Intrusion Detection in the Cloud Environment Using Multi-Level Fuzzy Neural Networks

H. Akramifard[1], **L. Mohammad Khanli** [1], **M.A Balafar**[1], **R. Davtalab**[1]
[1] Faculty of Electrical and Computer Engineering, Tabriz University, Tabriz, East Azerbayejan, Iran

Abstract - *Today virtualization is one of last innovations in computer's world. Enterprises are attempting to reduce their computing cost using virtualization. Cloud computing is ultimate response to this request of the market. Growth in the number of companies, who want to employ cloud resources, turns the user's data protection into a significant issue. Concentration of this paper is on the security of enterprise's data by intrusion detection while employing cloud computing. The goal of this research is recognizing the security threats and introducing a security method to mitigate them in the cloud computing environment. The intrusion detection will be responsible for anomaly detection on the generated data from the collected transactions through the cloud. The captured data will be classified using Multi-Level Fuzzy Neural Networks to detect the appearance of intruders on the cloud computing network. This approach will consider different attributes of the data to investigate the user's behavior. The evaluations show Multi-Level Fuzzy Neural Networks have more efficiency and better accuracy in intrusion detection.*

Keywords: Cloud computing, security, classification, intrusion detection, anomaly-detection, Multi-level Fuzzy Neural Networks.

1 Introduction

Nowadays cloud computing [1] provides computing and data storage services through the Internet. The cloud computing has scalability, elasticity and speed, etc. The internet started to offer meaningful bandwidth in the nineties. Cloud computing is a general term for anything that involves delivering hosted services over the Internet and managed by the cloud service provider. The Cloud services allow businesses and people to use software and hardware infrastructure that are managed by third parties at remote locations. The cloud services include online massive computing, file storage, social networking sites, webmail, and online business applications. The cloud computing provides remote access to information and computer resources from anywhere that an internet connection is available. Figure 1 shows architecture and layers of a cloud computing environment.

Incr Increasing in amount of cloud users, raises the privacy and Security concerns. Data protection became the major issue as the user's data managed by a third party [2]. We have to design and implement Intrusion Detection Systems (IDS) to detect the malicious activity on a cloud environment that could detect intruders and generate the alarms at the occurrence of any illegitimate activity. The intrusion detection systems train with the both normal and malicious data.

Garcia-Teodoro et al. [3] divides anomaly detection techniques as below:

1. Statistical based
    a) Univariate
    b) Multivariate
    c) Time series model
2. Knowledge based
    a) Finite State Machines
    b) Description languages
    c) Expert systems
3. Machine learning based
    a) Bayesian networks
    b) Markov models
    c) Neural networks
    d) Fuzzy logic
    e) Genetic algorithms
    f) Clustering & outlier detection

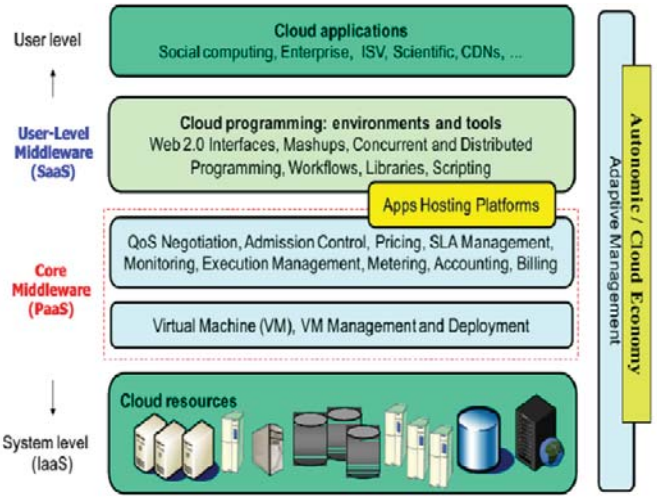

Fig. 1. Cloud Computing Architecture and its layers [4].

Section 2 represents the intrusion detection and related works in the cloud. In section 3, focus is on Multi-Level Fuzzy Neural Networks. An overview of the concepts of intrusion detection in cloud using MLF-NN is mentioned in section 4. In section 5 the proposed method and the evaluation are

explained. Finally the conclusion has been presented in section 6.

2 Intrusion Detection In Cloud

Recognizing malicious activities against the networking resources is known as intrusion detection. The recognition of any suspicious activity on the devices or networks is raised by an alert [5]. An Intrusion Detection System (IDS) in a cloud computing environment is for protecting each VM against the threat of malicious accesses. An Intrusion Detection System is a program that monitors the events at a machine or at a network automatically. It monitors the traffic at each machine, also monitors the network and makes records, to provide security to all the devices in the network [5], [6]. Environment of an IDS one of following groups [7]:

1) Host-based Intrusion Detection System:

It monitors a specific host to detect if any program accesses some resources, it acts like a firewall.

2) Network-based Intrusion Detection System:

It monitors the network packets for specific network segments or points to recognize any suspicious action.

2.1 Categories of IDS

Intruder identification is one of the basic IDS operations [8]. The two main identified methods of IDS [5], [6] are as bellow:

1)Misuse-based Detection:

A misuse-based Intrusion Detection System stores signatures depicting attacks into a database. Signature of such attacks widely used systems where security threats are common. The pattern (signature) based IDS performs a depth inspection of the packets, for any spiteful patterns in the load or header.

2)Anomaly-based Detection:

An anomaly-based Intrusion Detection System protects a statistical model of custom patterns, patterns that describe the normal behavior of monitored users [5]. At the first training stage of this Intrusion Detection System, a similarity metric is used to compare an input with the normal model, then generates alerts for large deviation values.

Misuse-based IDS look like efficient and effective, but it shows two main problematic conditions, one, mistakes in detection of unknown attacks [9], [10], and second, pattern analysis defect. The first is due to the fact that misuse-based IDS relies on string comparison of previous attack patterns [11], thus the unknown attacks can show deviation from the comparison string to already known attacks, and thus are ignored from being detected, that is false negatives [12], and second, misuse-based IDSs have weakness in pattern analysis, and in rule writing methods as to capture all the defenselessness of attacks it mainly relies on the human ability.

We distinguish misuse and anomaly based principles mainly in the way of modeling of their behavior and way of defining of their normalcy. Those two method specify how is further processing of observed data too [13], [14].

Here is a short history of some related work about intrusion detection in literature:

Massimo Meneganti et al. used fuzzy logic for classification and detection of anomalies firstly in 1998 [15]. They utilized fuzzy neural networks to find anomalies in the cooling system of a blast furnace.

Pei-Te Chen et al. proposed the concept of security auditors, to discover the system weaknesses and modify the tested packets using fingerprints that can be detected and recognized by IDS in 2007, [16].

Jun-Ho Lee et al. proposed a multi-level method for IDS [17] in cloud computing system in February 2011. In their method all the users were bound to a security system on the basis of anomaly status. The system decides about anomalies on user's IP coverage, amount of ID/password failures, vulnerable ports, and etc. In June 2011, an intrusion detection model baseed on anomaly in an environment of SaaS application was presented by Gustavo Nascimento et al. [5].

Chirag N. Modi et al. integrated a Network based IDS system in Cloud that offered IaaS to detect network attacks in July 2012 [18]. In this system, they used Bayesian classification method with Snort. This module guarantees low false positives and negatives with acceptable cost. Ajeet Kumar Gautam et al. proposed a hybrid intrusion detection system in cloud computing, they used KFSensor and anomaly based IDS via FlowMatrix with honeypot technology, in 2012 [6]. They designed an architecture by providing and detecting various attacks. In September 2012, Amirreza et al. introduced a Cloud Intrusion Detection System Service (CIDSS) [19] to overcome the crucial challenge of securing the client from cyber-attacks. Three primary components of CIDSS:

1) A Service Agent for Intrusion Detection.
2) A Service Component (CCSC).
3) An Intrusion Detection Service Component (IDSC) that were used to incorporate information and after that test them.

In 2013, Ahmed Patel et al. proposed a model of Intrusion Detection and Prevention system (IDPS) in cloud computing [7]. The concepts of fuzzy theory, autonomic computing, risk management, and ontology were grabbed and combined acknowledge the requirements of an IDS. In that year P. Gupta et al. proposed behavior based IDS [20], the implementation was instructed in a real cloud IaaS environment. The framework was tested with NIDS to detect network based attacks.

In 2014, Harshit Saxena et al. proposed an intrusion detection system using K-means, PSO with SVM classifier [21] to detect various attacks at network. They has tried to design an IDS that is trained on the basis of Particle Swarm Optimization, executed on the KDD data.

3 Multi-Level Fuzzy Neural Networks

Here we will introduce two efficient types of fuzzy neural network, Fuzzy min-max neural network (FMM) and Multi-Level Fuzzy Min-Max Neural Network (MLF).

3.1 Fuzzy min-max neural network

Fuzzy min-max neural network (FMM) is a machine learning method that has been proposed by Simpson in 1992 [22]. It can be used for data classification. The learning phase includes only one pass over the learning data. In this method we use convex hyperboxes in the pattern space. Each hyperbox is determined by MN and MX points which, respectively, mention the min and max points of the hyperbox. A three-dimensional hyperbox has been shown in figure 1.

Each hyperbox covers a part of pattern space, and belongs to only one of the classes but can include more than one sample of that class, defined as (1) [23].

$$B_i = \{X, MN_j, MX_j, f(X, MN_j, MX_j)\} \qquad \forall X \in I^n \quad (1)$$

Where MN_j and MX_j are min and max corners of the hyperbox. X is an input vector, and n mention number of dimensions. Each class may have one hyperbox or more. Hyperboxes of the same class could overlap each other, but hyperboxes from different classes couldn't. Final hyperboxes of an example of FMM network in a 2-D binary classification have been shown in figure 2.

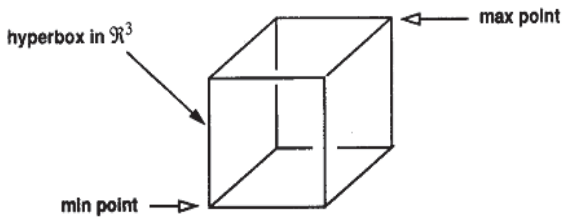

Figure 2. 3-D hyperbox and its min and max points [23].

Fuzzy set, that inputs classes belong to, includes union of hyperboxes of those classes. In the test stage, these hyperboxes and their membership function are used to detemine the classes. In this method, the size of hyperboxes is in range of [0, 1]. One of the possible membership functions is Simpson function is shown at (2):

$$b_j(X_h) = \frac{1}{2n}\sum_{i=1}^{n}\left[max\left(0, 1 - max\left(0, \gamma\, min\left(1, a_{hi} - mx_{j\,i}\right)\right)\right) + max\left(0, 1 - max\left(0, \gamma\, min\left(1, mn_{ji} - a_{hi}\right)\right)\right)\right] \quad (2)$$

Where $\boldsymbol{X_h} = (x_{h1}, x_{h2}, \ldots, x_{hn}) \in I^n$ is the *h*th sample and γ is in range of [0, 1] that determine how fast the membership values decrease as the distance between X_h and B_j increases. Figure 3 illustrate an example of two-classes fuzzy min-max hyperboxes, without overlapping between the classes.

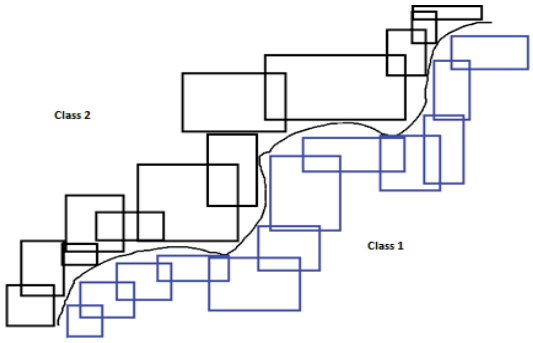

Figure 3. Final hyperboxes [23].

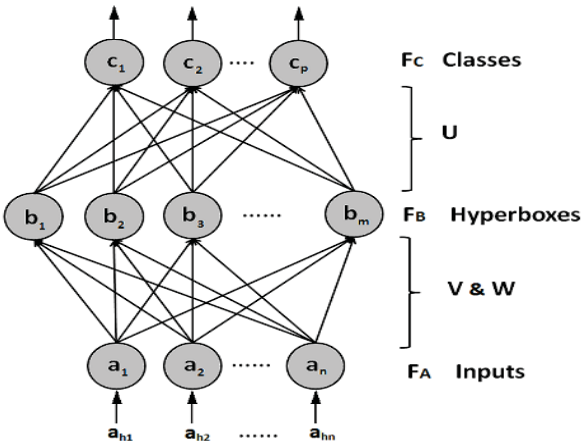

Figure 4. Structure of the classic FMM [23].

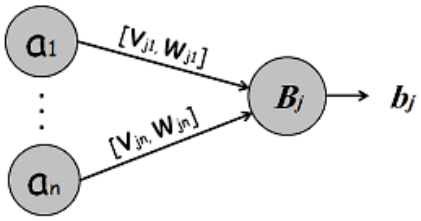

Figure 5. Details of a hyperbox [24].

FMM neural networks have three layers as shown in Figure 4, the first, input layer (FA), the second layer represents hyperboxes (FB), and third layer represents classes of each node (FC). Also each hyperbox locate in middle layer (FB), and the membership function of this hyperbox is the transition function of the Correspond node. Figure 5 demonstrate a hyperbox in details. Each node of input layer is connected to all nodes in the middle layer and each of these links has two weights (*Vji* and *Wji*), which are, respectively, the min and max points of the B_j hyperbox, and *i* is the index of the nodes in the first layer. Each node of the middle layer is also connected to all nodes in the output layer. Weights of those links are obtained from (3), and FC nodes outputs are provided by (2):

$$u_{ij} = \begin{cases} 1, & if\ b_j \in C_i \\ 0, & if\ b_j \notin C_i \end{cases} \quad (3)$$

All hyperboxes are created and adjusted in the learning step. The learning phase has three parts. Existance of a box that belongs to the same class and simultaneously the sample is in box area, will be checked for per sample (A_h). If a box is found, then no further processing is required and training goes on with the next sample. If there is no such hyperbox, following three steps are executed [24].

1. Expansion: In this stage, a hyperbox must be found to display the related class and also be capable of expansion to cover the input sample, the hyperbox size is limited to the θ parameter. If no such hyperbox is found, a new hyperbox is created with min and max points, relevant to this sample.
2. Overlap Test: In this step, the overlapping area of the extended hyperbox will check for all hyperboxes that belong to the other classes. In one case of (4), we can find overlap of two hyperboxes, after recognizing each dimension. To eliminate this overlap, the dimension Δ that has the least overlap will be selected for contraction.

Case 1: $v_{ji} < v_{ki} < w_{ji} < w_{ki}$

Case 2: $v_{ki} < v_{ji} < w_{ki} < w_{ji}$

Case 3: $v_{ji} < v_{ki} < w_{ki} < w_{ji}$

Case 4: $v_{ki} < v_{ji} < w_{ji} < w_{ki}$ (4)

For example of case 1: Min of 1st box (v_{ji}) less than min of 2nd box (v_{ki}), min of 2nd box (v_{ki}) less than max of 1st box (w_{ji}), max of 1st box (w_{ji}) less than max of 2nd box (w_{ki}), figure 6 shows the visualization of this case.

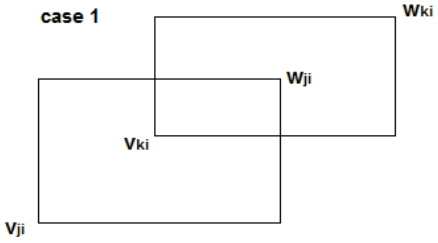

Figure 6. Illustration of contraction for case 1.

3. Contraction: If there is no overlap, this step is not necessary; else, considering the type of the overlap according to (4), one case of (5) will be executed.

Case 1: $v_{ji} < v_{ki} < w_{ji} < w_{ki}$

$v_{k\Delta}^{new} = w_{k\Delta}^{new} = \frac{v_{k\Delta}^{old} + v_{j\Delta}^{old}}{2}$ OR

$w_{j\Delta}^{new} = v_{k\Delta}^{old}$

Case 2: $v_{ki} < v_{ji} < w_{ki} < w_{ji}$

$v_{j\Delta}^{new} = w_{k\Delta}^{new} = \frac{v_{j\Delta}^{old} + v_{k\Delta}^{old}}{2}$ OR

$v_{j\Delta}^{new} = w_{k\Delta}^{old}$

Case 3: $v_{ji} < v_{ki} < w_{ki} < w_{ji}$

If $w_{k\Delta} - v_{j\Delta} < w_{k\Delta} - v_{j\Delta}$ Then

$v_{j\Delta}^{new} = w_{k\Delta}^{old}$

Else

$w_{j\Delta}^{new} = v_{k\Delta}^{old}$

Case 4: $v_{ki} < v_{ji} < w_{ji} < w_{ki}$

If $w_{k\Delta} - v_{j\Delta} < w_{j\Delta} - v_{k\Delta}$ Then

$w_{k\Delta}^{new} = v_{j\Delta}^{old}$

Else

$v_{k\Delta}^{new} = w_{j\Delta}^{old}$ (5)

Here, Δ denotes the selected dimension. These three steps are executed on every learning sample to obtain the required hyperboxes.

3.2 Multi-Level Fuzzy Min-Max Neural Network

In this article we will use multi-level fuzzy min-max neural network for IDS classification. This type of neural networks tries to better cover area of classes using more precise and smaller hyperboxes. Despite of classic FMM method, the contraction step do not handle the overlaps. The manner of MLF method is creation of hyperboxes in the first and the second levels, and the classification task are illustrated in figure 7.

Each node in the network of the MLF method is known as a subnet and is an independent classifier that classifies samples that belong to the defined region of pattern space. The first level classifier classify most of the region of pattern space, and the second level nodes take care of the remaining regions that are the same overlapped region of root subnet, as well each node in the ith level of the network classifies patterns of overlapped region in i−1th level of the network. Finally, the node that has the best output will select as the network's output.

In MLF, like in other FMM methods, all hyperboxes are created and adjusted during training phase and are used in test phase. FMM method handle overlap problem step by step just when an overlap is created; but in MLF overlap handling is done after creation and adjustment of all hyperboxes. This can reduce space and time complexity.

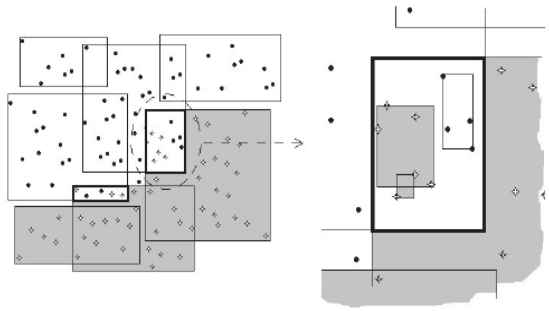

Figure 7. Different levels of classification in MLF [24].

4 Intrusion Detection In Cloud Using MLF-NN

For intrusion detection in this paper we proposed the concept of Multi-Level Fuzzy Min-Max Neural Network algorithm. Using MLF-NN we will classify criminal activities like unauthorized access and change in behavior of the user. This is accomplished by using data in the database. The algorithm design will be as follows:

START

1) Obtaining topology of devices on cloud environment.
2) Obtaining data set of the transaction through the running virtual machines.
3) Select the data to be investigated and normalize it on the basis of various attributes.
4) Training the MLF-NN using obtained dataset.
5) If classifier recognizes the activity as an attack, generate an alarm, else the user is genuine and no intrusion is detected.

END

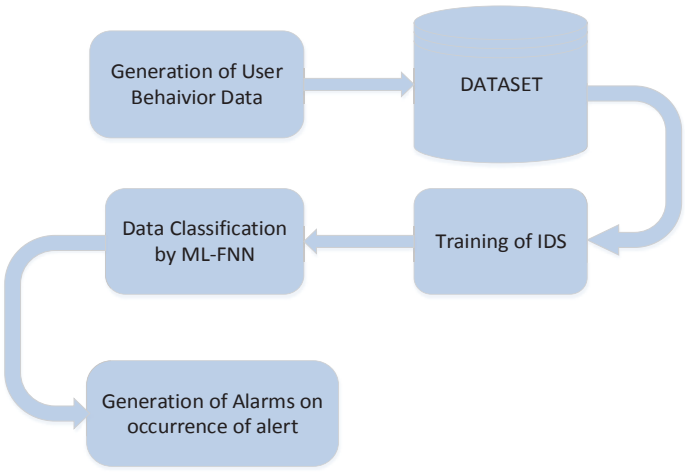

Figure 8. Architecture of intrusion detection using ML-FNN.

The proposed method comprises of three major steps of intrusion detection:

Step One: Data Generating

In this step a topology will be configured on an emulator. It lets the network to act on a virtual machine with the cloud environment. The operations will be logged for further observations.

Step Two: Dataset Making

In the second step generated data will be gathered. In this phase we will train the proposed IDS with the user behavior. The attributes types of the user's behavior in the cloud network are:

1. Basic features of individual TCP connections.
   a) Duration
   b) Protocol type
   c) Same host or not
   d) Number of data bytes from destination to source
   e) Number of data bytes from source to destination
2. Content features within a connection suggested by domain knowledge.
   a) Number of failed login attempts
   b) Success of login
   c) Number of "compromised" conditions
   d) Number of file creation operations
   e) Number of shell prompts
3. Traffic features computed using a two-second time window.
   a) Number of connections to the same host as the current connection in the past two seconds
   b) "SYN" errors
   c) "REJ" errors
   d) Number of connections to the same service as the current connection in the past two seconds
   e) Connections to different hosts
   f) Percent of connections to the current host having the same src port
   g) Percent of connections to the same service coming from different hosts
   h) Percent of connections to the current host that have an S0 error
   i) Percent of connections to the current host and specified service that have an S0 error
   j) Percent of connections to the current host that have an RST error
   k) Percent of connections to the current host and specified service that have an RST error

Step Three: Detecting

The final step will be the data analysis step where the data will be classified. Classifying is done with MLF-NN over a dataset. The result includes 21 attributes and two classes.

5 Evaluation Of Intrusion Detection Using MLF-NN

In this article the experimental data are from KDD dataset. We randomly select two groups without overlap from the data set; respectively denote them as INP and OUTP. INP uses to training and OUTP uses to test the model. There are intruders and normal users in the data set, we simulate the behavior of these two types of users for validating the ability of the model to identify the two types of users. Risk users' behaviors are normal in most cases, but they may be abnormal in some moments. The experiment simulate the behaviors by sending a large number of HTTP requests at a time, their behavior is similar to malicious users, but their attack time length is short.

We analyze the users' behaviors based on the data of INP obtain the evidences of users' behaviors. Behavior evidence including:

- ✓ Environmental attributes, such as network throughput, transmission delay, and IP loss ratio.
- ✓ Operational attributes, such as number of hits, pages accessed, important pages accessed, and time on page.

The environmental attributes principally used to determine the safety of the user's network environment. But operational attributes are mainly used to conclude consistency of user's behavior with his habits. About 95% of the users' behaviors are concentrated in the stable range [25]. User behavior hierarchical structure is shown in figure 9.

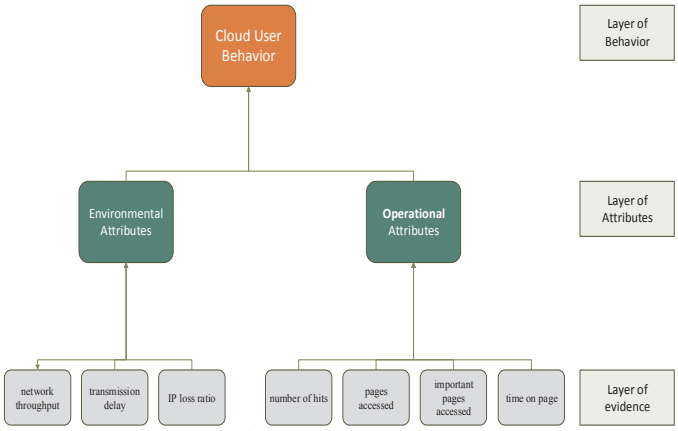

Figure 9. User behavior hierarchical structure [25].

The model have been tested by data set OUTP using MLF-NN and compare the model with the other classification methods. There are 25973 samples in OUTP, and the number of sample of attacks are 12075 and normal behaviors are 13898.

In anomaly detection, True Positive or Detection Ratio (DR) and False Positive Ratio (FPR) are two essential metrics. Here, the DR mainly mentions the amount of detected intrusions, and FPR mentions the false positives of recognized users as intruder. The DR and the FPR of the model using this model and comparing it with other classification methods, have been shown in figure 10 and figure 11.

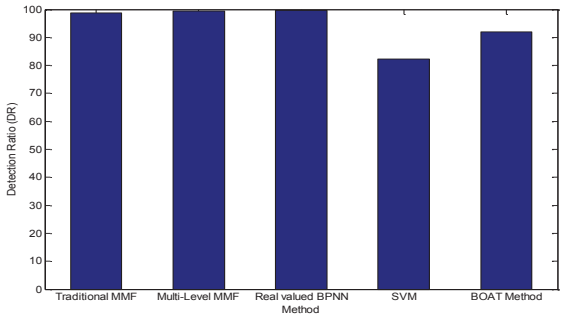

Figure 10. The DR of the four method.

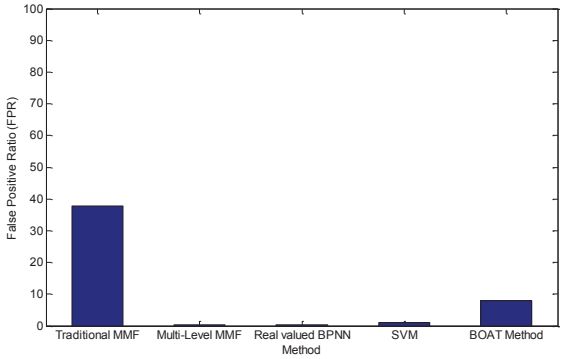

Figure 11. The FPR of the five methods.

As has been shown in figure 10 the Real valued BP-NN and MLF-NN respectively have 99.64% and 99.60% accuracy in DR and as has been figure 11 two above classification methods have 0.4% and 0.38% FPR, it shows the accuracy of the proposed model is better than other models like Bootstrapped Optimistic Algorithm for Tree Construction (BOAT) method came out to be 92.02% DR and 7.98% FPR.

Despite of near optimal ratio in detection and false positive, we chose MLF-NN over Real valued BP-NN because, because if there is some new data and if we want to train the network in Real valued BP-NN we must train all network, but using MLF-NN we can learn only new data to the network without changing all previous network trained data. Figure 12 has been represented comparation the Accuracy, Precision, Recal, and F-Score between four classification methods on proposed model. As we can see MLF-NN has the best result in comparision with other methods.

6 Conclusion

In this paper, we have presented a solution that detects malicious activities that masquerade in the system with the aim of violating the information. This solution uses MLF-NN to learn the behavior pattern of the user to detect malicious user in the system. The proposed solution proved to be effective in terms of reducing false positives rate and false negatives rate. The reduction of false positive and false negative rate indicates that, there is increasing in detection rate of intrusions. The results show that malicious users can be detected based on their behavior patterns.

7 References

[1] Anthony T. Velte, Toby J. Velte, Robert Elsenpeter, "Cloud Computing – A Practical Approach", Tata McGrawHill Edition, ISBN: 978-0-07-162695-8.

[2] Mell, Peter, and Tim Grance. "Effectively and securely using the cloud computing paradigm." NIST, Information Technology Lab 2009.

[3] P Garcia-Teodoro, J Diaz-Verdejo, G Macia-Fernandez, and E Vazquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28 (1 -2):18-28, 2009. doi: 10.1016/j.cose.2008.08.003.

[4] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, Rajkumar Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms", Software: Practice and Experience, Volume 41, Issue 1, pp. 23–50, January 2011.

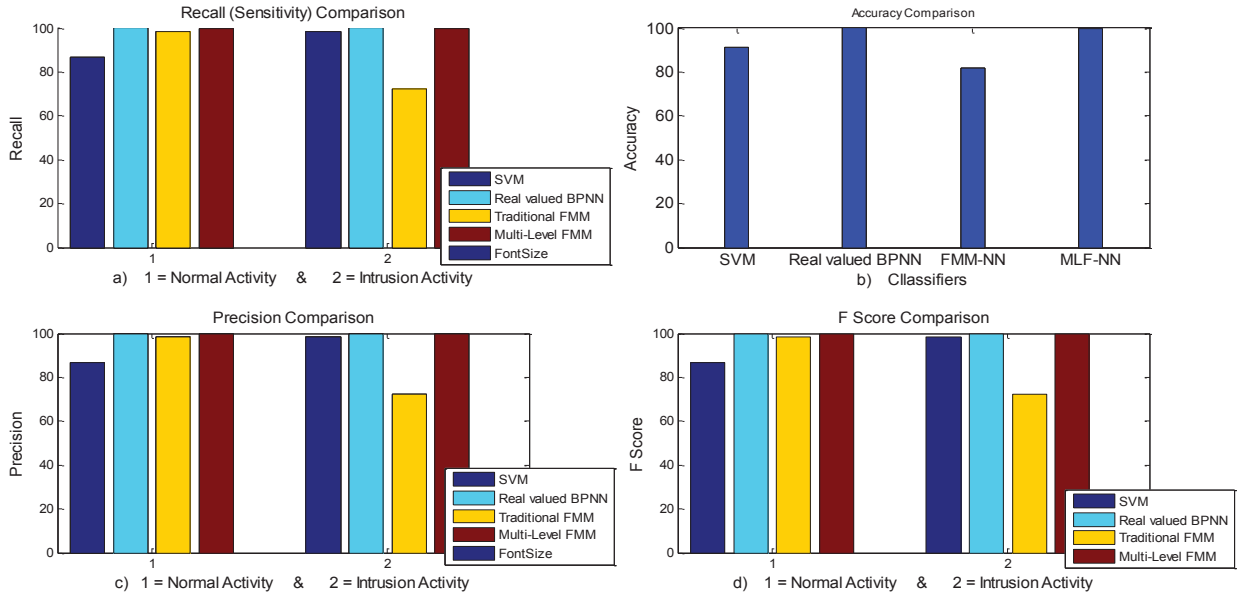

Figure 12. comparision between methods: a)Recal, b)Accuracy c)Precision d)F-Score.

[5] Nascimento, G., Correia, M., "Anomaly-based intrusion detection in software as a service", Dependable Systems and Networks Workshops (DSN-W), IEEE/IFIP 41st International Conference on, pp.19-24, June 2011.

[6] Ajeet Kumar Gautam, Vidushi Sharma, Shiva Prakash, "An Improved Hybrid Intrusion Detection System in Cloud Computing", International Journal of Computer Applications, Volume 53– No.6, pp. 1-13, September 2012.

[7] Ahmed Patel, Mona Taghavi, Kaveh Bakhtiyari and Joaquim Celestino Júnior, "An Intrusion Detection And Prevention System In Cloud Computing: A Systematic Review", Journal of Network and Computer Applications. Volume 36, Issue 1, pp. 25 -41, January 2013.

[8] Ms Deepavali P Patil, Prof.Archana C.Lomte, "Implementation of Intrusion Detection System for Cloud Computing", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 11, November 2013.

[9] David J. Day, Denys A. Flores, Harjinder Singh Lallie, "CONDOR: A Hybrid IDS to Offer Improved Intrusion Detection", IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 931-936, 2012.

[10] Choudhury, A.J.; Kumar, P.; Sain, M.; Hyotaek Lim; Hoon Jae-Lee, "A Strong User Authentication Framework for Cloud Computing", Services Computing Conference (APSCC), IEEE Asia-Pacific, pp.110-115, December 2011.

[11] Tupakula, U, Varadharajan, V., Akku, N., "Intrusion Detection Techniques for Infrastructure as a Service Cloud", Dependable, Autonomic and Secure Computing (DASC), IEEE Ninth International Conference on , pp.744-751, December 2011.

[12] Hari Om, Aritra Kundu, "A Hybrid System for Reducing the False Alarm Rate of Anomaly Intrusion Detection System", In Proceedings of 1st Int'l Conf. on Recent Advances in Information Technology (RAIT-2012),IEEE, pp. 131-136, 2012.

[13] Herve Debar, Marc Dacier, and Andreas Wespi. Towards a taxonomy of intrusion-detection systems. Computer Networks, (October 1998), 1999. Anita K. Jones and Rrobert S. Sielken. Computer system intrusion detection: A survey. Computer Science Technical Report, pages 1- 25, 2000.

[14] Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. Proceedings of the . . . , pages 25-36, 2003.

[15] Massimo Meneganti, Francesco S. Saviello, and Roberto Tagliaferri, "Fuzzy Neural Networks for Classification and Detection of Anomalies", IEEE transactions on neural networks, vol. 9, no. 5, pp. 848-861 september 1998.

[16] Pei-Te Chen, Chi-Sung Laih, "IDSIC: an intrusion detection system with identi fication capability", Springer-Verlag, pp.185-197, June 2007.

[17] Jun-Ho Lee; Min-Woo Park; Jung-Ho Eom; Tai-Myoung Chung, "Multi-level Intrusion Detection System and log management in Cloud Computing", Advanced Communication Technology (ICACT), 13th International Conference on , pp.552-555, February 2011.

[18] Chirag N. Modil, Dhiren R. Patell, Avi Patel, Rajarajan Muttukrishnan, "Bayesian Classifier and Snort based Network Intrusion Detection System in Cloud Computing", Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on, pp. 1-7, July 2012.

[19] Amirreza Zarrabi and Alireza Zarrabi, "Internet Intrusion Detection System Service in Cloud", International Journal of Computer Science Issues, Vol. 9, Issue 5, No. 2, pp. 308-315, September 2012.

[20] Punit Gupta, Deepika Agrawal, "Behavior Based IDS for Cloud IaaS", International Journal of Software and Web Sciences (IJSWS), pp. 31-36, June-August 2013.

[21] Harshit Saxena, Dr. Vineet Richariya, "Intrusion Detection System using K- means, PSO with SVM Classifier: A Survey", International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 2, pp. 653-657, February 2014.

[22] A. Joshi, N. Ramakrishman, E. N. Houstis, and J. R. Rice, "On neurobiological, neuro-fuzzy, machine learning, and statistical pattern recognition techniques," IEEE Trans. Neural Netw., vol. 8, no. 1, pp. 18–31, Jan. 1997.

[23] P. K. Simpson, "Fuzzy min-max neural networks. I. Classification," IEEE Trans. Neural Netw., vol. 3, no. 5, pp. 776–786, Sep. 1992.

[24] Reza Davtalab, Mir Hossein Dezfoulian, and Muharram Mansoorizadeh, "Multi-Level Fuzzy Min-Max Neural Network Classifier", IEEE transactions on neural networks and learning systems, vol. 25, no. 3, march 2014.

[25] Tian Junfeng, Cao Xun "A Cloud User Behavior Authentication Model Based On Multi-partite Graphs", IEEE, Innovative Computing Technology (INTECH), 2013 Third International Conference on , pp. 106-112, 2013.