Imperial Journal of Interdisciplinary Research (IJIR)
Vol-2, Issue-6, 2016
ISSN: 2454-1362, http://www.onlinejournal.in

Intrusion Detection System for Networks Using K-Means ANN Algorithm

Pankaja Alappanavar[1], Kumargaurav Shewale[2], Yogesh Markad[3]& Nilesh Patil[4]

[1]Asst. Prof., Department of Information Technology, SAOE, SPPU, Pune, India
[2,3,4]Department of Information Technology, SAOE, SPPU, Pune, India.

Abstract*: Intrusion detection is considered as a process of monitoring and analyzing the events occurring in a computer system in order to detect signs of security problems. Due to large increase in number of cyber attacks, building effective intrusion detection systems are essential for protecting information systems security, and till now it remains an elusive goal and a big challenge. We present an efficient approach of intrusion detection systems and address some of the research challenges to design efficient and effective intrusion detection systems. In existing IDS system, the rule sets of various attack patterns are stored in databases and the whole network traffic is matched against it to avoid any unauthorized and illegal activities. if any attack happens, and the pattern of that attack is not stored in IDS rule sets then that attack pattern is need to be manually updated in the database to prevent it next time. Hence we are proposing such a system in which there is no need to manually updating the attack pattern in IDS rule sets, the proposed 'K-Means ANN' algorithm will automatically capture the patterns of new attack and store it in IDS database which will reduces the human time as well as effort to learn new attacks patterns manually.*

Introduction

With the coming of Internet age, network security has become the key foundation to web applications, such as online retail sales, online auctions, etc. Intrusion detection attempts to detect computer attacks by examining various data records observed in processes on the network. It is one of the important ways to solve network security problems. Detection precision and detection stability are two main indicators to evaluate intrusion detection systems (IDS). In early stage, the research focus lies in using the rule-based expert systems and statistical approaches. But when encountering larger datasets, the results of rule-based expert systems and statistical approaches become worse. Therefore, a lot of data mining techniques have been introduced to solve the problem. Among these techniques, Artificial Neural Network (ANN) is one of the widely used techniques and has been successful in solving many complex practical problems. And ANN has been successfully applied into IDS. However, the main drawbacks of ANN-based IDS are: (1) lower detection precision, particularly for low-frequent attacks, e.g., Remote to Local (R2L), User to Root (U2R), and (2) weaker detection stability. For the above two aspects, the main reason is that the distribution of different types of attacks is imbalanced. For low-frequent attacks, the leaning sample size is too small compared to high-frequent attacks. It makes ANN difficult to learn the characters of these attacks and therefore detection precision is much lower. In practice, low-frequent attacks do not mean they are not important. Instead, serious consequences will be caused, if these attacks succeeded. For example, if the U2R attacks succeeded, the attacker can get the authority of root user and do everything he likes to the targeted computer systems or network device. Furthermore in IDS the low-frequent attacks are often outliers. Thus ANN is unstable as it often converges to the local minimum (Haykin, 1999). Although prior research has proposed some approaches, when encountering large datasets, these approaches become not effective.

To solve the above two problems, we propose a novel approach for ANN-based IDS, K-Means ANN, to enhance the detection precision for low-frequent attacks and detection stability. The general procedure of K-Means ANN approach has the following three stages. In the first stage, a K-Means clustering technique is used to generate different training subsets. Based on the different training sets, different ANNs are trained in the second stage. In the third step, the trained results and dataset is used for real time intrusion detection in on the networks. To

illustrate the applicability and capability of the new approach, the results of experiments on KDD CUP 1999 dataset demonstrated better performance compared in terms of detection precision and detection stability.

Methodology

The proposed system is based on the analysis of collected packets by using the Artificial Neural Network algorithms. Fig 1 shows the overall architecture of the proposed system. The IDS will have following components to achieve the required functionality:

Data Collection

This section gives an overview of the data set used for intrusion detection. This data set contains seven weeks of training data and it also contains two weeks of testing data. Raw data was about four gigabytes of compressed binary TCP dump data from the network traffic generated. This was TCP packets to and from some IP addresses, starting and processed into five million connection records. Each of these records is a vector of extracted feature values of that network connection. As we know, the connection is a sequence of ending at some well defined times. This data set containing five million connection records was used as the data set for the 1999 KDD intrusion detection contest and is called as the KDD Cup 99 data. Particularly, MIT Lincoln Labs' DARPA intrusion detection evaluation datasets have been employed

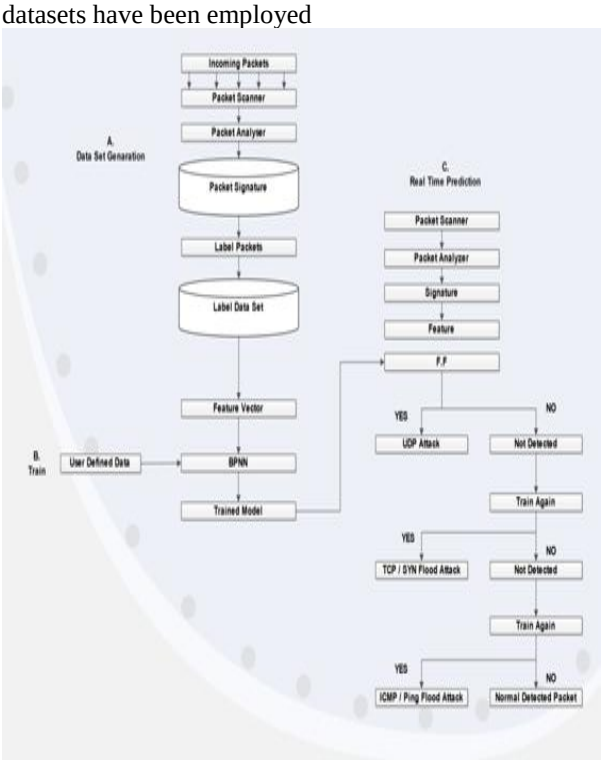

Figure1: Architecture of the System

design and test intrusion detection systems. In 1999, recorded network traffic from the DARPA 98 Lincoln Lab dataset was summarized into network connections with 41 features per connection. This formed the KDD 99 intrusion detection benchmark in the International Knowledge Discovery and Data Mining Tools Competition. The KDD 99 intrusion detection datasets are based on 1998 DARPA initiative, which provides designers of the intrusion detection systems (IDS) with a benchmark on to evaluate different methodologies [3]. To do this, a simulation is made of a factitious military network consisting of three "target" machines running various operating systems and services. Additional three machines are then used to spoof different IP addresses to generate traffic. Finally, there is a sniffer which records all the network traffic using the TCP dump format. The total simulation period is seven weeks. Each connection was labeled as normal or as one specific kind of attack. All the labels are assumed to be correct.

There were total 37 types of attacks in the data set. All the simulated attacks fell in exactly one of the four categories: Remote to Local; User to Root; Denial of Service; and Probe.

- Denial of Service (DoS): The attacker tries to prevent legitimate users from accessing a service.
- Remote to Local (r2l): The attacker does not have an account on the attacking machine. Hence, he tries to gain access to machine.
- User to Root (u2r): The attacker has local access to the victim machine and tries to gain super user privileges. Probe: Here, the Attacker tries to gain information about the target host.

Data pre-processing consists of following components including document conversion, feature selection and feature weighting.

The functionality of each component can be described as follows:

[1] Dataset prepared with DOS attack which include smurf, Neptune, back, teardrop and POD ping of death attacks /anomaly.

[2] Feature Selection: It reduces the dimensionality of the data space by removing irrelevant or less relevant feature selection criterion.

[3] Document Conversion: It will convert different types of documents such as gz, tcpdump to .csv file and .arff (Attribute-Relation File Format) data file format.

[4] Totally we considered 11850 data points for our experimentation.

KMeans Clustering

K-Means is one of the simplest unsupervised learning algorithms that solve the clustering problem. The objective is to classify a given data set into a certain number of clusters (assume initial clusters) fixed a priori.

The pseudo code for the adapted K-Means clustering algorithm is presented as below:

1. Choose random k data points as the initial Cluster Centroids.
2. Repeat
3. For each data point x from D
4. Then compute the distance of x from each cluster mean (centroid)
5. Assign x to the nearest cluster.
6. End for loop.
7. Again compute the mean for current cluster collections.
8. Until reaching stable cluster
9. Use these centroid for normal and anomaly traffic.
10. Calculate the distance of centroid from normal and anomaly centroid points.
11. If distance(X, Dj) > = 5
1. 12. Then anomaly found, exit
2. 13. Else
3. 14. X is a normal and it is not an Intrusion;
4. K-means clustering module can be summarized as follows:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

Where,

$\left\| x_i^{(j)} - c \right\|$ is a chosen distance measure between a data point and the cluster centroid, is an indicator of the distance of the n data points from their respective cluster Centroids.

2.3. Review of Artificial Neural Network

Feed forward neural network training is usually carried out using the called back propagation algorithm. Training the network with back propagation algorithm results in a non-linear mapping between the input and output variables. Thus, given the input/output pairs, the network can have its weights adjusted by the back propagation algorithm to capture the non-linear relationship. After training, the networks with fixed weights can provide the output for the training the network is based on the minimization of an energy function representing the instantaneous error.

2.4. KMeans ANN Framework:

For implementation of ANN algorithm on the dataset, training data is divided into several subsets using k-means clustering technique. Subsequently, it trains different ANN using different subsets. Then, it

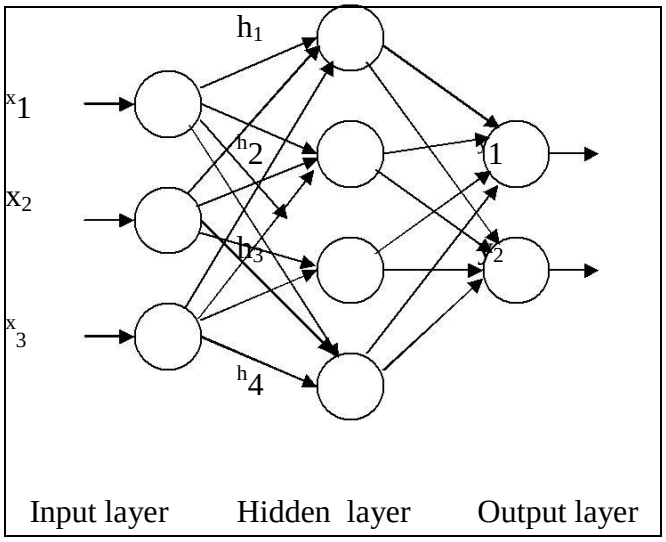

Figure 2: Architecture of Feed Forward Neural Network

determines membership grades of all these subsets and combines them using a new ANN to get final results. The whole framework of K-Means ANN is illustrated in Fig. 1. As typical machine learning framework, K-Means ANN incorporates both the training phase and testing phase. Training phase includes following three major stages:

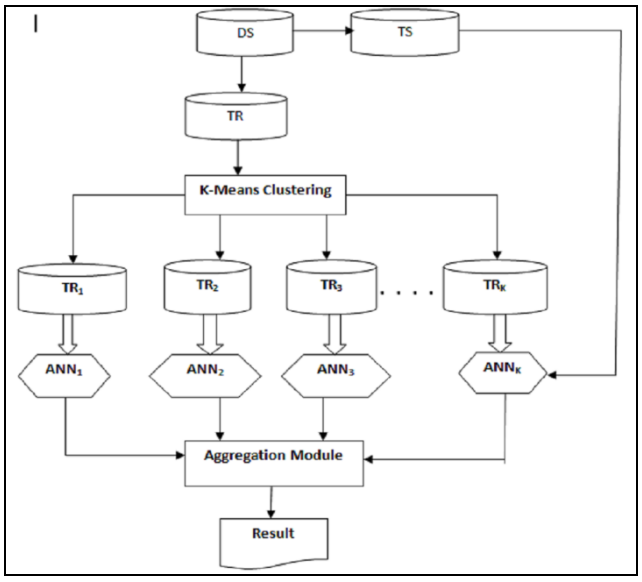

Figure 3: Framework for KMeans-ANN

Stage I :

For an arbitrary dataset DS, first it is divided into training set TR and testing set TS. Then, different training subsets TR_1, TR_2,... TR_k are created from TR with K-Means clustering module.

Stage II :

For each training subset TR_i (i=1, 2, ...k) , the ANN model, ANN_i ,(i=1,2,....k) is training by the specific learning algorithm to formulate k different base ANN models.

Stage III :

In order to reduce the error for every ANN_i, we simulate the ANN_i using the whole training set TR and get the results. Now, we use the membership grades, which were generated by k-means clustering module, to combine the results. Subsequently, we train another new ANN using the combined results. In the testing phase, we directly input the testing set data into the k different ANN_i and get outputs. Based on these outputs, the final results can then be achieved by the last aggregation module.

3. Mathematical Model

Let S be the Set of System,
S={s, e, X, Y, Fs, ANN, DS, success, failure}
Where,
S = s is the Start State of the System
E = e is the end State of the System
X = Set of Input Parameter
X= {X1}
Where,
X1= {input to the system is packets generated by the Packet Generator}
Fs = {Functions used in the Program}
Fs = {F1, F2 , F3, F4}
Let, F1 be used to detect the bad packet that is trying to intrude in the system.
F2 be used to generate the packets through packet generator.
F3 be used to maintain the control between the software design and the hardware component
F4 be used to capture and view the bad packets captured.
Y = {Output to the System}
Y = {successfully Display the bad packets that are being captured by our system.}
Success Case:
It is the case when all bad packets that are trying to intrude are detected and captured.
Failure Case:
It is the case when all bad packets that are trying to intrude are not detected and displayed.

3.1 State Diagram:

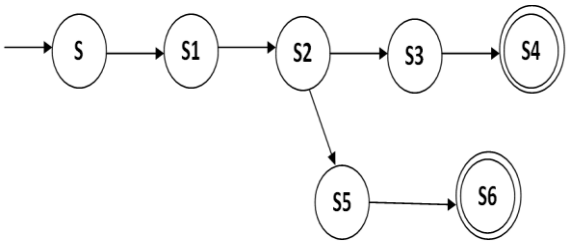

Figure 4: State diagram for system

4. Implementation and results

After the successful implementation of the proposed methodology, we were able to see the following results as expected:

4.1. Packet Capturing

This module will be implemented to capture the incoming packets from the network which can be a local network or web. the user interface of the form will show the detailed parameters of the captured packets like hop limit, acknowledgement flag as shown in the Figure 5.

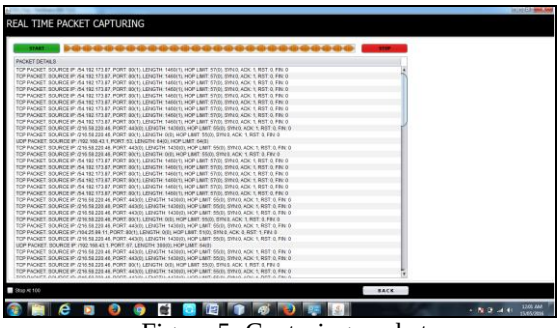

Figure 5: Capturing packets

4.2. Train data set

In the train data set module, signatures of captured packets are added into the training data set with the help of their parameter details. In the data set these signature are compared to decide the intrusions in the system. Implementation is shown in the Figure 6.

Figure 6: Train data set

4.3 Intrusion detection form

In this module, packets are analyzed to make the decisions if the detected incoming packets are intruded packets or not according to the trained data set which contain both normal packet and intruded packet signatures as shown in Figure 7.

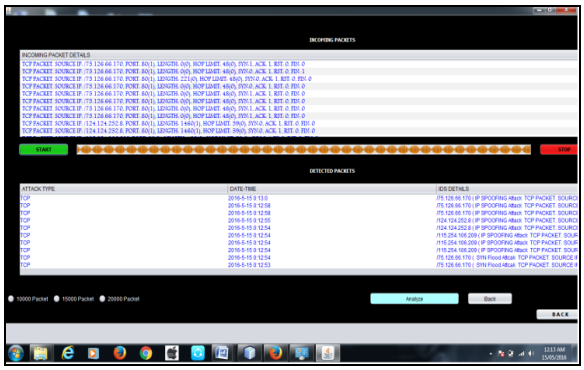

Figure 7: Intrusion detection process

4.4 Final Analysis

In the analysis the system will perform final analysis of the session to make the calculations to see which type of attack is detected more times in given time slot, as shown in Figure 8.

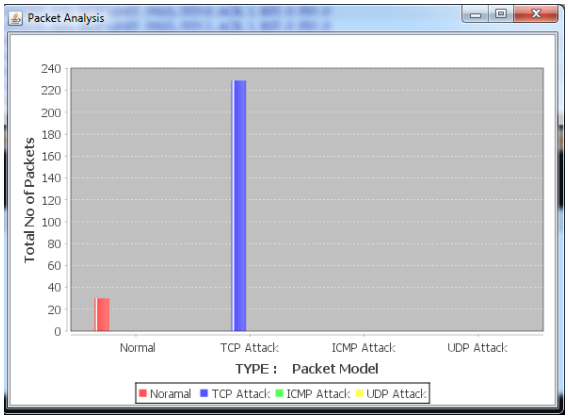

Fig 8 Final analysis

5. Future Scope

The proposed intrusion detection system based on IDS ANN can be used by antivirus software for pattern analysis and threat detection for various network based systems. The system can also be configured with windows firewall to detect and block access of malicious sites and resources.

6. Acknowledgements

I and my co-authors are really thankful to our guide for providing us all the guidance that we needed to complete the research work. We are also really thankful to our head of department for providing all the facilities and guidance to us.

7. References

[1] S. Sontakke, "Intrusion Detection System for Cloud Computing"," International Journal of Scientific & Technology Research" Volume **1**, Issue **4** (page no**. 67-71**), May **2012**.

[2] Swati Ramteke, Rajesh Dongare, Komal Ramteke, "Intrusion Detection System for Cloud Network Using FC-ANN Algorithm", International Journal of Advanced Research in Computer and Communication Engineering, Vol. **2**, Issue **4**, April **2013**.

[3] M J.M. Bonoficio, "Neural Networks Applied in Intrusion Detection Systems," IEEE World Congress on Computational Intelligence, Vol.**1, pp. 205-210, 1998**.

[4] X. L. Huang, "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering," School of Management, Fudan University, Shanghai 200433, PR China, Department of Information Systems, City University of Hong Kong, TatChee Avenue, Kowloon, Hong Kong.

[5] M. Hussain, "Distributed cloud intrusion detection model," International Journal of Advanced Science and Technology Vol**. 34** (page no. **71-82**), September, **2011**.

[6] J. Antony Jeyanna, E. Indumathi, Dr. D. Shalini Punithavathani, "A Network Intrusion Detection System Using Clustering and Outlier Detection", International Journal of Innovative Research in Computer and Communication Engineering, Vol. **3**, Issue **2**, February **2015.**