International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Optimizing trust, Cloud Environments Fuzzy Neural Network, Intrusion Detection System

[*1]Dr. Archana B., [2]Mrs. N. Jeebaratnam, [3]Dr. B. Nageswara Rao, [4]Dr. U. Sesadri, [5]Dr. N. Shirisha, [6]Dr. Nellore Manoj Kumar

Submitted: 21/12/2023 **Revised**: 27/01/2024 **Accepted**: 09/02/2024

Abstract: In the dynamic landscape of cloud computing, ensuring the security and integrity of services is paramount. This article introduces a novel approach to cloud intrusion detection by leveraging the synergies of fuzzy logic and neural networks. The proposed Fuzzy Neural Network Aided Cloud Intrusion Detection System (FNN-CIDS) integrates the adaptability of fuzzy systems with the learning capabilities of neural networks to enhance the detection accuracy of malicious activities within cloud environments. The system is designed to discern subtle patterns indicative of intrusion attempts, thereby fortifying the defense mechanisms for trusted services hosted in the cloud. The article presents the conceptual framework of FNN-CIDS, detailing the integration of fuzzy logic for rule-based inference and neural networks for pattern recognition. Experimental results demonstrate the system's efficacy in identifying diverse intrusion scenarios while minimizing false positives. This research provides a promising path for improving the reliability of cloud computing infrastructures and advances strong security frameworks for cloud-based applications. In this sense, the research effort provides a trust evaluation system to determine the reliability of cloud services and an intrusion detection system to guarantee intrusion-free cloud services. the construction of a cloud intrusion detection system using a neuro-fuzzy based self-constructing clustering algorithm. The performance of this method has been compared to other well-known clustering methods in the field of cloud intrusion detection using result analysis.

Keywords: *Cloud Computing, Intrusion Detection System, Neuro-Fuzzy, K-Means, Denial of Services, Cloud Security*

1. Introduction

Cloud computing offers an environment that facilitates conducive and ubiquitous access to a pool of integrated resources in an on-demand manner. It allows companies, organizations, and individuals to catch services by keeping themselves away from infrastructure costs and maintenance requirements. This convenience helps them in concentrating on their business progress without worrying about the manageability of infrastructure.

Cloud storage solutions offer several means to users for storing and processing their data. But these solutions provide a way to security hazards. These hazards may be classified into two dimensions: Security issues faced by cloud service providers in providing services to cloud consumers and security issues faced by cloud consumers in availing services from cloud service providers. Since consumers leave their valuable data with the providers, they do not have physical access to the hosting server. To satisfy them in security aspect, service providers employ several mechanisms for ensuring the provision of secured services. But still, security issues seem to exist.

An efficient security monitoring system should be clever in identifying security vulnerabilities and take appropriate measures in handling them by adopting any of the following modes [1]:

Deterrent or Preventive Mode

By using strong authentication mechanisms, only authorized users are permitted to access resources. Unauthorized users are prevented from access and administrators are informed an alarm of adverse reactions.

Detective Mode

By monitoring user behavior on systems and networks, the security system will differentiate normal and abnormal accesses. Based on this information, it will

[1]Associate Professor,Department of CSE,Vidya Vikas Institute of Engineering and Technology, Mysuru, Karnataka, India, Pincode: 570028. Email id: archanab.research@gmail.com
[2]Assistant Professor, Department of ECE, Centurion University of Technology and Management,R. Sitapur, Paralakhemundi, Odisha, Pincode: 761200. Email: jeevaratnam@cutm.ac
[3]Faculty of Mathematics,School of Technology, Apollo University,Murukambathu, Chittoor (Dist),Andhra Pradesh, India, Pincode: 517127. Email: bendi151977@gmail.com
[4]Associate Professor, Department of CSE, Vardhaman College of Engineering, Shamshabad, Hyderabad, Telangana, India, Pincode: 501218. Email : sesadri1601@vardhaman.org
[5]Associate Professor, Department of Computer Science and Engineering, MLR Institute of Technology, Hyderabad, India, Pincode:500043. Email: nallashirisha@mlrinstitutions.ac.in
[6]Department of Mathematics, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (SIMATS), Thandalam, Chennai, Tamilnadu, India, Pincode: 602 105. Email id: nelloremk@gmail.com

detect malicious behaviors and inform to the administrator.

Corrective Mode

In addition to the detection of malicious behaviors, this system involves in some restoration activities for reconstructing a compromised system.

Intrusions and Intrusion Detection Systems

Any action which tries to illegally access cloud resources is termed as intrusion. Several intrusion detection techniques exist to identify such illegal actions. An intrusion detection system (IDS) can trigger alarms based on the most common triggering mechanisms named misuse detection and anomaly detection. Misuse detection-based IDS is a system depending on set of rules which may be built either by the system or by the administrator manually. This system monitors packets on network and system operations and the rules search for known signatures to identify an attack. The advantage of this method is that signatures can easily be developed and recognized when the network behavior is known and it can be done rapidly with the contemporary systems. This system requires signature for each attack and detects only known attacks based on fixed behavioral pattern. To achieve maximum true positive rate, constant revision of rule set is essential. So, new signatures have to be added to the IDS as and when required. This will increase the size of rule set, in turn, increase resource consumption.

Anomaly detection-based IDS is a system that can detect intrusions by observing system activities and classifying them as either normal or anomalous, based on predefined metrics or rules. Any deviating activity that falls out of normal category is identified as an attack. In order to resolve attack traffic, the system must be educated to be familiar with normal system activity. This can be accomplished in numerous ways, most often with soft computing techniques. In strict anomaly detection, a mathematical model is used to define the normal usage of the system. Such a system needs a profile of the network or system to build a profile for reflecting a normal usage. This method needs the detailed knowledge of normal network behavior for correct detection. Once the behavior is defined, the IDS can be scaled up easily. This system is better than the misuse detection-based IDS in the sense that it can detect new attacks without requiring signatures for them, provided they fall out of normal usage patterns. However, it suffers from the fact that malicious activity that falls within normal usage patterns may not be detected, resulting in false errors.

In addition to implementing a triggering mechanism, monitoring of intrusive activity may occur at specific points within the network. The two common monitoring locations are host-based and network-based intrusion detection systems. Host-based intrusion detection system detects intrusive behaviors of users at either host level or operating system level. It identifies intrusions by analyzing system calls, system and application log files, suspicious file-system modifications, network attacks upon the specific system, known signature attacks, port scans and backdoor checking. This method is benefited from the handy determination of either success or failure of an attack. But it suffers from the difficulty of establishing an accurate picture of a network or regulating the events happening across the entire network. And also it has to run on every monitored host by supporting various operating systems.

Network-based intrusion detection system attempts to spot unauthorized access to a computer network by studying network traffic for signs of malicious activity by either comparing the packet to a set of known- attacks signatures or looking for anomalous packet activity that might designate malicious behavior. This system does not have to support every type of operating system used on a network. However, when networks become larger, a network-based IDS at a single place on a network may not capture all of the traffic successfully. This, in turn, requires large number of sensors in the network, increasing the cost of IDS.

Hybrid intrusion detection system can be developed by combining various intrusion detection techniques discussed previously. This system may theoretically be benefited from multiple approaches, while winning many of their disadvantages. But, since different IDS technologies work in different ways, incorporating them into a single system is a very complicated task.

IDS is categorized into centralized, hierarchical and distributed models based on the architecture. In centralized system, data collected from single or multiple hosts are shipped to a central location for analysis and that central unit node is responsible for detecting malicious activities. The drawback is that, any failure at the central unit leads to deactivating intrusion detection. In addition, it should quickly handle large amount of data received from multiple hosts. The network is divided into number of clusters in hierarchical architecture, with cluster heads identifying intrusions. Cluster-heads in the lowest level send alerts to heads in the higher level which send alerts to next higher-level heads based on alerts from both their own level and lower level.

This approach is more extensible than the centralized approach, but still the central unit is a bottleneck. Each host runs an intrusion detection system in a distributed architecture without any central coordinator, allowing more scalability. But the information of all alerts may

not be available at the time of decision making, causing less accuracy.

Motivation

In spite of all efficiencies and conveniences offered by cloud computing, it faces many barriers so that end users hesitate to have faith on it. Some of its hurdles are issues with cloud forensics, loss of physical control, transitive accountability, privacy laws, contingency arrangements, disaster management, multi-tenancy, security of hypervisors and operating systems, data ownership and retention decisions, trust and privacy problems. All these issues revolve around a single common determinant termed 'cloud security' and it is of two types:

1. Physical security - Protection against natural disasters and hazards.
2. Internet security - Protection against man-made attacks that may occur through internetworking.

Due to the openness of clouds, there is always a chance of intrusions. Intrusion is an activity where unauthorized elements try to gain access to cloud resources. It keeps cloud security under a question at all times. Consequently, several researches are being done in the field of cloud security to achieve intrusion-free accesses. But still, realization of a cloud intrusion detection system (CIDS) with satisfactory results in terms of detection accuracy, speed, resource consumption and scalability is not completed. Due to the dynamic behavior of both consumers and service providers, cloud security has become one of the major areas of research. By an enormous study of existing intrusion detection methods, it is found that, though each of the existing solutions achieves success with some accuracy, they typically lack in certain other essential parameters. Hence, it is planned to devise an efficient cloud intrusion detection system for achieving cloud security.

Usually, cloud service consumers (CSCs) need to assess the trustability of cloud service providers (CSPs). This is due to the reason that service providers themselves may be malicious in nature. This suspicion arises as a result of possible occurrences of insider attacks. Similarly, cloud service providers wish to know the trustability of service consumers. Knowing this information will help service providers to deny subscription to the doubtful consumers so that their intrusion detection methodology gets simplified. Hence, there should exist mutual trust between service consumers and service providers so as to ensure smooth delivery and consumption of cloud services.

2. Related Works

Though cloud computing attracts individuals and organizations toward its flexibility and convenience, it has several obstacles which prohibit certain users in availing some cloud services. Among all the obstacles, security is the most dominating issue. Consumers usually have the habit of storing personally identifiable information, sensitive information, sensitive personally identifiable information, behavior information and device identification information in cloud [2]. Though many researchers have come up with several security-related solutions, security is still a crisis. This section narrates existing and ongoing research works related to security, privacy and trust of cloud computing. It primarily investigates the major threats to the growth and adoption of cloud computing from the aspect intrusions and discusses various intrusion detection and trust assessment techniques.

Due to the open and virtualized resources of cloud, security controversies arise which include:

1. Privacy issues due to multi-tenant architecture
2. Loss of consumers' domination over their own data and resources
3. Increased probability of security attacks

Academic and industrial researchers have suggested several cloud security measures. Following a suitable combination of one or more security mechanism is fundamental for cloud service providers so that they can achieve the trust of individuals and organizations which will be helpful in increasing their business reputation [3]. In this regard, several issues of cloud security have been discussed by researchers. The risks involved in cloud are categorized into three groups [4], namely, Security risks, Privacy risks and Consumer risks. Preventing a system from a set of attacks will reflect in a secured system. The most significant elements that may cause risk in cloud security are availability, data integrity, data location, data access and network load. Achieving confidentiality of consumers' data is termed as privacy. Certain consumers may not have interest in sharing their data with others. But data stored in clouds are prone to attacks. Or otherwise, cloud service providers themselves may include some loopholes in terms of privacy policies defined in Service Level Agreements. And also, there is a possibility that cloud providers may incorporate some modifications to the terms and conditions of their service. When consumers are unaware of these modifications, it may lead to consumer risks, thereby affecting their security and privacy. Hence, consumers should be aware of terms and conditions and solutions offered by CSPs in ensuring privacy.

Cloud Security

In spite of authoritative and reliable cloud architecture, internal and external attacks always seem to happen. Either consciously or unconsciously, cloud service

providers themselves may cause harm to consumers' data. Any individual who has physical means of entry to the cloud server can cause damages to consumers' data. As consumers may not have archived copy of their data, integrity verification of their outsourced data becomes a challenge

By considering various security problems of cloud models, a Unified Modeling Language (UML) based methodology has been suggested by [5]. Generally, user privacy is achieved through access control and encryption techniques. Access control-based approaches can be used for ensuring privacy in clouds, where data are encrypted using several keys. But the problem with this approach is that consumers have to be given keys at the time of registration itself. This problem in turn introduces complexities in maintaining the secrecy of keys as consumers move around different clouds. Methods for achieving consumers' privacy in cloud storage have been developed [6]. These methods, however, require compound encryptions which are inefficient in terms of resource consumption. Furthermore, they find complexities in managing the inclusion and removal of consumers with the cloud environment.

Various security models exist to hold Provable Data Possession (PDP) property which ensures the integrity of consumers' data stored in cloud [7]. But these methods omit some of the security requirements such as authentication, privacy, dynamics, scalability, and public verifiability from their consideration. Hence, [8] employed a cooperative PDP technique which mainly concentrates on two main attacks namely, Data Leakage Attack and Tag Forgery Attack. Further, it is mentioned that each of the data integrity checking mechanism should be consumer-friendly and economic in terms of communication and computation. But still, this method incurs some minimum overhead by means of communication and computation.

Intrusion in Cloud

Authorized users of cloud services try to avail unauthorized services. In particular, insiders of clouds will carry out fraudulent activities and cause threats to security in all dimensions. Researches show that these types of attacks have to be dealt seriously.

Flooding Attacks

Intruders will first get an access to any of the legitimate hosts. From that host, intruders begin to send large number of packets. These kinds of actions will lead to Denial of Service (DoS) attacks. Direct attacks will prohibit a single server from offering expected services. On another side, indirect attacks not only affect the services of a single server; it makes other interrelated services which are tied up with the disturbed server unavailable to consumers.

User to Root Attacks

Using some password-guessing techniques, keyloggers, or phishing mechanisms, intruders gain access to genuine consumers' accounts. From there, they will try to achieve root level admittance to systems or virtual machines.

Port Scanning

Attackers will go through a set of ports for learning information like IP addresses and physical addresses along with details of gateways, routers and firewalls. Then, they locate some open ports whereat some specific services are accessible and perform illegitimate activities.

Backdoor Channel Attacks

Intruders utilize the disturbed resources as launching pad for accomplishing Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. These types of attacks are silent in the sense that they perform port scanning only. But it makes confidentiality requirements of legitimate consumers in question.

An audit based cloud intrusion detection technique is suggested by [11]. Several similar approaches exist which lack from cloud- specific attacks, high level data and middleware vulnerabilities. Each node is equipped with an intrusion detection system whose job is to cooperate with other nodes in intrusion detection. The behavior analyzer validates the activities of each consumer by comparing with that of normal consumer. The knowledge analyzer detects attacks with the help of knowledgebase which stores information about previously-happened attacks. Experiments are conducted using an artificial neural network using three categories of behavioral data: Legitimate actions, Malicious actions and Policy violations. But the results are not discussed in terms of scalability, speed and accuracy of detecting each category of attack during training and testing stages. Moreover, this approach necessitates large amount of training data.

Feasible methodologies have been suggested by [12-14] for cloud intrusion detection and prevention by adopting various techniques such as autonomic computing, risk management, fuzzy logic, and ontology. Various characteristics which are to be possessed by an efficient cloud intrusion detection system have been mentioned in their study as follows:

1. Should require no or negligible human intervention to work in a dynamic, large-scale, real-time environment.

2. Should be able to achieve optimized accuracy in detecting new types of attacks that could happen over time, that is, the system should have self-learning capability.
3. Time taken for intrusion detection should be as minimal as possible since early detection will avoid potential damages.
4. Should include self-configuration competence to deal with configuration changes of a cloud computing environment.
5. Should be reliable to offer satisfactory level of services in opposition to failures, with minimum computation and communication overheads.
6. Should be able to cooperate with other intrusion detection systems which execute concurrently in a distributed environment.
7. Should have self-defense ability.
8. Should adopt itself to changes in the behavior of users, systems and networks over time.

By comparing the features of traditional IDS with the requirements of cloud IDS, it is strongly recommended that the traditional IDS is not suitable to the cloud environment. Based on the differences between these two types of intrusion detection systems, an intrusion detection model using threads have been developed by [15]. Unfortunately, this model consumes some additional resources for thread scheduling and lacks in detecting host-based attacks.

A similar approach has been put forward where each user is offered with an individual intrusion detection system. A separate controller is used to coordinate these individual IDSs. This method essentially follows signature-based detection. And it too suffers from the consumption of additional resources and issue of detecting anomalous attacks.

An analogous approach which overcomes the deficiencies of conventional IDS has been suggested by [16] which works in SaaS delivery model. Set of lightweight IDS agents are embedded into the network with the centralized detection controller. But this method cannot be adopted for a network which incurs high traffic. In that sense, its cost of communication and computation will be high.

In [17] recommended an IDS in which separate IDSs are installed in each virtual machine. To have coordination between them, a notion of cloud alliance has been introduced using an integrated knowledge and behavior-based intrusion detection techniques. This method is capable of achieving the detection of malicious activities in the presence of single point failures.

The authors in [18] suggested the framework which combines the features of both intrusion detection and prevention in cloud model. In spite of an integrated anomaly and signature-based detection of their approach, their model has not been proven with experimental results. An analogous CIDPS has been developed by [18] and it too has been explained from theoretical aspects only.

An in-depth study of various cloud security issues has been presented in [19]. Regarding data and application security, Backdoor and Debug Options, CAPTCHA breaking attacks, Cookie Poisoning, Cross Site Scripting attacks, Denial of Service (DoS) attacks, Dictionary attacks, Hidden Field Manipulation, Man in the Middle attacks and SQL injection attacks have been identified as major threats. DNS Attacks, Sniffer Attacks, Issue of Reused IP Addresses, and BGP Prefix Hijacking are the chief risks which may affect the network security. By considering these kinds of attacks, the pros and cons of different security schemes based on storage security, consumer privacy, trust in clouds and virtualization have been analyzed. The authors concluded that a security mechanism should be capable of protecting data against all types of possible hazards.

A virtualization-based intrusion prevention system is framed in [20] which works well against certain types of network-based attacks. This model is situated in a virtual machine and monitors the flow of packets for detecting intrusions. This system is designed by considering the dynamic nature of clouds where the current state and state transition of each virtual machine is illustrated by the deterministic finite automata (DFA). But this method incurs some computational complexity in the presence of high network traffic.

The user behavior based CIDPS has been developed in [21]. According to this method, a profile is created for each user on each virtual machine. This profile is constructed based on the past behavior of users and by acquiring the control of virtual machines at random periods. Then the detection module analyses the network traffic which crosses virtual machines. Based on the stored profile, comparisons are done between the current behavior of traffic and the profile database. The results of comparison inform about the presence of intrusions, if any. The profile database is periodically updated so that new attacks can also be identified. But, at times, this approach fails to detect impersonation attacks.

This research work aims to design and analyze the performance cloud intrusion detection systems using fuzzy self-constructing clustering algorithm with statistical analysis. The major objectives of the research work are stated as follows:

1. To design an intrusion detection system for identifying malicious users whose behavior affects

the security and privacy of legitimate users and resources of cloud.

2. To analyze the performance of the designed system in terms of mean square error.
3. To compare the performance of the system against other clustering algorithms.
4. To acquire maximum accuracy for detecting intrusions.

3. Proposed Methodology

This section addresses the use of a self-constructing clustering technique based on neuro-fuzzy in the construction of a cloud intrusion detection system. In the realm of cloud intrusion detection, statistical analysis was performed to compare the performance of this technique to other well-known clustering algorithms.

Fuzzy Neural Network Aided Cloud Intrusion Detection System (FNN-CIDS)

For an IaaS model, a cloud IDS is developed in this work. It is also designed as a hybrid system which adopts ANN and Fuzzy systems as shown in Figure 1.

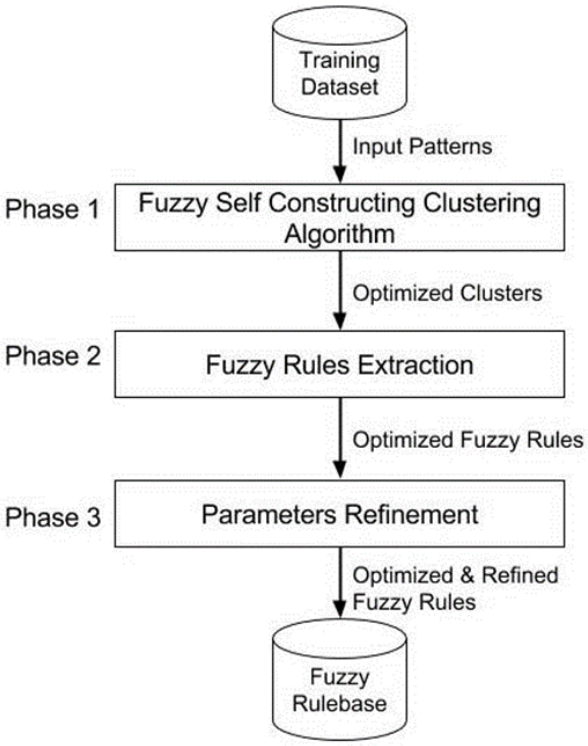


Fig 1. Fuzzy Neural Network Aided Cloud Intrusion Detection System Architecture

This system is constructed in three stages, namely:

Stage 1: Construction of clusters

Stage 2: Extraction of fuzzy rules

Stage 3: Refinement of antecedent and consequent parameters.

Algorithm: Fuzzy Neural Network Aided Cloud Intrusion Detection System (FNN-CIDS) Algorithm
Begin{Algorithm} Step 1: Fetch the first input pattern Step 2: Compute the input similarity between current pattern and all existing clusters Step 3: If the current pattern satisfies the input similarity condition Then Compute the output similarity between current pattern and all existing clusters If the current pattern satisfies the output similarity condition

Then

If only one cluster exist that satisfies both tests

Then

Associate current pattern with the matching cluster

Update its membership function

Else if several clusters exist that satisfies both tests

Then

Choose the cluster with the highest membership degree

Mark it as a winning cluster

Associate current pattern with the winning cluster

Update the mean, deviation and altitude o the winning cluster

End if

End if

End if

Else

Create a new cluster

Mark it as a matching cluster

Initialize its membership unction

End if

Step 4: Check whether all patterns have been processed

Step 5: If all clusters are processed

Then

Return the set o clusters and stop

Else

Read the next pattern and mark it as a current pattern

Goto step (3)

End if

End{Algorithm}

The input similarity between each pattern $I_i = [I_{i1}, I_{i2},\ldots, I_{iN}]$, $1 \leq i \leq K$, and each cluster Cj is computed by Equation (1), where m_j is the mean of cluster C_j and σ_j is the standard deviation of cluter C_j.

$$\mu(I_i, C_j) = \prod_{l=1}^{N} exp\left\{-\left[\frac{I_{il}-m_{jl}}{\sigma_{jl}}\right]^2\right\} \quad \ldots\ldots. (1)$$

Input similarity test between current pattern I_i and cluster C_j results in success if and only if the condition specified in Equation (2) holds.

$$\mu(I_i, C_j) \geq \omega_{in}$$

Here ω_{in} is a predefined input threshold which lies in the range [0.0, 1.0]. Number of clusters is affected by the value of ω_{in}. As ω_{in} tends to reach 1.0, the number of clusters also increases. Here, each cluster encloses a small number of patterns. Similarly, as ω_{in} tends to be close to 0.0, the number of clusters decreases. In this case, each cluster encloses a large number of patterns.

If the input similarity test is not succeeded for the current pattern, it becomes unnecessary to take output similarity test for that pattern. Hence, it is determined that no cluster currently exists which contains patterns similar to the current pattern. Hence existing clusters cannot be

used to represent the current pattern. In this scenario, a new cluster C_{new} has to be constructed. This cluster C_{new} follows Equations from (3) to (7) for the initialization of its parameters.

$$m_{new} = [I_{i1}, I_{i2}, \ldots\ldots, I_{iN}] \ldots\ldots (3)$$

$$\sigma_{new} = \sigma_0 \ldots\ldots\ldots (4)$$

$$alt_{new} = O_i \ldots\ldots\ldots (5)$$

where i indicates the current pattern I_i and σ_0 is the initial deviation.

Otherwise, if the input similarity test is succeeded for the current input pattern, the output similarity test is conducted. For a cluster C_j, the mean of the expected outputs of all the patterns belonging to that cluster is calculated and it is called the altitude alt_j. It is expressed in Equation (6) for $1 \leq j \leq CC$, where $PCNT_j$ is the total number of training patterns belonging to cluster C_j and CC is the total number of clusters currently existing.

$$alt_j = \frac{\sum_{l=1}^{PCNT_j} O_l}{PCNT_j} \ldots\ldots\ldots (6)$$

The difference between the expected (desired) output of the current pattern I_i and the altitude of each cluster C_j is calculated by Equation (7).

$$diff_{ij} = |O_i - alt_j| \ldots\ldots (7)$$

Here, we use parameters O_{high} and O_{low} which represent the highest and the lowest values of desired outputs, respectively. The difference between them is indicated by $DIFF$ and it is defined by the Equation (8).

$$DIFF = |O_{high} - O_{low}| \ldots\ldots\ldots (8)$$

Pattern I_i is said to have passed the output similarity test for cluster C_j if Equation (9) is satisfied.

$$diff_{ij} \leq \omega_{out}(DIFF) \ldots\ldots (9)$$

where ω_{out} is a predefined output threshold that too lies within the range out [0.0, 1.0].

Here, the value $|O_{high} - O_{low}|$ fixes a threshold for making a decision on the divergence between the expected and actually obtained output. If the error in difference between the two outputs ($diff_{ij}$) does not be positioned between O_{high} and O_{low}, then the probability of getting success in the output similarity test is very low for the current pattern. So, these two parameters (O_{high} & O_{low}) impose a constraint on the output similarity test for achieving high accuracy. As for as the condition mentioned in Equation (9) holds, the pattern I_i said to have cleared the output similarity test.

In addition to the input threshold ω_{in}, the output threshold ω_{out} also has an impact on the number and size of clusters. As the value of ω_{out} increases, it makes the product ($\omega_{out}. DIFF$) of Equation (9) larger. Hence, several patterns will clear the output similarity test and will belong to the same cluster. This will reduce the total number of possible clusters. Since these reduced number of clusters have to accommodate all patterns of the detection dataset, size of each cluster is increased. This situation leads to gaps between clusters. Unluckily, these gaps represent patterns for which the output is unidentified.

On the contrary, as ω_{out} decreases, the value ($\omega_{out}. DIFF$) of Equation (9) gets reduced. This makes the output similarity test tough. Hence the pattern will clear this test, if and only if it closely resembles the set of patterns which already belongs to that cluster. Hence the probability of clearing the output similarity test for cluster C_j by pattern I_i is reduced. This situation gives way to the creation of several new clusters. Hence the number of clusters will increase. Since each cluster contains only those patterns which are very similar to each other, each cluster will have a smaller number of patterns. So, size of each cluster gets reduced. So as ω_{out} decreases, the system will have large number of smaller clusters. When number of clusters increases, the possibility of cluster overlapping is also high. Consequently, a pattern may belong to more than one cluster. In regard to the input and output similarity tests, the following three cases exist:

Case 1: When the pattern clears both tests, it means that there exists a cluster to which the current pattern can be associated.

Case 2: When the pattern fails to clear the input similarity test, output similarity test is not conducted for that pattern and hence a new cluster is created.

Case 3: However, the probability for the pattern to fail in the output similarity test, provided the input similarity test is cleared, is very low. This does not mean that it is not necessary to conduct output similarity test for those patterns which have cleared the input similarity test. This case also leads to the creation of new clusters which are optimized in the parameter refinement phase.

If the constraints mentioned in Equations (1) and (9) are satisfied for several clusters for the pattern [I_i, O_i], a cluster which has the highest degree of membership is elected as a winning cluster C_w. Then the mean, standard deviation and altitude of Cw are revised by the Equations from (10) to (13).

$$PCNT_w^{rev} = PCNT_w + 1 \ldots\ldots (10)$$

$$m_w^{rev} = \frac{\sum_{l=1}^{PCNT_w^{rev}} I_{li}}{PCNT_w^{rev}} \ldots\ldots (11)$$

$$\sigma_w^{rev} = \sqrt{\frac{\sum_{l=1}^{PCNT_w^{rev}} (I_{li} - m_w^{rev})^2}{PCNT_w^{rev} - 1}} \quad \ldots\ldots\ldots (12)$$

$$alt_w^{rev} = \frac{\sum_{l=1}^{PCNT_w^{rev}} O_l}{PCNT_w^{rev}} \quad \ldots\ldots (13)$$

When an input pattern fails to clear similarity tests, it means that existing clusters cannot be used to represent this pattern. Hence a new cluster has to be created. Sine this cluster in initially going to have this input pattern alone, its deviation will be zero. But it cannot be employed in fuzzy similarity measurements. So σ_0 is adopted as its initial deviation. Later, when new patterns are associated with this cluster, its size, mean, deviation and altitude are updated by Equations from 10 to 13.

Similar to ω_{in} and ω_{out}, σ_0 also influences the number of clusters. For small values of σ_0, this clustering requires highly similar patterns alone to be associated with the same cluster. This results in the formation of large number of smaller clusters. Large values of σ_0 allows several patterns to be in the same cluster where each cluster may include some dissimilar patterns that are related to the deviation. This leads to the formation of small number of larger clusters.

4. Results and Discussion

The performance of the proposed FNN-CIDS cloud intrusion detection system to detect various attacks. The KDD Cup 1999 contains four types of attack Denial of Service attack (DoS), Probe attack, User to Root attack (U2R), and Root to Local attack (R2l) as discussed. Performance is measured based on Cluster Size, number of features, and percentage of training data used to train the FNN-CIDS. The matrices used for evaluation are True Positive Rate (TPR), False Positive Rate (FPR), Accuracy, Precision and F-Score. The performance of Adaptive Lion Neural Network cloud intrusion detection system is compared with K-Means (Kulhare and Singh, 2013), FCM (Pandeeswari and Ganeshkumar ,2015) and WLI (Wu et al., 2015).

Detection of DoS Attack

DoS attack is launched by an attacker by preventing a genuine user to access the authorized resources by making them unavailable. DoS attacks contribute up to 80% of the total attack space. Fig 1 and Table 1 shows the results obtained on the basis of Cluster Size.

Table 1. Performance-based on Cluster Size for DoS Attack

A. TPR based on Cluster Size				
Cluster Size	FCM	KM	WLI	Proposed FNN-CIDS
3	0.90331197	0.90866001	0.92175926	0.92760342
4	0.90121169	0.90235378	0.90334758	0.92571225
5	0.90024929	0.90185185	0.91786733	0.92811832
B. FPR based on Cluster Size				
Cluster Size	FCM	KM	WLI	Proposed FNN-CIDS
3	0.26370023	0.22939866	0.27069292	0.16262997
4	0.26344086	0.25871927	0.25707763	0.25448276
5	0.29675406	0.25975904	0.24615385	0.23859922
C. Accuracy based on Cluster Size				
Cluster Size	FCM	KM	WLI	Proposed FNN-CIDS

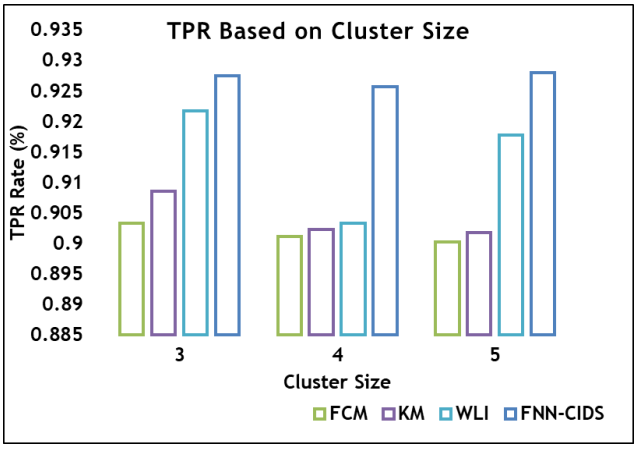
3	0.80769231	0.83050999	0.84264601	0.84265859
4	0.84456909	0.84944801	0.84955453	0.84976819
5	0.82332621	0.8456669	0.85001781	0.85522096

D. Precision based on Cluster Size

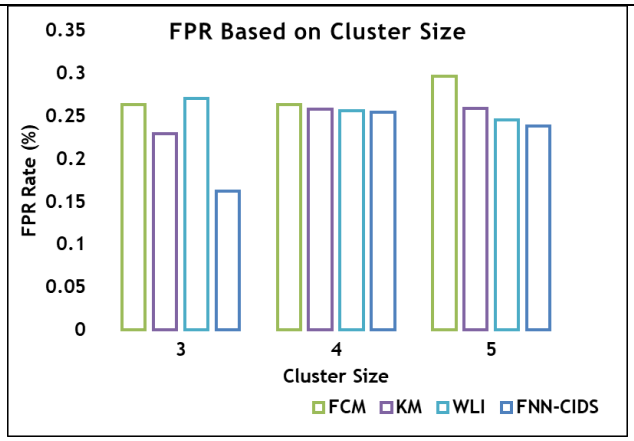
Cluster Size	FCM	KM	WLI	Proposed FNN-CIDS
3	0.80362998	0.82706013	0.82930708	0.83737003
4	0.83655914	0.84128073	0.84292237	0.84551724
5	0.80324594	0.84024096	0.85384615	0.86140078

E. F-Score based on Cluster Size

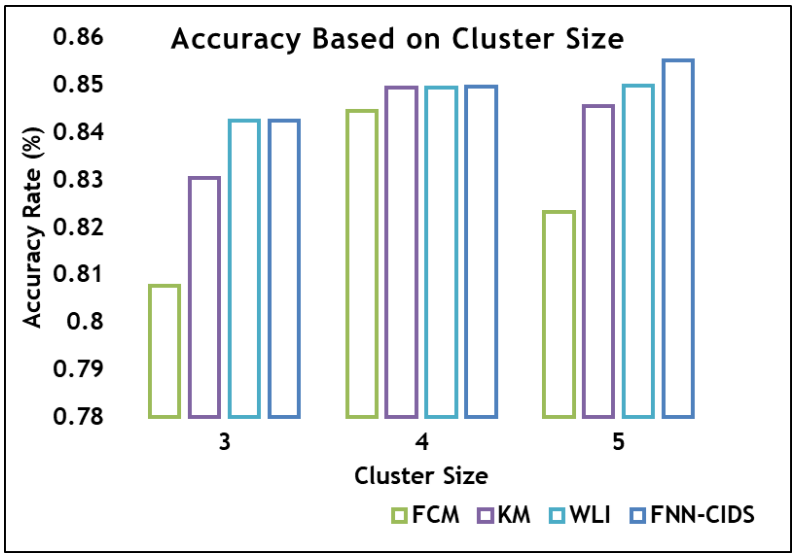
Cluster Size	FCM	KM	WLI	Proposed FNN-CIDS
3	0.85056	0.865942	0.873093	0.88018
4	0.867683	0.870748	0.87209	0.883799
5	0.848986	0.869957	0.8847	0.893516



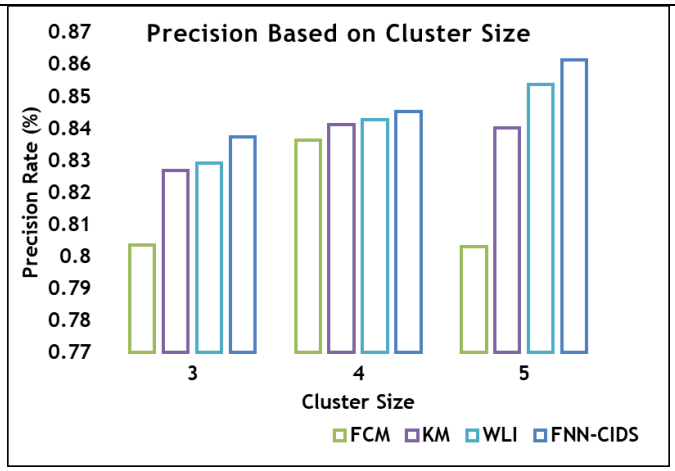
(A)


FPR Based on Cluster Size
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
FPR Rate (%)
3
4
5
Cluster Size
FCM KM WLI FNN-CIDS

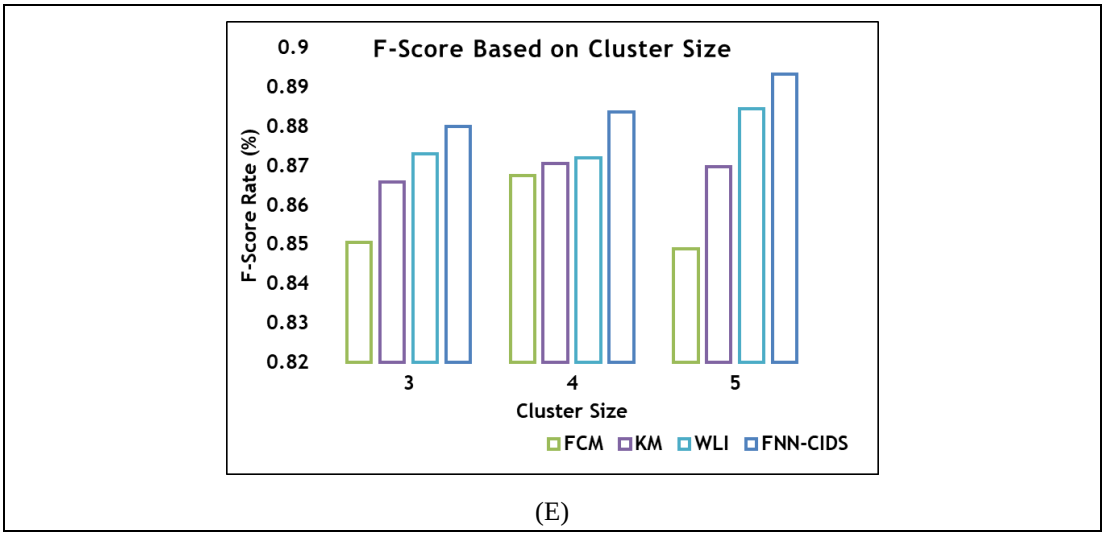
(B)


Accuracy Based on Cluster Size
0.86
0.85
0.84
0.83
0.82
0.81
0.8
0.79
0.78
Accuracy Rate (%)
3
4
5
Cluster Size
FCM KM WLI FNN-CIDS

(C)


Precision Based on Cluster Size
0.87
0.86
0.85
0.84
0.83
0.82
0.81
0.8
0.79
0.78
0.77
Precision Rate (%)
3
4
5
Cluster Size
FCM KM WLI FNN-CIDS

(D)



(E)

Fig 2. Performance-based on Cluster Size for DoS Attack

Fig 2(a) shows the True Positive Rate where FCM attains 90.33%, K-means attains 90.8%, WLI attains 92.7% and FNN-CIDS attains 92.7% True Positive Rate. From the Fig 2(a) it is evident that proposed FNN-CIDS achieves highest True Positive Rate. The Fig 2(b) shows the False Positive Rate where FCM records 26.3%, K-means records 22.9%, WLI records 24.6% and FNN-CIDS records 16.2% False Positive Rate. From the Fig 2(b) it is evident that proposed FNN-CIDS achieves lowest False Positive Rate. Fig 2(c) shows the detection Accuracy where FCM attains 84.4%, K-means attains 84.5%, WLI attains 85.0% and FNN-CIDS attains 85.5% detection Accuracy. From the Fig 2(c) it is evident that proposed FNN-CIDS achieves highest detection Accuracy. Fig 2(d) shows the Precision where FCM attains 83.6%, K-means attains 84.1%, WLI attains 85.38% and FNN-CIDS attains 86.0 % Precision. From the Fig 2(d) it is evident that proposed FNN-CIDS achieves highest Precision. Fig 2(e) shows the F-Score value where FCM attains 86.7%, K-means attains 87.0%, WLI attains 88.4% and FNN-CIDS attains 89.3 % F-Score value. From the Fig 2(e) it is evident that proposed FNN-CIDS achieves highest F-Score value.

Fig. 2 and table 1 indicate that proposed FNN-CIDS is more capable to detect DoS attacks than FCM, K-means, and WLI. FNN-CIDS detects DoS attacks with 92.8% true positive rate, 16.2% false-positive rate, 85.52% accuracy, 86.14% precision and 89.35% F-score value. And it is also evident that DoS attacks don't have any dependence on the number of clusters. For Cluster Size 3 DoS attacks have the lowest false positive rate.

Detection of Probe Attack

Probe attack defines the attackers attempt to do host or port scanning to gather information or to discover known vulnerabilities. Probe attacks contribute near about 1% of the total attack space. Table 2 and Fig 3 shows the results obtained on the basis of Cluster Size.

Table 2. Performance-based on Cluster Size for Probe Attack

A. TPR based on Cluster Size				
Cluster Size	FCM	KM	WLI	Proposed FNN-CIDS
3	0.80017806	0.80538133	0.80751425	0.82018545
4	0.80217236	0.80566643	0.80712251	0.81233951
5	0.80245726	0.80263909	0.80890313	0.85459729
B. FPR based on Cluster Size				
Cluster	FCM	KM	WLI	Proposed FNN-

Size				CIDS
3	0.29821937	0.28985215	0.2887045	0.2376652
4	0.2871567	0.28127341	0.2460177	0.23409524
5	0.28672331	0.2787037	0.23508772	0.21134259

C. Accuracy based on Cluster Size

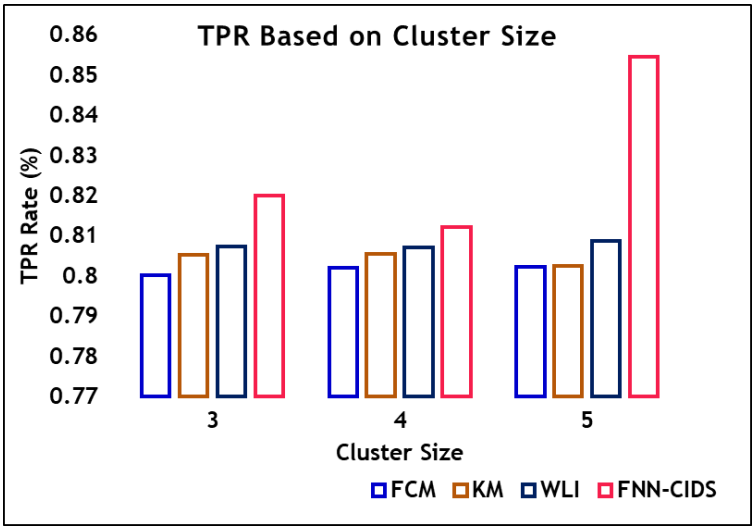
Cluster Size	FCM	KM	WLI	Proposed FNN-CIDS
3	0.80017806	0.82425214	0.82886671	0.85399429
4	0.83360656	0.83810541	0.85016026	0.85297789
5	0.82537393	0.84668803	0.85060628	0.87380613

D. Precision based on Cluster Size

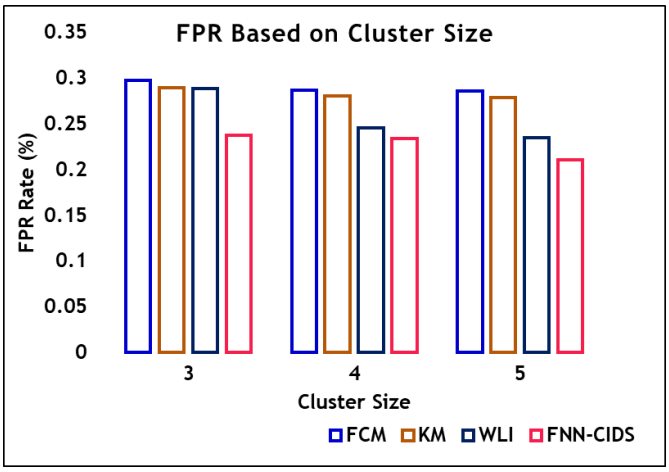
Cluster Size	FCM	KM	WLI	Proposed FNN-CIDS
3	0.80017806	0.81014785	0.8112955	0.8623348
4	0.8128433	0.81872659	0.8539823	0.86590476
5	0.81327669	0.8212963	0.86491228	0.88657407

E. F-Score based on Cluster Size

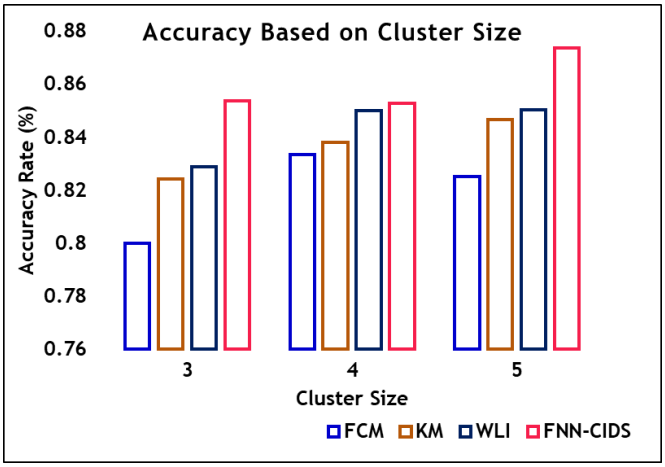
Cluster Size	FCM	KM	WLI	Proposed FNN-CIDS
3	0.80017806	0.807758	0.8094	0.840732
4	0.80747258	0.812144	0.829891	0.838267
5	0.80783075	0.811861	0.835971	0.870292


TPR Based on Cluster Size
TPR Rate (%)
0.86
0.85
0.84
0.83
0.82
0.81
0.8
0.79
0.78
0.77
3
4
5
Cluster Size
FCM KM WLI FNN-CIDS

(A)


FPR Based on Cluster Size
FPR Rate (%)
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
3
4
5
Cluster Size
FCM KM WLI FNN-CIDS

(B)


Accuracy Based on Cluster Size
Accuracy Rate (%)
0.88
0.86
0.84
0.82
0.8
0.78
0.76
3
4
5
Cluster Size
FCM KM WLI FNN-CIDS

(C)

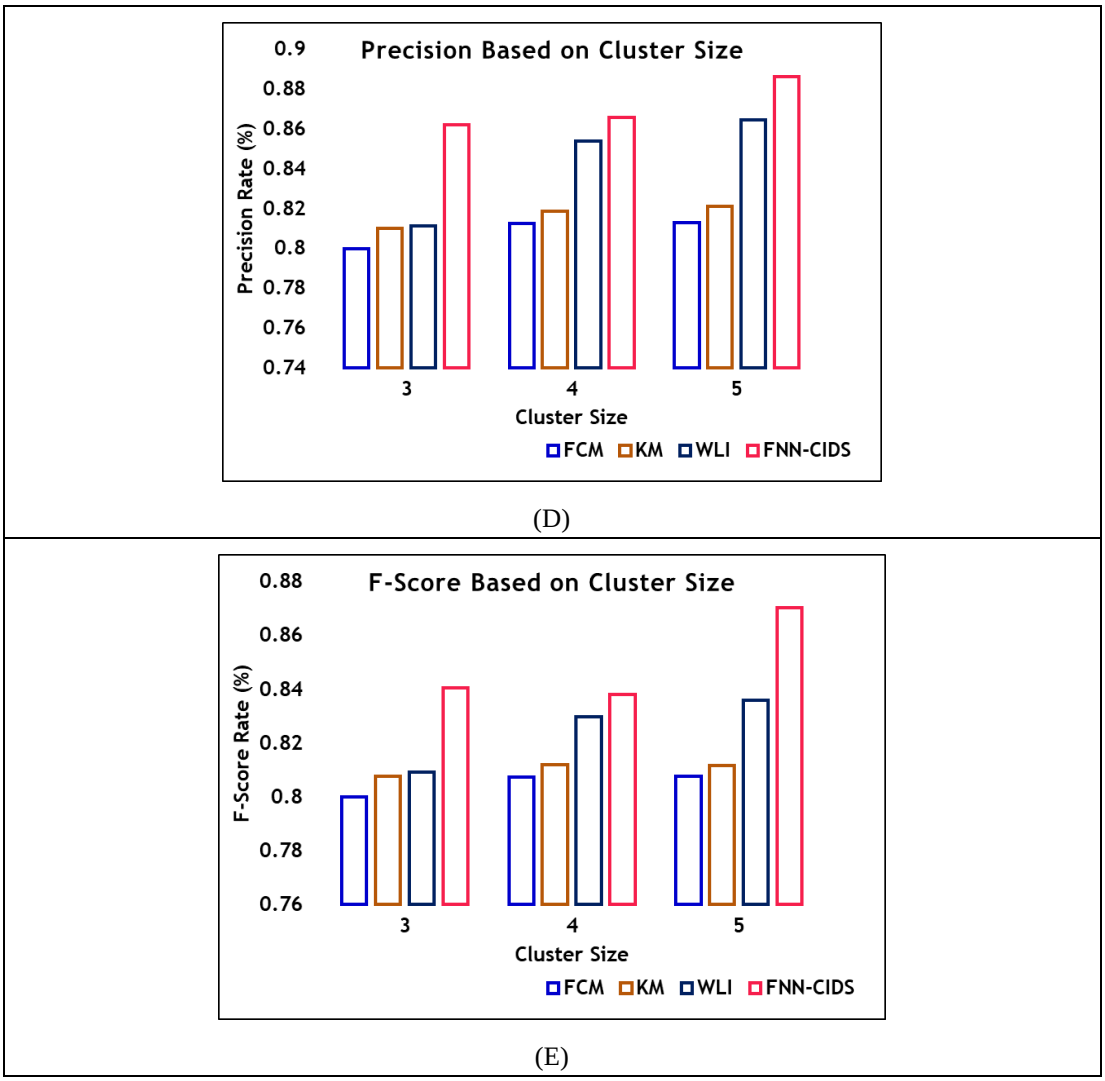


Fig 3. Performance-based on Cluster Size for Probe Attack

Fig 3 (a) shows the True Positive Rate where FCM attains 80.24%, K-means attains 80.56%, WLI attains 80.89% and FNN-CIDS attains 85.45% True Positive Rate. From the Fig 3(a) it is evident that proposed FNN-CIDS achieves highest True Positive Rate. The Fig 3(b) shows the False Positive Rate where FCM records 28.6%, K-means records 27.87%, WLI records 23.50% and FNN-CIDS records 21.13% False Positive Rate. From the Fig 3(b) it is evident that proposed FNN-CIDS achieves lowest False Positive Rate. Fig 3(c) shows the detection Accuracy where FCM attains 83.36%, K-means attains 84.66%, WLI attains 85.06% and FNN-CIDS attains 87.38% detection Accuracy. From the Fig 3(c) it is evident that proposed FNN-CIDS achieves highest detection Accuracy. Fig 3(d) shows the Precision where FCM attains 81.32%, K-means attains 82.12%, WLI attains 839% and FNN-CIDS attains 88.65 % Precision. From the Fig 6.5(d) it is evident that proposed FNN-CIDS achieves highest Precision. Fig 3(e) shows the F-Score values where FCM attains 80.78%, K-means attains 81.21%, WLI attains 83.59% and FNN-CIDS attains 87.02 % F-Score value. From the Fig 3(e) it is evident that proposed FNN-CIDS achieves highest F-Score value. Table 2 shows the results obtained for cluster size 3, 4 and 5 and shows the comparative analysis of detection of probe attack based on cluster size.

5. Conclusion

This research work employs a fuzzy self-constructing clustering algorithm to incorporate intrusion detection in a cloud environment which has been used to detect Denial of Service (DoS), and Probe attacks. Comparative analysis among the proposed CIDS and other methods (FCM, K-Means, WLI) based on the number of clusters, number of features, and training data percentage consequently assess their performance through a series of testbed experiments based on KDD dataset. These activities served the purpose of revealing the circumstances by which hypervisor detector outperformed other methods and consequently, constituted as an empirical proof for justifying the hypervisor detector. The overall performance of FNN-

CIDS to detect malicious activities present in the cloud environment is good.

References

[1] Abraham, A & Jain R 2005, 'Soft computing models for network intrusion detection systems', Classification and Clustering for Knowledge Discovery. Studies in Computational Intelligence, Springer, Berlin, vol. 4, pp. 191–207.

[2] Ahmad, S, Ahmad, B, Saqib, SM & Khattak, RM 2012, 'Trust Model: Cloud's Provider and Cloud's User', International Journal of Advanced Science and Technology, vol. 44, pp. 69-80.

[3] Alsafi, HM, Abduallah, WM & Patha, AK 2012, 'IDPS: an integrated intrusion handling model for cloud computing environment', International Journal of Computing and Information Technology, vol. 4, no. 1, pp. 1-16.

[4] Alsharafat, W 2013, 'Applying Artificial Neural Network and eXtended Classifier System for Network Intrusion Detection', The International Arab Journal of Information Technology, vol. 10, no. 3, pp. 230-238.

[5] Amirreza, Z & Alireza, Z 2012, 'Internet intrusion detection system service in a cloud', International Journal of Computer Science Issues, vol. 9, no. 5, pp. 308-315.

[6] Ang, JH, Tan, KC & Mamun, AA 2010, 'An evolutionary memetic algorithm for rule extraction', Expert Systems with Applications, vol. 37, no. 2, pp. 1302-1315.

[7] Aziz, ASA, Salama, MA, Hassanien, AE & Hanafi, SEO 2012, 'Artificial Immune System Inspired Intrusion Detection System using Genetic Algorithm', Informatica, An International Journal of Computing and Informatics, vol. 36, no. 4, pp. 347-357.

[8] Bakshi, A & Yogesh, B 2010, 'Securing cloud from ddos attacks using intrusion detection system in virtual machine', Proceedings of Second International Conference on Communication Software and Networks, pp. 260-264.

[9] Balazs, K, Koczy, LT & Botzheim, J 2008, 'Comparison of Fuzzy Rule-based Learning and Inference Systems', Proceedings of ninth International Symposium of Hungarian Researchers on Computational Intelligence and Informatics, pp. 61-75.

[10] Bapuji, V, Kumar, RN, Govardhan, A & Sarma, SSVN 2012, 'Soft Computing and Artificial Intelligence Techniques for Intrusion Detection System', Network and Complex Systems, vol. 2, no. 4, pp. 24-31.

[11] Bezdek, JC 1981, 'Pattern Recognition with Fuzzy Objective Function Algorithms', Kluwer Academic PublishersNorwell, MA, USA.

[12] Bhadauria, R & Sanyal, S 2012, 'A survey on security issues in cloud computing and associated mitigation techniques', International Journal of Computer Applications, vol. 47, no. 18, pp. 47-66.

[13] Bhat, AH, Patra, S & Jena, D 2013, 'Machine learning approach for intrusion detection on cloud virtual machines', International Journal of Application or Innovation in Engineering & Management, vol. 2, no. 6, pp. 57-66.

[14] Calheiros, RN, Ranjan, R, Rose, CAFD & Buyya, R 2009, 'CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services', Technical Report, GRIDS-TR-2009-1, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia.

[15] Calheiros, RN, Ranjan, R, Beloglazov, A, Rose, CAFD & Buyya, R 2011, 'CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms', Software – Practice & Experience, vol. 41, no. 1, pp. 23-50.

[16] Chandran, SP & Angepat, M 2010, 'Cloud Computing: Analysing the risks involved in cloud computing environments', Proceedings of Natural Sciences and Engineering, Sweden, pp. 2-4.

[17] Chiu, SL 1994, 'Fuzzy model identification based on cluster estimation', Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology, vol. 2, no. 3, pp. 267-278.

[18] Chung, CJ, Khatkar, P, Xing, T, Lee, J & Huang, D 2013, 'NICE: network intrusion detection and countermeasure selection in virtual network systems', IEEE Transactions on Dependable Secure Computing, vol. 10, no. 4, pp. 198-211.

[19] Cozar, J, Ossa, L & Gamez, JA 2014, 'Learning TSK-0 linguistic fuzzy rules by means of local search algorithms', Applied Soft Computing, vol. 21, pp. 57-71.

[20] Dhage, SN, Meshram, BB, Rawat, R, Padawe, S, Paingaokar, M & Misra, A 2011, 'Intrusion detection system in cloud computing environment', Proceedings of International Conference and Workshop on Emerging Trends in Technology, pp. 235-239.

[21] Dolev, S, Gilboa, N & Kopeetsky, M 2014, 'Efficient private multi- party computations of trust in the presence of curious and malicious users', Journal of Trust Management, vol. 1, no. 8, pp. 1-21.

[22] Dunlap GW, King ST, Cinar S, Basrai M, & Chen PM 2002, 'Revirt: enabling intrusion analysis through virtual machine logging and replay', Proceedings of fifth symposium on operating systems design and implementation, USENIX, Boston, vol. 36, no. SI, pp. 211-224.

[23] Eom, JH & Park, MW 2013, 'Design of Internal Traffic Checkpoint of Security Checkpoint Model in the Cloud Computing', International Journal of Security and Its Applications, vol. 7, no. 1, pp. 119-128.

[24] Eucalyptus. Available from: http://www.eucalyptus.com [5 March 2015].

[25] Fan, W & Perros, H 2014, 'A novel trust management framework for multi-cloud environments based on trust service providers', Knowledge-Based Systems, vol. 70, no. C, pp. 392 - 406.