. This is developed using the entropy based Hurst Parameter analysis.	The authors developed an intrusion detection system on KDD 1999 dataset.	The authors designed multiple classifier for determining the various known network attacks.	Automated system for Forensic identification is developed that uses network and management log for identifying the type of attack and attacker details
Based on classical algorithm	Fuzzy Logic	Genetic Algorithm	Multi Classifier (K-Means, Gaussian and Multiplayer Perception)	Fuzzy Logic
Training Dataset used	Real time cloud dataset for simulation	Numeric dataset of KDD 1999	Numeric dataset of KDD 1999	Real time cloud dataset for simulation
Nature of dataset	Network information	Network information	Network information	Cloud Network and management information
Deployed Prototype	Datacenter with VM	NA	NA	Private cloud system along with VM and log server
Features				
Detection of DoS Attack	✓	✓	✓	✓
Detection of R2L Attack	×	✓	×	✓
Detection of U2R Attack	×	✓	×	✓
Detection of Probe Attack	×	✓	×	✓
Model for Cloud Environment	✓	×	×	✓
Fuzzy Intelligent System	×	✓	✓	✓
Automatic Fuzzy Rule Generation	×	×	×	✓
Post Attack Forensic Model	×	×	×	✓
Forensic Investigation Report	×	×	×	✓

ACKNOWLEDGMENT

The research idea was supported by Department of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology, West Bengal. I personally thank my guide and seniors for their several therotical expertise, although they may not agree with all the views and conclusions of this paper. All the opnion in this paper do not represent views of my previous organization "Maulana Abul Kalam Azad University of Technology, West Bengal" as well as my current organization "Criminal Investigation Department, West Bengal".

REFERENCES

[1] Buyya R, Yeo C S, Venugopal S, Broberg J, Brandic I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, 25(6), (pp. 599-616).

[2] T OGRAPH B, MORGENS Y R. (2008). Cloud computing. Communications of the ACM, 51(7), (pp. 9-11).

[3] Rimal B P, Choi E, Lumb I. (2009). A taxonomy and survey of cloud computing systems. INC, IMS and IDC, (pp. 44-51).

[4] Mell P, Grance T. (2011). The NIST definition of cloud computing.

[5] Market Research Media. Global cloud computing market forecast 2015-2020. http://www.marketresearchmedia.com/2012/01/08/global-cloud- computing-market/ [Accessed July 5th, 2012]

[6] Krutz R L, Vines R D (2010). Cloud security: A comprehensive guide to secure cloud computing. Wiley Publishing.

[7] Computing C. (2011). Cloud computing privacy concerns on our doorstep. Communications of the ACM, 54(1), (pp. 36-38).

[8] Viega J. (2009). Cloud computing and the common man. Computer, 42(8), (pp. 106-108).

[9] Wei J, Zhang X, Ammons G, Bala V, Ning P. (2009). Managing security of virtual machine images in a cloud environment, ACM workshop on Cloud computing security (pp. 91-96).

[10] Zhang, X, Wuwong N, Li H, Zhang X. (2010). Information security risk management framework for the cloud computing environments. Computer and Information Technology (CIT), IEEE International Conference, (pp. 1328-1334).

[11] Subashini S, Kavitha V. (2011). A survey on security issues in service delivery models of cloud computing. Journal of network and computer applications, 34(1), (pp. 1-11).

[12] Kandukuri B R, Rakshit A. (2009). Cloud security issues. Services Computing, IEEE International Conference, (pp. 517-520).

[13] So K. (2011). Cloud computing security issues and challenges. International Journal of Computer Networks, 3(5), (pp. 247-255).

[14] Ren K, Wang C, Wang Q. (2012). Security challenges for the public cloud. IEEE Internet Computing, 16(1), (pp. 69).

[15] Clavister. Security in the cloud. http://www.clavister.com/documents/resources/white-papers/clavister-whp-security-in-the-cloud-gb.pdf, Clavister White Paper, [Accessed July 5th, 2012].

[16] Ruan K, Carthy J, Kechadi T, Baggili I. (2013). Cloud forensics definitions and critical criteria for cloud forensic capability: An overview of survey results. Digital Investigation, 10(1), (pp. 34-43).

[17] Shah J J, Malik L G. (2014). An approach towards digital forensic framework for cloud. Advance Computing Conference (IACC), IEEE International, (pp. 798-801).

[18] Dykstra J, Sherman A T. (2012). Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques. Digital Investigation, 9, (pp. S90-S98).

[19] Zawoad S, Hasan R. (2013). Cloud forensics: a meta-study of challenges, approaches, and open problems. arXiv preprint arXiv:1302.6312.

[20] Ruan K, Carthy J, Kechadi T. (2011). Survey on cloud forensics and critical criteria for cloud forensic capability: A preliminary analysis. Proceedings of the Conference on Digital Forensics, Security and Law, (p. 55).

[21] Zawoad S, Dutta A K, Hasan R. (2013). SecLaaS: secure logging-as-a-service for cloud forensics, Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, (pp. 219-230).

[22] Sang T. (2013). A log based approach to make digital forensics easier on cloud computing. In: Intelligent System Design and Engineering Applications (ISDEA), Third International Conference, (pp. 91-94).

[23] Thorpe S, Ray I, Grandison T, Barbir A. (2012). Cloud log forensics metadata analysis, Computer Software and Applications Conference Workshops (COMPSACW), IEEE 36th Annual (pp. 194-199).

[24] Vo H T, Wang S, Agrawal D, Chen G, Ooi B C (2012). LogBase: a scalable log-structured database system in the cloud. Proceedings of the VLDB Endowment, 5(10), (pp. 1004-1015).

[25] Patrascu A, Patriciu V V. (2014). Logging framework for cloud computing forensic environments. Communications (COMM), 10th International Conference, (pp. 1-4).

[26] Kim J S, Kim D G, Noh B N (2004). A fuzzy logic based expert system as a network forensics. Fuzzy Systems, Proceedings. IEEE International Conference on (2), (pp. 879-884).

[27] Dickerson J E, Dickerson J A. (2000). Fuzzy network profiling for intrusion detection. Fuzzy Information Processing Society, NAFIPS. International Conference of the North American (pp. 301-306).

[28] Iyengar N C S, Banerjee A, Ganapathy G. (2014). A fuzzy logic based defense mechanism against distributed denial of service attack in cloud computing environment. International Journal of Communication Networks and Information Security, 6(3), (pp. 233).

[29] Sabhnani M, Serpen G. (2003). Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context. In MLMTA (pp. 209-215).

[30] Hoque, Mohammad Sazzadul, et al. (2012). An implementation of intrusion detection system using genetic algorithm. arXiv preprint arXiv:1204.1336 .

[31] Stoffel K, Cotofrei P, Han D. (2010). Fuzzy methods for forensic data analysis. SoCPaR (pp. 23-28).

[32] Singh S. (2014). Cloud computing attacks: a discussion with solutions. Open Journal of Mobile Computing and Cloud Computing, 1(1).

[33] Lerman L, Bontempi G, Markowitch O. (2011). Side channel attack: an approach based on machine learning. Center for Advanced Security Research Darmstadt, (pp. 29-41).

[34] Portnoy L, Eskin E., Stolfo S. (2001). Intrusion detection with unlabeled data using clustering. Proceedings of ACM CSS Workshop on Data Mining Applied to Security DMSA-2001.

[35] Sharma A., Panda S N. (2009). Intrusion detection system. Enterprise Information Systems in 21st Century: Opportunities and Challenges, (pp. 194).

[36] Garcia-Teodoro P, Diaz-Verdejo J, Maciá-Fernández G,Vázquez E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. computers & security, 28(1), (pp. 18-28).

[37] Vo, H. T., Wang, S., Agrawal, D., Chen, G., &Ooi, B. C. (2012). LogBase: a scalable log-structured database system in the cloud. Proceedings of the VLDB Endowment, 5(10), 1004-1015.
[38] SaadAlqahtany, Nathan Clarke, Steven Furnell, Christoph Reich. (2014). "A forensically-enabled IAAS cloud computing architecture", 12th Australian Digital Forensics Conference, , http://ro.ecu.edu.au/adf/136/
[39] Meera G, Alluri B K, Powa, D, Geethakumari G. (2015). A strategy for enabling forensic investigation in cloud IaaS. In Electrical, Computer and Communication Technologies (ICECCT), IEEE International Conference on (pp. 1-5).
[40] Munz G, Carle G. (2008). Distributed network analysis using TOPAS and wireshark. In Network Operations and Management Symposium Workshops, 2008. NOMS Workshops 2008. IEEE (pp. 161-164).
[41] Dabir A, Matrawy A. (2007). Bottleneck analysis of traffic monitoring using wireshark. In Innovations in Information Technology, 4th International Conference on (pp. 158-162).
[42] Kayacik H G, Zincir-Heywood A N, Heywood M. I (2005). Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets. Proceedings of the third annual conference on privacy, security and trust.
[43] Devaraju S, Ramakrishnan, S. (2014). Performance comparison for intrusion detection system using neural network with KDD dataset. ICTACT Journal on Soft Computing, 4(3), (pp. 743-752).
[44] J B MacQueen (1967): Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-297
[45] Faraoun K M, Boukelif A. (2006). Neural networks learning improvement using the K-means clustering algorithm to detect network intrusions. INFOCOMP Journal of Computer Science, 5(3), (pp. 28-36).
[46] Soumi G, Dubey S K. (2013). Comparative analysis of k-means and fuzzy c-means algorithms." IJACSA) International Journal of Advanced Computer Science and Applications 4.4
[47] David C. Hoaglin, Frederick Mosteller, John W. Tukey. Understanding robust and exploratory data analysis". Wiley, 1983. ISBN 0-471-09777-2
[48] Suryawanshi S, Jodhe P, Chawhan S, Kuthe A M. (1999). Apriori Algorithm Using Data Mining.
[49] Pasquier N, Bastide Y, Taouil R., Lakhal L. Efficient mining of association rules using closed itemset lattices. Information systems, 24(1), (pp. 25-46).
[50] Jafarzadeh H, Sadeghzadeh M, Improved. (2014). Apriori Algorithm Using Fuzzy Logic, International Journal of Advanced Research, Computer Science and Software Engineering,
[51] Han J, Pei J, Yin Y. (2000). Mining frequent patterns without candidate generation. ACM Sigmod Record (Vol. 29, No. 2. (pp. 1-12).
[52] Baset S A. (2012). Cloud SLAs: present and future. ACM SIGOPS Operating Systems Review, 46(2), (pp. 57-66).
[53] Andrzejak A., Kondo D, Yi S. (2010). Decision model for cloud computing under sla constraints. IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (pp. 257-266).
[54] Sefraoui, O, Aissaoui M, Eleuldj M. (2012). OpenStack: toward an open-source solution for cloud computing. International Journal of Computer Applications, 55(3).
[55] Wuhib F, Stadler R, Lindgren H. (2012). Dynamic resource allocation with management objectives—Implementation for an OpenStack cloud. 8th international conference on network and service management (cnsm) and 2012 workshop on systems virtualiztion management (svm) (pp. 309-315).

Authors Profile

Mr. Palash Santra passed Diploma in Computer Science & Technology from The Calcutta Technical School in year 2011 and Bachelor of Technology (IT) from West Bengal University of Technology in year 2014. He also passed Master of Technology (Software Engineering) from Maulana Abul Kalam Azad University of Technology, West Bengal in year 2017. Currently he is working as Computer Analyst in Criminal Investigation Department, West Bengal, India. He has several research paper in reputed international conferences. His main research area focuses on Cloud Security, Cloud & Cyber Forensics and IoT Techniques & Security.