Evaluation of IDS using Neural Network over Cloud

Manoj Kumar Soni[1] Mrs. Megha Singh[2]
[1]M.Tech Scholar [2]Head of Dept.
[1,2]Department of Computer Science
[1,2]CIIT, Indore (M.P.), India

Abstract— In this modern world evaluation of cloud computing has come forward for combination of logical entities like data and software which are accessible through internet. Innovations are necessary to ride the inevitable tide of change. Most of organizations are discord to decreases their computing cost through the means of virtualization. This demand of reducing the cost has led to the moderns of Cloud Computing. Cloud Computing offers better computing through improved exploitive and diminished administration and infrastructure costs. Cloud Computing is the sum of Software as a Service and subservience Computing. Cloud Computing is still at its infant stage and a very new technology for the enterprises. An ID is a vital component to maintain network security. Also, as the cloud platform is speedily evolving and become most popular in our day to day life, it is helpful and necessary to build effective IDS for the cloud computing. However, existing IDS will be likely to face challenges when deployed on the cloud platform. The predestined Intrusion Detection System architecture may lead to overburden of a part of the cloud due to the extra detection overhead. This thesis proposes a neural network based Intrusion Detection System that is a distributed system with an adaptive architecture, so as to make full use of the available resources without overloading any single machine in the cloud. Moreover, with the machine learning ability from the neural network, the proposed IDS can detect new types of attacks with fairly exact results. Evaluation of the introduced IDS with the KDD dataset on a physical cloud tested shows that it is a liberal of promises approach to detecting attacks in the cloud computing infrastructure.

Key words: Cloud Computing, SaaS, IaaS, PaaS, Elasticity, KDD, Security, ANN

I. INTRODUCTION

Cloud computing is a large-scale distributed computing paradigm driven by economies of scale and outsourcing, where a pool of abstracted, virtualized, dynamically-scalable, managed computing, storage, services are delivered on demand over the Internet. Cloud computing technology is enabling IT professionals to disposal services to user‘s quick and a more flexible and Economical way without having to redesign the infrastructure. [1] Moreover, cloud consumers can enjoy on-demand service with the Pay-As-You-Go plan. The National Institute of Standard and Technology (NIST) defines Cloud Computing as the subscription for convenient, on request mechanism to a common pool of configurable computing resources that can be rapidly provisioned and released with minimum effort or interaction [2] Security in the cloud is a joint responsibility between the cloud provider and the cloud clients who own the data. The clients also need to make sure that they are in full control regarding the protection methods for their data. Furthermore, the reputation of the cloud providers can be damaged due to misconduct behaviors from the cloud customers themselves in such a fate-sharing environment. Therefore, cloud providers need to protect their customers and themselves against these new types of risks. The Cloud Security Alliance has defined seven top threats to cloud computing systems [3] namely:

1) Abuse and Nefarious Use of Cloud Computing.
2) Insecure APIs.
3) Malicious intruders.
4) Shared Technology Issues.
5) Data Loss or Leakage.
6) Account or Service Hijacking.
7) Unknown Risk Profile.

II. BACKGROUND

A. Intrusion Detection System

Intrusion Detection System (IDS) is a security technology, which can detect, prevent and possibly react to computer attacks. IDSs have proven to be effective tools in conventional local and wide area networks. In a typical network scenario, IDS generates alerts regarding security threats and logs them for further analysis. Then a network administrator can decide to rely on the IDS judgment and take an action or let the system react through a predefined plan. The concept of IDS was first introduced in 1986 by Dorothy Denning [4]. Since then, IDSs have evolved from standalone hardware to a piece of software to a virtual machine (VM) instance which can run on virtual environments like the clouds. Typically, many organizations require an automated process to monitor various events occurring on their network assets. Therefore, it can provide the needed protection against external intruders and internal users who are taking advantage of their privileged accounts. Accordingly, the incorporation of IDS in any network is vital. The location of the IDS is an important factor to in achieving efficient monitoring. In a typical network layout, the IDS box can be placed along with other essential security tools like the access control module and anti-virus server just behind the corporate firewall (Fig 1).

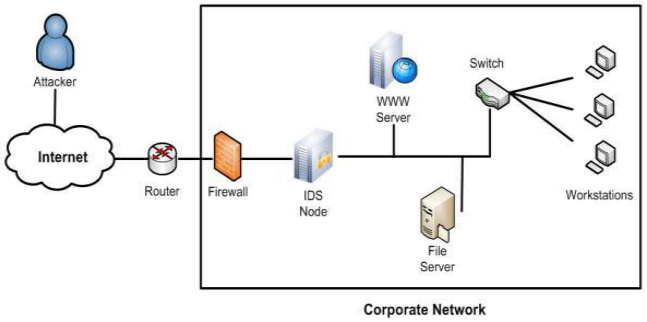

Fig. 1: IDS Placement in Typical Network

A major distinction between the IDS and a firewall is that the former will continuously monitor the internal part of the network as well as protecting it from internal and external threats. On the other hand, firewalls act like a conditional barrier that only allow defined services, ports or IP addresses to pass through them. However, once an

intruder bypasses the firewall, it is hard to stop or recognize the origin of the attack.

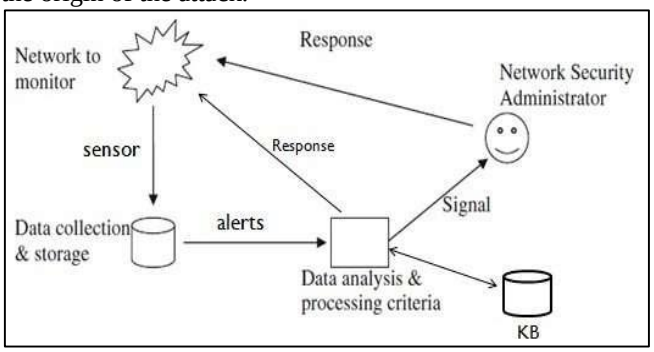

Fig. 2: IDS Detection Process

IDS technology has been proposed as an efficient security measure and is nowadays widely adopted for securing critical IT-Infrastructures. According to the protected objectives, IDSs can be categorized to three main categories namely [5]:

1) Network-based Intrusion Detection Systems (NIDS).
2) Host-based Intrusion Detection Systems (HIDS).
3) Distributed Intrusion Detection Systems (DIDS).

A few papers have proposed some IDSs frameworks for cloud systems. Some of these frameworks target SaaS service model, the others adapt some traditional techniques such as mobile agents.

B. Artificial Neural Network

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been provide to analyze. This specialist can then be used to give projections given new situations of interest and answer "what if" questions. Other advantages include:

1) Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
2) Self-Organization: An ANN can create its own organization/structure or representation of the information/data it receives during learning time.
3) Real Time Operation: ANN computations may be carried out in parallel process, and for that special kind of hardware devices are being designed and manufactured which take advantage of this capacity.
4) Fault Tolerance through Redundant Information/data Coding: Partial destruction of a network leads to the corresponding fall of performance. However, some network capacities may be retained even with major network damage.

III. RELATED WORKS

For learning process [6], supervised learning technique is efficient to build classifiers. As previously mentioned, it can take advantage of the known target outputs to train the classifier to perform classification. Supervised learning method based on support vector machine was proposed by Yang et al. [7]. The results showed the high detection rate whereas low false alarm rate, but there are some crucial problems on selects of the best attribution and reduction the feature space. Then, such problems can be resolved by using selecting best minimum feature. Terrence [8] applied genetic algorithm to feature subset of data for generating fuzzy rule. Then, fuzzy logic is applied to calculate the fitness function used to define the normal or abnormal behavior of network system. Results show that performance of such technique could reduce the false alarm rate. But, complexity of algorithm is high. Li [9] proposed the neural network classifier including two parts of process. The first part used 41features for training data and second part classified data by using 3 layers feed-forward neural network model. But there are 41 features used that are in a very large amount. Preecha Somwang [10] proposed SFAM (Simplified Fuzzy Adaptive resonance theory Map) classifier and PCA (Principal Component Analysis) use for selecting features that reduces the dimensionality of KDD dataset 21 features were selected out of 41 features. Amjad Hussain Bhat[11] proposed Naïve Bayes Tree (NB Tree) hybrid approach of NB Tree and Random Forest. NB Tree algorithm applied to Data Set and its result is provided to Random Forest Algorithm which is a classification algorithm. M.Madhavi[12] proposed the architecture of IDS over Cloud scenario to secure cloud services.

IV. INTRUSION DETECTION DATASET

The KDD 99 intrusion detection datasets are based on the 1998 DARPA initiative that provides all designers of intrusion detection systems with a benchmark on which to evaluate various methodologies. For that a simulation was made of a factitious military network consisting of three target machines running different operating systems and services. Additional three machines are then used to spoof various IP addresses to produce traffic. At last, there is a sniffer that records all network traffic using the TCP dump format. All simulated duration is seven weeks. Normal and small connections are created to profile that expected in a military network and then all attacks comes into one of four categories: User to Root, Remote to Local, Denial of Service, and Probe.

- Denial of Service (dos): Intruders tries to fend legal users/client from using a service.
- Remote to Local (r2l): Attacker doesn`t have any account on victim machine, here upon tries to gain access.
- User to Root (u2r): Attacker has local access to the victim machine and endeavors to gain super user perquisite.
- Probe: Attacker tries to gain information about the target host.

In Year 1999, the original TCP dump files were preapproved for utilization in the Intrusion Detection System benchmark of the International Knowledge Discovery Tools and Data Mining Tools Competition. To do so, packet information in the TCP backlash log file is summarized into bond. Specifically, a connection is a sequence of TCP packets starting and ends at any well allocated times, between which data flows from a source IP address to a target IP address under some well-defined protocol.

Features are grouped into four categories:

A. Basic Features:

Basic features can be retrieved from data packet headers without observing the payload. Content Features: Domain knowledge is used to define the payload of the original TCP packets. This includes features such as the number of failed login attempts; Time-based Traffic Features: These features are designed to capture properties that mature over a 2 second temporal window. One example of such a feature would be the number of connections to the same host over the 2 second interval.

There are total 41 Features available for KDD Training Data Set. They all classified into various categories. A complete list of the set of features defined for the connection records is given in the four tables, basic features, content features, traffic features and host based features table.

Table 1 shows information for the basic features of 9 individual features of TCP connections.

No.	Feature Name	Description
1	Duration	Length (number of seconds) of the connection
2	Protocol_type	Type of the protocol, e.g. tcp, udp, etc.
3	Service	Network service on the destination, e.g., http, telnet, etc.
4	Flag	Normal or error status of the connection
5	Src_bytes	Number of data bytes from source to destination
6	Dst_bytes	Number of data bytes from destination to source
7	Land	1 if connection is from/to the same host/port; 0 otherwise
8	Wrong_fragment	Number of ``wrong'' fragments
9	Urgent	Number of urgent packets

Table 1: Basic features of individual TCP connections [10]

No.	Feature Name	Description
10	Hot	Number of ``hot'' indicators
11	Num_failed_logins	Number of failed login attempts
12	Logged_in	1 if successfully logged in; 0 otherwise
13	Num_compromised	Number of ``compromised'' conditions
14	Root_shell	1 if root shell is obtained; 0 otherwise
15	Su_attempted	1 if ``su root'' command attempted; 0 otherwise
16	Num_root	Number of ``root'' accesses
17	Num_file_creations	Number of file creation operations
18	Num_shells	Number of shell prompts
19	Num_access_files	Number of operations on access control files
20	Num_outbound_cmds	Number of outbound commands in an ftp session
21	Is_hot_login	1 if the login belongs to the ``hot'' list; 0 otherwise
22	Is_guest_login	1 if the login is a ``guest'' login; 0 otherwise

Table 2: Content features by domain knowledge. [10]

Table 2 shows information for the content features within a connection suggested by domain knowledge.

The data schema of the traffic features computed using a two-second time window, as shown in Table 3.

No.	Feature Name	Description
23	Count	Number of connections to the same host as the current connection in the past two seconds
24	Srv_count	Number of connections to the same service as the current connection in the past two seconds
25	Serror_rate	% of connections that have ``SYN'' errors, S0 error rate
26	Srv_serror_rate	% of connections that have ``SYN'' errors, S0 error rate for the same service as the current one
27	Rerror_rate	% of connections that have ``REJ'' errors, RST error rate
28	Srv_rerror_rate	% of connections that have ``REJ'' errors, RST error rate for the same service as the current one
29	Same_srv_rate	% of connections to the same service
30	Diff_srv_rate	% of connections to different services
31	Srv_diff_host_rate	% of connections to different hosts

Table 3: Traffic features. [10]

Table 4 shows information for the host-based features from the communication of source address to destination address connection.

No.	Feature Name	Description
32	Dst_host_count	Count of connections having the same destination.
33	Dst_host_srv_count	Count of connections having the same destination host and using the same service.
34	Dst_host_same_srv_rate	% of connections having the same destination host
35	Dst_host_diff_srv_rate	% of different services on the current host.
36	Dst_host_same_src_port_rate	% of connections to the current host having the same src port.
37	Dst_host_srv_diff_host_rate	% of connections to the same service coming from different host.
38	Dst_host_serror_rate	% of connections to the current host that have an S0 error.
39	Dst_host_srv_serror_rate	% of connections to the current host and specified service that have an S0 error
40	Dst_host_rerror_rate	% of connections to the

		current host that have an RST error.
41	Dst_host_srv¬_rerror_rate	% of connections to the current host and specified service that an RST error.

Table 4: Host-based features [10]

There are 4 attacked class types of IDS of this experimental model, presented in the Table 5 [13].

Class	Attack
DoS	back, land, neptune, pod, smurf, teardrop
Probe	ipsweep, nmap, portsweep, satan
U2R	buffer_overflow,loadmodule, perl, rootkit
R2L	ftp_write, guess_passwd, phf, imap,multihop, warezmaster, Warezclient

Table 5: Data attack type [13].

V. THE PROPOSED IDS

Based on the cloud platform has been introduced above, the ANN-based IDS will be established. In the architecture, there is one manager VM and multiple worker VMs in the network. The manager VM monitors the load information for the worker VMs and decides the mapping of ANN on the worker VMs dynamically. That is, those worker VMs having certain amounts of resources available will be chosen to perform the intrusion detection task, and the worker VMs are assigned to the input layer, hidden layer and output layer to form an ANN. The input layer in the proposed ANN structure is responsible for collecting data from the network. All the requests or data flow in the network should first be collected by those nodes and then be passed through the whole neural network for any malicious activities. The hidden layer receives the raw data from the input layer and processes them based on the ANN mechanism discussed in ANN, and forwards the results to the output layer. This layer will also modify weight values of the input layer after each iteration and pass those updated values to the input layer. The output layer derives the final result based on the intermediate results received from the hidden layer. It also updates weight values for the hidden layer and sends them to the hidden layer to improve the overall network behavior. As mentioned previously, the architecture shown in Figure 3 is proposed for improving the system flexibility, which is also important to enhance the robustness of IDS. When one node in the IDS is unavailable due to situations such as deadlock, power off, and scarce resources, the IDS is able to adjust itself accordingly to form a new capable architecture.

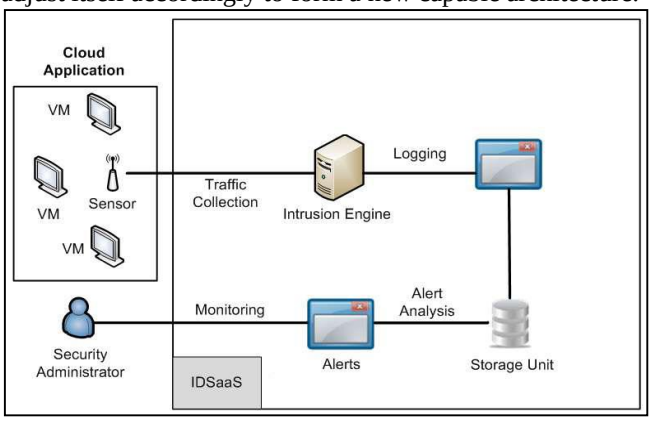

Fig. 3: Architecture of the proposed IDS

Figure 3 shows the process flow for the multi-threaded manager process. When a client joins the IDS, it will raise a thread and connect to the server, the server will then store the thread into the queue with the address and port number. Once the network connection is established, all the clients will send the resource usage information periodically to the manager so as to select the most appropriate nodes to construct the IDS. After the IDS is built, all the other IDS nodes will receive the message from the manager and run the corresponding (input, hidden, output or wait) function based on the conditional statement. The structure of the IDS can be adjusted through the manager, which sends messages to the candidate nodes to build a new IDS structure as requested. All the models are trained off-line before they are deployed.

A. Data Pre-Processing

Data pre-processing is the process of cleansing incomplete data of involved mapping symbolic-valued attributes to numeric-valued attributes. This process is implemented non-zero numerical features of variables for intrusion detection dataset [14].

B. Feature Selection

Feature selection is the process of selecting a subset of relevant features for use in model construction. Given the benchmark data from KDD Cup'99 dataset, which is an original complete 41feature.but some more important features used for attack include feature number 13, 15, 17, 22 and 40 namely Num_compromised, Su_attempted, Num_file_creations, Is_guest_login and Dst_host_rerror_rate. These features are used as an input for Neural Network in Training and Testing.

VI. RESULT AND ANALYSIS

As per the below cited Table shows that the result of KDD Data Set applying on our IDS using Neural Network over Cloud found that the Accuracy of Normal Data type is 97%, DOS is 96%, Probe is 94%, R2L is 98% and U2R found 92% accurate.

Attack	Original Number of Samples	Number of Accurate Samples	Actual % of Accurate Samples
Normal	97277	94358	96.99 %
DoS	391458	375799	95.99 %
Prob	4107	3860	93.98 %
R2L	1126	1103	97.95 %
U2R	52	48	92.30 %
Total Accuracy:			96.44 %

Table 6: Result Analysis

The proposed method and the other techniques were simulated on Microsoft windows 7 operating system by using MATLAB toolbox. Classification accuracy is usually measured in terms of precision, recall, and F-measure. First of all, two basic measures (precision and recall) are explained for a given document.
These are computed as follows:

$$Precision = \frac{\text{categories found and correct}}{\text{total categories found}} = \frac{TP}{TP + FP}$$

$$Recall = \frac{\text{categories found and correct}}{\text{total categories correct}} = \frac{TP}{TP + FN}$$

True Positive (TP) refers to the number of examples that are classified correctly as belonging to the class, while False Positive (FP) stands for the number of incorrectly classified examples. False Negative (FN) is the number of examples we incorrectly classify as negative.

For our evaluation, the F-measure is used as a metric for effectiveness of classification. The F-measure is defined as follows:

$$F\ measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

The F-measure was designed to balance weights of precision and recall. The F measure values are in the interval (0, 1) and larger F-measure values correspond to higher classification quality. The measure is a popular metric for evaluating classification systems and is most often used to compare the performance of classifiers.

Attack	Precision	Recall	F-measure
Normal	0.96	0.19	0.31
DoS	0.95	0.79	0.86
Prob	0.93	0.74	0.82
R2L	0.97	0.53	0.68
U2R	0.92	0.48	0.63

Table 7: Identification results

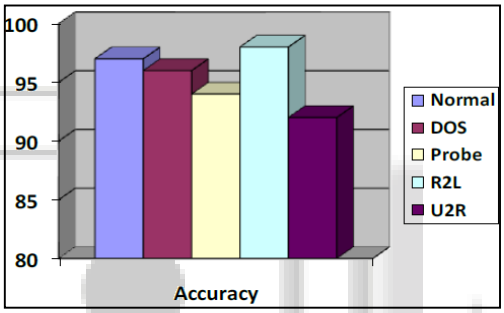

Fig. 1: Graph 1 Accuracy of System over Data Set

VII. Conclusion

Advanced soft computing and artificial intelligence methods/techniques are being used widely in Intrusion. Detection Systems (IDS) for acquiring the ability to learn and evolve, which makes them more accurate and efficient in the presence of enormous number of unpredictable attacks. In this thesis, a neural network based IDS are built on a cloud platform. The accuracy of the implemented IDS is shown to be high and the time expense is acceptable. Implementation of the neural network in the cloud for intrusion detection is a promising direction.

References

[1] Hisham A. Kholidy, Abdelkarim Erradi, Sherif Abdelwahed and Fabrizio Baiardi, "Autonomous Response, Self-Resilience, and Prediction in a Cloud IDS", International Journal of Science, Engineering and Technology Research (IJSETR), Volume 2, pp. 1403-1406, April 2013.

[2] NIST, "Cloud Computing Synopsis and Recommendations", Available: http://csrc.nist.gov/publications/nistpubs/800-146/sp800-146.pdf, pp. 554-561, May 2012.

[3] Hisham A. Kholidy and Fabrizio Baiardi, "CIDS: A framework for Intrusion Detection in Cloud Systems", The 9th International Conf. on Information Technology: New Generations (ITNG), Las Vegas, Nevada, USA, pp. 379-385. Available: http://www.di.unipi.it/~hkholidy/projects/cids/,2012.

[4] Sundas Juma, Zaiton Muda, M.A. Mohamed and Warusia Yassin, "Machine Learning Techniques For Intrusion Detection System", Journal of Theoretical and Applied Information Technology, Vol.72 No.3, pp. 422-429 28th February 2015.

[5] Roschke, S., Cheng and F., Meinel, "Intrusion Detection in the Cloud", The 8th International Conference on Dependable, Autonomic and Secure Computing (DASC) Chengdu, China, pp. 12-14, December 2009.

[6] Meyn S., Surana A., Lin Y., and Narayanan S.,"Anomaly Detection Using Projective Markov Models in a Distributed Sensor Network," in Proceedings of the 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference, Shanghai, pp. 4662-4669, 2009.

[7] Yang C., Yang H., and Deng F., "Quantum- Inspired Immune Evolutionary Algorithm based Parameter Optimization for Mixtures of Kernels and its Application to Supervised Anomaly IDSs," in Proceedings of the 7th World Congress on Intelligent Control and Automation, Chongqing, pp. 4568-4573, 2008.

[8] Terrence F., "Evolutionary Optimization of a Fuzzy Rule-based Network Intrusion Detection System," in Proceedings of Annual Meeting of the North American Fuzzy Information Processing Society, Toronto, pp. 1-6, 2010.

[9] Li X., "Optimization of the Neural-Network- Based Multiple Classifiers Intrusion Detection System," in Proceedings of International Conference on Internet Technology and Applications, Wuhan, pp. 1-4, 2010.

[10] Preecha Somwang and Woraphon Lilakiatsakun, "Anomaly Traffic Detection Based on PCA and SFAM", The International Arab Journal of Information Technology, Vol. 12, Issue 03, pp. 253-260, May 2015.

[11] Amjad Hussain Bhat, Sabyasachi Patra and Dr. Debasish Jena, "Machine Learning Approach for Intrusion Detection on Cloud Virtual Machines", IJAIEM, Volume 2, Issue 6, pp. 57-66, June 2013.

[12] M.Madhavi, "An Approach for Intrusion Detection System In Cloud Computing", IJCSIT, Volume 3, Issue 5, 2012.

[13] Devaraju S. and Ramakrishnan S., "Performance Analysis of Intrusion Detection System using Various Neural Network Classifiers," in Proceedings of International Conference on Recent Trends in Information Technology, Chennai, Tamil Nadu, pp. 1033-1038, 2011.

[14] Gou S., Wang Y., Jiao L., Feng J., and Yao Y., "Distributed Transfer Network Learning based Intrusion Detection," in Proceedings of International Symposium on Parallel and Distributed Processing with Applications, Chengdu, pp. 511-515, 2009.

