Design of Host based Intrusion Detection System using Fuzzy Inference Rule

Hari Om
Department of Computer Science & Engineering
Indian School of Mines, Dhanbad
Dhanbad 826004,India

Alok Kumar Gupta
Department of Computer Science & Engineering
Indian School of Mines, Dhanbad
Dhanbad 826004,India

ABSTRACT
The security of a system is an important issue due to the latest advancements in information technology. Intrusion Detection Systems are used to identify the attacks and malicious activities in the computer systems. This paper discusses a new host based intrusion detection system for detecting changes in hardware profile using fuzzy inference rule. The proposed system is able to analyze and detect the unauthorized access in a computer system by generating a set of fuzzy IF-THEN rules with the help of frequent item set. These fuzzy inference rules are used to find the misuse of the system. The experiments of the proposed system are carried out on the system performance log.

Keywords
Intrusion Detection System (IDS), Fuzzy logic, System performance log, Fuzzy inference rules

1. INTRODUCTION
Due to technology emerges day by day, there is a need to have a security mechanism to protect the systems from unauthorised users and malicious activities. For this, the intrusion detection systems (IDS) are used. An intrusion detection system is a device or software application that collects information from a variety of network sources or computer systems for analysis in order to detect the signs of malicious activities. An intrusion is defined as a set of actions that attempts to compromise the integrity, confidentiality, or availability of the system resources [1]. Integrity refers to maintain and assure the accuracy and consistency of data over its entire life cycle. Confidentiality refers to maintain the secrecy of data into system so that unauthorized user cannot access. Availability refers to availability of information resources. There are two common approaches to develop an intrusion detection model: misuse detection model and anomaly detection model [2]. The misuse detection model refers to detection of intrusions that follow well-defined intrusion patterns. Every intrusion has some pattern e.g. number of packets, number of connection, bytes sent, duration etc. It matches the packets with the database of pattern. Whenever there is a match, alarms are raised. It is very useful in detecting known attack, but not suitable for unknown attacks. The anomaly detection model refers to detection performed by detecting changes in the patterns of utilization or behaviour of the system. Whenever there is any deviation from the normal behaviour activity, alarms are raised. Normal behaviour can be developed using different techniques such as statistical analysis, data mining algorithms, genetic algorithms, artificial neural network approach, fuzzy logic and rough set etc. The anomaly detection systems can detect new intrusions unlike the misuse detection systems. The IDSs can be network based or host based as far as the source of data is concerned. Network based IDS (NIDS) collects raw packets as the data source from the network and analyse for signs of intrusions. The host based IDS (HIDS) operates on information collected from within an individual computer system such as operating system audit trails, C2 audit logs, and System logs.

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. This mapping provides a basis from which decisions can be made, or patterns are discerned. The process of fuzzy inference involves all of the pieces that are described in membership functions, logical operations, and fuzzy IF-THEN rules. The proposed work describes to design a host based intrusion detection system using fuzzy inference rules. The performance log of a computer system acts as input to proposed system. The input is divided into two subsets: one is called as training data and other as testing data. The training dataset is further classified into two subsets: abnormal and normal data. After that, perform data mining technique to select frequent items from each attribute in the abnormal data as well as normal data. These mined frequent items are used to find the important attributes of the input dataset, which in turn are used to develop a set of definite and indefinite rules using a deviation method. Then, indefinite rules must be ignored and definite rules are used to generate fuzzy inference rules by fuzzifying it in such a way that we obtain a set of fuzzy IF-THEN rules with consequent parts that represent as either normal or abnormal data. These rules are given to the fuzzy inference system to effectively learn[16][18]. In testing phase, the testing data is matched with fuzzy inference rules to detect abnormal and normal behaviour of data. In this proposed work, Mamdami fuzzy inference system is used which is implemented in MATLAB 7.5.

The remaining paper is organized as follows: Section 2 reviews the related work. Section 3 discusses our proposed work. The performance log analysis is given in section 4. The experimental methodology and the results are discussed in section 5. Finally, the paper is concluded in section 6.

2. RELEATED WORK
Several researchers have discussed different designs for developing intrusion detection systems. In the last couple of years, intrusion detection has received a lot of interest among the researchers since it is being widely applied for preserving the security within a network and computer systems. Denning has described a rule based intrusion detection system that can detect security violations attempted by outsiders to system penetrations and misuse by insiders [5]. In this system, the generated audit record is matched with the defined rules and checked for abnormal behaviour. Srinivasa et al. have presented a rule based intrusion detection system in which

they use genetic algorithm to make IDS more efficient [6]. The genetic algorithm has been used to prune the best rules from the generated rule set. They use DARPA dataset for training and testing purpose. Siraj et al. discuss an intelligent alert clustering model for network intrusion analysis [7]. They use principal component analysis with expectation maximization technique to aggregate similar alerts and reduce the number of low quality alerts. Shanmugavadivu et al. use KDD Cup99 for their proposed anomaly based network intrusion detection system [8]. They use fuzzy logic for identifying the intrusion activities in a network. This system generates fuzzy IF-THEN rules and with the help of fuzzy decision module the system identifies the appropriate classification of the test data. Dhanalakshmi discusses a system in which the fuzzy logic is integrated with the data mining methods using genetic algorithm for intrusion detection [9]. This system uses data mining to extract rules and Mamdami fuzzy inference system to determine the behaviour of the test data. Om et al. have designed a PCA based anomaly detection system for outlier detection in a computer system [10]. They use principal component analysis (PCA) to reduce the dimensions of data recorded by the computer system (performance log). Bharti et al. have proposed an intrusion detection model in which they use feature selection algorithm to select the non-redundant attributes [11]. They use fuzzy K-mean clustering algorithm to calculate the membership of every data point and J48 classification techniques for assigning a cluster to a particular class. Han et al. describe an evolutionary neural network based intrusion detection system, which has good detection performance and also reduces the training time [12]. Om et al. discuss a neural network based model, which can detect changes in the hardware profile of a computer system [13]. They use back propagation network (BPN) for detection and reported that the very high and very low values of the learning rate have bad effect on the results.

3. PROPOSED WORK

Recently, several researchers have focused on fuzzy logic for developing an effective intrusion detection system. This paper also proposes a fuzzy logic based intrusion detection system for detecting the changes in hardware profile. The model of our system is shown in Fig. 1. The preprocessor generates the rules and provides them to fuzzy rules generator to generate

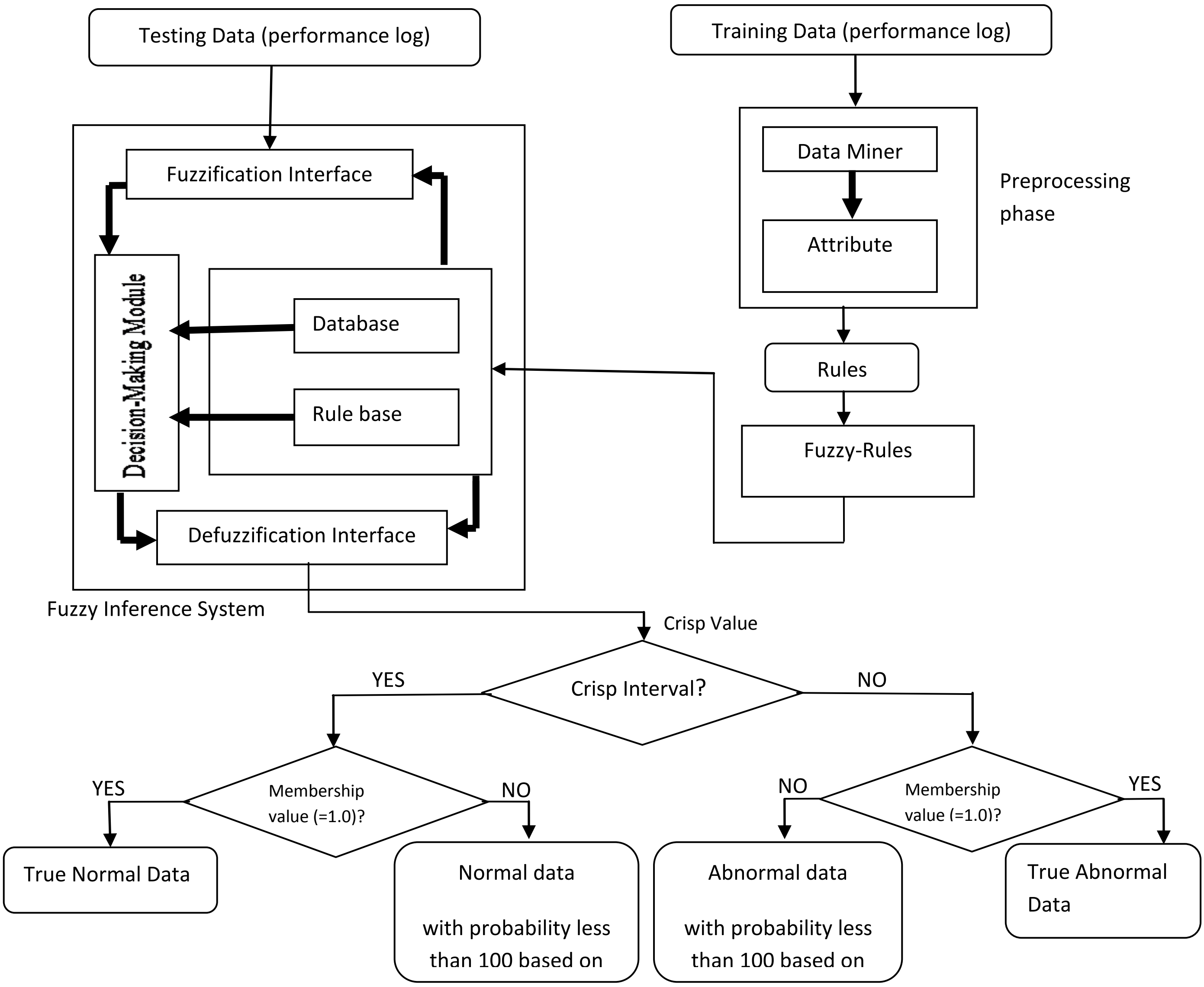

Fig. 1. Intrusion Detection System based on Fuzzy Inference rule

fuzzy IF-THEN rules for the proposed system. A fuzzy inference system is used to generate the output for detection.

In this figure, a bold line denotes the interconnection inside the module and a normal line denotes the data flow in the proposed system. The different steps involved in the proposed system for intrusion detection are described as follows:

i. Preprocessing module
   a. Data miner
   b. Attribute selector
ii. Fuzzy rules generator
iii. Fuzzy inference system
iv. Finding behavior for a test input

3.1 Preprocessing Phase

The first component of our proposed system is preprocessing module. This module is responsible for accepting training data as system log and categorizes it into two classes: abnormal and normal data. These data are used to generate rules. The preprocessing module contains two sub-modules: data miner and attribute selector.

a. Data miner

The data miner integrates FP-Growth algorithm property to mine the frequent items in each attribute [3]. By considering the frequency of items in each attribute, the data miner identifies the frequent items by inputting sufficient support. The frequent items are identified for both the classes namely, abnormal and normal classes.

b. Attribute Selector

This module selects the suitable attributes from the input data as all the attributes are not useful for detection. For identifying the suitable attributes, the deviation method is used. Each class (abnormal and normal) is represented as $Class_i = [A_1, A_2, A_3, ...A_n]$, where A_1, $A_{2,...}$ A_n are the attributes in i^{th} class. In our proposed system, the $class_1$ refers to normal and $class_2$ to abnormal class. Each attribute A_j contains items I_k that have frequency greater than the minimum support.

$$A_j = \{I_k; 1 \le k \le m\} \text{ and } \mathrm{support}(\mathrm{I}_k) > \min \mathrm{support}$$

Here, $\mathrm{support}(I_k)$ is the frequency of k^{th} item in A_j attribute of $class_i$. Then, for each attribute, the deviation range(D) {min, max} of the frequent items is calculated as follows.

$$D_{A_j} = \{ f_{\min}, f_{\max} \} \text{ ,where } f_{\min} = \min[\mathrm{support}(\mathrm{I}_k)]$$

$$\text{and } f_{\max} = \max[\mathrm{support}(\mathrm{I}_k)]$$

The attributes containing non-identical {max, min} range for both the classes are chosen as effective attributes, which will give significant detection rate. The chosen attributes are used to generate rules for the proposed system. Compare the deviation range of the effective attributes for both abnormal and normal data to calculate intersection point. After that, intersection points are used to generate IF-THEN rules.

3.2 Fuzzy Rules Generator

The fuzzy rules are generated from the definite rules obtained from above step. The definite rules are the rules that contain only one classification label in THEN part. The fuzzy rules contain linguistic variables only, hence the membership functions are used to fuzzify the numerical values in definite rules. The triangular and trapezoidal membership functions are used in this step. For example, if an attribute1 is N_1, the data is abnormal; if an attribute1 is N_2, the data is normal. Here, N_1 and N_2 are linguistic variables. These fuzzy rules are given to Rulebase of the fuzzy inference system.

3.3 Fuzzy Inference System

This step describes the fuzzy logic system for finding the suitable class label of the input test dataset. In the proposed system, we use Mamdami inference system [14], which is based on Zadeh’s paper on fuzzy algorithms for complex system and decision process [17]. In our proposed system, nine inputs and one output Mamdami fuzzy inference system with the centroid of area defuzzification strategy is used.

3.4 Finding Behaviour for a Test Input

For testing phase, a test data from the testing data is given to the Mamdami fuzzy inference system. It compares the input variable with the membership functions on the premises part to obtain the membership value for each linguistic label. The output of fuzzification interface is fed to the decision making unit which in turn compares that particular input with Rulebase. The output of knowledgebase is fed to defuzzification interface, which converts the fuzzy result into a crisp value. If the crisp value lies in the user defined intervals for respective behavior, the data can be classified as normal and abnormal. In our proposed system, if the crisp value is between 0 to 50, i.e. $(0 \le crisp \text{ value} \le 50)$, the data is normal; and if the crisp value is between 50 to 100, i.e. $(50 < crisp \text{ value} \le 100)$, the data is abnormal. If the membership value for a test data is 1.0, the test data is either true normal or true abnormal.

ALGORITHM

Input: Performance log recorded by the computer system
Output: Abnormal data or Normal data
Assumptions:

a. Training data is divided into two classes: abnormal class and normal class.
b. Testing data contains abnormal data, normal data or both.

step 1: Training data is given as input to preprocessing module that contains data miner and attribute selector:
   a) Data Miner extracts frequent itemsets in each attribute from both the classes: abnormal and normal classes.
   b) Attribute selector selects effective attributes, which are suitable for rules generation.
   c) With the help of effective attributes, IF-THEN rules are generated.

step 2: Rules obtained from preprocessor are given as input to fuzzy rules generator module.

step 3: Fuzzy rules generator generates fuzzy IF-THEN rules corresponding to each rule obtained in step 1, and feeds it into rulebase of fuzzy inference system.

step 4: In database of fuzzy inference system, define membership function of fuzzy sets used in fuzzy rules.

step 5: Testing data is given as input to fuzzy inference system.

step 6: For each test data, fuzzy inference system generates a crisp value as an output.

step 7: If crisp value lies in the user defined intervals for respective behavior, the data can be classified as normal or abnormal. In our proposed system, if crisp value is between 0 to 50, the data is normal and, if crisp value is between 50 to 100, the data is abnormal.

step 8: If the membership value for a test data is 1.0, the test data is true normal or true abnormal, otherwise test data has respective behavior with probability less than 1.0 based on the membership value.

4. PERFORMANCE LOG ANALYSIS

The performance log has been generated of patterns with errors and without errors. The proposed system has been applied to analyse the log and find the result.

(a) Performance log

The performance logs are generated by taking some of the process attributes for normal and abnormal behavior of the system. The performance of the personal computer can be measured by using the performance log. The hardware profile of the system that has been used in experiments as follows:

- Intel Core 2 Duo CPU @ 2.33 GHz
- 1024 MB RAM
- Microsoft Windows XP

(b) Attributes used in performance log

The used attributes for Performance Log analysis are as follows.

- ***% committed byte in use:*** It is the ratio of memory committed bytes to the memory commit limit. The commit limit is determined by the size of the paging file.
- ***Available Mbytes:*** It is the amount of physical memory in Megabytes immediately available for allocation to a process or for the system use.
- ***System Driver Resident Bytes***: It is the size (in bytes) of the pageable physical memory being used by device drivers.
- ***System Driver Total Bytes:*** It is the size in bytes of the pageable virtual memory currently being used by the device drivers.
- ***I/O Write Operations/sec:*** The rate at which the process is issuing write I/O operations.
- ***File Control Operations/sec:*** It is the combined rate of file system operations that neither reads nor writes.
- ***File Data Operations/sec:*** It is the combined rate of read and write operations on all logical disks on the computer.
- ***File Write Operations/sec:*** It is the combined rate of the file system write requests to all devices on the computer, including requests to write data in the file system cache.
- ***Threads:*** It is the number of threads in the computer at the time of data collection.

5. EXPERIMENT METHODOLOGY AND RESULT

For experimental evaluation of the proposed system, the performance log of the computer system needs be generated. The steps for generating the performance logs are as follows:

- On the start menu, point to settings, point to Control Panel, double click Administrative Tools, and double click Computer Management.
- Explore performance Logs and Alerts, right click Counter Logs, and then click New Log Settings.
- Type a name for the counter log and then click OK.
- Click Add Counters.
- In the Performance object box, select a performance object that need be monitored.
- Counters added for experiment.
- On the General tab under Sample data every sampling interval of 15 seconds is configured.
- On the Log Files tab log files properties is configured as Comma delimited files that can be viewed later in reporting tools such as Microsoft Excel.

The training dataset has been divided into two subdatasets: normal dataset and abnormal dataset. Firstly, the normal dataset have been generated. Samples of normal dataset are shown in Table 1. After that, the abnormal dataset have been generated by disabling graphics driver, audio driver, and Ethernet driver. This generates the logs for the system performance that have been considered as intrusions. Samples of abnormal dataset are shown in Table 2. For testing dataset, we have taken normal data as well as abnormal data, i.e. mixed data. Samples of the testing patterns are shown in Table 3.

Table 1. Training Dataset (Normal dataset)

% Committed Bytes In Use	Available MBytes	System Driver Resident Bytes	System Driver Total Bytes	IO Write Operations /sec	File Control Operations /sec	File Data Operations /sec	File Write Operations /sec	Threads
10.07679182	646	1114112	8830976	105.9535826	1407.184756	324.4265362	111.2242327	535
9.287402465	659	1118208	8830976	11.94779465	851.0300656	89.97557086	13.61648105	514
9.271461777	688	1118208	8830976	20.24620835	147.1846067	72.46011408	20.24620835	507
9.232098038	693	1118208	8728576	0.066668724	12.40038269	0.066668724	0.066668724	504
9.207861687	692	1118208	8728576	20.06729463	157.6716006	68.86882176	20.06729463	499
9.198427402	694	1118208	8728576	0.066668704	13.80042174	0.133337408	0.066668704	495
9.070251261	698	1118208	8728576	16.80053295	144.1379057	65.66874981	20.06730324	486
9.065859439	698	1118208	8728576	0.066668731	13.86709614	0.333343657	0.066668731	482
9.061955597	700	1118208	8728576	20.66731268	146.0045638	66.33540684	20.66731268	481
9.061630277	700	1118208	8728576	0.066668763	11.20035221	0.066668763	0.066668763	481
9.059515696	700	1118208	8728576	20.06729652	145.8712452	65.93540286	20.06729652	480
9.059190376	701	1118208	8728576	0.066668741	11.33368589	0.066668741	0.066668741	480
9.247062766	680	1118208	8728576	27.80086509	532.8832487	249.8744421	147.4045869	491

9.149792039	682	1118208	8728576	0.200006348	155.1382573	6.866884615	0.400012696	488
9.150442679	699	1118208	8728576	20.06729462	144.2711813	65.73539035	20.06729462	489
9.150442679	699	1118208	8728576	0.066668738	13.86709753	0.333343691	0.066668738	489
9.144424257	698	1118208	8728576	20.06729919	150.7380846	66.06874918	20.06729919	486
8.742165884	701	1118208	8728576	20.06729919	37.20116907	2.933425518	1.13336895	479
8.73289426	701	1118208	8728576	20.06729501	149.9380282	67.33544174	20.06729501	475
8.73289426	701	1118208	8728576	0.06666874	11.26701703	0.06666874	0.06666874	475
8.728177117	702	1118208	8728576	20.06730434	146.6046587	66.00209735	20.06730434	473
8.732406279	702	1118208	8728576	0.066668745	11.33368665	0.066668745	0.066668745	469
8.721833374	702	1118208	8728576	20.06729527	143.2711546	65.66872373	20.06729527	463
8.720857414	702	1118208	8728576	0.066668754	13.86710078	0.333343769	0.066668754	462
8.715652291	701	1118208	8728576	20.06729895	153.271496	68.66883031	20.06729895	458
8.71272441	702	1118208	8728576	0.066668742	11.26701732	0.066668742	0.066668742	456
8.707844608	702	1118208	8728576	20.06729966	146.0712743	66.00208193	20.06729966	454
8.707844608	703	1118208	8728576	0.066668741	11.33368595	0.066668741	0.066668741	454
8.786246765	678	1118208	8728576	29.73427583	452.1476651	107.3367356	29.73427583	454
8.73452086	703	1118208	8728576	20.20063344	855.2268176	436.4136848	20.20063344	453
8.7343582	703	1118208	8728576	20.06729848	143.4045151	65.66873424	20.06729848	453
8.746557706	702	1118208	8728576	0.06666875	23.40073138	3.333437518	0.06666875	453
8.764287655	702	1118208	8728576	22.60071308	289.6758063	82.60260623	23.06739448	453
8.747858987	702	1118208	8728576	0.066668763	11.33368974	0.066668763	0.066668763	453
8.746557706	702	1118208	8728576	20.06729443	145.8045612	65.93539597	20.06729443	452
8.746720366	702	1118208	8728576	0.066668774	11.33369156	0.066668774	0.066668774	452
8.75387741	702	1118208	8728576	21.57819834	160.8374969	70.26234337	21.57819834	455
8.752413469	702	1118208	8728576	0.066668746	11.26701807	0.066668746	0.066668746	455

Table 2. Training Dataset (abnormal dataset)

% Committed Bytes In Use	Available Mbytes	System Driver Resident Bytes	System Driver Total Bytes	IO Write Operations /sec	File Control Operations /sec	File Data Operations /sec	File Write Operations /sec	Threads
7.423384034	700	1617920	2150400	102.1167592	1627.371182	312.9875432	107.081218	486
7.148329421	705	1617920	2150400	31.36551108	517.8646084	150.0873073	31.76592186	484
7.087332686	733	1617920	2150400	0.400010744	463.2791101	7.533535681	0.400010744	478
7.091073819	735	1617920	2150400	20.06719968	152.1373743	68.66849061	20.06719968	475
7.082778263	736	1617920	2150400	0.066668489	14.5337306	0.333342445	0.066668489	474
7.083266237	736	1617920	2150400	27.80074732	578.8822277	98.26930826	29.40079033	474
7.073832075	736	1617920	2150400	0.066599087	15.38438913	0.399594523	0.066599087	471
6.970218954	739	1617920	2150400	17.53380821	144.0705686	65.6684452	20.06721016	465
6.968592374	741	1617920	2150400	0.066668435	11.93364993	0.133336871	0.066668435	465
6.963224662	741	1617920	2150400	20.06721729	144.5372993	65.93514254	20.06721729	464
6.962086056	740	1617920	2150400	0.066668456	20.20054218	3.066748978	0.066668456	464
6.964851241	742	1617920	2150400	3.400093518	58.93495431	27.66742765	3.400093518	461
7.118400357	738	1617920	2150400	22.20059521	329.4088317	201.2720629	137.0703416	470
7.040649851	739	1617920	2150400	0.80002152	148.6706658	4.533455281	1.0000269	467
7.040487193	741	1617920	2150400	20.40055848	145.87066	66.3351493	20.40055848	467
7.040487193	741	1617920	2150400	0.066668473	11.33364035	0.066668473	0.066668473	467
6.622944207	744	1617920	2150400	16.667118	165.671153	67.00181438	20.80056327	457
6.624733444	727	1617920	2150400	5.266809775	176.938141	27.467413	11.93365759	457

6.622618891	727	1617920	2150400	20.06721418	143.7372551	65.66845838	20.06721418	456
6.61952839	727	1617920	2150400	0.066668462	11.3336386	0.066668462	0.066668462	455
6.619853705	744	1617920	2150400	20.06721714	145.3373202	66.00181053	20.06721714	455
6.631077105	743	1617920	2150400	0.066668439	20.40054244	3.066748209	0.066668439	452
6.62587205	743	1617920	2150400	20.06721692	144.6706336	65.93514132	20.06721692	450
6.624570786	743	1617920	2150400	0.066668482	11.33364196	0.066668482	0.066668482	449
6.614323335	744	1617920	2150400	20.06721429	143.203908	65.66845872	20.06721429	442
6.605377147	744	1617920	2150400	0.066668464	13.06701891	0.333342319	0.066668464	436
6.599684118	745	1617920	2150400	20.06721678	143.5372682	65.73513538	20.06721678	432
6.599684118	745	1617920	2150400	0.066668473	11.26697186	0.066668473	0.066668473	432
6.896209582	714	1626112	2150400	41.20111634	699.4189507	249.5400946	41.20111634	432
6.634980896	745	1626112	2150400	7.000191124	582.6825754	289.3412331	7.000191124	431
6.634980896	744	1626112	2150400	20.06721757	144.6706383	65.80180645	20.06721757	431
6.644577716	744	1626112	2150400	0.066668477	19.73386925	3.066749951	0.066668477	430
6.641161898	745	1626112	2150400	20.06721537	143.5372581	65.73513074	20.06721537	429
6.641324556	744	1626112	2150400	1.266700927	17.26713369	1.800048686	1.266700927	429
6.640185951	745	1626112	2150400	20.0672155	143.3372535	65.66846267	20.0672155	428
6.639860635	744	1626112	2150400	0.066668472	19.66719923	3.06674971	0.066668472	428
6.644415058	744	1626112	2150400	21.73392725	159.404356	70.66859782	21.73392725	430
6.650433402	743	1626112	2150400	2.466734036	157.004288	16.60045338	2.933413449	430

Table 3. Testing Dataset

% Committed Bytes In Use	Available MBytes	System Driver Resident Bytes	System Driver Total Bytes	IO Write Operations /sec	File Control Operations /sec	File Data Operations /sec	File Write Operations /sec	Threads
9.284637244	652	1097728	8830976	108.6147969	1339.254896	319.2395796	114.1205674	513
9.278130841	657	1101824	8728576	23.3536425	819.6461272	109.0281481	23.3536425	510
9.226730256	687	1101824	8728576	0.066668887	12.40041307	0.133337775	0.066668887	507
9.223965035	688	1101824	8728576	20.06732944	145.1381269	65.66883553	20.06732944	504
9.200379323	688	1101824	8728576	0.066668863	14.80048762	0.333344316	0.066668863	501
9.189481098	688	1101824	8728576	20.66734708	159.4719168	69.46895373	20.66734708	496
9.184438636	688	1101824	8728576	0.066668857	14.3338043	0.333344286	0.066668857	494
9.064720818	691	1101824	8728576	16.80055114	143.8713863	65.66882091	20.06732497	488
9.063094218	694	1101824	8728576	0.066668849	11.93372394	0.133337698	0.066668849	487
9.064720818	678	1101824	8728576	24.8674805	307.8767425	90.0029455	28.86761141	487
9.063419538	678	1101824	8728576	0.066668842	11.26703434	0.066668842	0.066668842	487
9.06732338	678	1101824	8728576	20.13398502	152.8716148	69.20223991	20.13398502	487
8.996403586	679	1101824	8728576	0.133337731	130.4043013	3.866794209	0.333344328	487
9.157925043	676	1101824	8728576	25.20081934	363.5451531	224.2739584	140.0712207	495
9.147840118	693	1101824	8728576	0.133337658	32.60105736	0.66668829	0.133337658	493
9.148653419	693	1101824	8728576	20.40066379	144.2713609	66.00214754	20.40066379	492
9.139056474	693	1101824	8728576	0.466681817	22.6007337	0.866694803	0.466681817	489
8.738262042	696	1101824	8728576	18.06725269	162.0052548	66.73549795	20.73400586	480
8.73305692	695	1101824	8728576	0.066668828	20.93401205	3.066766096	0.066668828	478
8.73338224	695	1101824	8728576	20.06731572	143.2713006	65.66879066	20.06731572	478
8.735171501	696	1101824	8728576	0.066668828	14.60047333	0.400012968	0.066668828	477
8.731755639	696	1101824	8728576	20.0673133	143.2712833	68.73554822	20.0673133	473
8.785596125	690	1101824	8728576	31.68734062	410.2442755	105.3349715	37.29014275	473

8.770794058	690	1101824	8728576	20.80517306	163.5073216	70.95097479	20.80517306	471
8.760546473	690	1101824	8728576	1.598386615	18.1150483	2.131182153	1.598386615	467
8.75566667	694	1101824	8728576	20.06730937	144.6046313	65.73543867	20.06730937	464
8.745907066	695	1101824	8728576	0.06666879	11.33369432	0.06666879	0.06666879	458
8.990547823	663	1105920	8728576	38.33455603	673.8881606	227.5405908	38.33455603	459
8.787710706	693	1105920	8728576	11.80038201	770.691616	331.8107416	14.46713501	459
8.765914255	694	1105920	8728576	20.06730534	147.6713665	66.0687694	20.06730534	458
8.777137801	693	1105920	8728576	0.066668812	20.93400694	3.066765348	0.066668812	458
8.778601741	694	1105920	8728576	20.06730737	144.2712729	65.7354321	20.06730737	459
8.774047259	694	1105920	8728576	0.066668791	14.20045253	0.333343956	0.066668791	457
8.772095338	694	1105920	8728576	20.067307	143.2712384	65.66876211	20.067307	456
8.771932678	694	1105920	8728576	0.066668779	11.26702366	0.066668779	0.066668779	456
8.769167457	694	1105920	8728576	21.6673614	159.6717863	70.66893256	21.6673614	454
8.769655437	694	1105920	8728576	0.066668788	11.33369404	0.066668788	0.066668788	454
8.769004797	695	1105920	8728576	20.0673076	146.3380073	66.00210807	20.0673076	454

For testing purpose, the testing dataset is given to the proposed system, which finds the behaviour of the input data as normal or abnormal. Results of some tested dataset are given in the form of a table (shown in Table 4). The last two columns denote the results of the tested dataset. The normal data is represented as N and the abnormal data as A. In the last column, the membership value of each dataset is given for the respective behaviour. The membership value 1.0 denotes that it is completely member of normal or abnormal data.

Table 4. Result of test dataset of the proposed system

% committed bytes in use	Available Mbytes	System Driver Resident Bytes	System driver total bytes	IO write operations /sec	File control operations /sec	File data operations /sec	File write operations /sec	Threads	Normal (N) or abnormal(A)	Member-ship value
8.7690047	695	1105920	8728576	0.0666687	11.533699	0.0666687	0.0666687	454	**N**	**1.0**
6.5995214	729	1687552	2150400	0.7992243	33.167809	1.9314588	0.7992243	424	**A**	**1.0**
8.7556666	694	1101824	8728576	20.067309	144.60463	65.735438	20.067309	464	**N**	**1.0**
6.5915512	734	1687552	2150400	20.067295	143.40449	65.668725	20.067295	427	**A**	**1.0**
9.1390564	693	1101824	8728576	0.4666818	22.600733	0.8666948	0.4666818	489	**N**	**1.0**
7.1449136	728	1679360	2150400	5.2001596	197.13938	47.401454	5.2001596	469	**A**	**1.0**
7.5839623	716	1105920	8728576	0.0666687	11.267024	0.0666687	0.0666687	450	**A**	**0.12**
7.4222454	699	1675264	2150400	15.871385	206.06125	98.762694	15.871385	476	**N**	**0.2**

Graphical results of the tested dataset of the proposed system are given below (refer Figs. 2 to 5):

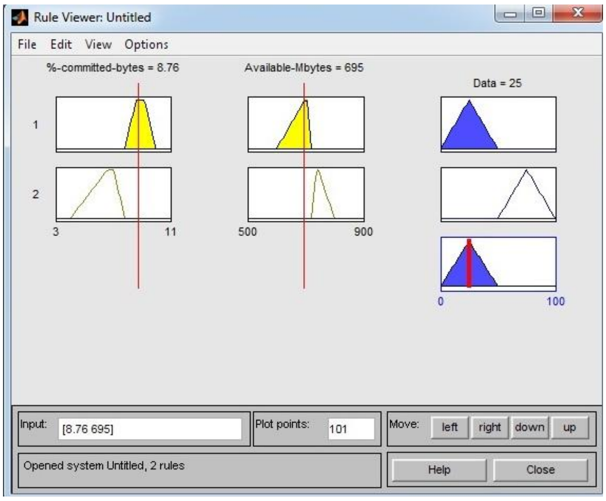

Fig. 2. Normal dataset with membership 1.0

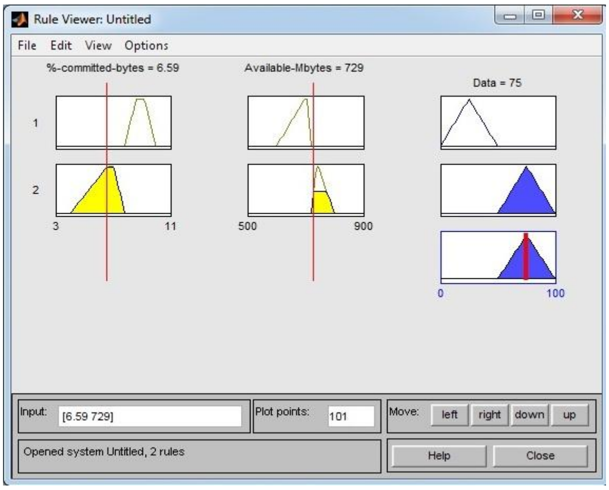

Fig. 3. Abnormal dataset with membership 1.0

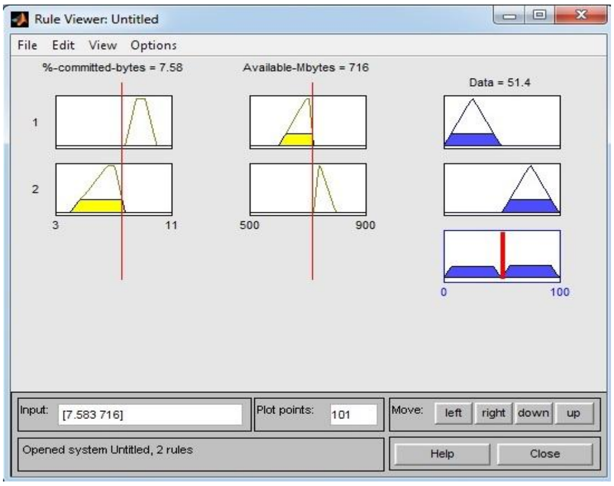

Fig. 4. Abnormal dataset with membership 0.12

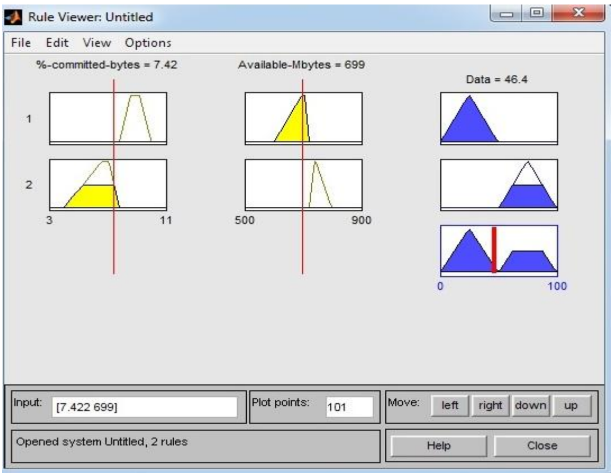

Fig. 5. Normal dataset with membership 0.2

6. CONCLUSION

Sometimes, a user attempts to change the hardware profile of a computer system without the knowledge of the administrator, this can be a sign of intrusion. This paper discusses an intrusion detection system using fuzzy logic, which can detect the changes in the hardware profile of a computer system. A Mamdami fuzzy inference system has been implemented to identify the accurate behavior of the generated system log. Mamdami fuzzy inference system works on the basis of fuzzy IF-THEN rules; so the fuzzy rules generator module has been used to generate the rules for all combinations of the selected attributes. System performance log of a computer system has been used to evaluate the performance of the proposed system. The experimentation results show that the proposed system can be applicable for detecting changes in hardware profile of a computer system.

7. REFERENCES

[1] Adetunmbi A. O., Zhiwei S., Zhongzhi S., Adewale O. S., "Network Anomalous Intrusion Detection using Fuzzy-Bayes," International Federation for Information Processing, Vol. 228, pp. 525-530, 2006.

[2] Biswanath, M., Todd L.H. AND Karl, N. L., "Network Intrusion Detection," IEEE Network, Vol. 8(3), pp. 26-41, 1994.

[3] Han J., Pei J., Yin Y., Mao R., "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach," Data Mining and Knowledge Discovery, Vol. 8(1), pp. 53-87, 2004.

[4] Anderson J. P., Computer Security Threat Monitoring and Surveillance. Technical report, Fort Washington, PA, Apr. 1980.

[5] Denning D. E., "An Intrusion Detection Model," IEEE Trans. on Software Engineering, Vol. 13(2), pp. 222-232, 1987.

[6] Srinivasa K. G., Chandra S., Kajaria S., Mukherjee S., "IGIDS: Intelligent Intrusion Detection System Using Genetic Algorithms," World Congress on Information and Communication Technologies, pp. 852-857, 2011.

[7] Siraj M. M., Maarof M. A., Hashim S.Z.M., "Intelligent Alert Clustering Model for Network Intrusion Analysis," Int. J. Advance Soft Comput. Appl. Vol. 1 (1), pp. 33-48, 2009.

[8] Shanmugavadivu R., Nagarajan N. , "An Anomaly Based Netwok Intrusion Detection System Using Fuzzy logic," IJCSIS, Vol. 8(8), pp. 185-193, 2010.

[9] Dhanalakshmi Y., Babu I. R., "Intrusion Detection Using Data Mining Along Fuzzy Logic and Genetic Algorithms," IJCSNS, Vol. 8(2), pp. 27-32, 2008.

[10] Om H., Hazra T., "Design of Anomaly Detection System for Outlier Detection in Hardware Profile Using PCA," IJCSE, Vol. 4(9), pp. 1623-1632, 2012.

[11] Bharti K., Jain S., Shukla S., "Fuzzy K-mean Clustering Via J48 For Intrusion Detection System," IJCSIT, Vol. 1(4), pp. 315-318, 2010.

[12] Han S. J., Cho S. B., "Evolutionary Neural Network for Anomaly Detection Based on the Behaviour of a Program," IEEE Trans. on Systems, Man and Cybernetics-Part B, Vol. 36(3), pp. 559-570, 2006.

[13] Om H., Sarkar T. K., "Neural network based intrusion detection system for detecting changes in hardware profile," Journal of Discrete Mathematics and Cryptography, Vol. 12(4), pp. 451-466, 2009.

[14] Mamdani, E.H. and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," Int. Journal of Man-Machine Studies, Vol. 7(1), pp. 1-13, 1975.

[15] Stallings, W. Cryptography and Network Security Principles and Practices: Prentice Hall, 1998.

[16] Lamba,V.K., Neuro Fuzzy System, University Science press: 2008

[17] Zadeh, L.A., "Outline of a new approach to the analysis of complex systems and decision processes," IEEE Trans. on Systems, Man, and Cybernetics, Vol. 3(1), pp. 28-44, 1973.

[18] Cirstea, Dinu A., Khor, Mccormick M., Neural and Fuzzy logic control of Drives and Power System, Elsevier : 2002