Cloud Intrusion Detection System Using Fuzzy Clustering and Artificial Neural Network

Avinash Appasha Chormale  
School of Computer Science,  
Yashwantrao Chavan Maharashtra Open University Nashik (MS), India.  
mr.avinash.chormale@gmail.com

Prof. (Dr.) Arjun P. Ghatule  
Controller of Examinations  
Yashwantrao Chavan Maharashtra Open University Nashik (MS), India.  
arjun1671@gmail.com

Abstract. Intrusion detection system (IDS) is a security layer used to detect suspicious activities and generate alerts when such activities are identified in systems. Artificial Neural Networks (ANN) can be used to detect the intrusion in the system but there is small problem that ANN lacks in certain areas that are detection precision for low frequent attacks and detection stability. So we have decided to implement FC-ANN approach based on fuzzy clusters and artificial neural network, to solve the problem. The general procedure of FC-ANN is as follows: initially fuzzy clustering technique is used to generate different training subsets. Subsequently, based on different training subsets, different ANN models are trained to make different base models. Finally, a meta-learner, fuzzy aggregation module, is engaged to combine these results. In addition to this we are going to add restore point which allows for the registry keys, rolling back of system files, installed programs and the project data base etc.

Keywords: Intrusion detection, Cloud computing, fuzzy clusters, artificial neural network, fuzzy aggregation, cloud instance monitor, heterogeneous log handler, audit log processor.

1. Introduction

Now a day in information technology, the sharing of resources and information in interconnected network is essential. But to secure this information from unauthorized uses and manipulation, it is necessary to impose some restrictions. There are some tools developed to do the same like Firewall, Anti-virus and Intrusion detection system [15].

Intrusion detection system is used for detection of intrusion in the system. The use of an intrusion detection system is becoming common due to the increase in complexity of attack and that of the computer systems themselves [1]. Generally intrusion detection system works in pre-defined manner in spite of implementation mechanism selected. These are some common steps followed by the intrusion detection system:

i. Information is acquired in the form of IP packets.

ii. The information are decoded and transformed into a uniform format, through the process of feature extraction.

iii. The information is then dissected in a method which is exact to the individual IDS, and classified as threatening or not.

iv. Alerts are generated if a threatening pattern is encountered [2].

The main two types of intrusion detection system: Signature-based detection, anomaly-based detection.
1.1. Signature-Based Detection:
It depends on exact matching of system or network activity. This system can only detect an attack if there exact matching activities found against existing stored patterns, known as signatures. Snort uses this type of detection. Snort is an open source IDS which implements a range of pattern matching algorithms of the input data and produces alerts based on the matching of the input to a signature base [2]. This type of system can restrict the false alarms from happening but potential threats likely to get missed as new techniques of attacks keeps on evolving. As well as the expenditure of maintenance is too much as the signature set needs to keep on upgrading.

1.2. Anomaly-Based Detection:
Anomaly-based detection is the impressively recommended technique in today’s situation where latest attacks are being recognized every day. This technique develops itself by understanding and gathering the information about the system and determines the behavior of the system based on it [3]. There are three types of algorithms basically used to develop the system and that are fuzzy logic, genetic algorithm, artificial neural networks. New attacks can be detected in this type of detection system unlike the signature based detection system.

In this paper we propose to increase anomaly based detection system having both the fuzzy logic and artificial neural networks. Among these techniques, ANN is one of the broadly used techniques and has been successful in solving many complex practical problems. And ANN has been successfully applied into IDS. However, the most important disadvantage of ANN-based IDS have in two aspects: (1) lower detection precision and (2) weaker detection stability. To solve these two problems, we propose a novel approach for ANN-based IDS, FC-ANN, to enhance the detection precision for low-frequent attacks and detection stability.

The fuzzy clustering is being used for separation of the datasets into dissimilar clusters and fuzzy aggregation is used to aggregate different ANN’s results. ANN used for to find out the pattern of every subset and decide firm actions on it.

2. Our proposal
In this study, we are using the similar Fuzzy Clusters – Artificial Neural Network (ANN) approach which is based on FC-ANN with the addition of system restores point. System Restore is a module that allows for the rolling back of registry keys, system files, installed programs and the project data base etc. which is stored in the cloud server, to a prior state in the result of system failure or malfunction of the system or if any Intrusion is found in the system.

By fuzzy cluster method, the heterogeneous training set is divided to several homogenous subsets. Thus difficulty of each sub training set is reduced and therefore the detection performance is improved. The experimental outcome using the KDD CUP 1999 dataset shows the efficiency of our new approach particularly for low-frequent attacks, i.e., R2L (remote to local) and U2R (user to root)attacks in terms of finding accuracy and finding stability. Data mining techniques, such as support vector machine, outlier detection, evolutionary computing; genetic algorithm may be introduced into Intrusion Detection System. Assessment of different data mining methods will provide clues for constructing more effective hybrid Artificial Neural Network for finding intrusions.

3. Structure of IDS based on Fuzzy Clustering and Artificial Neural Network
Fuzzy Clusters and Artificial Neural Network initially divides the training data into various subsets using fuzzy clustering method, after it trains the different Artificial Neural Network using different subsets. Then it determines membership grades of these subsets and combines them via a new ANN to get finishing outcome. The complete structure of FC-ANN is demonstrated in figure 1.
Figure 1: Structure Intrusion Detection System using FC-ANN and Restore Point Facility Model

3.1. Fuzzy cluster module

The purpose of fuzzy clustering module is to separate a given set of information into clusters, and it should have the following properties: related information in same cluster, homogeneity within the clusters and heterogeneity between clusters, where information belonging to dissimilar clusters should be as different as possible. Through fuzzy clustering module, the training set is clustered into several subsets. Owing to the details that the size and difficulty of every training subset is reduced, the proficiency and efficiency of subsequent Artificial Neural Network module can be enhanced. There are two types of clustering techniques soft clustering techniques and hard clustering techniques. At the side of Partition of training set, we also require to aggregate the outcome for fuzzy aggregation module. Therefore, we opt the soft clustering method, for fuzzy clustering module.

3.2. Artificial Neural Network module

Artificial Neural Network modules purpose to be trained the pattern of each subset. Artificial neural network is a biologically motivated form of distributed computation. It is collected of simple processing units, and links between them. In this research work, we will make use of classic feed-
forward neural networks qualified with the back-propagation algorithm to forecast intrusion. A feed-forward neural network has an input layer, an output layer, with one or more concealed layers in between the input and output layer.

3.3. Restore Point for system backup:

System Restore is a part that allows for the rolling back of system files, registry keys, installed programs and the project data base etc. [1] which is stored in the cloud server, to a previous state in the event of system malfunction or failure of the system or if any Intrusion is detected on the system.

In System Restore boundary is based on Shadow Copy technology. In prior period it was based on a file filter that observes the changes for a given some set of file extensions, and then copied files before they were overwritten. Shadow Copy has the advantage that block-level changes in files located in any directory on the volume can be watched and backed up in spite of of their location.

In System Restore, the user may generate a new restore point manually, roll back to available restore point, or modify the System Restore configuration. Furthermore, the restore itself can be undone. For the Intrusion Detection System this can afford restore points casing the last some weeks.

System Restore backs up system files on the server on which the generally Intrusion Detection System training database is saved and saves them for future recovery and use it again when the system is restored and apply the Artificial neural network for the similar type of attack is found again on the system. In addition it stores data backup about the drivers and registry which are installed on the cloud server.

3.4. Backup strategy used for the system

Backup plan starts with a concept of an information repository. The backup information needs to be stored in some way to be structured to a scale. It can be as straightforward as a page of paper with a list of all backup tapes and the dates they were written or a more refined setup with a computerized catalogue, index, or relational database. Various warehouse models have different advantages. Some of the types of repositories are described as follows.

3.4.1. System imaging

A warehouse of this type contains entire system images from one or more specific points in time. This technology is often used by computer technicians to record known good configurations. Imaging is usually more useful for deploying a standard configuration to many systems rather than as a tool for making continuing backups of diverse systems.

3.4.2. Unstructured

An unstructured warehouse may just a pile of floppy disks or CD-R/DVD-R media with minimum data about what was backed up and when. This is the easiest to implement, but maybe the smallest amount likely to achieve a high level of recoverability.

3.4.3. Differential

A differential technique warehouse saves the information since the last full backup. It has the benefit that only utmost of two data sets are required to restore the data. One drawback, at least as compared to the incremental backup method, is that as time from the last full backup increase so does the time to
perform the differential backup. To act upon a differential backup, it is first essential to execute a full backup. After that, each differential backup made will contain all the changes since the last full backup. Restoring an entire system to a certain point in time would require locating the last full backup taken previous to the point of the failure or loss plus the last differential backup since the last full backup.

3.4.4. Incremental

An incremental style repository plans to make it more feasible to accumulate backups from multiple points in time by classifying the information into increments of change between points in time. This eliminates the need to store duplicate copies of unaffected data, as would be the case with a portion of the data of consequent full backups. Typically, a full backup (of all files) is made which serves as the reference point for and incremental backup set. After that, any number of incremental backups is made. Restoring the entire system to a certain point in time would involve locating the final full backup taken previous to the data thrashing plus each of the incremental backups that cover the phase of time among the full backup and the point in time to which the system is supposed to be restored. Furthermore, a few backup systems can rearrange the repository to produce full backups from a series of incremental.

4. Essentials of this system architecture

The architecture of CloudIDS is been considered as a ‘Generic Cloud Security Framework’ due to the following reasons. CloudIDS is designed such that it uses two expert engines – uX-Engine and sX-Engine[5]. uX-Engine makes use of an unsupervised machine learning technique, whereas sX-Engine makes use of a supervised machine learning technique. Due to the advancements in Artificial Intelligence field, a wide range of machine learning algorithms are available and there is no single best algorithm for a specific problem. CloudIDS gives the flexibility to choose the algorithm depend upon the need and business requirement of an organization(s). This choice will decide the scope, accuracy and efficiency of CloudIDS as an end product.

Further, the main motive for designing the ‘CloudIDS Generic Cloud Security Framework’ is to ensure the security of cloud virtual machine (VM), which is resourced to customers for deployment of their applications and data. CSP puts emphasis on the underlying cloud infrastructure to provide effective services to customers through VMs. The security of infrastructure as well as VM’s ensures the smooth and hassle free operation between the customer and the CSP. The customer companies (Cloud Users or CUs), however, prefer the private or hybrid cloud for less probability of attacks and hence, highly secure configuration[6]. This is because of a belief that in-premise computing is safer than cloud. Pay-as-you-use model provides the foundation for the cloud based services after the deployment of the user VM(s) on CSPs cloud infrastructure. CSP segregates the infrastructure logically with the help of virtualization through hypervisors. Hypervisors are software tools which play a significant role in the deployment and maintenance of virtual machines based on the requirements of the customer. Customers’ deploy their data and applications on the VMs allocated to them. The client or end users can access the VM/Instance with authenticated credentials over Internet. This availability over the Internet fascinates and provokes the attackers and hackers to try to attack the VM. To sum up this discussion, the security of VM has become the need of the hour. Hence, CloudIDS provides a solution
to meet this requirement of the present scenario. Furthermore, based on the business requirements of the customer this framework extends the support for multiple VM/Instances.

5. CloudIDS – System Architecture

CloudIDS is a state-of-the-art generic security framework with the next generation hybrid two-tier expert engine based IDS for cloud computing environment [16]. Figure 2 shows the process workflow of CloudIDS.

This system architecture is designed for a particular VM/Instance (Virtual Machine) on cloud IaaS infrastructure. The goal of CloudIDS is that the virtual machine on which the customer’s applications and information are deployed is safe and sound. This system is self-possessed of a number of modules/subsystems. These modules/subsystems as can be seen in figure 2. Each module have dependency with the other and work synchronously. The detailed functionality of each of the subsystem is explained further.

![CloudIDS System Architecture](image)

**Figure 2: CloudIDS System Architecture**

5.1. CloudIDS C3 Subsystem

CloudIDS C3 stands for Configuration & Customization Console (C3) subsystem. C3 is used to configure and customize the remaining multiple subsystems of CloudIDS architecture with respect to the user’s requirements. C3 allows users to configure the file system paths and databases that need to be monitored by CloudIDS. For instance, a user might want a particular database related to transactions need to be monitored; then he can specify it via C3. This selective monitoring gives the
users an added flexibility to decide which data needs greater security. Using C3, users can customize
the two-tier subsystems uX-Engine and sX-Engine with suitable algorithms and its associated values,
etc. The user can decide which type of machine learning algorithm as well as the hyper parameters
should be used for intrusion detection. This will also be used to configure the allowed and non-allowed
transactions on their VMs, and the action that needs to be taken by the system when any kind of
tampering of data is found. A warning system or an action based system can then be formulated based
on the policies agreed upon by the users. In general, CloudIDS C3 includes a huge amount of
configuration and customization properties and activities in all subsystems of CloudIDS Framework.

5.2. CIM Subsystem
Cloud VM/Instance Monitor (CIM) [1] subsystem is responsible for observing the actions performed
on the user specified virtual machines or instances. It continuously monitors the VM on the user
specified application-related directories and files, and database tables as configured in C3. CIM is
responsible for observing all the behavior of both authorized and unauthorized users and update it to
H-log-H for immediate recording of each and every activity for the configured VMs. This includes the
recording of both normal and anomalous behaviors. CIM might result in heterogeneous audit logs
generation based on the nature of item accessed in selected VMs by any subject/user, irrespective of
whether it is authorized or unauthorized.

5.3. H-log-H Subsystem
Heterogeneous Log Handler (H-log-H) subsystem meticulously associates with CIM for recording
each and every activity occurred on user configured VMs. Generally, files and directory related logs
are managed and maintained by the concerned Operating Systems in their own formats (for example,
UbuntuOS and MacOS audit logs are not in the same formats). Similarly, tables and database related
logs are managed and maintained by different DBMS utilities with differences in formats (for
example, MySQL’s and Firebird’s audit logs are not in same format). Likewise, depending on the type
of and number of applications, tools, databases systems, etc. configured on VM, it produces multiple
type of audit logs to H-log-H. After recording these heterogeneous assorted audit logs will be
surrendered to Audit Log Preprocessor subsystem by H-log-H subsystem, for further process.

5.4. Audit Log Preprocessor (ALP) Subsystem
Audit Log Preprocessor (ALP) subsystem accepts the input from H-log-H with integrated
heterogeneous audit logs from multiple types of application, tools, etc. Upon the reception of input,
ALP rearranges, processes, and converts unstructured audit logs into structured data. During this
conversion, ALP is expected to utilize numerous techniques and mechanisms to handle these
dissimilar audit logs. Separation of these heterogeneous audit logs to the corresponding log handling
mechanism will help ALP in better management of huge amount of logs. ALP’s processed output (i.e.
uniformly formatted structured audit logs) will be given as input to both uX-Engine and sX-Engine.
Hence, ALP in CloudIDS becomes a mainstay subsystem which demands utmost accuracy in
conversion with extreme computation complexity in implementation.

6. CloudIDS uX-engine tier-1
CloudIDS Tier-1 accommodates its functionality with two subsystems and its associated audit
repositories. The tier as a whole verifies the normalcy of the audits being produced by the user
configured VMs.
6.1. uX-Engine

uX-Engine is based on intelligent learning systems and uses machine learning algorithms. It analyses all the activities that are carried out on the VM. This works in two stages – learning phase and testing phase (deployment phase). When the algorithm is being used for learning, all the numerically converted audit logs are fed to it through ALP, generally these logs come after a dry run of few days in VM. This engine clusters these logs based on some type of similarity in behavior or relative feature mapping. However later, the system can be configured to work on any type of unsupervised algorithm (like K-Mean, SOM, GHSOM, etc.). After the learning is finished, the system learns standard activities with respect to CIM configured VMs. The logs which were used for the learning are stored in the Standard Audit Repository (SAR). Now whenever a VM is deployed, the CIM will monitor it, and all the information about the activities carried out on that VM will be sent to this tier via ALP after being preprocessed. Now, whenever a new audit comes, it will be compared against the clusters which were formed during learning. If the new audit couldn’t be resolved to any of the cluster, then it is sent further to CloudIDM for verification. Now there are two possible cases, either it can become intrusion attack or a normal activity that the uX-Engine has not learnt yet. If the audit is found to be normal, it is stored in SAR (collection of normal behavior audit logs).

6.2. CloudIDM

Identity Management (IDM) – a means of achieving utmost security and self-reliance in deployed applications at any global enterprise. IDM has shown the new way of building trust and confidence across various applications deployed on different systems and platforms, while providing ultimate innumerable identity management capabilities such as provisioning/de-provisioning, enterprise SSO, authentication and authorization, delegated administration, etc. [11][12].

In case of cloud environment, each and every cloud deployments are dynamic in nature. Such as,

- Servers being launched or terminated
- IP addresses dynamically assigned & reassigned
- Services started or decommissioned or re-started, etc.

This dynamic landscape of cloud, without option, needs to have appropriate extended IDM competencies as built-in to serve the users need. In addition, the inclusion of proper IDM functionalities helps cloud to deliver its promised properties like pay-as-you-use, metered services, elasticity, etc.

On the other hand, it is also the fact that today’s (traditional) IDM tools and platforms are capable of protecting applications/projects/users up to application level. Considering all the above mentioned facts, cloud is really in a need of a new paradigm ‘Cloud IDM’, which is capable of handling strong IDM capabilities at the following 3 levels.

1. Cloud Application Level
2. Cloud VM Level
3. Cloud Infrastructure Level

CloudIDM (Intelligent Cloud IDM) [13][14] is targeted to be a complete identity management framework, which provides all the above mentioned functionalities to any cloud environment through its 3-layer architecture.
In CloudIDS, after uX-Engine fails to classify a log into any of the existing clusters, CloudIDM comes into play. CloudIDM is used as a decision making subsystem in CloudIDS, with its selected functionalities; not all. CloudIDM is a tightly coupled association with Cloud IdP (Identity and Access Management Provider) of the specific VMs. During the course of the CloudIDS process workflow, CloudIDM takes appropriate decisions at important situations. After uX-Engine forwards the outlying, dissimilar audits to CloudIDM subsystem, it analyzes the log and classifies it into one of the following categories:

**Category #1** – Normal Behavior: When a new log settles with existing clusters.

**Category #2** – Special Permission Behavior: When a new log doesn’t fit with any of the existing clusters and that log’s behavioral analysis turn TRUE with the verification of Cloud IdP as exceptions/special permissions/temporary provisioning by admin, or due to the dynamic change in user configurations (for example, if user admin change some configurations at VM but don’t update the CloudIDS C3), etc.

**Category #3** – Anomalous Behavior: When a new log doesn’t fit with any of the existing clusters and that log’s behavioral analysis turn FALSE with Cloud IdP in all cases.

As a result, if the monitored audit log(s) are found under the category #1 i.e. Normal Behavior, then they will be safely sent to the Standard Audit Repository SAR. Those found under the category #2 i.e. Special Permission Behavior, are moved into Special Permission Audit Repository (SPAR). Finally, those found under the category #3, i.e. Anomalous Behavior, are directly sent to the CloudIDS architectures tier-2 subsystem sX-Engine.

For instance there is an admin who delegates the authority to login to some of his/her employee for three days. So this will be reflected in IdP. Normally if the employee tries to login using admin’s credentials, it is considered as an attack. But now this employee has special permissions, and CloudIDM will classify such a log into category #2. After the stipulated time of three days has finished, the same log will be classified as an attack. And it will be removed from SPAR.

6.3. **Standard Audit Repository (SAR)**

This audit repository is tightly-coupled with uX-Engine & CloudIDM subsystems. Standard Audit Repository contains the category #1 type logs i.e. Normal Behavioral Activity Logs that have been analyzed and verified using CloudIDM. SAR stores only those audit logs after the clearance acceptance from CloudIDM. Logs contained in SAR will be fed to uX-Engine for further learning, this ensures that the uX-Engine is up to date with activity identification on the VM. It will also help in making the uX-Engine, a dynamic learning machine.

6.4. **Special Permissions Audit Repository (SPAR)**

It is also an audit repository, which will be tightly-coupled with CloudIDM subsystem of CloudIDS & Cloud IdP services. Special Permission Audit Repository contains the category #2 type logs i.e. Special Permission Behavior Event Logs that are not normal behavior but have passed as acceptable through the CloudIDM after its verification with Cloud IdP system services. These activities can be of any sort of special permission on a specific subject/user/object/identity in an organization due to the dynamic nature of business requirements. These kinds of exceptional permissions can be given at almost all the SDLC phases of both in project and in business workflow. SPAR will not be used in
either uX-Engine or sX-Engine; but simply present in the system as repository for auditing and analysis purpose.

7. CloudIDS sX-Engine Tier-2

CloudIDS Tier-2 comprises of supervised machine learning-based engine along with the acute audit repository. This tier handles the outlying event logs, which are passed to it by the CloudIDM subsystem of CloudIDS Tier-1, and classifies it to a threat pattern. Further, this tier concludes its functionality by incorporating Warning Level Generator and Alert System in their respective processes.

7.1. sX-Engine

It is a supervised machine learning-based expert engine subsystem that focuses on input provided to it by CloudIDS Tier-1 CloudIDM subsystem’s category #3 Anomalous Behavior type of outlying audit logs. It classifies these logs into appropriate threat pattern classification. Supervised learning is often used for pattern recognition and classification tasks. And that’s what the sX-Engine aims to do here. Training and testing procedures needed for it to be able to successfully classify pattern into appropriate category. For its training purpose, the corpus contains all feasible and finite collection of host/network-based attacks, virus, worms, vulnerabilities, threats signatures. However creating such a training corpus requires a massive human effort. After such a corpus is built, the organization can choose any supervised machine learning technique based on their intuition and requirements. And this appropriate selection of suitable algorithm decides the quality and accuracy of sX-Engine’s overall execution. Further, sX-Engine will be trained using the selected algorithm and it will take as input the generous corpus which contains all possible anomalous activities, henceforth referred as Acute Audit Repository (AAR). After the system is finished training, it develops a model which it uses to classify the new attacks into the categories of normal or anomalous behavior. In testing phase, the trained sX-Engine accepts the category #3 Anomalous Behavior types of outlying audit logs as input from CloudIDM and checks the patterns for classification. Classification will be performed based on the features extracted from the input audit logs and the same will be compared with the AAR corpus logs features. If the engine classifies it into an attack, then appropriate warning is generated through Warning Level Generator. But all those events that could not be matched to a class of threats in AAR will undergo manual inspection by the CloudIDS admin. The admin would examine and verify the activity recorded in the log so as to take the decision and feedback to the system. If the intrusion is found to be vulnerable by CloudIDS admin then a new class would be assigned by the CloudIDS admin via the Warning Level Generator and the same would be subsequently used to train the sX-Engine dynamically. In addition, sX-Engine stores back the input anomalous audit log into AAR with generated warning level information as label.

7.2. Acute Audit Repository (AAR)

It is an audit repository that is tightly-coupled with sX-Engine subsystem. This is basically a huge collection of all possible attacks, virus, worms, vulnerabilities and threats signatures. This helps sX-Engine during its training and testing phases. In addition, it stores back the input audit log after the testing along with warning level information label.
8. Warning Level Generator (WLG) Subsystem

Warning Level Generator (WLG) subsystem takes the feedback given by the sX-Engine and works on it further to classify them under priority warning levels based on the user configuration of CloudIDS C3. Events that were mapped as an intrusion after classification by sX-Engine will be assigned to a class of threat. Based on the class, the warning level generator will make a decision for the type of alert that needs to be given. This process helps in making the system to not respond to intrusion/attack/anomalous behavior in the same way. WLG involves in segregation of different anomalous behavior under various priority warning levels, for example high-medium-low, or red-blue-green, etc. In other way WLG helps both CU (Cloud User) and CSP in achieving an appropriate SLA process workflow automation.

9. Alert system

Alert System (AS) is a subsystem in CloudIDS which takes care of notifying the concerned stakeholders at the customer side and at the CSP organization. Once a warning level has been generated, the list of concerned stakeholders for the particular level of warning will be notified using this module. For example, if the warning level for the intrusion is found to be low, then only the CSP and CU admins will be notified of the intrusion. For medium and high priority levels, the CU as well as CSP stakeholders will be notified of the intrusion. Based on the AS notification, CSP needs to provide appropriate solution for the intrusion and explanation to CU. This will help maintain transparency between the CU and the CSP at all times. Alert functionalities can be different for different situation. If a new type of attack comes, the application gives added functionality to inform admin via pop ups depending upon the severity level. If the attack is of high severity, for example 5 (assuming 5 is high) then the alert will go to admin, CSP as well as the user. The alert may go in the form of SMS, email or a call with prerecorded message. The screen can also be blocked preventing the intruder from undergoing further harm to VM. On the similar lines, it can deal with attacks of low severity, informing the concerned personnel and taking appropriate actions.

10. Conclusion

We propose an intrusion recognition mechanism, called FC-ANN, based on ANN and fuzzy clustering. In fuzzy clustering method, the heterogeneous training set is divided into number of homogenous subsets. Therefore difficulty of each sub training set is reduced and subsequently the detection performance is improved. And in this way system turns out to be more efficient and stable and we efficiently overcome the drawbacks – weaker detection stability, lower detection precision. And beside with this we can effectively take backup of system using restore point facility.

CloudIDS Generic Cloud Security Framework is responsible for various security related activities with respect to current cloud computing environment. CloudIDS performs the following roles to handle the security of VM/Instance.

CloudIDS Role #1: VM Monitor

CloudIDS Role #2: System Auditor

CloudIDS Role #3: Intrusion Detection System

CloudIDS Role #4: Behavioral Analyzer
CloudIDS Role #5: Alert Agent

Various cloud computing security issues resolved by CloudIDS architecture.

11. References
[1] Gang Wang, Jinxing Hao, Jian Ma, Lihua, “A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering”, ELSEVER, Expert System with applications 37 (2010) 6225–6232.

[2] Julie Greensmith and Uwe Aickelin “Firewalls, Intrusion Detection Systems and Anti-Virus Scannersl ASAP Group, University Of Nottingham, UK, 1-17.

[3] Zhenwei Yu, Jeffrey J. P. Tsai, Fellow, IEEE, and Thomas Weigert –An Automatically Tuning Intrusion Detection System IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 37, NO. 2, 373

[4] www.darpa.mil/

[5] Madhan Kumar Srinivasan, K. Sarukesi, Revathy P, “eCloudIDS Tier-1 iCloudIDM Layer-I (iCloudIDM-LI) Subsystem Design and Implementation through User-centric Identity Management Approach for Secure Cloud Computing Environment”, IEEE 14th International Conference on Mobile Data Management, 978-0-7695-4973-6/13 2013 IEEE, DOI 10.1109/MDM.2013.95, 206-211.

[6] Avinash Appasha Chormale, Prof. (Dr.) Arjun Ghatule, “A Review on Intrusion Detection Techniques for Cloud Computing and Security Challenges”, EECCMC, Technical Sponsor by IEEE, CFP18037-PRJ, ISBN:978-1-5386-4303-7,2018.

[7] Mrs.Anshu Gangwar, Mr. Sandeep Sahu, “A survey on anomaly and signature based intrusion detection system (IDS)” Int. Journal of Engineering Research and Applications ISSN : 2248-9622, Vol. 4, Issue 4( Version 1), April 2014, pp.67-72.

[8] Kusum Kumari Bharti, Sanyam Shukla, Sweta Jain, “Intrusion detection using clustering”, IJCCT Vol.1 Issue 2, 3, 4; 2010 for International Conference [ACCTA-2010], 3-5 August 2010, 158-165.

[9] Sutapat Thiprungsri, “Cluster Analysis for Anomaly Detection in Accounting Data, Nineteenth Annual Strategic and Emerging Technologies Research Workshop San Francisco, California, USA. July 31, 2010, 1-8.

[10] R. Shanmugavadivu, Dr.N.Nagarajan, “NETWORK INTRUSION DETECTION SYSTEM USING FUZZY LOGIC”, IJCSE, ISSN : 0976-5166, Vol. 2 No. 1, 101-111.

[11] Jer Min Jou, Shiann Rong Kuang, and Ren Der Chen, “A New Efficient Fuzzy Algorithm for Color Correction”, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: FUNDAMENTAL THEORY AND APPLICATIONS, VOL. 46, NO. 6, 773-775.

[12] Muna Mhammad T. Jawhar, Monica Mehrotra, “Design Network Intrusion Detection System using hybrid Fuzzy-Neural Network”, International Journal of Computer Science and Security, Volume (4): Issue (3), 285-294.

[13] Madhan Kumar Srinivasan, K. Sarukesi, Revathy P, “eCloudIDS Tier-1 iCloudIDM Layer-I (iCloudIDM-LI) Subsystem Design and Implementation through User-centric Identity Management Approach for Secure Cloud Computing Environment”, 14th International Conference on Mobile Data Management, 978-0-7695-4973-6/13 $26.00 © 2013 IEEE, DOI 10.1109/MDM.2013.95, 206-211.
[14] Madhan Kumar Srinivasan, K. Sarukesi, Revathy P, “Architectural Design for iCloudIDM Layer II (iCloudIDM-LII) Subsystem of eCloudIDS Generic Security Framework”, International Conference on Advances in Computing, Communications and Informatics (ICACCI), 978-1-4673-6217-7/13/$31.00 © 2013 IEEE 2013, 1668-1674.

[15] Prof. D.P. Gaikwad, Sonali Jagtap, Kunal Thakare, Vaishali Budhawant, “Anomaly Based Intrusion Detection System Using Artificial Neural Network and Fuzzy Clustering”, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 1 Issue 9, November-2012, 1-6.

[16] Madhan Kumar Srinivasan, K Sarukesi, Ashima Keshava, and P. Revathy, “eCloudIDS – Design Roadmap for the Architecture of Next-Generation Hybrid Two-Tier Expert Engine-Based IDS for Cloud Computing Environment”, S.M. Thampi et al. (Eds.): SNDS 2012, CCIS 335, pp. 358–371, 2012. © Springer-Verlag Berlin Heidelberg 2012, 358-359.

Avinash Appasha Chormale he has completed his M.Tech. (CSE) form JNTU, Hyderabad, India. He is assistant lecturer in Greenfingers College of Computer and Technology, affiliated to Solapur University Solapur. At present perusing Ph.D in Computer Science, Yashwantrao Chavan Maharashtra Open University,Nashik. His research topic is ‘Development of Cloud based IDS: A security enhancement framework using FC-ANN’.

Prof. (Dr)Arjun P. Ghatule he has completed his Ph.D., MCA, MBA(HR), M.Sc.(Elex.), DCS,DBM, MDP(IIM, Ahmedabad). He is approved faculty for MCM & MCA of Shivaji University and Bharati Vidyapeeth Deemed Universit. He is approved Director for MCA of Solapur University, Solapur. At present he is COE of Yashwantrao Chavan Open University, Nashik, India.

He has served tenure post of Director of Exam in University of Mumbai Appointed by Higher and Technical Education, Government of Maharashtra on recommendations of Hon’ble Chancellor/Governor of Maharashtra.

Dr. Ghatule has a member of various professional bodies such as ISTE, IACSIT, IEDRC. SCIEI,UACEE, IAENG, CBES, ISOC Etc. He is Editor-in-Chief for International Journal of Research in Computer Science and Information Technology.

Under his guidance 12 M.Phil/PhD students perusing their M.Phil/PhD, also 5 students completed their M.Phil/PhD.