Propose Multi level Network Intrusion Detection System to detect intrusion in Cloud Environment

Shawq malik Mehibes* Soukaena H. Hashim*,Ph.D.(Ass.prof.)

Abstract

Cloud computing is one of the popular technologies, which can used by most organizations because of its attractive properties such as availability, flexibility, integrity. The open and distributed structure of Cloud Computing and the services provided by it make it attractive aim for potential cyber-attacks by intruders. Network intrusion detection system (NIDS) represents important security mechanism, provides defence layer which monitors network traffic to detect suspicious activity and policy violations. This work proposed Multi-level-NIDS to detect intrusions and the type of intrusion in traditional/Cloud network. The proposed system evaluated with kdd99 dataset, the experimental results shows the efficiency and capability of the proposed system in detect attack and type of attack.

Keywords: Cloud Computing, Network Intrusion Detection, Data mining, Fuzzy C-mean algorithm, Back Propagation algorithm.

* University of Technology

1. Introduction

Nowadays, Cloud Computing (CC) represents a revolution in Information Technology (IT) where computing resource, storage and other services are offered over the internet. CC ensures data and service availability, and fast accessibility and possibility of expanding [1]. According to National Institute of Standards and Technology (NIST) Cloud Computing (CC) is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [2]. Cloud Computing is experiencing conventional attacks like Address Resolution Protocol spoofing, IP spoofing, DNS poisoning, Flooding, Routing Information Protocol attacks, Denial of Service (DoS), Distributed Denial of Service (DDoS), etc. [3]. So that, it is important to deploy intrusion detection for Cloud Computing. Intrusion Detection System (IDS) is an important part of network security used to maintain system availability and data integrity [4]. Intrusion Detection System (IDS) is an important security solution which monitors computer system to detect suspicious activity and volatiles policy. NIDS identify suspicious activity through monitor network traffic for certain network segments and analysis of the information obtained to detect potential intrusion. IDS is based on data mining techniques which increase the accuracy and efficiency notably with altitude. It can help in revealing new intrusions and policy violations, promoting decision support for intrusion management, and discovers behaviour patterns of attackers that were unknown previously [5]. FCM clustering algorithm is used in IDS; there are several attributes which make fuzzy systems suitable for IDS and can be explained as follows; the ability of Fuzzy systems to combine inputs from different sources. There is a class of intrusions which is not clearly recognized (e.g. at what threshold should an alarm be set?). Alarm rates that might happen with intrusion are usually unknown [6]. Back Propagation (BP) artificial neural network has the ability to solve several problems confronting the other present techniques used in intrusion detection. There are three advantages of intrusion detection based on neural network; Neural network provides elasticity in intrusion detection process, where the neural network has the ability to analyse and ensure that data right or partially right. Likewise, neural network is capable

of perform analysis on data in nonlinear fashion. Neural network has the ability to process data from a number of sources in a non-linear fashion. This is very important especially when coordinated attack by multiple attackers is conducted against the network. Neural networks are characterized by high speed in processing data [7]. The proposal trend is to build multilevel NIDS using FCM and BP as consequent phases supported by Information Gain (IG) as a feature selection. Training and testing of the proposed NIDS are under KDD99 dataset which is popular dataset that has ever been used in the intrusion detection field. The rest of paper is organized as the following: in section 2 we present the previous related work, in section 3 the KDD cup 99 dataset is included and described in addition to pre-processing operations, in section 4 feature selection algorithm illustrate, in section 5 Fuzzy C-Mean algorithm explain in details, in section 6 Back Propagation algorithm explain in details, section 7 evaluation performance is illustrated , in section 8 the proposed Multi-level-NIDS are illustrated in details, in section 9 the experiments and results illustrated, finally the conclusion.

2. Related Work

1. Deshmukh V. G. et al., 2013 [4], proposed FC-ANN approach for intrusion detection in cloud environment. The proposed intrusion detection model in the first places the training dataset is divided into homogenous subgroups via fuzzy clustering technique. The resulting subsets are used to train ANN learning algorithms. Then emulation of the ANN models to reduce the error is used by using the whole dataset. The membership grades are created using fuzzy clustering model used to merge the results. Finally, New ANN is trained using the merge results.

2. Khazaee S. et al., 2013 [8], proposed intrusion detection based on Fuzzy-ARTMAP neural network and used FCM algorithm as prepossessing step. The FCM algorithm is used to cluster the training data and separate the inappropriate sample from the training set. In this approach the sampling with membership smaller than 0.5 well moved to new set called inappropriate data1. Then after clustering the sampling that does not match the clusters is moved to new set called inappropriate data2. Then both inappropriate data1 and inappropriate

data2 class labels change to abnormal. Then the Fuzzy-ARTMAP neural network is used as classifier.

3. Modi C**.** et al., 2013 [3], developed network intrusion detection system for cloud computing system. The authors integrate the snort signature network intrusion detection with decision tree classifiers, where packet capture module is used to capture network traffic where it resides. Snort is used to detect known attack by matching between the captured packets with rules stored in knowledge base. If the snort decides that the packet is normal traffic then decision tree classifiers are used to detect unknown attack after preprocessing the packet. The system ensures low false positive with reasonable price and high accuracy.

4. Selman A. H. et al., 2013 [9], proposed intrusion detection using Neural Network Committee Machine. The system uses principle component analysis algorithm for feature reduction. Intrusion detection consists of three back propagation algorithm where the dataset divided to three subsets. Voting scheme is used to detect the output of the system. The system update is offline as soon as the threshold of recorded unknown packets reached, where the system retrains the neural networks with unknown new attacks. The classification rate of the system is equal to 99.8% and false positive and false negative equal to 0.1%.

5. kumarl S. et al., 2014 [10], proposed Multi-layer perceptron neural network which increases the performance of intrusion detection . The proposed algorithm uses Gradient descent with momentum back propagation algorithm for learning process. The proposed work uses KDD99 dataset to train and test the proposed algorithm, the accuracy of the proposed system for binary class is equal to 93% and 91.9% for attack class.

3. Dataset Description

The KDD cup 99 was popular dataset used for evaluating intrusion detection algorithms. This dataset consists of TCP connections, each connection has 41 features with a label to determine the class of a connection whether normal connection or class of attack connection. The feature of dataset is divided to numeric and symbolic features, classified

into the following four categories (Basic features, Content features, Time-based traffic features, Time-based traffic features). Attack class is classified to four main categories [5]:

- Denial of Service (DOS) attacks: The attacker attempts to make the system resource occupied to prevent the legitimate user from using the system.
- Probe attack: The attacker scans the network to collect information and find fragility. Then he uses this fragility to attack at later time.
- Remote-to-Local (R2L) attack: Hacker sends packets to the victim machine through network. After that he exploits fragility to get unauthorized local access to that machine.
- User-to-Root (U2R) attack: In this attack, hacker at first gets access to normal user then exploits fragility in the system to get root level access. The aim of this attack is to get illegitimate super-user privileges.

The KDD cup 99 datasets consist of training and testing datasets see table(1). There are 4,940,000 data sample in the training set, these samples are distributed between normal behaviour and 24 attacks. On the other hand, there are 311,029 data samples which include normal network traffic and 38 class of attack, 24 attack existing in training set besides 14 new attack. As the training set contains large number of data samples, other training sets formed include 10% of data samples used in wide range.

Table (1): Number of samples in KDD cup 99

dataset	normal	Dos	Probe	U2R	R2L	Total
corrected KDD 99"	60593	229853	4166	70	1126	311029
"10% KDD	97277	391458	4107	52	1126	494020

Standardization of the dataset has been done so that it would be appropriate to be used by the proposed algorithm. The following steps show the standardization process:

A. Convert the value of symbolic feature to sequential integer value from [1….N], which are the three types of protocols (tcp, udp, icmp), 68 types of service, and 11 types of flag in KDD cup 99 dataset. Since FCM algorithm and BP algorithm accept numerical value.

B. Data normalization to avoid the bias problem some larger features values can be caused. This leads to improve the efficiency and the accuracy of mining algorithms, because these algorithms provide better result when the data to be analysed fall between [0 and 1]. Min-max normalization method which is a linear transformation is used to scale data between [0 and 1] .The following formula is used to find the new value [11]:

$$X_n = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (1)$$

4. Feature selection

In this work Information Gain used as feature selection algorithm. The IG evaluates attributes by measuring their information gain with respect to the class. Let C be a set consisting of c data samples with m distinct classes. The training dataset C_i contains sample of class I. Expected information needed to classify a given sample is calculated by [12]:

$$\mathrm{I}(c_1, c_2, c_m) = -\sum_{i=1}^{m} \frac{c_i}{c} \log_2 \left(\frac{c_i}{c}\right) \qquad \text{Eq. (2)}$$

Where $\frac{c_i}{c}$ is the probability that an arbitrary sample which belongs to class C_i. Let feature F have v distinct values $\{f_1, f_2, \ldots, f_v\}$ *which* can divide the training set into v subsets$\{C_1, C_2, \ldots, C_v\}$, where C_i is the subset which has the value f_i for feature F. Let C_j contain C_{ij} samples of class i. The entropy of the feature F is given by

$$\mathrm{E(F)} = \sum_{j=1}^{v} \frac{C_{ij} + \cdots + C_{mj}}{C} \times \left(C_{ij} + \cdots + C_{mj}\right) \qquad \text{Eq. (3)}$$

Information gain for F can be calculated as:

$$Gain(F) = \mathrm{I}(C_1, \ldots\ldots, C_m) - E(F) \qquad Eq. (4)$$

5. Fuzzy C-Mean (FCM) algorithm

The aim of fuzzy clustering algorithm is partition the data to various clusters according to degree membership value, in a way that each cluster

contains data more similar to each other and different from data in other clusters. In fuzzy c-means algorithm the data point data point belongs to more than cluster with some degree of membership which is known as soft clustering. The data is assigned to clusters based on fuzzy membership function [12]. The fuzzy clustering algorithm depends on minimizing objective j and calculates using equation (2.1). This is based on the following object function minimization j as shown in equation (2.1) [13,14].

$$J_m(U, C)=\sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}^{m}\ d_{ij}^{2}(x_i, c_j) \qquad Eq.(5)$$

Where

1. *m – real number in domain ($1 \leq m < \infty$).*
2. *k –number of cluster.*
3. *n – number of data samples.*
4. *ui – membership degree that indicates the probability that data sample xi belongs to j^{th} Cluster.*
5. *c_j – centre of cluster.*

The fuzzy clustering can be done by repeatedly modified cluster centre c_j and fuzzy membership uij using equations (2.2) (2.3) [13, 14].

$$u_{ij} = \frac{1}{\sum_{j=1}^{k}(d_{ij}/d_{ik})^{2/m-1}}, \qquad \forall i \quad Eq.(6)$$

$$c_j = \frac{\sum_{i=1}^{n} u_{ij}^{m} . x_i}{\sum_{i=1}^{n} u_{ij}^{m}}, \qquad \forall i \quad Eq.(7)$$

Where u_{ij} indicates the membership degree of the data samples that belongs to specific cluster, and satisfies the following conditions, see equations (2.4) and (2.5) [13, 14]:

$$\sum_{j=1}^{k} uij = 1 \qquad \forall k\ Eq.(8)$$

$$\sum_{j=1}^{k} uij > 0 \qquad \forall i\ Eq.(9)$$

6. Back Propagation (BP) algorithm

Back propagation is multilayer feed forward neural network containing one input layer, one or more hidden layers and one output layer, neurons are arranged in layers. The learning course of back propagation consists of two phases: the forward phase where the input is presented and propagation is towards the output layer: the backward phase where the error is computed and the weight adjusted to reduce the error so that the ANN learns the data [13]. In the forward each neuron in the input layer has input multiplied by weights between input and hidden layers. Each hidden neuron (j) in hidden layer receives value Zj(j) using equation (2.6) [15, 16]:

$$z_j=\theta+\sum_{i=1}^{n} x_i\, w_{ij} \qquad Eq.\ (10)$$

The binary sigmoid function actviation function is used to process the output of hidden layer using equation (2.7):

$$f(x) = \frac{1}{1+\exp^{-x}} \qquad Eq.\ (11)$$

$$z_j = f(z_j)$$

The output of hidden layer is broadcast to the nodes in the output layer as in equation (2.8) [15, 16]:

$$y_k=\theta + \sum_{j=1}^{p} z_j\, w_{jk} \qquad Eq.\ (12)$$

$$y_k = f(y_k)$$

Where θ is the basis between layers.

The Mean Square Error value E is considered quantative measure of learning which reflects the degree of learning. generally, an MSE under (0.1) indicates that the net learned its training set. The Mean Square Error value E is calcualted according the following equation (2.9) [15, 16]:

$$E = \frac{1}{2} \sum_{p} \sum_{k} (T_{pk} - Y_{pk})^2 \qquad Eq.\ (13)$$

Where

1. *p – represents the number of samples.*
2. Y_{pk} *- the value of actual output.*
3. T_{pk} *- the value of target output.*

In the backward phase if the output of network is different from target output then the output error will be calculated, the error is then propagated towards the input layer to update the weight between neurons in layers. The error between the output and hidden layer can be calculated using the following equation (2.10) [15, 16]:

$$\delta_{2k} = y_k(1 - y_k)(T_k - y_k) \qquad Eq.\ (14)$$

The error between hidden and input layer can be computed using equation (2.11) [15, 16]:

$$\delta_{1j} = z_j\,(1 - z_j)\sum_{k=1}^{m} \delta_{2k}\, w_{jk} \qquad Eq.\ (15)$$

Then the weights are updated to reduce the error according the following equations (2.12) and (2.13) [15, 16]:

$$w_{jk}(new) = \eta * \delta_{2k} * z_j + \propto * w_{jk}(old) \qquad Eq.\ (16)$$

$$w_{ij}(new) = \eta * \delta_{1k} * x_i + \propto * w_{ij}(old) \qquad Eq.\ (17)$$

Where

1. w_{jk} - *the weights between the output and hidden layers.*
2. w_{ij} - *the weights between the hidden and input layers.*
3. η – *learing rate.*
4. $\propto$ – *momentum cofficient.*

The goal of backward phase is to find the global optimum of network weights and reduce the gradient error. The learning course is achieved by minimizing mean absolute error value E_m.

7. Evaluation Measures

The measure efficiency of IDS depends on its ability to make the right detection depending on the nature of the given status compared with the result of intrusion detection system (IDS). In the first level of the proposed system four possible results can be obtained and called confusion matrix described in table (2).

Table (2): Confusion Matrix

Actual class	Predicted Class	
	Negative class(normal)	**Positive class(attack)**
normal	True negative (TN)	False positive (FP)
attack	False negative (FN)	True positive (TP)

The four outcomes are true negative (TN) which indicates the correct prediction of normal behaviour, true positive (TP) which indicates the correct predication of attack behaviour, false positive (FP) which indicate the wrong predication of normal behaviour as attack, false negative (FN) which indicate mistaken predication of attack behaviour as normal. Both (TN) and (TP) are considered guide of the correct operation of the IDS. To evaluate the performance of the second level of the proposed system Detection Rate of Class (DRC) used as evaluation measure which represent the rate of correctly classified attack samples for each type of attack. Moreover, there are additional suggested result which is unknown in the case of normal connections improperly clustered as abnormal. In addition, each of (FP), (FN), reduces the effectiveness of IDS. Therefore (FP) and (FN) should be minimize to increase the efficiency of IDS system. The performance of the proposed system evaluated using the following measures [17]:

Accuracy (ACC): It measures performance representing the rate of samples which are properly detected as normal or attack to the overall number of samples and calculated using the equation:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \qquad Eq.(18)$$

Detection Rate (DR): It measures performance which indicates the ratio of the number of samples that are correctly classified as attack to the total number of attack samples and is calculated using equation:

$$DR = \frac{number\ of\ correctly\ detected\ samples}{total\ number\ of\ samples} \qquad Eq.\ (19)$$

False alarm rate (FAR): It measures performance which represents the rate of samples which are improperly categorized as attack to the overall number of samples of normal behaviour and is calculated using equation:

$$\mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}} \qquad Eq.(20)$$

Detection Rate of the class (DRC): It measures performance representing the rate of samples of specific class that are correctly classified to its class to the total number of samples of the specific class. This measure is used to evaluate the second phase of the proposed system.

$$DRC = \frac{The\ number\ of\ correctly\ classfied\ samples}{total\ number\ of\ samples} \qquad Eq.(21)$$

In addition, to detect the number of normal traffic that is incorrectly classified as attack in first phase, counter suggested called unknown to count the number of normal traffic which is detected in second phase.

8. The Proposed Multi-level-NIDS

The proposed Multi-level-NIDS system consists of two phases; in the first level the proposed system train with Fuzzy C-Mean (FCM) algorithm. FCM is one clustering algorithm that uses membership to determine a certain degree of clustering which each data point belongs to. Its core idea is to divide n vector *x* (*i*= 1, 2… *n*) into *c* fuzzy group and seek the clustering centre for each group. Fuzzy C-means using fuzzy partition, that would make us use the membership to determine the degree each given data point belonging to, the value of each data point is between zero and one. FCM algorithm is soft clustering algorithm which allow data point to belong more than one cluster. In the second level the proposed system train with Back propagation algorithm. Back propagation (BP) algorithm is a supervised learning technique which means the algorithm training with samples of input and output that network should be calculated. BP is multilayer feed forward neural network containing one input layer, one or more hidden layers and one output layer, neurons are arranged in layers. The learning course of back propagation consists of two phases: the forward phase where the input is presented and

propagation is towards the output layer: the backward phase where the error is computed and the weight adjusted to reduce the error so that the ANN learns the data. The proposed multi-level-NIDS illustrate in Fig. 1. Algorithm 1 show the proposed Multi-level-NIDS. The two level of Multi-level-NIDS is explained as follows:

Level 1: FCM algorithm used to detect the normal traffic from abnormal. The result is two clusters normal and attack clusters.

Level 2: BP algorithm used to classify the type of attack or indicates unknown in case of normal connection detect as attack.

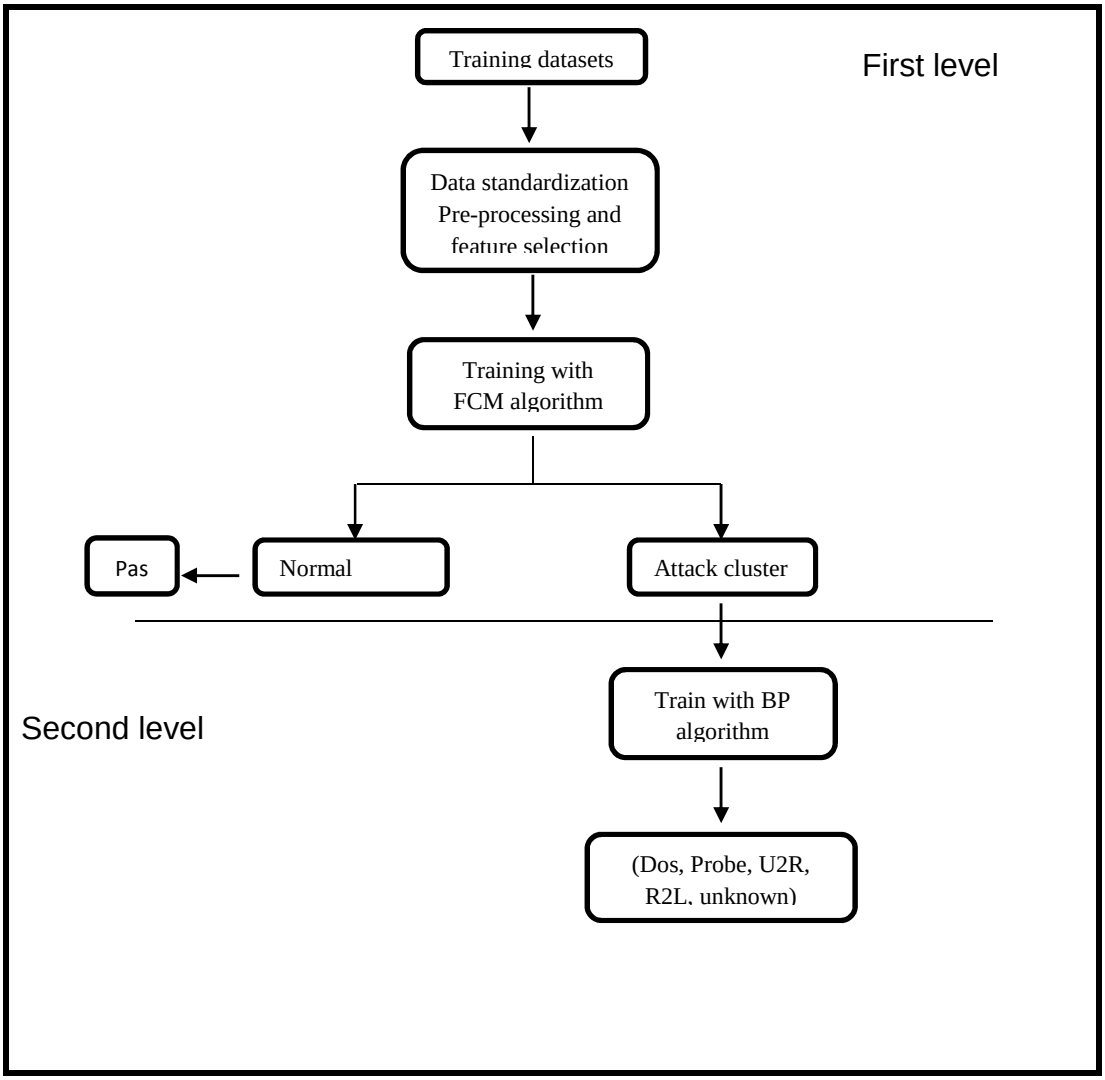

Figure 1: Proposed multi-level-NIDS

Algorithm (1): Multi-level-NIDS
Input: Training dataset.
Output: Detect the normal samples from attack samples and detect the type of attack.
Steps
//first level of the proposed system.//

1. Initialize the cluster centre randomly.
2. Compute the membership matrix for each sample using

$$membership_{of\ sample\ i\ regard\ to\ clsuter\ j} = \frac{1}{\sum_{j=1}^{k}(sample\ i - cluster\ \ cenetr\ j/samplei - clsuetr\ centerk)^{2/fuzzienss-1}}$$

3. Update the cluster center using

$$cluster\ center\ j = \frac{\sum_{i=1}^{number\ of\ sample} membership_{of\ sample\ i\ regard\ to\ clsuter\ j}^{fuzzienss} . sample\ i}{\sum_{i=1}^{number\ of\ sample} membership_{of\ sample\ i\ regard\ to\ cluster\ j}^{fuzzienss}}$$

4. Repeat until stopping criteria.
5. Assign the cluster type for each sample with highest membership value.
6. Segregate attack sample from normal sample according to label type of each sample.

//Second level of the proposed system.//

7. Initialize the weights vectors from input units to hidden units and from hidden units to output units randomly.
8. Compute the desired output for attack cluster from labelled attack samples.
9. Do
10. Each feature in attack cluster sample represents input unit (X_i) then broadcasts to all neurons in hidden layer.
11. Compute the output of each unit in the hidden layers (z_j) using

$$hidden\ j = bais + \sum_{i=1}^{number\ of\ input\ unit} input\ unit_i\ weight_{ij}$$

Then the output of hidden layer is processed using

$$hidden(j) = \frac{1}{1 + \exp^{-\text{hidden j}}}$$

12. Compute the output of each unit in the output layer (Yk) using equation

$$output\ of\ _k = bais + \sum_{j=1}^{\text{number of hidden units}} hidden\ _j . \text{weight}_{jk}$$

then the output processed using equation

$$output(k) = \frac{1}{1 + \exp^{-\text{output k}}}$$

13. Compute the error value between output units and hidden units using equation

$$error\ of\ output\ _k = \text{ouptut}_k(1 - \text{outptu}_k)(\text{target}_k - \text{output}_k)$$

14. Compute the the error value between hidden units and input units using equation (2.11).

$$error\ hidden\ _j = \text{hidden}_j\left(1 - \text{hidden}_j\right) \sum_{k=1}^{\text{output number}} \text{error output}_k\ \text{weight}_{jk}$$

15. Update the weights w_{ij} and w_{jk} between layers according to equations:

$$\text{weight}_{jk}(\text{new}) = \text{learning rate} * \text{error output}_k * \text{hidden}_j + \text{momentum} * \text{weight}_{jk}(\text{old})$$

$$\text{weight}_{ij}(\text{new}) = \text{learning rate} * \text{error ouptu}_k * \text{input}_i + \text{momentum} * \text{weight}_{ij}(\text{old})$$

16. Calculate the mean square error value E_m using equation

$$\text{error} = \frac{1}{2} \sum_{\text{number of sample}} \sum_{k} (\text{target}_{pk} - \text{output}_{pk})^2$$

17. Repeat until E_m <=Minerror.

End

9. Experimental results

The proposed system evaluated with KDD99 dataset. The proposed system is trained with samples selected from KDD 99 dataset includes normal behaviour samples besides the other four types of attack (Dos, Probe, U2R, R2l) to specify normal samples from attack samples and also to detect the type of attack. The proposed system train with 41 features of KDD99 dataset. then the proposed system train with 25, 20, 15 features of KDD99 dataset which representing the features with highest Information Gain. In the first level of multi-level-NIDS three evaluation criteria used to assess the proposed system which is (ACC, DR, FAR), in the second level of multi-level-NIDS Attack Detection Rate (ADR) used as evaluation criteria. To check the efficiency of the proposed module three experiments are conducted, in the first experiments the algorithm is tested with dataset called dataset1 consist of (1000) records contain normal behaviour in addition to four attack types. The second and third experiments are conducted with datasets set called dataset2 and dataset3 consist of (500) (1500) records respectively and they also include four types of attack. In the first level the fuzziness parameter set to 2, number of cluster centre set to 5 and number of iteration set to 17. As for Back propagation parameters values which is the second level of the proposed system show in table 4. The result of the first level shown in table 5 and the results of second level shown in table 6.

Table (4): Parameters of back propagation algorithm

Parameters name	Value parameters
basis	1
Learing rate	1
Momentum cofficient	1
No.of unit in input layer	41,25,20,15
No.of unit in hidden layer	10
No.of unit in output layer	5
Mean square error	0.001
Maximum number of iteration	1500

Table (5): the result of the first level of Multi-level-NIDS

Dataset	Number of features	ACC	DR	FAR
Dataset1	41	0.99	0.99	0.01
	25	0.91	0.89	0
	20	0.90	0.88	0.01
	15	0.91	0.89	0.01
Dataset2	41	0.99	1	0.01
	25	0.89	0.87	0.01
	20	0.99	1	0.01
	15	0.99	1	0.01
Dataset3	41	0.96	0.93	0
	25	0.91	0.89	0
	20	0.95	0.93	0
	15	0.95	0.93	0.002

Table (6): the result of the second level of Multi-level-NIDS

Type of Attack	Number of Feature	Dataset1	Dataset2	Dataset3
Dos	41	0.99	0.99	0.99
	25	0.99	0.99	0.99
	20	0.99	0.99	0.99
	15	0.98	0.99	0.99
probe	41	0.99	0.98	0.98
	25	0.99	0.98	0.98
	20	0.99	0.98	0.98
	15	0.98	0.98	0.98
U2R	41	0.96	0.96	0.96
	25	0.95	0.96	0.96
	20	0.96	0.96	0.96
	15	0.94	0.95	0.96
R2L	41	0.98	0.98	0.98
	25	0.98	0.98	0.98
	20	0.98	0.98	0.98
	15	0.97	0.98	0.97
Unknown	41	0.99	0.99	0
	25	0.98	1	0
	20	0.98	0.98	0
	15	0.98	1	1

10. Conclusion

In this work Multi-level-NIDS system proposed to detect the intrusion and type of intrusion in traditional/cloud network. The proposed work in the first level can detect the normal traffic from intrusion traffic with high accuracy and detection rate and low false positive rate, and the second level can detect the class of attack with high detection rate even with low frequency of attack (U2R). The best result of the Detection Rate of Attack (DRA) is achieved with Denial of Service (DOS) attack, and the worst Detection Rate of Attack (DRA) is achieved with User to Root (U2R) attack, where the KDD99 dataset contains a large number of records of Denial of Service (DoS) and the number of records of U2R in KDD99 dataset is very few. The normal traffic that is misclassified as attack in first level of the proposed system can be recognized as unknown attack in the second phase of the proposed system where the BP algorithm trained to classify the normal connection to unknown. The experiments show that the number of input features has impact on the speed of the FCM algorithm training whenever the fewer features the training time will be reduce. However, the accuracy be affected in a negative way where the FCM algorithm the clusters identified via similarity measure therefore all features values is important even zero values. The impact of feature selection on BP algorithm is reduce the training time without impact on its detection rate where BP algorithm trained with maximum Mean Square Error (MSE) equal to 0.001. Four experiments was conduct on the proposed system with 41,25,20,15 features of the KDD99 dataset. Feature selection in Intrusion Detection System increase the speed and in some times the accuracy and the feature selection result depend on offline traffic, in online traffic the features values change therefore the features selection result may be changed. Therefore, it is better to use all features in implement Network Intrusion Detection System (NIDS).

Reference

1. Ghosh P., Mandal A. K., and Kumar R., "**An Efficient Cloud Network Intrusion Detection System**", springer, volume 339, pp. 91-99, 2015.
2. **Mell P., Grance T., "The NIST Definition of Cloud Computing", Recommendations of the National Institute of Standards and Technology, Special Publication 800-145, Available at < http://csrc.nist.gov/publications/PubsSPs.html#800-145>.**
3. Modi C., Patel D., Borisaniya B., Patel H., Patel A., Rajarajan M. K., **"A Survey of Intrusion Detection Techniques in Cloud"**, Journal of Network and Computer Applications ,volume 36 ,pp. 42–57,2013.
4. Deshmukh V. G., Borkut A. G., Agam N. A., **"Intrusion Detection System For Cloud Computing"**, International Journal of Engineering Research & Technology (IJERT) Vol. 2, Issue 4,pp.1-5, 2013.
5. Nadiammai G.V., Hemalatha M., **"Effective Approach Toward Intrusion Detection System using Data Mining Techniques"**, Elsevier, Egyptian Informatics Journal, Vol.15, Issue.1, pp.37-50, 2014.
6. Dickerson J. E., Juslin J., Koukousoula O., Dickerson J. A. , **"Fuzzy Intrusion Detection "**, IEEE, Joint 9th IFSA World Congress and 20th NAFIPS International Conference,pp.1506-1510, 25-28 July,2001.
7. Wu S. X., Banzhaf W., **"The Use of Computational Intelligence in Intrusion Detection Systems: A review",** Applied Soft Computing, Vol. 10, issue. 1, pp.1-35, 2010.
8. Khazaee S., Rad M. S., **"Using Fuzzy C-Means Algorithm for Improving Intrusion Detection Performance"**, IEEE, 13th Iranian Conference on Fuzzy Systems (IFSC), 27-29 Aug, 2013.
9. Selman A. H., Koker R., Selman S., **"Intrusion Detection Using Neural Network Committee Machine"**, IEEE, XXIV International Symposium on Information, Communication and Automation Technologies (ICAT), 30 Oct-1 Nov , Sarajevo, Bosnia and Herzegovina, 2013.

10. kumarl S., Yadav A., **"Increasing Performance of Intrusion Detection System Using Neural Network"**, IEEE, International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 8-10 May ,pp.546-550,2014.
11. Sahu S. K., Sarangi S., Jena S. K**.," A Detail Analysis on Intrusion Detection Datasets"**, IEEE, pp.1348-1353,2014.
12. Saurabh Mukherjee, Neelam Sharma, **"Intrusion Detection Using Naive Bayes Classifier with Feature Reduction"**, Elsevier, Procedia Technology, vol.4, pp. 119 – 128, 2012.
13. Gao H., Zhu D., Wang X., **"A Parallel Clustering Ensemble Algorithm for Intrusion Detection System",** IEEE, Ninth International Symposium on Distributed Computing and Applications to Business Engineering and Science (DCABES), 10-12 Aug, pp. 450-453, 2010.
14. Pandeeswari N., Kumar G., **"Anomaly Detection System in Cloud Environment Using Fuzzy Clustering Based ANN**", Springer US, Mobile Networks and Applications, Volume 21,issue 3, pp. 494-505, 2015.
15. Hogo M. A**.,"Temporal Analysis of Intrusion Detection",** IEEE, International Carnahan Conference on Security Technology (ICCST), 13-16 Oct, 2014.
16. Esmaily J., Moradinezhad R., Ghasemi J., **"Intrusion Detection System Based on Multi-Layer Perceptron Neural Networks and Decision Tree",** IEEE, 7th International Conference on Information and Knowledge Technology (IKT), 26-28 May, 2015.
17. Wang H., Zhang Y., Li D., **"Network Intrusion Detection Based on Hybrid Fuzzy C-Mean Clustering"**, IEEE, Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010), 10-12 Aug ,pp.483-486,2010.
18. Hu L., Zhang Z., Tang H., Xie N., **"An Improved Intrusion Detection Framework Based on Artificial Neural Networks"**, IEEE, 11th International Conference on Natural Computation (ICNC),pp. 1115-1120, 15-17 Aug,2015.

19. Hogo M. A**.," Temporal Analysis of Intrusion Detection",** IEEE, International Carnahan Conference on Security Technology (ICCST), 13-16 Oct, 2014.

اقتراح نظام كشف تطفل شبكي متعدد المستوى لكشف التسلل في بيئة الحوسبة السحابية

شوق مالك محيبس أم.د. سكينة حسن هاشم

المستخلص

الحوسبة السحابية هي واحدة من التقنيات الشائعة،التي تستخدم في معظم المؤسسات لما لها من خصائص مميزة مثل التوافر، المرونة ، التكامل. لهيكلية المفتوحة والموزعة للحوسبة السحابية والخدمات المقدمة جعلتها هدف محبب للهجمات الالكترونية المحتملة. نظام كشف التطفل الشبكي (NIDS) يمثل الية امنية مهمة،توفر طبقة دفاعية التي تراقب حركة مرور الشبكة للكشف عن نشاطات مشبوهة او انتهاك للسياسات. هذا العمل يقترح نظام كشف تطفل شبكي متعدد المستوى لكشف التطفل ونوع التطفل في الشبكة التقليدية / السحابية. النظام المقترح قيم باستخدام مجموعة البيانات القياسية KDD99، النتائج التجريبية اظهرت كفاءة وقدرة النظام المقترح في كشف الهجوم ونوع الهجوم .

الكلمات المفتاحية: الحوسبة السحابية، كشف التطفل الشبكي، تنقيب البيانات، خوارزمية العنقدة المضببة، خوارزمية الانتشار العكسي

*الجامعة التكنولوجية