An Efficient Fuzzy Self-Classifying Clustering based Framework for Cloud Security

Sivakami Raja[1*], Jaiganesh M[1], Saravanan Ramaiah[2]

[1] Department of Information Technology, PSNA College of Engineering and Technology,
Dindigul, Tamilnadu, India
E-mail: rsivakami@psnacet.edu.in, jaidevlingam@gmail.com

[2] Department of Computer Science and Engineering, RVS Educational Trusts Group of Institutions,
Dindigul, Tamilnadu, India
E-mail: directorrvsetgi@revsgroup.com

Received 29 August 2016

Accepted 1 December 2016

Abstract

Though cloud computing has become an attractive technology due to its openness and services, it brings several security hazards towards cloud storage. Since the distributed nature of clouds is achieved through internetworking technologies, clouds suffer from all the vulnerabilities by which networking also suffers. In essence, data stored in clouds are vulnerable to attacks from intruders. But, no single technique can provide efficient intrusion detection. In this paper, we propose fuzzy self-classifying clustering based cloud intrusion detection system which is intelligent to gain knowledge of fuzzy sets and fuzzy rules from data to detect intrusions in a cloud environment. Its efficiency is explained by comparing with other three cloud intrusion detection systems. Using a standard benchmark data from a CIDD (Cloud Intrusion Detection Dataset), experiments are conducted and tested. The results are presented in terms of success rate accuracy.

Keywords: Fuzzy neural networks, Intrusion detection, Hybrid intelligent systems, Partitioning algorithms, Pattern analysis.

1. Introduction

Cloud storage system is a concept that has become very popular since it provides extensive storage capacity for end users. Users can avail services on rental basis without fear about the cost of data storage and maintenance. Although cloud service providers (CSP) can provide more vigorous infrastructures and honest security measures, this data outsourcing makes data security and privacy a big question. To answer this question, several techniques and policies are being used. But intruders and hackers take pleasure in breaking security fence. So cloud users still face the problems of confidentiality, integrity and availability from internal as well as external factors. If these concerns are not solved properly, the growth of cloud computing may not be in a positive direction.

Intrusion is an act of threatening data confidentiality and integrity through impersonation, data corruption, data theft and other similar kind of fraudulent activities. An intrusion detection system (IDS) fights against confidentiality, integrity and availability issues during intrusions. A successful intrusion can put the entire network out of action. Moreover, intruders are smart in not leaving any sign of identification. But, intrusions can be realized as patterns of monitored sequence. Using these patterns, intrusions can be identified by measuring the deviation between intrusive and non-intrusive patterns. In regard to this, significant researches are going ahead. Hence, it is still a thrust area of research. Data mining concepts can be implemented in IDS to scrutinize valuable information from huge amount of noisy and dynamic data. It can also be used to make perfect predictions for future scenarios.

In our work, a cloud intrusion detection system (CIDS) is designed based on fuzzy neural network in two phases, namely, training phase and testing phase. The training phase consists of clustering, fuzzy rulebase generation and fuzzy rulebase optimization stages. The testing phase is used to measure the success rate which is defined as the degree to which each pattern is classified correctly. The clustering stage of training phase is implemented with our proposed fuzzy self-classifying clustering algorithm and the corresponding success rates are measured based on which the comparison is made with other three systems, namely, K means (KM) clustering based CIDS, modified K means (MKM) clustering based CIDS and fuzzy self-constructing clustering based CIDS.

The rest of this paper is organized as follows: Section 2 outlines IDS in cloud. Our system model with proposed method is described in Section 3. Experimental results and analysis are presented in Section 4. Finally, Section 5 concludes our work.

2. Intrusion Detection in Cloud

Several intrusions which can threat cloud services and security issues are examined in Ref. 1. It concludes that soft computing based intrusion detection techniques can assure security. Various cloud security requirements are identified and cryptography based approach is presented to ensure security in cloud environments.[2] Prime security worries in cloud computing are classified as storage, virtualization, and networks.[3] Classical security measures may not act sound in cloud environments since they are composed of a combination of different technologies. In the field of network intrusion detection, T-S model based on fuzzy neural network[4] is tested and achieved better results for intrusion detection dataset analysis. A genetic algorithm and formal concept analysis based approach is presented in Ref. 5 for creating clusters from intrusion detection dataset by using Minkowski distance. This approach can be used in cloud environment integrated with data mining based intrusion detection. An automatic approach is proposed in Ref. 6 to detect intrusions in a dynamic environment by using data mining concepts.

Even though pattern-based IDS is straightforward to realize and very successful to scrutinize known attacks, its yield in the detection of unknown attacks is not optimum. On the other case, rule-based techniques perform well to detect unknown attacks. But they may come across the problem of contemporary observation for attacks. Likewise, heuristic approaches can also detect unknown attacks, but lacks performance in real-time functions due to computational complexities.[7] Autonomic computing, fuzzy theory, ontology, and risk management are suggested[8] as notions to design an effective cloud intrusion detection and prevention system. Based on the knowledge base of behaviours of each cloud user, an IDPS[9] is designed to secure cloud from known attacks. In Ref. 10, distributed security architecture is proposed to attain cloud security. Though this approach enhances cloud security, it introduces some additional complexities. Security issues in IaaS clouds incorporating multi-tenancy are discussed[11] and at the end, arrived at the point that several cloud environments use access control mechanisms and cryptography to security. In Ref. 12, security challenges in cloud service delivery models are studied. The majority of the recently proposed techniques on cloud operates at each of the infrastructure, platform, and application layers independently, and provides detection free from other layers. In Ref. 13, a study on some of the general hazards to cloud security is presented together with the means of handling them. Data administration and security practices of some of the cloud service providers are also discussed.

The effectiveness of intrusion detection heavily depends on the construction of fuzzy rules. Since fuzzy rules are generated from clustering of intrusion detection dataset, the behavior of clustering algorithms influences the effectiveness of intrusion detection. Each active cluster has to be formed in a way to include an adequate amount of data for ensuring fuzzy input – output association in the input data space. Fuzzy C Means algorithm[14] and Kohonen's Self-Organizing Map[15] are the well known algorithms which can be employed in this regard. But the initialization of total number and location of cluster centers directly affect the competence of solutions. To estimate these values, Mountain method[16] can be employed for its clarity and efficacy. But its order of computational growth is exponential, as dataset's size increases. An alternative approach, called subtractive clustering[17] can be adopted to fix cluster centers, where no guess is made in the initialization of cluster centers. This approach treats all data as cluster centers. But problem arises when the actual center deviates from all data points. Besides this, the radius of cluster affects the membership of data. If the radius is large, it will encompass all data. Else if it is very small, it will neglect neighbor data. K means clustering[18] realizes desirable precision than other methods. Fuzzy C Means clustering can also offer desirable performance relative to that of K means clustering. Unfortunately, it encounters speed related problems due to fuzzy measures associated calculations. Hence, it is essential to devise a new clustering algorithm that can be successfully applied in intrusion detection.

Since the openness of clouds introduces security problems, several researchers invite themselves toward formulating solutions to improve and guarantee cloud security. Machine learning is one of such methods which works with negligible human intervention. Since intrusions are the major bottlenecks of cloud security, it is decided to design a cloud intrusion detection component with the application of machine learning approach. Existing intrusion detection methodologies show poor performance in detecting less-frequent attacks. Moreover, they suffer from excessive consumption of resources, increased false rates, inability in detecting long-penetration attacks, utilization of additional mechanisms and speed-related issues. For this purpose, we have designed a fuzzy neural network and genetic algorithm based mechanism[19] which can overcome all these drawbacks with 98.598% accuracy in detecting intrusions.

Even though this genetic algorithm based CIDS[19] produces satisfactory results, the time taken to identify the occurrence of intrusion is found to be high. Early detection of intrusions is very much essential since late detection will lead to some disasters which may not be repaired. From the aspect of cloud intrusion detection, scalability is another important factor which influences time and accuracy of intrusion detection. The intrusion detection system developed in our previous work[19] is based on genetic algorithm (GA) and is of four phases. Hence, it is tried in this work to avoid the need of genetic algorithm (by reducing number of phases) for achieving early detection of intrusions with an application of efficient fuzzy self classifying clustering algorithm.

3. System Model

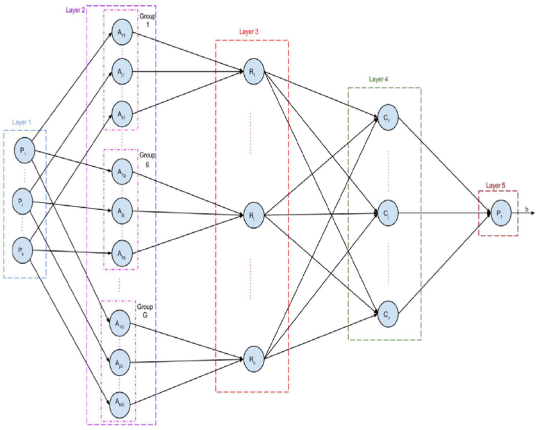

Fig. 1. Fuzzy Neural Network

The proposed architecture is a five-layered neural fuzzy system as shown in Fig 1. Layers from 1 to 5 consist of the input nodes, antecedent nodes, rule nodes, consequent nodes and output nodes, respectively. The function of the first layer is to simply forward the input vector to the second layer which in turn matches antecedents of the input training value with the corresponding labels. The next layer generates the set of type 2 fuzzy TSK rules from the training data by measuring the overall similarity between the input training vector and the antecedent part of the fuzzy rule. The purpose of layer 4 is to extract consequent for the fuzzy rules from the input vector. The last layer generates a crisp output through defuzzification.

3.1. *Proposed methodology*

As the word ‘cloud’ of ‘cloud computing’ speaks for internet, cloud computing is observed as an internet based computing. It is built on resource sharing for offering services to its users through virtualization, where consumers have the benefit of services on pay-per-use basis. As cloud services are offered over the structure and functionalities of internet, cloud encounters all the types of security and privacy concerns which exist in the internet too.[1] But, intrusion monitoring within virtual machines is different from intrusion monitoring within physical machines. Moreover, the security risks of remote data centers of clouds are not clearly defined so far. Hence, developing an efficient methodology for achieving cloud security with maximum accuracy is often considered as a challenging task. This challenge leads to the requirement of inventing an idea which can keep various cloud services away from security hazards. Since the conventional authentication and confidentiality approaches become weak in detecting malicious activities of cloud users, an integrated knowledge and behavior based approach is gaining importance. This is due to the reason that attacks to cloud will be silent so that traditional host-based and network-based intrusion detection techniques cannot identify them.

Since the cloud attackers do not leave any sign of intrusion, its detection becomes a challenge. The malicious users of cloud are usually identified by their following behaviours[20]:

- Trying to heavily access providers' resources such as Memory, CPU and I/O
- Abrupt increase in incoming traffic at a specific port
- Usage of a specific resource for a time longer than its average time of usage
- Increased number of login attempts within a particular time period
- Multiple logins with the common ID at same time
- Significant increase in the frequency of system calls
- Heavy information flow with same source port and / or destination port
- Trying to gain administrator privileges
- Increased bill amount to a specific consumer for having accessed the services

Hence a Cloud Intrusion Detection Dataset (CIDD) is vital to design efficient CIDS for real world detection systems. Current dataset becomes unsuitable here due to the diverse operating systems of the virtual machines, the different range of consumers' requirements, and the size of data in cloud systems. So, a Log Analyzer and Correlator System (LACS)[21, 22] has been employed to the logs that are obtained from the DARPA Intrusion Detection Evaluation Group of MIT Lincoln Laboratory to generate CIDD. This CIDD consists of both knowledge-based and behavior-based audit data that are gathered from UNIX and Windows users.

During training, as in figure 2, the proposed fuzzy self classifying clustering algorithm is used to separate the training dataset into a group of fuzzy clusters. Then, these clusters are transformed into a rulebase of type-2 fuzzy TSK IF-THEN rules. Then, each fuzzy rule in the rulebase is assigned a weightage depicting its significance in the modeling of the application environment. Conflicting rules with low influences are deemed as outliers and are subsequently deleted from the system. This approach ensures that the proposed system maintains a consistent rulebase that is able to provide an aptly description to the application problem. Then, during testing, for every pattern from test dataset, a comparison is made against the set of rules in the fuzzy rulebase. Based on the result of this comparison, either an alarm is generated to indicate the occurrence of an intrusion or the service request is granted permission to access cloud resources to indicate that it is a valid request. Since our system is trained with both normal and intrusive patterns, it is capable of detecting both intrusive and nonintrusive patterns. To establish the correctness and efficiency of our proposed algorithm, we have implemented three more cloud intrusion detection systems with other clustering algorithms, namely, K means,[18] Modified K means,[19, 23] and Self constructing clustering[24] in our work. The same dataset which is employed in our proposed system is used for these systems also with the same distribution. Detection accuracies during training and testing are

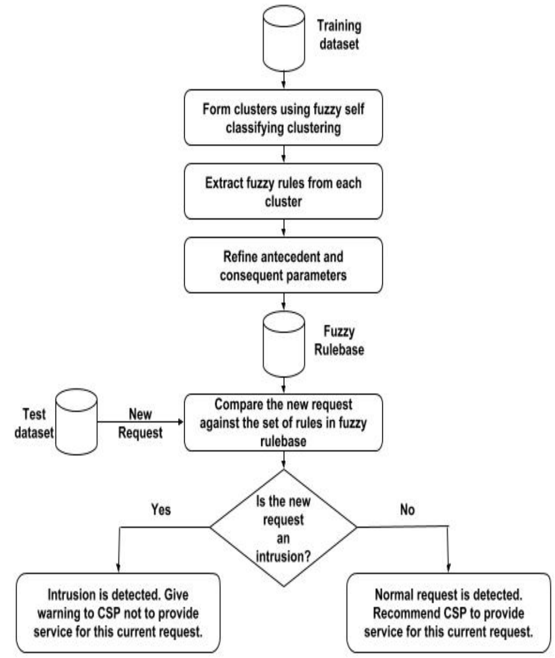
Training dataset
Form clusters using fuzzy self classifying clustering
Extract fuzzy rules from each cluster
Refine antecedent and consequent parameters
Fuzzy Rulebase
Test dataset
New Request
Compare the new request against the set of rules in fuzzy rulebase
Is the new request an intrusion?
Yes
No
Intrusion is detected. Give warning to CSP not to provide service for this current request.
Normal request is detected. Recommend CSP to provide service for this current request.

Fig. 2. Fuzzy Neural Network

measured. These results are compared with the results of our proposed system.

3.2. *Proposed fuzzy self-classifying clustering*

Clustering is employed to partition the intrusion detection dataset into number of clusters, where similar patterns are associated with the same cluster. Efficient clustering will improve the learning capability of FNN. Hence this phase is essential in this work. The intrusion detection dataset contains *K* patterns in the form $[I_1, O_1]$, $[I_2, O_2]$, …, $[I_K, O_K]$ where each $I_i = [I_{i1}, I_{i2}, \dots, I_{iN}]$ and *N* is the number of variables and it is illustrated in Eq. (1).

Each pattern $[I_i, O_i]$ refers to a specific record of intrusion detection dataset and each such record has *N* number of variables as its input variables and one variable as its output variable. This output variable indicates one of the five classes namely, normal, Denial of Service (DoS), Probing, Remote-to-Local (R2L) and User-to-Root (U2R).

For each pattern in the training set, the following steps illustrate the proposed fuzzy self classifying clustering algorithm.

```
Begin{algorithm}
Read the training pattern.
If cluster(s) exist previously,
  For every existing cluster j, j>0,
      Estimate the deviation ratio $(dr_j)$ between the current pattern and cluster j by $dr_j = (1 - Match_j) / Match_j$
  End For
  Find the cluster (CC) which has the least $dr_j$ with the current pattern and assign this $dr_j$ to $dr_{\min}$
If $dr_{\min}$ does not exceed deviation threshold (dt),
    Associate the training pattern to the cluster CC.
  Else
    Create a new cluster and update existing clusters.
  End if
Else
  Create a new cluster and associate the training pattern to the new cluster.
End if
End(algorithm}
```

Here, $Match_j = e^{-\rho D_j^M}$ is a resemblance measure between current pattern and the cluster *j*, D_j^M is a Minkowski distance between current pattern and cluster *j* and ρ is a numerical element

By the completion of this algorithm, all the patterns of intrusion detection dataset will be classified and grouped into a number of clusters. Hence each cluster will contain patterns of interrelated features. Mean and deviation of *i*th cluster are given as $[m_{i1}, m_{i2}, \dots, m_{iN}]$ and $[\sigma_{i1}, \sigma_{i2}, \dots, \sigma_{iN}]$, respectively. The values of variables of CIDD are at first normalized into the range [0, 1]. Since all fuzzy rules have the same set of variables, they can be denoted as $v_1, v_2, \dots, v_N$. A type-2 fuzzy rule *i* is now extracted from each cluster C_i and it is of the form:

If v_1 is $\widetilde{A}_{i1}$ and v_2 is $\widetilde{A}_{i2}$ and … and v_N is $\widetilde{A}_{iN}$

Then op is $c_i = \omega_{i0} + \omega_{i1} v_1 + \cdots + \omega_{iN} v_N$

Here, $\widetilde{A}_{ij}$ are type-2 fuzzy sets corresponding to variables v_j of *i*th rule for $1 \le j \le N$. op *is* one of the five output classes and ω_{ij} is a real valued weight associated with variable v_j of *i*th rule. The membership function of $\widetilde{A}_{ij}$ is defined as $\mu_{\widetilde{A}_{ij}}(v_j) = gauss(u; gauss(v_j; m_{ij}^I, \sigma_{ij}^I), \sigma_{ij}^s$ and $u \in [0,1]$. Each fuzzy rule has mean and deviation as its antecedent parameters and rule weight as its consequent parameter. This rule weight value along with the values of antecedent parameters is used to compute the degree of fulfilment.

3.3. *Rulebase optimization*

Each rule includes mean, deviation and the scaled deviation as its antecedent parameters and the rule weights as its consequent parameters. The weight is tied to a rule to determine a degree of fulfilment by multiplying with antecedents. The generated T2FNN is based on neurons. Each neuron possesses a center vector and a width vector, thus symbolizes one cluster in the input space. This initial structure of a T2FNN may be redundant and may lead to unnecessary rules in the extracted set of fuzzy rules. Mean (m) and deviation (σ) are the antecedent parameters and rule weight (ω) is the consequent parameter of each fuzzy rule. Hence, we proceed to improve the precision of these rules, by refining the antecedent and consequent parameters involved, through the application of a dynamical optimal learning algorithm. An iteration of learning involves the presentation of all training patterns. In all iterations of training, we begin to improve antecedent parameters by keeping consequent parameters as constant. After that, consequent parameters are refined by holding antecedent parameters as constant. This refinement technique is repeated until this phase results in preferred approximation precision. The general process of this phase is explained with the corresponding algorithm as given below:

```
Begin{algorithm}
  Repeat
    For each fuzzy rule
      Finetune antecedent parameters, where consequent
                                parameters are kept constant
      Finetune consequent parameters, where antecedent
                                parameters are kept constant
    End for
  Until $MSE<MSE_{predefined}$
End{algorithm}
```

For K training patterns of the form $[I_1, O_1]$, $[I_2, O_2]$, …, $[I_K, O_K]$, the objective of this phase is the minimization of the objective function given by Eq. (2) which defines the mean-squared-error (MSE) value, for $1 \le j \le K$.

$$e_j = \frac{1}{2}\left[y(I_j) - O_j\right]^2 \tag{2}$$

In Eq. (2), $y(I_j)$ is the actual output and O_j is the desired output of j^{th} pattern. Eq. (3) expresses a way in which the mean of i^{th} rule can be fine-tuned by keeping ω values as constant. A similar kind of equation is followed in the refinement of deviation of rule *i*.

$$m_{ij}(l+1) = m_{ij}(l) - \left(\frac{\alpha\left(y(I_l) - O_l\right)\left(Ip_{il} - m_{ij}\right)F\left(m_{ij}, \sigma_{ij}; I_{il}\right)}{2\sigma_{ij}^{2}} \times \frac{\left(\prod_{\substack{q=1 \\ q\neq j}}^{N_{R_i}} \overline{u}_{\tilde{A}_{iq}}\right)\left(\omega_{iq} - y(I_j)\right)}{\left(\sum_{k=1}^{L} \overline{g}^{k} + \sum_{k=L+1}^{M} \underline{g}^{k}\right)} \right) \tag{3}$$

Eqs. (4) and (5) are used to fine-tune the consequent parameters by maintaining mean and deviation values fixed.

if $j \le L$,

$$\omega_{ij}(l+1) = \omega_{ij}(l) - \alpha\left\{ \frac{\left(y(I_l) - O_l\right)}{2} \times \frac{\prod_{q=1}^{N_{R_i}} \overline{u}_{\tilde{A}_{iq}}}{\sum_{k=1}^{L} \overline{g}^{i} + \sum_{k=L+1}^{M} \underline{g}^{i}} \right\} \tag{4}$$

if $j > L$,

$$\omega_{ij}(l+1) = \omega_{ij}(l) - \alpha\left\{ \frac{\left(y(I_l) - O_l\right)}{2} \times \frac{\prod_{q=1}^{N_{R_i}} \underline{u}_{\tilde{A}_{iq}}}{\sum_{k=1}^{L} \overline{g}^{i} + \sum_{k=L+1}^{M} \underline{g}^{i}} \right\} \tag{5}$$

Here, α is the learning rate parameter, N_{R_i} is the number of inputs to i^{th} rule, $\underline{u}_{\tilde{A}_{iq}}$ is the lower bound membership value of fuzzy sets $\tilde{A} - q^i$ and $\underset{\sim}{u}_{\tilde{A}_{iq}}$ is the upper bound membership value of fuzzy sets $\tilde{A} - q^i$. L and M values are derived from an iterative Karnik-Mendel method. F, $\overline{g}^{k}$ and $\underline{g}^{k}$ of equations from (3) to (5) are defined in equations (6), (7) and (8), respectively.

$$F\left(m_{ij}, \sigma_{ij}; I_{il}\right) = \exp\left(-\frac{1}{2}\left(\frac{I_{il} - m_{ij}}{\sigma_{ij}^{2}} \right)^{2} \right) \tag{6}$$

$$\overline{g}^{k} = \prod_{q=1}^{N_{R_i}} \overline{u}_{\tilde{A}_{iq}}\left(I_q\right) \tag{7}$$

$$\underline{g}^{k} = \prod_{q=1}^{N_{R_i}} \underline{u}_{\tilde{A}_{iq}}\left(I_q\right) \tag{8}$$

This phase constructs the compact fuzzy rulebase which holds the set of parameters-refined fuzzy rules. This rulebase is used to train the system about the four types of intrusive accesses and also about normal accesses. Once the training is completed, this CIDS is employed in the field of cloud intrusion detection. Whenever cloud consumer requests access to the cloud service, consumer's activity pattern is compared against this fuzzy rulebase. Inference of this comparison decides whether the consumer is an intruder or a legitimate user.

4. Experimental Results

The CIDS developed in this work is of two entities: Cloud and Consumers. Cloud is defined as an entity for hosting consumers' data. It provides IaaS services to consumers. Consumers are the entities that request and avail services from Cloud. A Cloud environment is created with Intel core 2 Duo CPU, 2.5 GHz, 4GB memory and GNU / Linux kernel 2.6.32 and designed with Eucalyptus.[25] Experiments are conducted using CIDD.[21, 22] Our implementation consists of two phases labelled, training phase and testing phase. In the training phase, Cloud is loaded with the two-third of intrusion detection dataset (named as training dataset). A set of fuzzy rules is extracted from the fuzzy neural network using this training dataset and refined to achieve the highest precision on the dark data. These fuzzy rules are stored in a fuzzy rulebase and this rulebase is available in Cloud. This step makes Fuzzy Neural Network (FNN) to learn normal and abnormal patterns of CIDD, in several iterations. In addition to this learning of known patterns, it also becomes capable of identifying new or mysterious patterns. The deviation between the actual output and the desired output can then be used to train the ANN further so that the error in its pattern classification can be reduced to an acceptable low level. This kind of learning capability of FNN makes it suitable for detecting both known and unknown attack patterns. A backpropagation neural network is adopted in this work to train the system for learning intrusion detection dataset. After the training is over, for a set of randomly chosen trained patterns, the deviation between the actual output and the desired output is measured from which the accuracy of intrusion detection (pattern classification) during training is calculated.

In the testing phase, the remaining one-third dataset (named as test dataset) is loaded in the Consumer system. For each pattern (service request) of test dataset sent from Consumer to Cloud, the fuzzy rulebase is consulted to make the decision of allowing or denying the request. With the help of knowledge which is already gained through learning phase, the FNN now identifies various attack patterns of testing data and produces the attack type as its result. As the desired results for each pattern are available in the testing dataset, the deviation between actual and desired results is measured based on which the accuracy of intrusive and non-intrusive pattern classification can be measured. Since the testing dataset contains unknown patterns, the accuracy of experiments during testing gives a sign of correctness in detecting real time intrusions.

Accuracy measurements during training and testing are carried out for K means clustering based CIDS, modified K means clustering based CIDS, self constructing clustering based CIDS and self classifying clustering based CIDS models. The results of K means, modified K means and self constructing clustering based models are compared with that of our proposed model, for training and testing stages independently. The differences between these accuracy values show the improvements in training and testing accuracies of the proposed model. The same distribution of data is used to train and test all the four cloud intrusion detection models, namely, K means clustering based CIDS, modified K means clustering based CIDS, self constructing clustering based CIDS and self classifying clustering based CIDS models.

For the number of patterns varying from 10000 to 20000, detection accuracies for intrusion detection are measured during training and testing phases. The observations are presented in table 1 for the four

Table 1. Detection accuracy of clustering algorithms in CIDS.

No. of patterns	K Means[18]		Modified K Means[19]		Fuzzy Self Constructing[24]		Fuzzy Self Classifying	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
10000	89.33	89.18	92.41	92.26	94.70	94.64	97.69	97.64
12000	90.77	90.48	92.69	92.52	95.09	95.07	98.10	98.07
14000	91.50	91.24	93.51	93.48	96.11	95.96	99.00	98.96
16000	92.18	92.06	94.55	94.36	96.32	96.25	99.27	99.25
18000	92.88	92.78	95.49	95.32	96.50	96.48	99.52	99.48
20000	93.70	93.64	95.12	95.09	97.29	97.26	99.31	99.26

clustering algorithms. For the higher number of patterns, we obtain better success rates during training in all cases. But there exists significant deviations between training and testing success rates of the all four types of CIDS we have considered. Still, self-classifying clustering based CIDS achieves the least difference of 0.05%. In addition to that, our proposed system gives 8.35%, 4.51%, & 3.09% detection accuracy improvement during training and 8.52%, 4.46%, & 3.17% improvement during testing with reference to K means clustering based CIDS, modified K means clustering based CIDS, and fuzzy self constructing clustering based CIDS, respectively.

Table 2. Difference between training and testing.

Clustering	Error in Success rate
K Means[18]	0.16
Modified K Means[19]	0.12
Fuzzy Self Constructing[24]	0.05
Fuzzy Self Classifying	0.03

Table 2 gives the difference between the peak performances of the four clustering algorithms during training and testing phases in cloud intrusion detection. It shows that our proposed fuzzy self classifying clustering based CIDS achieves lower error among the four methods. Even though the improvement is small, it is an essential progress in the field of cloud intrusion detection which can identify new malicious activities. It reduces the possibility of permitting intrusive patterns to access cloud resources and guarantees that even a less-frequent intrusion cannot go undetected. Satisfactory results are obtained in detecting each of the four categories of attacks with 99.31% efficiency during training and 99.26% efficiency during testing. Literature reviews prove that this improvement could lead to a positive direction toward cloud intrusion detection by not allowing even a less-hazardous activity.

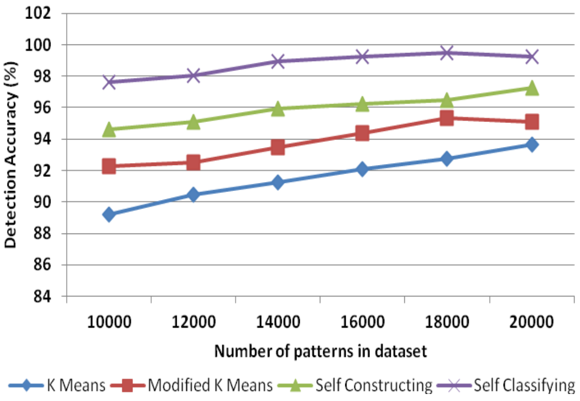

Fig. 3. Detection accuracy for varying number of patterns

The results of our experiments are further explained by Statistical analysis with ANOVA using Tukey post hoc method. The results of the ANOVA random effects model in terms of success rate accuracy for the four different cloud intrusion detection systems are presented in Fig 3 for varying number of patterns where our proposed system achieves significant increase in success rate accuracy than other systems. A one-way ANOVA was conducted to test for detection accuracy differences among the four types of clustering algorithms in cloud intrusion detection.

Table 3 presents the results of one-way ANOVA. There was a significant effect in the detection accuracy for varying number of patterns with [F(3,20) = 39.395 and p = 0.000] for the four methods at the p<0.05 level. This suggests that the four types of cloud intrusion detection systems are significantly different.

Table 3. Results of ANOVA between detection accuracies of fuzzy self classifying clustering based CIDS and other systems.

Analysis of results	Sum of squares	df	Mean square	F	P value	F_{crit}	% of contribution
Between systems	169.8572	3	56.6191	39.3950	1.39E-08	3.0983912 24	85.52
Within systems	28.7443	20	1.4372				14.47
Total	198.6015	23					99.99

Table 4. Multiple comparisons table for table 3

System 1	System 2	Mean Difference	SE	Tukey HSD Q Statistic	Tukey HSD p value	95% Confidence Interval		Tukey
						Lower bound	Upper bound	HSD inference
Modified K Means	K Means	2.2750	0.69215	4.6483	0.0177992	0.831201	3.718799	Significant
Self Constructing	K Means	4.380	0.69215	8.9493	0.0010053	2.936201	5.823799	Significant
Self Constructing	Modified K Means	2.1050	0.69215	4.3010	0.0301906	0.661201	3.548799	Significant
Self Classifying	K Means	7.2133	0.69215	14.7384	0.0010053	5.769534	8.657133	Significant
Self Classifying	Modified K Means	4.9383	0.69215	10.0901	0.0010053	3.494534	6.382133	Significant
Self Classifying	Self Constructing	2.8333	0.69215	5.7891	0.0029254	1.389534	4.277133	Significant

Table 5. Bonferroni and Holm table for table 3.

System	Bonferroni and Holm T-statistic	Bonferroni p-value	Bonferroni inference	Holm p-value	Holm inference
K Means[18]	10.4216	9.4367E-09	P<0.01	9.4367E-09	P<0.01
Modified K Means[19]	7.1348	3.9100E-06	P<0.01	3.2583E-06	P<0.01
Self Constructing[24]	4.0935	0.0033919	P<0.01	0.0016960	P<0.01

To determine further which pairs of the cloud intrusion detection systems are significantly different from each other, Tukey HSD multiple comparison test is conducted. The p-value of one-way ANOVA is lower than 0.01 which firmly suggests that one or more pairs of cloud intrusion detection systems are significantly different. We first set up the critical value based on the four systems and degrees of freedom for the residual, for significance level 0.01 and 0.05 (p-values) in the Studentized Range distribution. Then, the appropriate critical values of the Studentized Range distribution are compared to establish a Tukey test statistic. Table 4 presents the Multiple Comparisons between the detection ratios of our proposed system and other systems for different dataset size. Tukey post hoc comparisons of the protocols indicate that our system gave statistically significant detection ratio than other systems.

In table 5, we analyze pairs of cloud intrusion detection systems relative to fuzzy self-classifying clustering based CIDS for simultaneous comparison. This test is conducted to show the differences of methods relative to our proposed method. This observation strongly shows that the proposed system is significantly different from the other discussed systems based on the p-value. Table 6 presents the Scheffe p-value for the comparison between the self classifying clustering based CIDS and the other three systems. Based on the p-value, it is inferred that the proposed system is significantly different.

Table 6. Scheffe table for Table 3.

System	Scheffe T-statistic	Scheffe p-value	Scheffe inference
K Means[18]	10.4216	2.8356E-08	P<0.01
Modified K Means[19]	7.1348	1.0171E-05	P<0.01
Self Constructing[24]	4.0935	0.0059774	P<0.01

For the five class classification problem, figure 4 gives the detection accuracy values of our fuzzy self classifying clustering based cloud intrusion detection From these measurements which are obtained when the intrusion detection system is presented with the maximum number of patterns, it is shown that we got satisfactory results in detecting each of the four categories of attacks with 99.31% efficiency during training and 99.26% efficiency during testing.

T-S FNN based algorithm network intrusion detection is used in Ref. 4 which achieves the detection accuracy of 95.2%. In Ref. 5, a genetic algorithm based approach is used to detect intrusions. Results of this

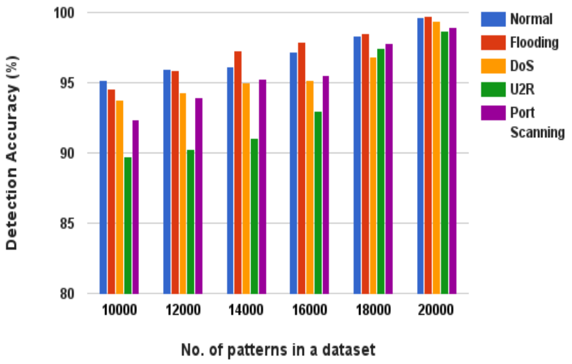

Fig. 4. Detection accuracy of fuzzy self classifying clustering based CIDS for various attacks

approach are compared by testing Minkowski distance function against Euclidean distance function. It is shown that the former achieves 82.13 % success rate and the later produces 81.76% only. A means to intrusions detection is recommended in Ref. 6 with a mining algorithm that gives better detection ratio. But it is compromised, when the system activities vary considerably. In Ref. 26, a data mining technique for intrusion detection is proposed which has speedy detection of new attacks. But its performance is limited by quality of training dataset. A control-theoretic approach is suggested in Ref. 27 for intrusion detection by means of distributed multiple nodes. Though it reduces the rate of false positives, its deficiency lies in generating automated response and in computational complexity. In Ref. 28, a data pre-processing methodology is proposed to expedite a hidden Markov model training for intrusion detection and it scales down training time by up to 50%. But it acquires performance deterioration in terms of false alarms. For the same intrusion detection problem, 79.89% is achieved in Ref. 29 by genetic algorithm (GA) and fuzzy Logic based method. 90.04% and 92.82% accuracies are shown in Refs. 30 and 31, respectively, by employing GA based intrusion detection approaches. With these comparisons, we show that our system gives significantly improved result of 99.26% detection accuracy in the field of cloud intrusion detection.

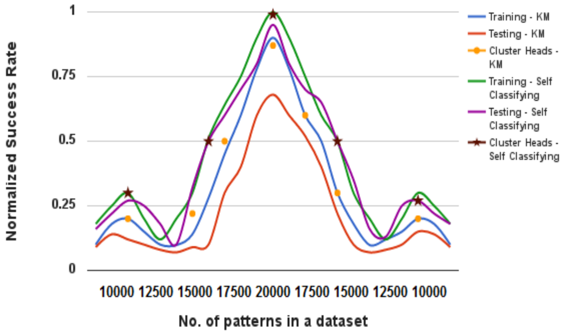

Fig. 5. Comparison between K Means Clustering based CIDS and Self Classifying Clustering based CIDS

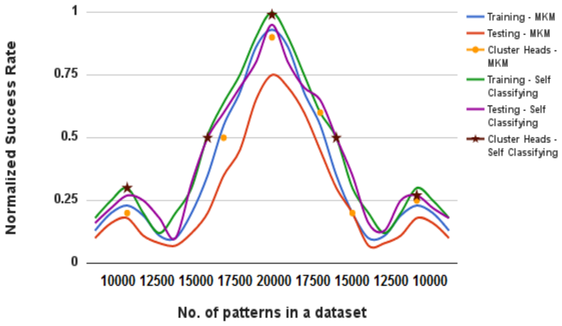

Fig. 6. Comparison between Modified K Means Clustering based CIDS and Self Classifying Clustering based CIDS

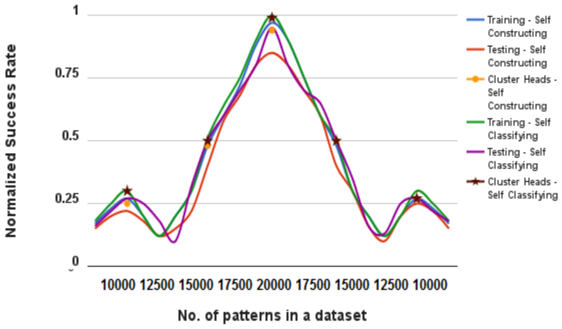

Fig. 7. Comparison between Self Constructing Clustering based CIDS and Self Classifying Clustering based CIDS

Further, the comparison between the training and testing accuracies of the fuzzy self classifying clustering based CIDS with that of the other three clustering algorithms based CIDS are shown in figures from 5 to 7 with the same set of randomly chosen sample patterns.

Figures 5 and 6 show that K means clustering based CIDS and Modified K means clustering based CIDS exhibit almost similar performance during training with respect to the number of cluster centers. Moreover, there exists significant deviation between the performances measured during training and testing. But, Modified K means clustering based CIDS achieves higher success rate during testing. Above all these inferences, the proposed fuzzy self classifying clustering based CIDS encounters noteworthy improvements in both training and testing stages than that of remaining systems. Figure 7 shows the comparison between the success rate of fuzzy self constructing clustering based CIDS and the proposed CIDS. Though these two systems offer similar performance during training phase, better success rate in testing phase is experienced in the application of proposed CIDS only.

From these figures, we see that the later gives higher success rate than all the remaining three models. Moreover, each of these systems except fuzzy self classifying clustering based CIDS have potential difference in their training and testing success rates. But fuzzy self classifying clustering based CIDS contributes no probable difference in the success rates of training and testing phases. Results show that the CIDS model based on fuzzy self-classifying clustering achieves better results than the remaining three models, even in the situation of least number of patterns considered.

5. Conclusion

In this paper, we have proposed the fuzzy self-classifying clustering algorithm to incorporate intrusion detection in a cloud environment. The results of our proposed approach are compared with other cloud intrusion detection systems based on K means, modified K means, and fuzzy self constructing clustering algorithms. Using each of these algorithms, the intrusion detection training dataset is partitioned into several clusters with similar patterns belonging to same cluster. Each of the resulting clusters is defined with the membership function by statistical mean and deviation which results in a type-2 fuzzy TSK-rule. A fuzzy neural network is constructed accordingly and the associated parameters are refined by a type-2 fuzzy neural network. For a new input from the test data set, a corresponding crisp output of the system is obtained by combining the inferred results of all the rules into a type-2 fuzzy set which is then defuzzified by applying a type reduction algorithm. The results are then compared. This method is repeated by varying the number of patterns in intrusion detection data set. Statistical analysis with ANOVA using Tukey post hoc method is carried out to demonstrate that the proposed method achieves statistically significant performance than other methods.

References

1. C. Modi, *et al.*, A survey of intrusion detection techniques in Cloud. *Journal of Network and Computer Applications*, 36, (2013) pp. 42-57.
2. D. Zissis and D. Lekkas, Addressing cloud computing security issues. *Future Generation Computer Systems*, Science Direct, Elsevier, (2012) pp. 583-592.
3. K. Hashizume, *et al.*, An analysis of security issues for cloud computing. *Journal of Internet Services and Applications*, 4(5) (2013).
4. H. Liang, An Improved Intrusion Detection based on Neural Network and Fuzzy Algorithm. *Journal of Networks*, 9(5) (2014) pp. 1274-1280.
5. A.S.A. Aziz, *et al.*, Artificial Immune System Inspired Intrusion Detection System Using Genetic Algorithm. *Informatica*, 36, (2012) pp. 347-357.
6. F. Zhao and H. Jin, Automated Approach to Intrusion Detection in VM-Based Dynamic Execution Environment. *Computing and Informatics*, 31, (2012) pp. 271-297.
7. H.-J. Liao, *et al.*, Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36, (2013) pp. 16-24.
8. A. Patel, *et al.*, An intrusion detection and prevention system in cloud computing: A systematic review. *Journal of Network and Computer Applications*, 36, (2013) pp. 25-41.
9. S. Gupta, *et al.*, A Profile Based Network Intrusion Detection and Prevention System for Securing Cloud Environment. *International Journal of Distributed Sensor Networks*, Article ID 364575 (2013).
10. F. Sabahi, Secure Virtualization for Cloud Environment Using Hypervisor-based Technology. *International Journal of Machine Learning and Computing*, 2(10) (2012).
11. L.M. Vaquero, *et al.*, Locking the sky: a survey on IaaS cloud security. *Computing*, 91(1) (2011) pp. 93-118.
12. S. Subashini and V. Kavitha, A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, Elsevier, 34(1) (2011) pp. 1-11.
13. S.P. Ahuja and D. Komathukattil D, A survey of the state of cloud security. *Netw. Commun. Technol.* 1(2) (2012) pp. 66-75.

14. J. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981).
15. T. Kohonen, The self-organizing map. *Proceedings of IEEE*, 78(9) (1990) pp. 1464-1480.
16. R. Yager R and D. Filev, Generation of fuzzy rules by mountain clustering. *Journal of Intelligent and Fuzzy Systems*, 2(3) (1994) pp. 209-219.
17. S. Chiu, Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, 2, (1994) pp. 267-278.
18. T. Kanungo, *et al.*, An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7) (2002) pp. 881-892.
19. S. Raja and S. Ramaiah, An Efficient Fuzzy-based Hybrid System to Cloud Intrusion Detection. *International Journal of Fuzzy Systems*. (2016) pp. 1-16.
20. J.H. Eom, and M.W. Park, Design of Internal Traffic Checkpoint of Security Checkpoint Model in the Cloud Computing, *International Journal of Security and Its Applications*, 7(1) (2013) pp. 119-128.
21. H.A. Kholidy and F. Baiardi, CIDD: A Cloud Intrusion Detection Dataset for Cloud Computing and Masquerade Attacks. *Ninth International Conference on Information Technology - New Generations*, Las Vegas, Nevada USA April 16-April 18, (2012).
22. S. Raja and S. Ramaiah, Performance Comparison of Neuro-Fuzzy Cloud Intrusion Detection systems. *The International Arab Journal of Information Technology*, 13(1A) (2016) pp. 142-149.
23. M.-C. Su and C.-H. Chou, A Modified Version of the K-Means Algorithm with a Distance Based on Cluster Symmetry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6) (2001) pp. 674-680.
24. C.-Y. Yeh, *et al.*, Data-Based System Modeling Using a Type-2 Fuzzy Neural Network with a Hybrid Learning Algorithm. *IEEE Transactions on Neural Networks*, 22(12) (2011) pp. 2296-2309.
25. Eucalyptus. http://www.eucalyptus.com.
26. R. Perdisci, *et al.*, Classification of Packed Executables for Accurate Computer Virus Detection. *Pattern Recognition Letters*, 29(14) (2008) pp. 1941-1946.
27. R. Khanna and H. Liu, Control Theoretic Approach to Intrusion Detection Using a Distributed Hidden Markov Model. *IEEE Wireless Communications*, 15(4) (2008) pp. 24-33.
28. J. Hu, *et al.*, A Simple and Efficient Hidden Markov Model Scheme for Host-Based Anomaly Intrusion Detection. *IEEE Network*, (2009) pp. 42-47.
29. M.M.M. Hassan, Network Intrusion Detection System using Genetic Algorithm and Fuzzy Logic, *International Journal of Innovative Research in Computer and Communication Engineering*, 1(7) (2013).
30. M.S. Hoque, *et al.*, An implementation of intrusion detection system using genetic algorithm, *International Journal of Network Security & Its Applications*, 4(2) (2012).
31. H.M. Shirazi and Y. Kalaji, An Intelligent Intrusion Detection System Using Genetic Algorithms and Features Selection, *Majlesi Journal of Electrical Engineering*, 4(1), (2010) pp. 33-43.