Mobile Netw Appl (2016) 21:494–505
DOI 10.1007/s11036-015-0644-x

# Anomaly Detection System in Cloud Environment Using Fuzzy Clustering Based ANN

N. Pandeeswari [1] · Ganesh Kumar [1]

Published online: 16 August 2015

**Abstract** Cloud computing affords lot of resources and computing facilities through Internet. Cloud systems attract many users with its desirable features. In spite of them, Cloud systems may experience severe security issues. Thus, it is essential to create an Intrusion Detection System (IDS) to detect both insider and outsider attacks with high detection accuracy in cloud environment. This work proposes an anomaly detection system at the hypervisor layer named Hypervisor Detector that uses a hybrid algorithm which is a mixture of Fuzzy C-Means clustering algorithm and Artificial Neural Network (FCM-ANN) to improve the accuracy of the detection system. The proposed system is implemented and compared with Naïve Bayes classifier and Classic ANN algorithm. The DARPA's KDD cup dataset 1999 is used for experiments. Based on extensive theoretical and performance analysis, it is evident that the proposed system is able to detect the anomalies with high detection accuracy and low false alarm rate even for low frequent attacks thereby outperforming Naïve Bayes classifier and Classic ANN.

**Keywords** Anomaly detection system · Artificial neural network · Cloud computing · False alarm rate · Fuzzy clustering · Naïve Bayes classifier

## 1 Introduction

Many organizations have begun to upload their vast quantity of essential information into public cloud. The sensitive information uploaded into public cloud [1] is vulnerable to security risks such as availability, confidentiality and integrity of those organizations. Besides, the uninterrupted service of cloud technology attracts the intruders to gain access and misuse resources and services provided by Cloud service provider (CSP). The anomaly or intrusion may be an attack [2] to end user's private data, CPU utilization, bandwidth usage, processing power and storage capacity of the cloud system. To protect the user's data and cloud resources from malicious activities, firewall and intrusion detection systems are the only permanent solutions. Firewall is not suitable for detecting insider attacks. Some of the Denial of Service attacks (DoS) and Distributed Denial of Service attacks (DDoS) are too complex to detect with firewall. In order to protect the cloud computing environment, it is imperative to develop an anomaly detection system. A traditional network-based or host-based intrusion detection system [1, 3] does not suit virtual cloud environment. So, it is necessary to develop an anomaly detection component which is suitable for detecting the malicious activities in cloud computing systems. In this work, an anomaly detection system at hypervisor layer [4, 5] is developed and is named Hypervisor Detector to detect the anomalous activities.

For inventing intrusion detection systems, many research works have applied various data mining and machine learning approaches [5]. Also, the existing intrusion detection systems use rule sets of different attack patterns which are stored in databases. To avoid any unauthorized and illegal activities, the whole network traffic and user behaviours are matched against the attack patterns. If the attack pattern is already stored in the database, then it can be detected. Due to this, the database should be updated manually. Therefore, the proposed system,

✉ N. Pandeeswari
tparameshsee@gmail.com

[1] Department of Information Technology, PSNA College of Engineering and Technology, Dindigul, India

namely Hypervisor Detector is designed and implemented by using Fuzzy C-Means clustering –Artificial Neural Network (FCM-ANN) for which there is no necessity of manually updating the database. FCM-ANN can automatically capture the patterns of new attacks. The proposed model is trained and tested with DARPA's KDD cup dataset which is specifically designed for intrusion detection system. Finally, a comparison is made among various anomaly detection systems which are created by using Naïve Bayes classifier and Artificial Neural Network.

## 2 Related works

Most researchers have developed intrusion detection systems to detect the intruders in cloud environment by employing various machine learning approaches [3], log based [6] approaches and data mining techniques [7, 8]. Since, cloud specific attacks do not necessarily leave any traces in a node's operating system and virtual machines change their states dynamically, it is very hard to detect intrusions by using the traditional IDS. So, by considering the dynamic nature of cloud virtual machines, many research works [4, 9, 10] have applied Virtual Machine Monitor (hypervisor) layer.

The authors, Modi et al. [7] have developed a Network Based Intrusion Detection (NIDS) component in cloud computing system which uses Snort and signature Apriori algorithm. The NIDS module is integrated into cloud computing environment to monitor the traditional and the virtual network. This system is developed to detect the known and derivative of known attacks by monitoring the network traffic. The authors, A. Bakshi et [11] al have developed an intrusion detection component to detect the Denial of Service attack. In this methodology, snort is installed in the virtual switch to capture the network traffic. The traffic details are analyzed and the detection system determines the nature of the attack and informs the virtual server. If any threat is found, the virtual server stops the malware actions by blocking communication from that IP address. If impersonation takes place, this system cannot detect the machine which causes intrusion.

The authors, Vereia et al. [3] have proposed a Grid and Cloud Computing Intrusion Detection System (GCCIDS) that employs an audit system. GCCIDS integrates knowledge and behavior analysis to discover the intrusions. This system makes use of an event auditor that captures data from various resources like system logs, node messages and services. Based on the captured data, the IDS service can be used to detect intrusions by using behavior based and knowledge based techniques. GCCIDS uses artificial neural network for behavior analysis.

The authors, Hai Jin et al. [1] developed an intrusion prevention system, named VMFence. The VMFence is designed to monitor the flow of network and the integrity of files. The VMFence monitors the virtual bridge on the privileged VM, where all communications among the virtual machines should pass. This system suffers from computational complexity. The researchers, C. Mazzariello et al. [12] have developed IDS for eucalyptus cloud. This work deploys Snort on cloud controller and on physical host machines to detect outside attacks only.

The researchers K. Jones et al. [13] have developed an intrusion detection system for Distributed architecture (DIDS) to detect known and unknown attacks since it extracts the benefits of both NIDS and HIDS. In virtual Cloud environment, DIDS can be located at host device or at the processing server (in backend). Still, the system lacks with detection accuracy. The authors, Feng et al. [8], have applied Hidden Markov model for abnormality detection by using frequent system call sequences to detect the signature based attacks in virtual dynamic execution environment. This system uses automated mining algorithm (AGAS) to create frequent system calls. The performance of data mining based detection techniques depend on the quality of training and the detection models. For very large datasets, this system does not work well because of its low level learning capability.

The authors, Sanjay Ram [4] et al., have developed an IDS by using two approaches namely: Performance approach and Information approach. The Performance approach is used to create user profile based on user behaviours. The information approach is used to analyze the user actions. The audited data is sent to the IDS service core which uses classis Artificial Neural Network. In [5], the authors, Amjad et al. have developed an Intrusion detection system at Virtual Machine Monitor layer by using two different approaches namely: Naïve Bayes classifier and a hybrid approach which is a mixture of Naïve Bayes classifier and Random forest to control and analyze the network flow among the virtual machines in cloud environment. Since classic ANN and Naïve Bayes based IDS yields lower detection rate for least-frequent attacks such as Remote to Local (R2L) and User to Root (U2R). Therefore, to improve the performance of the anomaly detection system, the proposed method uses FCM-ANN. The proposed Hypervisor Detector is implemented at (Virtual machine monitor) hypervisor layer which can monitor the activities of virtual machines that are running on the dynamic environment.

## 3 Background

In this section, cloud security issues, motivation, contribution and various attacks to cloud are discussed as follows.

### 3.1 Security issues

Cloud computing has a number of security issues that fall into two categories. They are 1. Security issues to cloud service providers 2. Security issues to consumers. Cloud service providers have to ensure that the cloud infrastructure and services provided are safe and secure. Besides, they have to assure that the consumers' data are protected unless and otherwise they use strong passwords and authentication mechanism. The organization that hosts the sensitive information into public cloud does not have physical access to servers hosting those information. Trusting a cloud system depends strongly on the deployment model, as governance of data and applications are outsourced and are out of user's control. This may create problems such as Loss of control, Lack of trust (mechanisms) and Multi-tenancy. In [14] the authors, Keiko Hashizume et al. have discussed security issues to cloud computing such as 1. Lack of employee screening and poor hiring practices 2. Lack of customer background checks and 3. Lack of security education.

### 3.2 Motivation

With modern technological development, all organizations, whether big or small, have started to upload their details into public cloud. While uploading users' sensitive information onto the external online storage, the system integrity, confidentiality and availability have to be guaranteed. The open and distributed infrastructure of cloud computing attracts the intruders. Cloud computing involves service oriented architectures, multitenancy, multicasting group. The cloud computing system is vulnerable to various threats that include integrity, confidentiality and availability of resources and data. Besides, the virtualized infrastructure can be exposed to several threats. The problem becomes more critical when the stored data and computing power is abused by the inside intruder that makes cloud system itself a threat. Lack of control over the virtualized environment is the severe concern for cloud service consumers. This necessitates in importance of intrusion detection system. There may be possibility for a number of hackers like terrorists; and other organizations can steal users' information.

The information safety measures have to be assured while using Internet, cloud computing, and wireless applications. This raises the need for secure and safe security systems through the use of firewalls, intrusion detection and prevention system and other cryptographic primitives. The intrusion detection system has to perform early detection of malware activities and protect the system from serious damage. This work discuss about intrusion detection system to protect the cloud system from the severe attacks such as probe, DoS, User to Root (U2R) and Remote to Local (R2L) attacks. The performance of IDS can be measured in terms of detection accuracy.

### 3.3 Contribution

With technological improvement of Internet, the network security concerns have become one of the key issues in web applications. The illegitimate users can impersonate the authorized users and try to destroy the services and the resources provided for authorized users. To confront the intruders, IDS should provide security means by examining configurations, logs, network, and user behaviours. Perhaps, the IDS should be distributed to work with virtual cloud environment, so as to monitor each node and make an alert in that environment in case of any malware actions. The proposed system uses middleware layer of cloud called Virtual Machine Monitor (VMM) layer which is also referred as Hypervisor layer.

Intrusion detection systems attempt to discover the unauthorized and illegal actions by examining various user actions on the network. The optimal IDS should detect intrusions with high detection accuracy. In order to improve detection accuracy, many researchers have employed rule based expert systems and some statistical approaches. However for very large datasets, the rule based expert systems and statistical approaches become inferior. Hence, there are a number of data mining approaches that have been introduced to resolve this problem. Artificial Neural Network is one of the widely used approaches and it has been successful in solving complex problems.

There are some drawbacks existing in ANN based IDS. The main drawback is lower detection precision for low frequent attacks such as Remote to Local (R2L) and User to Root (U2R). Since the learning sample size of the low frequent attacks are too small compared to high frequent attacks, it is difficult to detect the low frequent attacks with high detection accuracy. Consequently to improve detection accuracy, the fuzzy clustering technique is incorporated into ANN based-IDS. FCM-ANN uses divide and conquer strategy to improve the detection accuracy of IDS. This work designs an intrusion detection system to detect malicious intruders in virtual cloud. The Design objectives are 1. Detection of anomaly in cloud 2. High accuracy 3. Low false positives 4. Scalability and 5. Compatibility.

### 3.4 Various types of attacks to cloud system

With desirable features and constant development, cloud computing system has to provide high quality service and secure user's sensitive information. This attractive technology may be the place for several intruders to gain uninterrupted services and resources. The attacks that may affect cloud computing system are 1. Insider attack 2. Flooding attack 3. User to root attack and 4. Port scanning 5. Attacks on virtualization and 6. Backdoor channel attacks.

#### *3.4.1 Insider attack*

The employers, entrepreneurs and associates who work currently or earlier have full permissible access to the user's information and are referred as insiders. The insiders may aim to disclose user's information to others, and obtain unprivileged access to the cloud resources. This type of attack causes severe security risks. The malicious insider attack is very difficult to realize.

#### *3.4.2 Flooding attack*

In this attack, the user sends huge amount of information continuously from innocent host (zombie) to the victim machine. Packets can be TCP, UDP, ICMP or a mixer of them. The main aim of this attack is to deny access of authorized users and hack the cloud resources. This type of attack can be realized or made to happen over illegitimate/ unauthorized network connections. This type of attack affects the service availability to the authorized users.

#### *3.4.3 User to root attacks*

In this type, the attackers [15] may steal the login credentials of authorized users and try to gain limitless access the whole system. An attacker, who obtains login and password of authorized information, can gain access to the server and the virtual machines to the root level. Buffer overflow method can be used to detect this attack [2, 15]. There are no security mechanisms to handle the weak password recovery techniques and phishing attacks.

#### *3.4.4 Port scanning*

Port scanning method provides a list of open, closed and filtered posts. Through port scanning, the attackers can find open ports and attack the services on these ports. Port scanning reveals the entire network related information like IP address, MAC address, router and gateway filtering and firewall rules. In cloud system, the services offered on the open ports are affected.

#### *3.4.5 Attacks on virtualization*

After compromising hypervisor, control of virtual machines of virtual environment will be captured. Zero day attack is one of the attacks for virtual machines.

#### *3.4.6 Backdoor channel attacks*

In this attack, hacker compromises a node and uses the compromised node as zombie process to do denial of service attack. This is the passive attack which affects confidentiality, and aims to gain remote access to the infected node. The anomaly based and signature based detection techniques can be used to detect the flooding attacks and backdoor channel attacks.

## 4 Anomaly detection system in virtual cloud environment

Cloud computing is a revolutionary model in offering computing facilities and storage resources as services through Internet to satisfy the requirements of users with lower cost. With its increasing improvements, the cloud computing system is more vulnerable against security threats. The IDC survey [16] concluded that the security of cloud services is the greatest challenge. As the cloud services are delivered over Internet; the security and privacy on the public network should be a severe concern. With the fast growing web based services, Cloud influences many users to seek web services as well as intruders to misuse the cloud resources.

Due to the open nature; the cloud system becomes the most attractive place for more vulnerability. A survey on business and technical issues, at design and implementation levels, have made [17] the issues directly affect the performance of cloud computing. Cloud applications execute beyond the firewall and move to public domain, and may have a severe consciousness on security. This makes a requirement to provide anomaly detection system in virtual cloud environment which overcomes the problem with traditional computing environments. However, with dynamic nature of cloud system, monitoring activities of virtual machines have become more difficult. Therefore, neither Host based nor Network based intrusion detection systems suits this virtual environment.

The host based intrusion detection [18] in hypervisor or host machine would allow IDS to examine the hypervisor and virtual operating systems on the same hardware platform. Here, if the host is compromised, the HIDS on the hardware platform would be neutralized. With rapid flow of large volume of data and dynamic change of virtual machines, network based IDS does not suit the cloud computing infrastructure. Therefore, to detect unauthorized and malware access in and around the cloud system, this paper proposes hypervisor based anomaly detection system in virtual distributed environment. This paper exemplifies a method to detect the unusual behavior in virtual network which uses FCM-ANN based analysis to detect anomalies. The representation of cloud architecture with Hypervisor Detector is shown in Fig. 1.

### 4.1 Hypervisor detector

Hypervisor is a piece of software, firmware that creates and executes virtual machines. Hypervisor is the software layer that executes on hardware platform. A machine on which

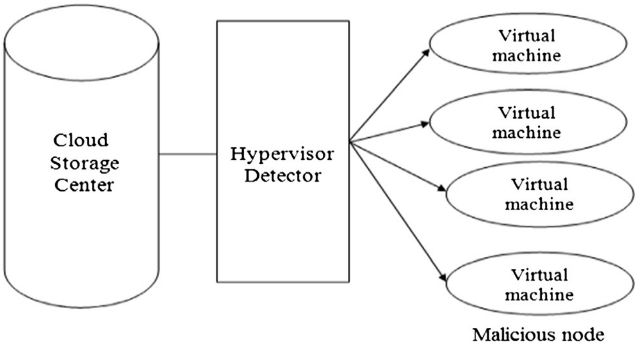


**Fig. 1** Cloud system with Hypervisor Detector

the hypervisor is running virtual machines is referred as host machine. Hypervisor (VMM) has capabilities to monitor and examine the network based and host based events on virtual environment. Hypervisor provides a virtual operating platform and manages the operation of guest machines. Every host machine is presented with a virtual hypervisor that runs separately from host machine. The virtual hypervisors monitor the real hardware systems which provide single platform for different VMs. This ability provides hypervisor based virtualization [19] to acquire a secure infrastructure. The hypervisor, as a hardware element, is used to discover network based intrusion. Hypervisor Detector monitors the virtual network traffic (network based events) to capture the network data and analyse it. The Hypervisor Detector is experienced with the database of normal activities; any deviation from this is notified as anomalous activity.

### 4.2 Virtual machine

Virtual machines execute user application process on single hardware platform. Due to its dynamic mature, the virtual machines can dynamically modify their states (new, execute, kill). VM can migrate from one platform to other without any conditions. VM can be clogged, poised and be infertile. Virtual machines on a single hardware platform run various applications that can be stored on different locations of the host machine. Due to dynamic nature of cloud infrastructure [20], it is possible for the virtual computation environment to relocate itself and scale its resources across a multi-domain infrastructure.

### 4.3 Virtualization

Virtualization [1] is to afford parallel and interactive access to a large pool of information centre that supports numerous instances of OS running on a single hardware platform and can control the multiple OS consecutively resulting in hardware virtualization. Hypervisor permits multiple instances of OSes to share hardware facilities on which it is hosted. The operating system installed and executed on a virtual machine is called Guest OS. The Hypervisor monitors and executes [19] the guest operating system running on the virtual machines functioning on it.

## 5 Design of hypervisor detector using FCM-ANN

Cloud computing system works with the concept of virtualization of application and storage resources. Hypervisor in cloud system monitors various guest operating systems executing on same hardware platform. With its open nature and enormous amount of traffic data, the cloud computing system becomes an attractive place for hackers. So, an efficient, well-designed and effective intrusion detection system is essential. Therefore, the proposed work is designed to examine the anomalies on virtual network by analyzing the network based events on multiple virtual machines. This Hypervisor Detector is trained by Fuzzy C-Means clustering Artificial Neural Network. This model is trained to observe the operations on virtual machines. A collection of fuzzy sets (fuzzy space) [21] defines the fuzzy linguistics values or fuzzy classes. The fuzzification parameter in the range [1, n] determines the degree of fuzziness among data points. Fuzzy clustering is the process of assigning these fuzzy membership functions and cluster the data points to the corresponding cluster group. The Fuzzy C-Means algorithm endeavors to cluster a finite collection of n elements $x=\{x_1,x_2,\ldots,x_n\}$ into a number of fuzzy clusters with respect to the membership values.

Given a set of data, the algorithm returns a list of C cluster centers $c=\{c_1,c_2,\ldots,c_k\}$ and a partition matrix $w=w_{ij}\in[0,1]$.

Where,

$$i=\{1,2,\ldots,n\}$$
$$j=\{1,2,\ldots,k\}$$

$w_{ij}$ Fuzzy membership value that determines the degree to which the data point is related.

There are two steps involved into Fuzzy C-Means clustering algorithm. The Framework of FCM-ANN is shown in Fig. 2. FCM-ANN [22] consists of three phases.

The three phases of FCM-ANN is explained as follows.

Phase I In the first phase, fuzzy clustering approach is used to create different clusters according to the fuzzy membership values.

Phase II Based on the various clusters, the different ANN is trained.

Phase III To evacuate the error of the different ANN, fuzzy aggregation module can be used to combine the results of different ANNs.

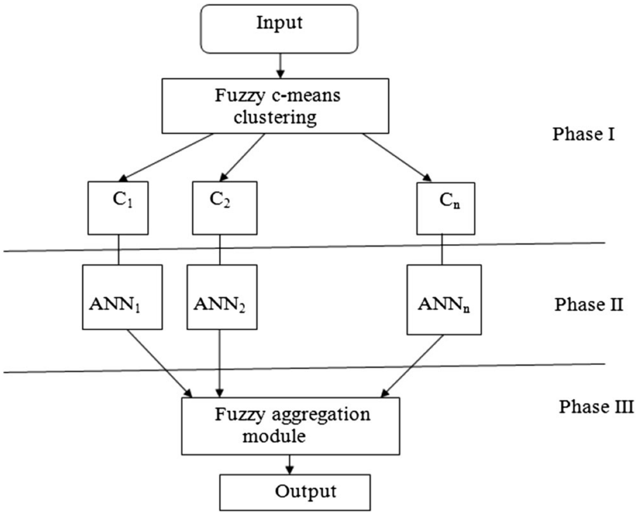


**Fig. 2** Framework of FCM-ANN

### 5.1 Phase I: fuzzy clustering module

The task of Fuzzy Clustering module is to segment the data set into different clusters based on fuzzy membership values. The data within the same cluster should have homogeneity. The data belonging to different clusters should have heterogeneity. In the first phase, the whole data undergoes clustering. Fuzzy C-Means clustering algorithm of soft clustering technique is used. Fuzzy C-Means algorithm is a text clustering technique, in which every data belongs to a cluster group. The cluster group is defined based on the fuzzy membership function. This is based on the following object function minimization *j* as in Eq. 1.

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

where

$1 \le m < \infty$
$m$–*real number* $(> 1)$
$u_{ij}$–*degree of membership for the data* $x_i$ *in the* $j^{th}$ *cluster*
$x_i$–$i^{th}$ *data*
$c_j$–*center of cluster*

The fuzzy clustering is an iterative approach which can be carried out by updating the fuzzy membership $u_{ij}$ and the clusters center $c_j$ by Eqs. 2 and 3 respectively.

$$u_{ij} = \frac{1}{\sum_{p=1}^{k} \left(\frac{x_i - c_j}{x_i - c_p}\right)^{\frac{2}{m-1}}} \qquad (2)$$

$$c_j = \frac{\sum_{i=1}^{n} u_{ij} x_i}{\sum_{i=1}^{n} u_{ij}} \qquad (3)$$

The iteration will stop at lower value of $\varepsilon$ which can be verified by using Eq. 4.

$$\max_{ij} \left\{ u_{ij}^{(q+1)} - u_{ij}^{q} \right\} < \varepsilon \qquad (4)$$

where

$\varepsilon$–*termination criterion*; $(0 < \varepsilon < 1)$
$q$ – *number of iteration*

Based on the above discussion, the fuzzy cluster module contains the following steps.

1. *Initialize* $U = [u_{ij}]$ *matrix* : $u(0)$ *and* $q = 1$ (5)

Table 1 Attribute description for dataset

| Sl. No | Attribute number | Attribute | Description | Type | Domain type |
|---|---|---|---|---|---|
| 1 | 1 | Duration | Length of the connection (number of seconds) | continuous | real |
| 2 | 3 | Service | Network service on the destination (TCP, UDP etc.) | discrete | Integer |
| 3 | 5 | src_bytes | number of data bytes from source to destination | continuous | real |
| 4 | 6 | dst_byte | number of data bytes from destination to source | continuous | real |
| 5 | 23 | Count | number of connections to the same host as the current connection in the past two seconds | continuous | real |
| 6 | 24 | srv_count | number of connections to the same service as the current connection in the past two seconds | continuous | real |
| 7 | 32 | dst_host_count | count for destination host | continuous | real |
| 8 | 33 | dst_host_srv_count | srv_count for destination host | continuous | real |
| 9 | 35 | dst_host_diff_srv_rate | diff_srv_rate for destination host | continuous | real |
| 10 | 36 | dst_host_same_src_port_rate | same_src_port_rate for destination host | continuous | real |
| 11 | 38 | dst_host_serror_rate | serror_rate for destination host | continuous | real |

**Table 2** Dataset for attack distribution

| Sl. No | Attack type | Number of records | % of occurrence |
|---|---|---|---|
| 1 | Normal | 5700 | 31.67 |
| 2 | Probe | 2155 | 11.97 |
| 3 | DoS | 3475 | 19.3 |
| 4 | U2R | 70 | 0.38 |
| 5 | R2L | 6600 | 36.67 |
| Summary | Total | 18,000 | 100 |

2. $q^{th}$ *iteration*

Calculate the centre vectors as Eq. 6.

$$c(q) = [c_j] \ \ with\ u(q) \tag{6}$$

$$c_j = \frac{\sum_{i=1}^{n} u_{ij} x_i}{\sum_{i=1}^{n} u_{ij}} \tag{7}$$

3. *Update* $U(q+1)$

$$u_{ij} = \frac{1}{\sum_{p=1}^{k} \left( \frac{x_i - c_j}{x_i - c_p} \right)^{\frac{2}{m-1}}} \tag{8}$$

4. *If* $\left( u_{ij}^{(q+1)} - u_{ij}^{q} \right) < \varepsilon$ (9)

else do
step 2.

then do
step 5.

5. Based on the *argmax*($u_{ij}$), every data value is allocated into the corresponding clusters.

After five steps of fuzzy clustering module, every data in the dataset is allocated to various clusters. Using the above mentioned steps, the data set can be classified into k separate clusters. In the next phase, the resultant clusters are trained by various Artificial Neural Network modules. The fuzzy clustering module is used to reduce the size and the complexity of the training dataset. Therefore, the performance of the artificial neural network module can be improved.

**Table 3** Results of hypervisor detector for various attacks

| Attack type | Detection rate (%) | False alarm rate (%) |
|---|---|---|
| Normal | 99.8 | 0.2 |
| Probe | 99.73 | 0.27 |
| DoS | 99.96 | 5.33 |
| U2R | 96.78 | 3.22 |
| R2L | 93.73 | 6.27 |

**Table 4** Performance comparison for Normal under various techniques

| | Naïve Bayes | ANN | FCM-ANN |
|---|---|---|---|
| Precision | 90.71 | 90.89 | 92.73 |
| Recall | 98.33 | 98.94 | 99.12 |
| *F*-value | 94.53 | 94.9 | 96.31 |

### 5.2 Phase II: ANN module

The result of fuzzy clustering module is a number of clusters that maintains homogeneity within the cluster and heterogeneity between the modules. In this phase, the resultant clusters of fuzzy clustering module can be used as input for various ANN. The back propagation algorithm is used in ANN module to train the various clusters. Since the size and the complexity of the training set are greatly reduced, the effectiveness of the consequent ANN module can be improved.

ANN is a biologically inspired form of distributed computation [23, 24]. ANN consists of simple processing units, neurons and interlinks (connections). This module aims to learn the patterns of each cluster. This module exploits feed-forward neural networks trained with back propagation algorithm to detect the intrusions.

The feed forward neural network consists of an input layer, an output layer and number of hidden layers. In each node *i*, the input layer has the value $x_i$ as the input, multiplied by a weight value between input layer and hidden layer. Each node j in the hidden layer receives input value ln(*j*) according to Eq. 10.

$$\ln(j) = \theta_j + \sum_{i=1}^{n} x_i w_{ij} \tag{10}$$

The bipolar sigmoid activation function is used to process the ln(*j*) using Eq. 11.

$$f(x) = \frac{2}{1-e^x} - 1 \tag{11}$$

**Table 5** Performance comparison for DoS under various techniques

| | Naïve Bayes | ANN | FCM-ANN |
|---|---|---|---|
| Precision | 99.23 | 99.52 | 99.94 |
| Recall | 95.65 | 97.30 | 97.2 |
| *F*-value | 98.78 | 99.13 | 99.32 |

**Table 6** Performance comparison for Probe under various techniques

| | Naïve Bayes | ANN | FCM-ANN |
|---|---|---|---|
| Precision | 58.7 | 65.71 | 53.61 |
| Recall | 89.1 | 89.73 | 82.23 |
| *F*-value | 68.39 | 76.36 | 63.05 |

The output of the activation function *f*(ln(*j*)) then broadcasts all of the neurons to the output layer as in Eq. 12.

$$y_k = \theta_k + \sum\nolimits_{j=1}^{m} w_{jk} f(\ln(j)) \tag{12}$$

where$\theta_j$ and $\theta_k$ are the biases in the hidden layer and the output layer.

To find out the mean absolute error value $E_m$, the output value is compared with the target value by using Eq. 13.

$$E_m = \frac{1}{2n}\sum\nolimits_k \sqrt{(T_k - y_k)^2} \tag{13}$$

where

$n-number\ of\ training\ samples$
$y_k-\quad output\quad value$
$T_k-\quad target\quad value$

To reduce the gradient error and find the global optimum of network weights, the partial derivatives are $\frac{\partial E}{\partial w}$ calculated for each of the weight in the network. The weight will be adjusted according to following Eq. 14.

$$w(t+1) = w(t) + \eta \frac{\partial E(t)}{\partial w(t)} \tag{14}$$

To speed up the convergence of the error in learning process, the momentum gain value is included into Eq. 14 to get Eq. 15.

$$w(t+1) = w(t) + \eta \frac{\partial E(t)}{\partial w(t)} + \alpha \Delta w(t) \tag{15}$$

where

$0 < \alpha < 1$

**Table 7** Performance comparison for R2L under various techniques

| | Naïve Bayes | ANN | FCM-ANN |
|---|---|---|---|
| Precision | 43.45 | 56.32 | 94.52 |
| Recall | 9.35 | 7.31 | 60.73 |
| F-value | 16.37 | 12.51 | 75.92 |

**Table 8** Performance comparison for U2R under various techniques

| | Naïve Bayes | ANN | FCM-ANN |
|---|---|---|---|
| Precision | 28.33 | 52.17 | 85.30 |
| Recall | 8.32 | 27.37 | 78.54 |
| *F*-value | 11.89 | 33.02 | 81.43 |

### 5.3 Phase III: fuzzy aggregation module

The Fuzzy aggregation module is used to combine the resultant ANN network modules into a single ANN module. The Fuzzy aggregation module is to aggregate different ANN module's result and reduce detection errors as every *ANN*$_i$. In ANN module only learns from that clusters. As the errors are nonlinear, optimization process uses another new ANN to learn the errors as follows.

1. Let the whole data set be the input for every trained *ANN*$_i$ and get the results as Eq. 16.

$$y_j = \left[y_{j1}, y_{j2}, \ldots, y_{jk}\right] \tag{16}$$

where

$j\ =\ 1\ ,\ 2\ ,\ \ldots\ ,\ n$
$n-is\ the\ number\ of\ input$
$y_{jk}-\ output\ \ of\ \ AN\ N_k$

2. Make input for new ANN as Eq. 17.

$$y_{input} = [y_1.u_1, y_2.u_2, \ldots, y_n.u_n] \tag{17}$$

where

$u_n-is\ the\ membership\ value\ belonging\ to\ C$

3. Train the new ANN with input value $y_{input}$ as in Eq. 17.

Based on the above three steps, the new ANN can learn the errors which are caused by individual *ANN*$_i$ in ANN modules. The proposed model is verified by using the DARPA's KDD

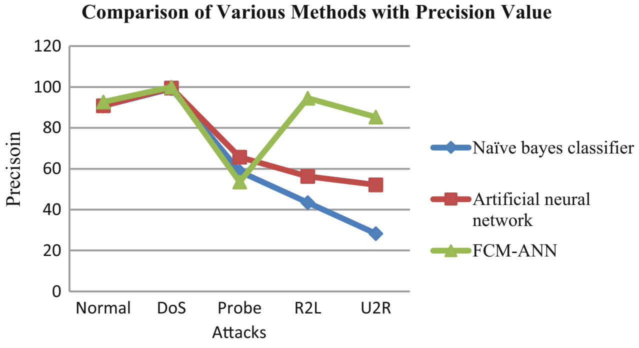
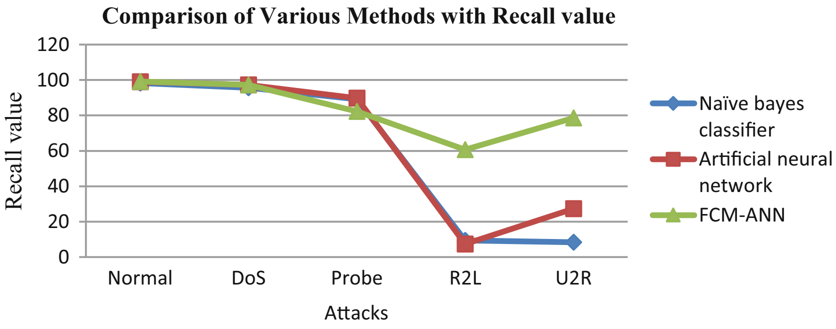


**Fig. 3** Performance comparison of FCM-ANN with NB, ANN with Precision value

Fig. 4 Performance comparison of FCM-ANN with NB, ANN with Recall value



cup dataset [25] which uses input attributes and output attributes from that dataset. KDD cup is specifically designed for intrusion detection system. In this paper, the 11 different attributes have been selected from KDD cup data set. The descriptions of attributes have been specified [26] in Table 1.

## 6 Experiments and results

To implement the Hypervisor Detector, this work uses cloud simulator; cloudsim 3.0. The Hypervisor Detector is trained and tested in cloudsim 3.0. To train and test the proposed system, the DARPA's KDD cup dataset 1999 is used. This dataset has 41 features and a label specifying the record as either normal or attack. Here, the attack types are categories such as 1. Denial of service: denying access to legitimate users by making some computing or memory resources too busy. 2. Probe (PRB): scanning the host and port to collect information or to find out the known vulnerabilities. 3. Remote to local (R2L): unauthorized users access from a remote machine in order to exploit the host's vulnerabilities. 4. User to root (U2R): unauthorized access to a root machine starting from simple host machine attack. For implementation, 18,000 records from KDD dataset are used and the same is explained in Table 2.

For testing the system model, the KDD test dataset is used. The measurements frequently proposed to evaluate the performance of anomaly detection system are as follows.1. True positives 2. True negatives 3. False positives and 4. False negatives. True positive implies that the anomaly detection system detects exactly the attack that has occurred. True negative: This value implies that the detection system has not made a mistake to detect the normal condition. False positives: This value implies that IDS has mistakenly marked the normal condition as abnormal. If this value is consistently high, this can cause the administrator to intentionally disregard the system warnings, which make the system in dangerous status. False negatives: it indicates that the anomaly detection system fails to detect intrusions after a particular attack has occurred.

### 6.1 Performance evaluation

This clustering and classification algorithm uses only continuous values. For implementation, the KDD dataset is used which has discrete and continuous values. Hence, the discrete values are converted into continuous values by doing fuzzification. The fuzzy clustering module is used to divide the training dataset into a number of small clusters. In this next stage, artificial neural network and fuzzy accumulator module is implemented. For ANN, three layer feed forward neural network is used. The ANN module uses 11 input nodes and 5 output nodes. Likewise, the fuzzy accumulator module uses five input and output nodes. The nodes in the input and hidden layers use sigmoid transfer function and the nodes in output layer use linear transfer function. While training the model, the training error obtained is 0.0013. The experiment results for detecting various attacks are shown in Table 3.

Fig. 5 Performance comparison of FCM-ANN with NB, ANN with F-value

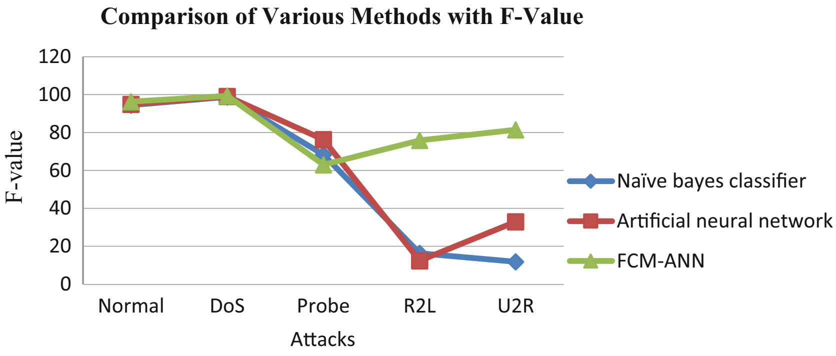
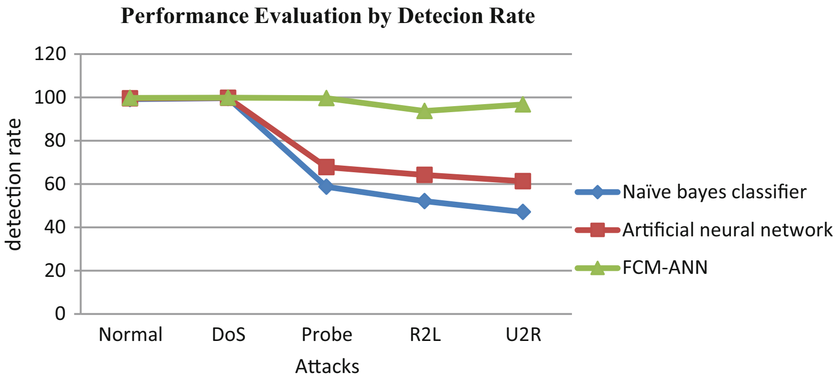


**Fig. 6** Comparison of FCM-ANN with NB and ANN based on Detection rate



Since minimum numbers of instance are used [22] for measuring the standard performance measure, these instances are not sufficient. Due to this reason, the precision, recall and F-values do not dependent on the size of the input samples that are used. They can be defined by using Eqs. 18, 19, and 20 respectively.

$$precision = \frac{TP}{TP + FP} \tag{18}$$

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

$$F\text{−}value = \frac{(1 + \beta^2)*recall*precision}{\beta^2*(recall + precision)} \tag{20}$$

where

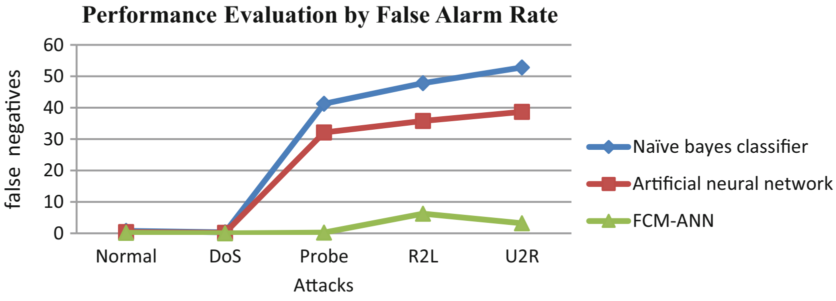
TP number of true positive
FP number of false positive
FN number of false negative
β Relative importance of precision vs. recall (β =1)

The results of proposed system with classic ANN [4] and Naïve Bayes [5] technique have been compared. The comparison results are shown in Tables 4, 5, 6, 7 and 8 and (Figs. 3, 4 and 5). The Tables 4, 5, 6, 7 and 8 values clearly show the difference of every evaluation criteria under various attacks. The Hypervisor Detector using FCM-ANN can get similar results as Naïve Bayes and ANN for detecting high frequent attacks such as normal, DoS and probe. The Hypervisor Detector yields highest detection accuracy for low-frequent attacks such U2R and R2L, whereas the detection systems using Naïve Bayes and ANN can't give better results.

Table 3 shows the detection rate and the false alarm rate of Hypervisor Detector under various attacks such as Normal, Probe, DoS, U2R and R2L. The Tables 4, 5, 6, 7 and 8) show the performance comparisons of various models such as Classis ANN, Naïve Bayes, and FCM-ANN under different evaluation criterion such as Precision, Recall, and F-value for various attacks. The Tables 5 and 6 show the performance comparisons for high frequent attacks such as DoS and probe. From the Tables 5 and 6, it can be observed that FCM-ANN yields same results compared to Classic ANN and Naïve Bayes. Furthermore, the Tables 7 and 8 show the comparisons results of low frequent attacks such as R2L and U2R respectively. These Tables' exhibit that performance of FCM-ANN is greatly higher when compared to Classic ANN and Naïve Bayes.

The (Figs. 3, 4 and 5) show the performance comparisons of the various attacks under different evaluation criterion. The performances of various models are compared by using precision value which is represented in Fig. 3. From Fig. 3, it is observed that the high frequent attacks such as DoS and probe have same precision value under various models. But, the lower frequent attacks such as R2L and U2R have higher values in FCM-ANN when compared to classic ANN and Naïve Bayes under precision value. The Fig. 4 shows the performance comparisons of Naïve Bayes, classic ANN and FCM-ANN using recall value. From Fig. 4, it is evident that

**Fig. 7** Comparison of FCM-ANN with NB and ANN based on False alarm rate



the FCM-ANN offers maximum recall value for lower frequent attacks when compared to classic ANN and Naïve Bayes.

The Fig. 5 shows the performance comparisons of classic ANN, Naïve Bayes, and FCM-ANN based on F-Value. The Fig. 5 shows the maximum F-value for various types of attacks while classic ANN and Naïve Bayes yield lesser value for lower frequent attacks. From the results, it is evident that the FCM-ANN can be worthful for detecting both high frequent and low frequent attacks. In (Figs. 3, 4 and 5), it is shown that the FCM-ANN is suitable for detecting both high frequent attacks and low frequent attacks with highest detection accuracy. The detection systems using classic ANN and Naïve Bayes can detect high frequent attacks with high detection rate; while giving low detection rate for low frequent attacks.

For evaluation and comparison, detection rate and false alarm rate are used. From Table 3 values it is understood that the anomaly detection system using FCM-ANN is best as it offers high detection accuracy and low false positives. The detection rate and false alarm rate of the proposed system is compared with classic ANN and Naïve Bayes classifier which are shown in Figs. 6 and 7 respectively. From the Fig. 6, it is observed that the proposed system gives high detection accuracy for both high frequent and low frequent attacks. The Fig. 7 shows the proposed system which produces low false alarms when compared to NB and classic ANN. The Fig. 6 shows the detection rate of Naïve Bayes, classic ANN and FCM-ANN. From Fig. 6, it is evident that the FCM-ANN gives same detection rates for DoS and Probe attacks under various models. But, low frequent attacks such as R2L and U2R can be detected with high detection rate by using FCM-ANN when compared to Classic ANN and Naïve Bayes. The Fig. 7 shows the performance evaluation of various models by using False alarm rate. The Fig. 7 exhibits that that proposed model gives minimum false alarm rate when compared to classic ANN and Naïve Bayes classifier.

## 7 Conclusion

This works deploys an anomaly detection system called Hypervisor Detector at the virtual machine monitor layer. The Hypervisor Detector is designed with a hybrid approach FCM-ANN which is a combination of Fuzzy C-Means clustering and Artificial Neural Network. This model works in three phases. The first phase of FCM-ANN is fuzzy clustering module which is used to divide the large dataset into small clusters so as to improve the learning capability of ANN. Fuzzy clustering module enhances the performance of artificial neural network. In second phase, various ANN modules are trained according to their cluster values. In third phase, Fuzzy aggregation module is used to combine the results of various ANN. Here, the Hypervisor Detector is compared with Naïve Bayes and classic ANN by using the various evaluation criterions such as precision, recall value and F-value under various attacks. The performance results of FCM-ANN confirm that it outperforms the Naïve Bayes and the classic ANN algorithms even for low frequent attacks. Hence, the proposed Hypervisor Detector is suitable for detecting various attacks with high detection rate and low false alarm rate.

## References

1. Hai J, Guofu X, Deqing Z (2013) A VMM-based intrusion prevention system in cloud computing environment. J Supercomput Springer Sci+Bus Media 66(3):1133–1151
2. Oktay U, Sahingoz, OK (2013) Attack types and intrusion detection systems in cloud computing. In: Proceedings of 6th International Information Security & Cryptology Conference, p 71–76, 2013, Ankara, Turkey
3. Vieira K, Schulter A, Westphall C, Westphall C (2010) Intrusion detection techniques in grid and cloud computing environment. IEEE IT Prof Mag 2010:38–43
4. SanjayRam M, Velmurugan N, Thirukumaran S (2012) Effective analysis of cloud based intrusion detection system. Int J Comput Appl Inform Technol 1(2):16–22
5. Amjad HB, Sabyasachi P, Debasish J (2013) Machine learning approach for intrusion detection on cloud virtual machines. Int J Appl Innov Eng Manag 2(6):57–66
6. Dunlap GW, King ST, Cinar S, Basrai M, Chen PM (2002) Revirt: enabling intrusion analysis through virtual machine logging and replay. In: Proceedings of 5th symposium on operating systems design and implementation. USENIX, Boston, pp 211–224
7. Chirag NM, Dhiren RP, Avi P, Muttukrishnan R (2012) Integrating Signature Apriori based Network Intrusion Detection System (NIDS) in Cloud Computing. In: Proceedings of 2nd International Conference on Communication, Computing & Security, Procedia Technology, 6:905–912. doi:10.1016/j.protcy.2012.10.110
8. Feng Z, Hai J (2012) Automated approach to intrusion detection in VM-based dynamic execution Environment. Comput Inform 31: 271–297
9. Garfinkel T, Rosenblum M (2003) A virtual machine introspection based architecture for intrusion detection. In: Proceedings of Network and Distributed Systems Security Symposium (NDSS), p 191–206, 2003
10. Kourai K, Chiba S (2005) HyperSpector: virtual distributed monitoring environments for secure intrusion detection. In: Proceedings of 1st ACM/USENIX international conference on virtual execution environments. ACM, Chicago, pp 197–207
11. Bakshi A, Yogesh B (2010) Securing cloud from DDOS attacks using intrusion detection system in virtual machine. In: Proceedings of second International Conference on Communication Software and Networks, p 260–264. doi:10.1109/ICCSN.2010.56
12. Mazzariello C, Bifulco R, Canonoco R (2010) Integrating a network IDS into an Open source Cloud computing environment. In: Information Assurance and Security (IAS), 2010 Sixth International Conference on, pp. 265–270. IEEE, 2010, Atlanta, GA, USA
13. Jones AK, Sielken RS (2000) Computer system intrusion detection: A survey. (Online) http:// www.cs.virginia.edu/_jones/IDS-research/Documents/jones-sielken-survey-v11.pdf
14. Keiko H, David GR, Eduardo FM, Eduardo BF (2013) An analysis of security issues for cloud computing. J Internet Serv Appl 4(5):1–13

15. Modi C, Patel D, Borisaniya B, Patel H, Patel A, Rajarajan M (2013) A survey of intrusion detection techniques in Cloud. J Netw Comput Appl 36(1):42–57

16. Gens F (2008) IT Cloud Services User Survey, pt.2: Top Benefits & Challenges. (Online) http://blogs.idc.com/ie/?p=210

17. Nirmala AP, Sridaran R (2012) Cloud computing issues at design and implementation levels-A survey. Int J Adv Netw Appl 3(6): 1444–1449

18. Vikrant GD, Atul GB, Nikhil AA (2013) Intrusion detection system for cloud computing. Int J Eng Res Technol (IJERT) 2(4):2149–2153

19. Farzad S (2012) Secure virtualization for cloud environment using hypervisor-based technology. Int J Mach Learn Comput 2(1):39–45

20. Vinothina V, Sridaran R, Padmavathi G (2012) A survey on resource allocation strategies in cloud computing. Int J Adv Comput Sci Appl 3(6):97–104

21. Witcha C, Abdul HA, Mohd NMS, Siriporn C, Surat S (2007) A rough fuzzy hybrid algorithm for computer intrusion detection. Int Arab J Inform Technol 4(3):247–254

22. Gang W, Jinxing H, Jian M, Lihua H (2010) A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. Expert Syst Appl 37(9):6225–6232

23. Anderson J (1995) An introduction to neural networks. MIT Press, Cambridge

24. Haykin S (1999) Neural networks: a comprehensive foundation, 2nd ed. Prentice-Hall Inc, Englewood Cliffs, NJ

25. Mahbod T, Ebrahim B, Wei L, Ali A, Ghorbani A (2009) Detailed analysis of the KDD CUP 99 data set. In: Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009, pp.1–7, Ottawa, Canada

26. Shanmugavadivu R, Nagarajan N (2011) Network intrusion detection system using fuzzy logic. Ind J Comput Sci Eng 2(1):101–111