

A Neuro Fuzzy Based Intrusion Detection System for a Cloud Data Center Using Adaptive Learning

Pandeeswari Nagarajan, Ganeshkumar Perumal

*Department of Information Technology, PSNA College of Engineering & Technology, Dindigul,
TamilNadu, India*

Emails: pandeeswari@psnacet.edu.in drpganeshkumar@gmail.com

Abstract: *With its continuous improvements, the cloud computing system leaves an open door for malicious activities. This promotes the significance of constructing a malware action detection component to discover the anomalies in the virtual environment. Besides, the traditional intrusion detection system does not suit for the cloud environment. So, the proposed scheme develops an anomaly detection system, named Hypervisor Detector at a hypervisor layer to detect the abnormalities in the virtual network. Besides, the fuzzy systems have the ability to detect the presence of uncertain and imprecise nature of anomalies; they are not able to construct models based on target data. One of the successful approaches, which integrate fuzzy systems with adaptation and learning proficiencies of a neural network, such as ANFIS (Adaptive Neuro Fuzzy Inference System) model, is based on target values. The Hypervisor Detector is designed and developed with an ANFIS and practised with a hybrid algorithm, a combination of the back propagation gradient descent technique with the least square method. For the experiments and performance analysis, DARPA's KDD cup data set is used. The performance analysis and results show that the model proposed is well designed to detect the abnormalities in virtual environment with the minimum false alarm rate and reduced overhead.*

Keywords: *Cloud computing, intrusion detection system, ANFIS, hypervisor, false alarm rate.*

1. Introduction

In modern years, the majority of the IT organizations have agreed to utilize the cloud computing technology. Cloud computing encourages the resource constraint clients to use the shared resources and the computing facilities by providing these facilities for low cost. This provides significance over online storage which affords storage, computational and shared resource facilities as services used on pay for usage basis. The virtualization technology [1] builds cloud computing environment which permits sharing of the computing servers, the storage space in order to increase the utilization. With the continuous enhancement of cloud technology, the rigorous concerns are regarding the security issues which are introduced by this cloud computing paradigm. Besides, the enhanced virtualized infrastructure and the distributed nature of a cloud system provide probability for the intruders to hack the resources. The problem becomes more critical when the stored data and computing power is abused by an inside intruder that makes the cloud system itself a threat. Also, the lack of control over the virtualized environment is a severe concern for the cloud service consumers. This raises the need [2] for secure and safe security systems through the use of firewalls, intrusion detection and prevention systems and other cryptographic primitives. Before using the cloud computing model, the organizations [3] must be aware of the security issues, such as: 1) privileged user access; 2) data segregation; 3) long term liability; 4) regulatory compliance; 5) recovery; 6) investigative support, and 7) data location. The cloud computing providers necessitate providing security against insider and outsider attacks. The malicious behaviours on the virtual machines have to be monitored to find the untrustworthy actions on the virtual environment. The virtual machines [4] are being dynamically changed, its states (like start, run, block/suspend and resume) and they can migrate from one hardware platform to another. Considering the dynamic nature of a cloud computing system and to overwhelm the drawbacks of the traditional IDS, the virtual machine technology can be used to determine the malware actions. Hence, this paper proposes an intrusion detection mechanism, named Hypervisor Detector to detect the abnormal behaviour in virtual environment. The proposed Hypervisor Detector is implemented with an Adaptive Neuro-fuzzy inference system; it uses back propagation gradient descent technique in combination with the least square scheme to detect the anomalies in a virtual cloud. The Hypervisor Detector is compared with the other VM-IDSEs that uses an artificial neural network [5], Naive Bayes [6] and NBRF (Naive Bayes Random Forest) [6].

2. Related works

Many researchers have developed intrusion detection elements to discover the intrusions in a cloud computing system. Regarding the dynamic nature of a cloud system, the virtual machine based monitoring element was first proposed by Garfinkel and Rosenblum [7]. They developed a prototype, called Livewire which uses the visibility of the Host based Intrusion Detection System (IDS) and

drags out the IDS to detect the network based attacks in the virtual network. Jin et al. [4] have designed a virtualization based detection element called VMFence to examine the network flow and integrity of a file and also to detect the real time attacks. VMFence was located in a privileged VM connected with the virtual bridges connecting the virtual machines. This system is computationally more complex. Amirreza and Alireza [8] have developed a Cloud based Intrusion Detection System (CIDSS) that is positioned inside the user network to capture the activities done by the user. CIDSS comprises of three components, namely: 1) an intrusion detection service agent; 2) a cloud computer service component, and; 3) an intrusion detection service component which in turn uses the pattern-matching technique to detect the malicious activity. This system realizes more communication overhead. The IDS for distributed architecture [9-11] considers audit logs, traffic data and virtual machine's behaviour to detect both known and unknown attacks. Various authors have used different techniques, such as behaviour and knowledge basis [5], virtual machine log and replay [12], hidden Markov model [13], FC-ANN [14], Principal Component Analysis Neural Network Algorithm (PCANNA) [15] to detect the anomalies. An integrated approach for detecting and preventing the intrusions was proposed by Ubhale and Sahu [16]. This work uses both signature-based and anomaly based intrusion detection system to do in-depth analysis to determine the known and unknown attacks. To improve the detection accuracy of both anomaly and signature based intrusion detectors, stacking methodology [17] can be used. Amjad, Sabyasachi, and Debasis [6] have developed a virtual machine monitor based intrusion detection system which uses two different machine learning algorithms. The two approaches are: 1) Naive Bayes classifier, and 2) hybrid approach which is a combination of Naive Bayes and Random Forest. The proposed work uses a hypervisor based intrusion detection system (Hypervisor Detector) to observe the activities of the virtual machines and to detect the anomalies in the virtual network. The hypervisor based intrusion detection system overcomes the drawbacks of isolated monitoring [9] which omits the separate hardware platforms. The Hypervisor Detector uses an adaptive neuro fuzzy inference system to implement the Hypervisor Detector and is compared with IDSes, designed by using an artificial neural network [5], Naive Bayes and hybrid approach (Naive Bayes Random Forest) [6].

3. Intrusion detection system

IDS is a process for detecting malicious activities in network environment. The intrusion detection system must offer security solutions by examining configurations, logs, user actions and network traffic. The IDS has to be designed with two requirements: 1) functional requirement; 2) performance requirement. The functional requirements are IDS means to monitor the virtual hosts' activities and describe the intrusion. The Intrusion detection system should maintain very low false alarm rate. The performance requirement of IDS is to detect intrusions accurately and intimate this immediately in order to handle additional computational and communication loads. Regarding the source of the data, the

intrusion detection system can be categorized as a Host Based Intrusion Detection System (HBIDS) and a Network Based Intrusion Detection System (NBIDS). HBIDS collects the system characteristic including system conditions, file system integrity, audit logs, system calls and network events from and to the host, on which it is deployed. HBIDS does not require any additional hardware, which uses the resources of the host machine. HBIDS is designed to monitor the single host; it is able to detect the low level local activities. NBIDS collects data packets by listening to the network traffic and analyzing the packets to detect the malware functionalities. NBIDS does require a hardware machine to act as a monitoring element and it is not able to analyze the encrypted network traffic. While comparing [13] HBIDS with NBIDS, HBIDS provides high visibility and is with low detection accuracy, whereas NBIDS offers high detection rate at a high cost. In accordance with the identification method, IDS can be designated as signature based and anomaly based IDS to detect the known and unknown attacks respectively. Signature based IDS (SIDS) uses the patterns of unauthorized users where it does pattern matching to identify the unauthorized accesses. SIDS can detect only known attacks. In Anomaly based IDS (AIDS), the usual behaviours of the users are monitored and also used to create a profile which explains the usual behaviours. The user behaviour that is accessing the system is monitored and used to create a profile, any deviation from the normal user profile is depicted as unusual behaviour. AIDS is designed to detect the unknown attacks. The Cloud Intrusion Detection System (CIDS) integrates [3] knowledge and behaviour analysis to detect intrusions. The CIDS must detect and response to distributed and coordinated attacks. CIDS must be adaptive to a dynamic network and configuration changes. Since the cloud is distributed, to monitor each node's activity, the proposed method uses a hypervisor based intrusion detection system, named Hypervisor Detector.

4. Intrusion detection system in virtual environment

With the modern technological development, from large scale organizations to small scale organizations have started to outsource their details into a public cloud. While outsourcing the users' sensitive information onto the external online storage, the system integrity, confidentiality and availability has to be guaranteed. The open and distributed infrastructure of the cloud computing attracts the intruders. With the faster growth of web based services, the cloud magnetizes many users to use the web services, as well as intruders to abuse the cloud resources. And also, due to the open nature, the cloud system becomes the most attractive place for more vulnerability. A survey on business and technical issues at design and implementation levels has been made [18], where the issues are directly affecting the performance of cloud computing. Cloud applications are executed beyond the firewall and moved to the public domain which may have severe consciousness on security. This makes a force to provide IDS in dynamic cloud environment which overcomes the problem with traditional computing environments. Perhaps, with the dynamic nature of a virtual machine, the monitoring process will be rather complex. Therefore, to detect the unauthorized and malware access on a cloud system, this

paper proposes a hypervisor based anomaly detection system. The host based intrusion detection [14] in a hypervisor or host machine would allow the IDS to examine the hypervisor and virtual operating systems on the same hardware platform. Then if the host is compromised, the HIDS on the hardware platform would be neutralized. With the rapid flow of a large volume of data, the Hypervisor based detection system acts as a host based intrusion detection system which is also configured to monitor the network activities in the cloud computing infrastructure. This paper describes the method to detect the unusual behaviour in a virtual network which uses anomaly based detection technique. The representation of a cloud system model is shown in Fig. 1.

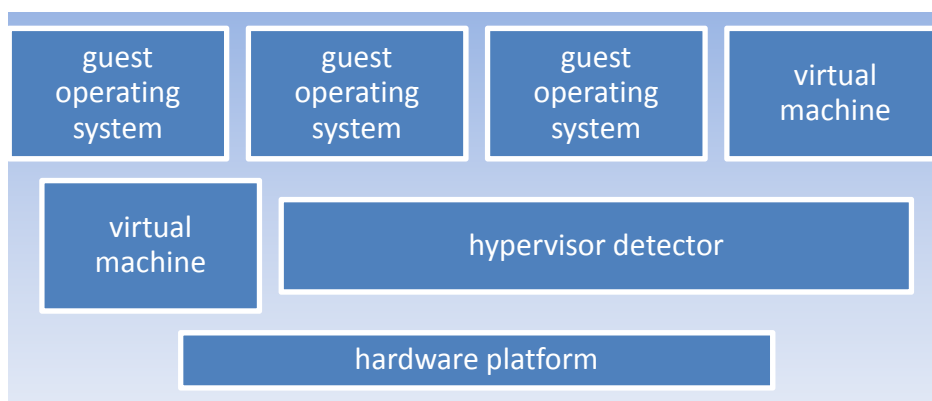


Fig. 1. Virtual cloud architecture with a Hypervisor Detector

4.1. Hypervisor Detector

The hypervisor is the software layer which is executed on a hardware platform. The hypervisor VMM has the potential to monitor and control the network based and host based events on the virtual environment. Every host machine is presented with the virtual hypervisor that runs separately from the host machine. The virtual hypervisors are monitoring the real hardware systems which provide a single platform for the different VMs. This ability provides hypervisor based virtualization [19] to acquire a secure infrastructure. The hypervisor as a hardware element is used to discover the network based intrusion. The Hypervisor Detector monitors the virtual network traffic (network based events) to capture network data and analyze it. The Hypervisor Detector is experienced with the database of normal activities; any deviation from this is notified as abnormal activity. The proposed method uses an adaptive neuro fuzzy based scheme to train and test the Hypervisor Detector.

4.2. Guest OS

This Operating System (OS) is installed and executed on a virtual machine. The Hypervisor monitors [19] the guest operating system running on the virtual machines.

4.3 Virtual machine

The virtual machine runs a user process on the single hardware platform. Virtual machines are dynamically changing their states. VM can migrate from one platform to another without any conditions. VM can be clogged, poised and infertile. The virtual machines on a single hardware platform run various applications that can be stored on different locations of the host machine. Due to the dynamic nature of the cloud infrastructure [20], it is possible for the virtual computation environment to relocate itself and scale its resources across a multi-domain infrastructure.

4.4. Virtualization

Virtualization [4] is a process to afford parallel and interactive access to a large pool of the information center that supports numerous instances of OS running on a single hardware platform and can also control the multiple OS which consecutively results in hardware virtualization. Hypervisor permits multiple instances of OSES to share the hardware facilities on which it is hosted.

5. Design of a Hypervisor Detector with ANFIS

The cloud computing system works with the concept of virtualization of the application and storage resources. The hypervisor in a cloud system monitors the various guest operating systems executed on the same hardware platform. With the open nature and enormous amount of traffic data, the cloud computing system becomes an attractive place for hackers. Thus, an efficient and effective intrusion detection system is required. Therefore, the proposed work is designed to examine the anomalies in the virtual network by analyzing the events on the multiple virtual machines. Soft computing is an innovative approach to build computationally intelligent systems which enhance the extraordinary abilities of the human to analyze and learn uncertainty and imprecision of an environment. The main task of constructing an anomaly detection system is to utilize a classifier that can separate a normal and intrusive dataset [21]. Adaptive Neuro Fuzzy Inference System (ANFIS) is also referred as a fuzzy classifier that provides a method for a fuzzy modeling procedure to observe the information about a dataset. In order to calculate the membership function parameters, it allows the fuzzy inference system to track the given input/output data. Hypervisor Detector is practiced with ANFIS which uses the back propagation gradient descent technique in combination with the least square method. This model is trained to observe the operations on the virtual machines. Neuro fuzzy systems [22, 23] are fuzzy inference systems that use artificial neural networks to determine their properties by processing the data samples. The particular development in a neuro fuzzy system is the adaptive [24] Neuro Fuzzy inference system to solve the non-deterministic procedures. The gradient descent with the least square technique is used to update the parameters of the membership functions in an adaptive inference system. ANFIS [22] uses the essence of fuzzy logic and neural network to adjust the parameters and to provide an optimized result. The intrusion detection system has to perform early detection

of the malware activities and protect the system from a serious damage. The performance of the detection system can be measured in terms of detection accuracy.

Let the inference system be with two inputs x and y ; a single output f ; then the two fuzzy if-then rules [24] are as follows:

Rule 1. If x is A_1 and y is B_1 then $f_1 = p_1x + q_1y + r_1$.

Rule 2. If x is A_2 and y is B_2 then $f_2 = p_2x + q_2y + r_2$.

Let the membership functions be μA_i and μB_i for fuzzy sets A_i and B_i respectively.

This model uses T -norms (logical) for rule creation.

By evaluating the rule premises, the result is as follows:

$$(1) \quad w_i = \mu A_i(x) \mu B_i(y), \quad i = 1, 2, \dots$$

Evaluation of the proposition and rule consequences yields is as follows

$$(2) \quad f(x, y) = \frac{w_1(x, y) f_1(x, y) + w_2(x, y) f_2(x, y)}{w_1(x, y) + w_2(x, y)},$$

shrunked to

$$(3) \quad f(x, y) = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2},$$

where f can be written as

$$(4) \quad f = \bar{w}_1 f_1 + \bar{w}_2 f_2,$$

where

$$(5) \quad \bar{w}_i = \frac{w_i}{w_1 + w_2}.$$

The ANFIS model is an information processing technique [26] that can use the training and knowledge of the neural networks to find the parameters of a fuzzy system. The performance of ANFIS is measured in terms of a training error. This Hypervisor Detector uses knowledge and behaviour analysis for detecting the anomalies in virtual environment which are shown in Fig. 2.

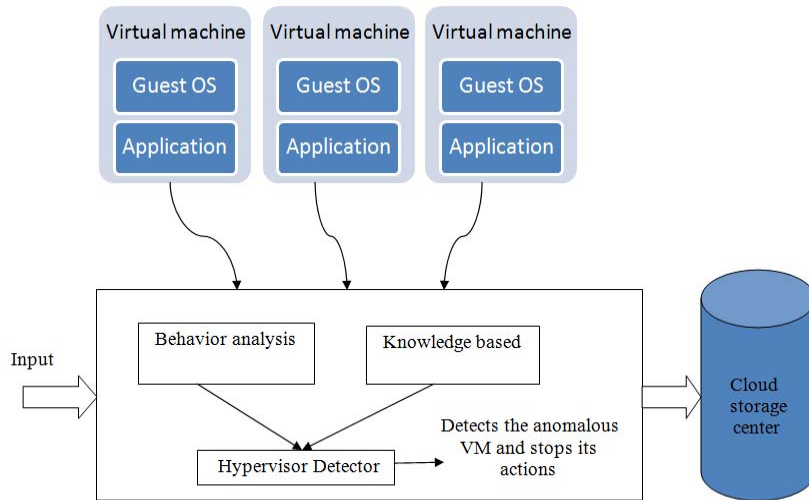


Fig. 2. Framework for Hypervisor Detector

The proposed model is verified using DARPA's KDD cup dataset [25] which uses the input attributes and output attributes from that dataset. The KDD cup is specifically designed for intrusion detection system. This paper uses the reduced 30 real attributes of KDD cup given in Table 1.

The output is classified into two: 1) normal; 2) abnormal.

Table 1. Attribute description for KDD dataset

No	Attribute name	Attribute number	Type
1	duration	1	Real
2	src_bytes	5	Real
3	dst_bytes	6	Real
4	wrong_fragment	8	Real
5	urgent	9	Real
6	hot	10	Real
7	num_failed_logins	11	Real
8	num_compromised	13	Real
9	root_shell	14	Real
10	su_attempted	15	Real
11	num_root	16	Real
12	num_file_creations	17	Real
13	num_shells	18	Real
14	num_access_files	19	Real
15	num_outbound_cmds	20	Real
16	count	23	Real
17	srv_count	24	Real
18	serror_rate	25	Real
19	srv_serror_rate	26	Real
20	rerror_rate	27	Real
21	srv_rerror_rate	28	Real
22	same_srv_rate	29	Real
23	diff_srv_rate	30	Real
24	srv_diff_host_rate	31	Real
25	dst_host_count	32	Real
26	dst_host_srv_count	33	Real
27	dst_host_same_srv_rate	34	Real
28	dst_host_diff_srv_rate	35	Real
29	dst_host_same_src_port_rate	36	Real
30	dst_host_srv_diff_host_rate	37	Real

The ANFIS model is trained for the number of iterations. Each iteration comprises of forward and backward passes. The forward move is to regulate the consequent parameters and the reverse move is to regulate the values of the activation function. In the following i – represents the layer number.

The first layer is an input layer which forwards inputs to the next layer (the first hidden layer),

(6) $x_i \rightarrow y_i$,
 where x_i is input and y_i is output of the first layer.

The second layer is the fuzzification layer. The fuzzification neurons collect inputs from its corresponding antecedent neuron in the input layer and determine the degree to which the inputs belong to the fuzzy set. Hence,

$$(7) \quad y_i = f(x_i),$$

f is the fuzzyfication function.

This model uses the Gaussian membership functions which can be represented as

$$(8) \quad \text{Gaussmf: } (x; \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}}.$$

The next hidden layer (third layer) is the fuzzy rule layer. The neurons in this layer correspond to a single first order sugeno-type model. This layer calculates the truth value of the rule using product (AND) operator. Therefore,

$$(9) \quad y_i = \prod_c^m x^3 c_i,$$

where: $x^3 c_i$ is the input from a fuzzification layer neuron; y_i – the output of layer i ; m – the number of antecedents of a fuzzy layer.

The fourth layer is the defuzzification layer. The logical values received from the rule layer are received and the strength of this layer is measured using

$$(10) \quad y_i = \frac{x_d}{\sum_{j=1}^n x_d},$$

where x_d is the input from fuzzy rule layer neuron; y_i – the output of i -th layer; n – the number of neurons in i -th layer.

Each neuron in this layer is associated with the normalized value and accepts the inputs, such as (x_1, x_2, \dots, x_n) .

The consequent neuron in a defuzzification layer is determined as

$$(11) \quad y_i = x_i (k_i + k_i x_1 + k_i x_2 + \dots + k_i x_n),$$

where k_i is the set of parameters from rule layer.

The final layer is the summation layer and the result can be obtained by

$$(12) \quad y = \sum_{i=1}^n x_i,$$

where n is the number of defuzzyfication neuron.

Training the network is meant to minimize the error rate (error function). This function describes the error while doing classification on the training data.

Suppose that there is a set of m data points $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ then the output function (curve model function) is given as,

$$(13) \quad y = f(x, \beta),$$

where $\beta = (\beta_1, \beta_2, \dots, \beta_n)$ with $m \geq n$.

Find the parameter β such that the curve model fits the best data in the least square sense, that is the sum of squares where s should be minimized as (14)

$$(14) \quad s = \sum_{i=1}^m r_i^2,$$

where r is the residual (error value), r_i is given by the equation

$$(15) \quad r_i = y_i - f(x_i, \beta), \text{ for } i = 1, 2, \dots, m.$$

The S value can reach the minimum when the gradient is zero. This model consists of n parameters. They are n gradient equations.

$$(16) \quad \frac{\partial s}{\partial \beta_j} = 2 \sum_{i=1}^m r_i \frac{\partial r_i}{\partial \beta_j} = 0 \text{ for } j = 1, 2, \dots, m.$$

The gradient equations do not have a closed solution. In order to obtain a closed solution, the initial values for the parameters must be chosen. The initial values can be set by using the convergence criteria:

$$(17) \quad \frac{\Delta\beta_j}{\beta_j} < 0.001, \text{ for } j = 1, 2, \dots, n.$$

After the parameters can be refined iteratively, the values can be attained by consecutive approximation

$$(18) \quad \beta_j \approx \beta_j^{k+1} = \beta_j^k + \Delta\beta_j,$$

where k is the number of iteration, $\Delta\beta_j$ is the shift vector.

The model is linearized by approximation to Taylor series (of first order) at each iteration:

$$(19) \quad f(x_i, \beta) \approx f(x_i, \beta^k) + \sum_j \frac{\partial f(x_i, \beta^k)}{\partial \beta_j} (\beta_j - \beta_j^k) \approx f(x_i, \beta^k) + \sum_j J_{ij} \Delta\beta_j,$$

where J is the independent variable (Jacobian constant) in linear model.

Thus,

$$(20) \quad \frac{\partial r_i}{\partial \beta_j} = -J_{ij},$$

and the residuals are given by

$$(21) \quad r_i = \Delta y_i - \sum_{s=1}^n J_{is} \Delta\beta_s,$$

$$(22) \quad \Delta y_i = y_i - f(x_i, \beta^k).$$

By substituting Equations (21) and (22) in Equation (16), the equation becomes

$$(23) \quad -2 \sum_{i=1}^m J_{ij} (\Delta y_i - \sum_{s=1}^n J_{is} \Delta\beta_s) = 0.$$

After some rearrangements, the above equation forms n simultaneous linear equations:

$$(24) \quad \sum_{i=1}^m \sum_{s=1}^n J_{ij} J_{is} \Delta\beta_s = \sum_{i=1}^m J_{ij} \Delta y_i, \text{ for } j = 1, 2, \dots, n.$$

The normal equation can be represented by using a matrix notation, such as

$$(25) \quad (J^T J) \Delta\beta = J^T \Delta y.$$

When these are not equally reliable, the weighted sum of squares may get minimized.

$$(26) \quad S = \sum_{i=1}^m W_{ii} r_i^2,$$

Where W – diagonal weight matrix, W must be equal to the reciprocal of the error variance of the measurement. The normal equations are specified as

$$(27) \quad (J^T W J) \Delta\beta = J^T W \Delta y.$$

6. Experimental results

To implement the Hypervisor Detector, the proposed work uses the cloud simulator cloudsim 3.0. The Hypervisor Detector is trained and tested on cloudsim 3.0. To train and test the proposed system, DARPA's KDD cup dataset 1999 is used. This dataset has 41 features and a label specifying the record as either normal or an attack. Here, the attack types are categories as: 1) denial of a service: Denying the access of legitimate users by making some computing or memory resources too

busy; 2) probe (PRB): Scanning the host and port, to collect the information or to find the known vulnerabilities; 3) remote to local (R2L): Unauthorized users can access from a remote machine in order to exploit the host's vulnerabilities; 4) user to root (U2R): unauthorized access to a root machine which starts from a simple host machine attack. The proposed system Hypervisor Detector model uses using an ANFIS. To configure ANFIS model for a problem, it is necessary to specify the fuzzy rules and the membership function of a fuzzification neuron. For a fuzzy rule it takes the antecedents as fuzzy sets and the consequents are formed and adjusted by the learning algorithm. This system has been trained with 10% of KDD cup data values. Table 2 shows the sample distribution of the training data and Table 3 shows the sample distribution of the test data. The measurements are frequently proposed to evaluate the performance of an anomaly detection system as follows: 1) true positives; 2) true negatives; 3) false positives, and 4) false negatives. True positive designates the anomaly detection system which exactly detects the attacks occurred. True negative: This value designates the detection system which does not make a mistake to detect the normal condition. False positive: This value specifies that ADS mistakenly marking the normal condition as abnormal. If this value is consistently high, this causes the administrator to intentionally disregard the system warnings, which brings the system in a dangerous status. False negative indicates that the anomaly detection system is failing to detect the intrusions after a particular attack has occurred.

Table 2. Sample distribution of 10% training data

Class	Number of samples	% of samples
Normal	99 400	20.29 19.69
DoS	383 560	78.3 79.24
Probe	5603	1.14 0.83
R2L	1208	0.24 0.23
U2R	72	0.01
Total	489 843	100

Table 3. Sample distribution of 10% testing data

Class	Number of samples	% of samples
Normal	70 593	22.69
DoS	219 553	70.58
Probe	5389	1.73
R2L	15 164	4.87
U2R	328	0.1
Total	311 027	100

6.1. Performance analysis

The proposed model of a Hypervisor Detector is designed by using Adaptive neuro-fuzzy technology. The virtual network is simulated. The Hypervisor Detector monitors and examines the virtual traffic and the virtual host activities. The Hypervisor Detector monitors the activities of the virtual machines and is designed to capture the packets intended for cloud services. The performance of this implementation is based on how well the system is trained to detect the intrusions. The result is obtained in cloudsim 3.0. The ANFIS model which uses a sugeno type model and Fuzzy inference system is generated by using Grid pattern analysis which is trained for 150 epochs; the training stops at a training error rate about 0.02, which is designated as the best. The KDD data set is an extensively used data set for anomaly detection system which is the observation of the data captured in DARPA'98 IDS assessment. The Hypervisor Detector is tested with the KDD cup test dataset. The Hypervisor Detector implemented with ANFIS provides the best detection accuracy rate of 99.8%. The performance analysis shows how well the Hypervisor Detector has been trained to detect the abnormal behaviour. This paper compares the performance of the Hypervisor Detector with the model proposed by Vieira et al. [5] and Amjad, Sabyasachi and Debasish [6]. The authors Vieira et al. [5] use an artificial neural network for detecting anomalies in grid and cloud computing. Amjad, Sabyasachi and Debasish [6] use two approaches: 1) Naive Bayes classification, and 2) hybrid approach which combines Naive Bayes classifier and random forest technique for the anomaly detection. It can be found that Hypervisor Detector designed with ANFIS shows fast convergence because of its hybrid learning approach and its simple interpretation. The results of the performance comparison in terms of detection accuracy and false negatives are shown in Figs 3 and 4. From this performance analysis, it can be shown that the Hypervisor Detector which is implemented with an Adaptive neuro learning fuzzy inference system can detect the unusual activities with high detection accuracy and minimum error rate.

Since we are using minimum number of instances [27] for measuring the standard performance measure, these instances are not sufficient. Due to this reason, the precision, recall and F -values which are not dependent on the size of the input samples could be used.

They can be defined as follows:

$$(28) \quad \text{precision} = \frac{TP}{TP+FP'}$$

$$(29) \quad \text{Recall} = \frac{TP}{TP+FN'}$$

$$(30) \quad F\text{-value} = \frac{(1+\beta^2)*\text{recall}*\text{precision}}{\beta^2*(\text{recall}+\text{precision})}$$

where TP is the number of true positive; FP – the number of false positive; FN – the number of false negative; β – the Relative importance of precision vs. recall ($\beta=1$).

Table 4. Performance comparison of normal under various techniques

Techniques	Naïve bayes	NBRF	ANN	ANFIS
Precision	90.21	92.12	90.69	93.72
Recall	97.04	98.52	97.32	99.26
F-value	92.41	94.32	92.56	96.67

Table 5. Performance comparison for DoS under various techniques

Techniques	Naïve bayes	NBRF	ANN	ANFIS
Precision	93.21	98.62	94.39	99.77
Recall	93.55	96.48	94.76	98.82
F-value	95.78	97.02	96.13	98.26

Table 6. Performance comparison for Probe under various techniques

Techniques	Naïve bayes	NBRF	ANN	ANFIS
Precision	56.23	74.34	63.2	77.3
Recall	84.27	87.65	85.73	91.3
F-value	65.8	76.42	72.3	78.7

Table 7. Performance comparison for R2L under various techniques

Techniques	Naïve bayes	NBRF	ANN	ANFIS
Precision	41.67	65.56	55.35	94.49
Recall	7.46	25.61	8.32	61.4
F-value	10.83	33.56	11.67	74.32

Table 8. Performance comparison for U2R under various techniques

Techniques	Naïve Bayes	NBRF	ANN	ANFIS
Precision	27.56	62.6	55.54	83.30
Recall	7.54	34.18	26.67	84.61
F-value	11.72	53.48	31.53	90.43

Besides, for the performance analysis and comparison, precision, recall and F-values can be used. Tables 4-8 show that the intrusion detection system using ANFIS is the best which has the highest detection accuracy and the minimum false positives. The detection rate and false alarm rate of the proposed system is compared with ANN [5], Naive Bayes classifier and NBRF [6] which are shown in Figs 3 and 4 respectively.

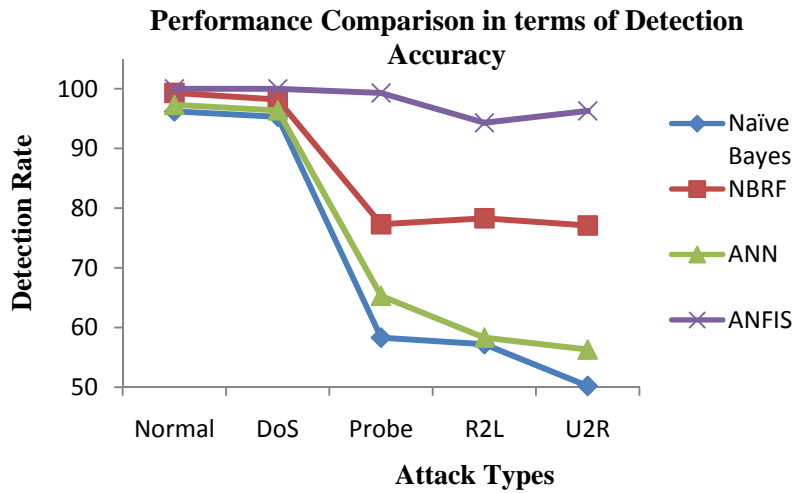


Fig. 3. Performance comparison of a Hypervisor Detector against models [5] and [6]

Performance Comparison in terms of False Alarm Rate

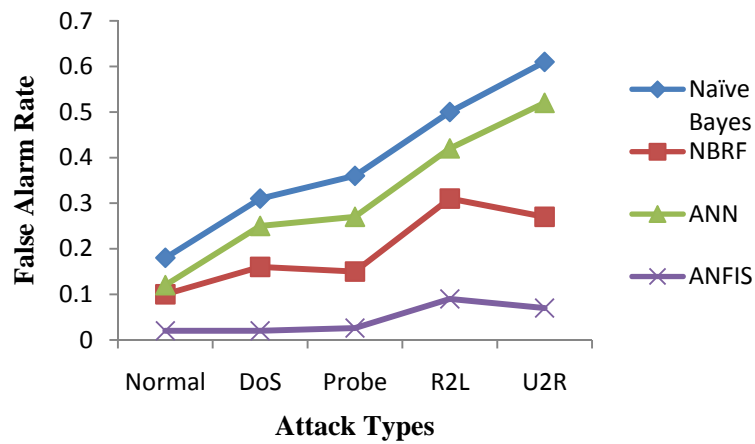


Fig. 4. Performance comparison of Hypervisor Detector against models [5] and [6]

From Fig. 3 it is observed that the proposed system gives high detection accuracy for both high frequent and low frequent attacks. Fig. 4 shows, that the proposed system produces low false alarms when compared to NB, NBRF and ANN. This shows that the proposed model Hypervisor Detector is very efficient and effective to detect the malware activities in cloud environment with a minimum mean squared error.

7. Conclusion

This work delineates the anomaly detection system (Hypervisor Detector) in cloud computing environment. The Hypervisor Detector is designed with ANFIS to detect anomalies in a cloud virtual network; ANFIS uses back propagation gradient descent technique in mixture with the least square scheme for training and testing the system. The experimental and performance comparison results indicate that the proposed model is well-designed, efficient and effective to disclose the anomalies in a virtual network. Hence, the proposed model can be a well suited model to detect the anomalies with minimum error (mean squared error < 0.03). Hence, having a Hypervisor Detector shows the best performance results that can be achieved in virtual environment to detect the anomalies with high detection accuracy.

References

1. Kevin, S. Security in a Virtualized World. – Journal of Network Security, Vol. 8, 2009, pp. 15-18.
2. Praveen Kumar, P., K. Bhaskar Naik. A Survey on Cloud Based Intrusion Detection System. – International Journal of Software and Web Sciences, Vol. 4, 2013, No 2, pp. 98-102.
3. Sanjay Ram, M., N. Velmurugan, S. Thirukumaran. Effective Analysis of Cloud Based Intrusion Detection System. – International Journal of Computer Applications & Information Technology, Vol. 1, 2012, No 2, pp.16-22.
4. Jin, H., G. Xiang, D. Zou, S. Wu, F. Zhoa, M. Li et al. A VMM-Based Intrusion Prevention System in Cloud Computing Environment. – Journal of Supercomputing, Springer Science Business Media, LLC, Vol. 66, 2013, No 3, pp. 1133-1151.
5. Vieira, K., A. Schuller, C. Westphall, C. Westphall. Intrusion Detection Techniques in Grid and Cloud Computing Environment. – Proc. IEEE IT Professional Magazine, Vol. 12, 2010, No 4, pp. 38-43.
6. Amjad, H. B., P. Sabyasachi, J. Debashish. Machine Learning Approach for Intrusion Detection on Cloud Virtual Machines. – International Journal of Application or Innovation in Engineering & Management (IJAIEM), Vol. 2, 2013, No 6, pp. 57-66.
7. Garfinkel, T., M. Rosenblum. A Virtual Machine Introspection Based Architecture for Intrusion Detection. – In: Proc. of Network and Distributed Systems Security Symposium, 2003, pp. 191-206.
8. Amirreza, Z., Z. Alireza. Internet Intrusion Detection System Service in a Cloud. – International Journal of Computer Science Issues, Vol. 9, 2012, No 5, pp. 308-315.
9. Kourai, K., S. Chiba. HyperSpector: Virtual Distributed Monitoring Environments for Secure Intrusion Detection. – In: Proc. of International Conference on Virtual Execution Environments, ACM, Chicago, 2005, pp. 197-207.
10. Jones, A. K., R. S. Sielken. Computer System Intrusion Detection: A Survey. – Techreport, September 2000, pp. 1-25. doi=10.1.1.24.7802.
11. Parag, K. S., S. Sneha, A. D. Gawande. Intrusion Detection System for Cloud Computing. – International Journal of Scientific & Technology Research, Vol. 1, 2012, No 4, pp. 67-71.
12. Dunlap, G. W., S. T. King, S. Cinar, M. Basrai, P. M. Chen. Revirt: Enabling Intrusion Analysis through Virtual Machine Logging and Replay. – In: Proc. of 5th Symposium on Operating Systems Design and Implementation, USENIX, Boston, 2002, pp. 211-224.
13. Feng, Z., H. Jin. Automated Approach to Intrusion Detection in VM-Based Dynamic Execution Environment. – Computing and Informatics, Vol. 31, 2012, pp. 271-297.

14. Vikrant, G. D., G. B. Atul, A. A. Nikhil. Intrusion Detection System for Cloud Computing. – International Journal of Engineering Research & Technology (IJERT), Vol. 2, 2013, No 4, pp. 2149-2153.
15. Zeenat, M., A. Chetan, S. H. Syed et al. Intrusion Detection in Cloud Computing Environment Using Neural Network. – International Journal of Research in Computer Engineering and Electronics, Vol. 1, 2012, No 1, pp. 1-4.
16. Ubhale, P. R., A. M. Sahu. Securing Cloud Computing Environment by Means of Intrusion Detection and Prevention System (IDPS). – International Journal of Computer Science and Management Research, Vol. 2, 2013, No 5, pp. 2430-2435.
17. Otte, C., C. Tormann. Improving the Accuracy of Network Intrusion Detectors by Input-Dependent Stacking. – Integrated Computer-Aided Engineering, Vol. 18, 2011, No 3, pp. 291-297.
18. Nirimala, A. P., R. Sridaran. Cloud Computing Issues at Design and Implementation Levels – A Survey. – International Journal of Advanced Networking and Applications, Vol. 3, 2012, No 6, pp. 1444-1449.
19. Farzad, S. Secure Virtualization for Cloud Environment Using Hypervisor-Based Technology. – International Journal of Machine Learning and Computing, Vol. 2, 2012, No 1, pp. 39-45.
20. Vinothina, V., R. Sridaran, G. Padmavathi. A Survey on Resource Allocation Strategies in Cloud Computing. – International Journal of Advanced Computer Science and Applications, Vol. 3, 2012, No 6, pp. 97-104.
21. Jang, J. S. R. ANFIS: Adaptive-Network-Based Fuzzy Inference Systems. – IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, 1993, No 3, pp. 665-685.
22. Guler, I., E. D. Ubeyli. Application of Adaptive Neuro-Fuzzy Inference System for Detection of Electrocardiographic Changes in Patients with Partial Epilepsy Using Feature Extraction. – Expert Systems with Applications, Vol. 27, 2004, pp. 323-330.
23. Guler, I., E. D. Ubeyli. Adaptive Neuro-Fuzzy Inference System for Classification of EEG Signals Using Wavelet Coefficients. – Journal of Neuroscience Methods, 2005, pp. 1-9.
24. Fallahpour, A. R., A. R. Moghassem. Yarn Strength Modeling Using Adaptive Neuro-Fuzzy Inference System (ANFIS) And Gene Expression Programming (GEP). – Journal of Engineered Fibers and Fabrics, Vol. 8, 2013, No 4, pp. 6-18.
25. Tavallaee, M., E. Bagheri, L. Wei, A. Ghorbani. Detailed Analysis of the KDD CUP 99 Data Set. – In: Proc. of IEEE Symposium on Computational Intelligence in Security and Defense Applications, Ottawa, 2009, pp. 1-6.
26. Loganathan, C., K. V. Girija. Hybrid Learning for Adaptive Neuro Fuzzy Inference System. – International Journal of Engineering and Science, Vol. 2, 2013, No 11, pp. 6-13.
27. Gang, W., H. Jinxing, M. Jian, H. Lihua. A New Approach to Intrusion Detection Using Artificial Neural Networks and Fuzzy Clustering. – Expert Systems with Applications, Vol. 37, 2010, No 9, pp. 6225-6232.