ISSN:0975-9646

A Review on: Advanced Artificial Neural Networks (ANN) approach for IDS by layered method

P. D. Somwanshi[1] and S. M. Chaware[2]
[1]*Research Scholar, Department of Computer Engineering, Bhivarabai Sawant College of Engineering and Research, Narhe, Pune, India-41*
[2] *Head of Department, Department of Computer Engineering, Bhivarabai Sawant College of Engineering and Research, Narhe, Pune, India-41*

Abstract— Security is a big issue for all networks in today's enterprise environment. Hackers and intruders have made many successful attempts to bring down high-profile company networks and web services. Many methods have been developed to secure the network infrastructure and communication over the internet; among them are the use of firewalls, encryption, and virtual private networks. Intrusion detection is a relatively new addition to such techniques. Using intrusion detection method we can find the intrusion signature in a network and must perform efficiently to manage with the large amount of network traffic. Existing system are label these two issues of correctness
and efficiency using Conditional Random Fields with Layered Approach. We show that high attack detection correctness can be achieved by using Conditional Random Fields and high efficiency by implementing the Layered Approach. In this review paper, we are modifying existing system and newly use modified ANN algorithm to improve the efficiency and correctness to detect intrusion than previous one.

Keywords—IDS, Layered Approach, ANN's method, Host Based.

I. INTRODUCTION

Now a day, intrusion detection is one of the high priority and demanding tasks for network administrators and security expert. There is a need to protect the networks from known intrusion (attacks) and also to find new and unseen attacks by developing more reliable and efficient intrusion detection systems. First purpose of intrusion detection system is to detect intruders; that is, unexpected, unwanted or unauthorized people or programs in a network. Since 1980's after the dominant research from Anderson [1] intrusion detection system was began. Intrusion detection systems are divided into network based, host based, or application based categorized. [2]. Also, intrusion detection systems can also be classified as signature based or anomaly based depending upon the method of attack detection. The signature-based systems to detect intrusion from previously known attacks while the anomaly-based systems learn from the normal data collected when there is no inconsistent activity [2].

After the introduction in Section I, related work with highlighted on various methods used for intrusion detection are described in Section II. We describe proposed method layered approach with ANN method in Section III. And in section IV, mention there advantages of proposed system. In Section V, describes conclusion and future work.

II. RELATED WORK

The domain of intrusion detection and network security has been around since late 1980s. Since then, a number of methods and frameworks have been proposed and many systems have been built to detect intrusions. In this section, we briefly discuss existing techniques and frameworks.

Naive Bayes classifier is one of the intrusion detection methods. It provides lower attack detection accuracy when the selection features are dependable [3]. Decision trees have also been used for intrusion detection [3]. The decision trees select the best features for each decision node during the construction of the tree based on some well-defined criteria [5]. In 2010[4], Kapil Kumar Gupta, Baikunth Nath and Ramamohanaroa Kotagiri presented "A frame work using a layered approach for intrusion detection". They have addressed two main issues of ID i.e. accuracy and efficiency by using conditional random fields and layered method [6]. They have shown that layered Conditional Random Fields (CRFs) have very high attack detection rate than previous one [7].

A. Limitations of Existing system

- Existing system is based on off-line system which was less efficient and accurate.
- Static off-line database.
- Layered approach of CRF is not support pipelining of layers in multi-core processors.
- Using Naïve Bayes when selection features are dependable then attack detection accuracy is lower.
- Using decision tree algorithms additional research is needed in data mining to improve, automate, and simplify decision tree method for use in industry.

III. PROPOSED METHOD

Due to the above limitations we have to develop modified or advanced artificial neural network (ANN) algorithm to improve intrusion detection system better than previous.

Neural network algorithms are merging now a day as a new artificial intelligence technique that can be applied to real life problems. Neural networks are a form of artificial intelligence that uses multiple artificial neurons, networked

work together to process information. This type of network has the capability to learn from structure, and extend results from data that has been previously entered into the network's knowledge base. This ability makes neural network applications really valuable in intrusion detection [8].

A. The Artificial Neuron:

A single artificial neuron can be implemented in many different ways. The general mathematic definition is as follows:

$$y(x) = g(\sum_{i=0}^{n} wi\ xi) \quad \text{.....1.1}$$

x is a neuron with **n** input dendrites ($x_0 \ldots x_n$) and one output axon y(x) and where ($w_0 \ldots w_n$) are weights determining how much the inputs should be weighted. **g** is an activation function that weights how powerful the output (if any) should be from the neuron, based on the sum of the input. If the artificial neuron should mimic a real neuron, the activation function **g** should be a simple threshold function returning 0 or 1. The output from the activation function is either between 0 and 1, or between -1 and 1, depending on which activation function is used. This is not entirely true, since e.g. the identity function, which is also sometimes used as activation function, does not have these limitations, but most other activation functions uses these limitations. The inputs and the weights are not restricted in the same way and can in principle be between -∞ and +∞, but they are very often small values centered around zero. The artificial neuron is also illustrated in figure 1.1

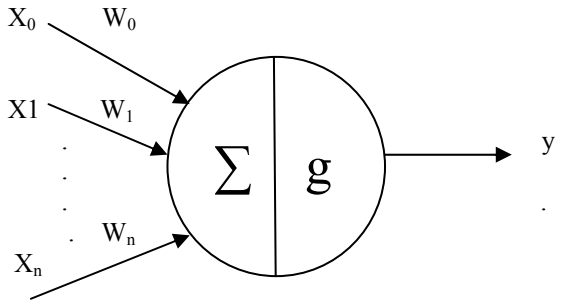

Fig 1.1 An artificial neuron.

According to Fig 1.1, the weights are not illustrated, but they are implicitly given by the number of pulses a neuron sends out, the strength of the pulses and how closely connected the neurons are.

B. The Artificial Neural Network (ANN):

In a multilayer feed-forward ANN, the neurons are ordered in layers, starting with an input layer and ending with an output layer. Between these two layers are a number of hidden layers. Connections in these kinds of network only go forward from one layer to the next.

Multilayer feed-forward ANNs have two different phases:

A training phase or learning phase and an execution phase. In the training phase the ANN is trained to return a specific output when given a specific input, this is done by continuous training on a set of training data. In the execution phase the ANN returns outputs on the basis of inputs.

The way the execution of a feed-forward ANN functions are the following:

An input is presented to the input layer, the input is propagated through all the layers (by equation 1.1) until it reaches the output layer, where the output is returned. In a feed-forward ANN an input can easily be propagated through the network and evaluated to an output. It is more difficult to compute a clear output from a network where connections are allowed in all directions (like in the brain), since this will create loops.

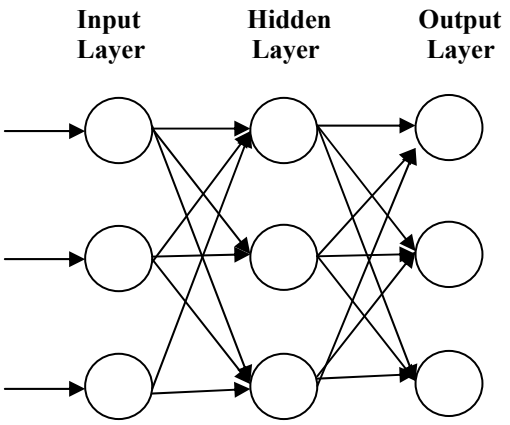

Fig 1.2 A fully connected multilayer feed-forward network with one hidden layer.

Fig 1.2 shows a multilayer feed-forward ANN where all the neurons in each layer are connected to all the neurons in the next layer. This is called a fully connected network and although ANNs do not need to be fully connected, they often are.

Two different kinds of parameters can be adjusted during the training of an ANN, the weights and the **t** value in the activation functions. This is impossible and it would be easier if only one of the parameters should be adjusted. To manage with this problem a bias neuron is invented. The bias neuron lies in one layer, is connected to all the neurons in the next layer, but none in the previous layer and it always emits 1. Since the bias neuron emits 1 the weights, connected to the bias neuron, are added directly to the combined sum of the other weights (by equation 1.1), just like the **t** value in the activation functions. A modified equation for the neuron, where the weight for the bias neuron is represented as w_{n+1}, is shown in equation 1.2.

$$y(x) = g(w_{n+1}\sum_{i=0}^{n} wi\ xi) \quad \text{..... 1.2}$$

Adding the bias neuron allows us to remove the **t** value from the activation function, only leaving the weights to be adjusted, when the ANN is being trained. We cannot remove the **t** value without adding a bias neuron, since this would result in a zero output from the sum function if all inputs where zero, regardless of the values of the weights. An ANN with added bias neurons is shown in Fig 1.3.

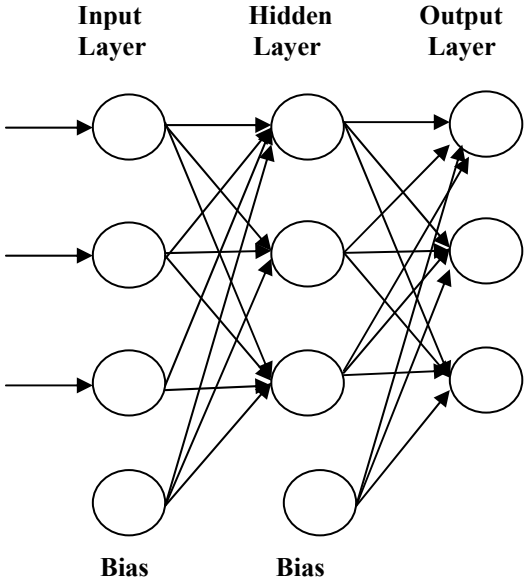

Fig 1.3 A fully connected multilayer feed-forward network with one hidden layer and bias neurons.

C. Algorithm of ANN:

Algorithm of ANN is divided into two parts:

1. Training
   Step 1: Select the number of **n** layers.
   Step 2: Select all features for each layer.
   Step 3: Train a separate model with ANN for each layer using the all features selected from Step 2.
   Step 4: Manage trained models sequentially such that only the connections labeled as normal are passed to the next layer.

2. Testing
   Step 5: For each (next) test instance perform Steps 6 through 9.
   Step 6: Test the instance and label it either as attack (intrusion) or normal.
   Step 7: If the instance is labeled as attack, block it and identify it as an attack represented by the layer name at which it is detected and go to Step 5. Else pass the sequence to the next layer.
   Step 8: If the current layer is not the last layer in the system, test the instance and go to Step 7. Else go to Step 9.
   Step 9: Test the instance and label it either as normal or as an attack (intrusion). If the instance is labeled as an attack, block it and identify it as an attack corresponding to the layer name.

IV. ADVANTAGES OF PROPOSED SYSTEM

- To detect all types of attacks (intrusion).
- More secure, reliable, accurate and efficient than previous one.
- Scope is unlimited.
- Host based online real time IDS advanced layered ANN's method.
- Dynamic real time database.
- Execution time is less.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed the new advanced artificial neural network (ANN) method to detect all types of attacks (intrusion) by host based online real time system. It is more accurate and efficient than previous one. The performance result shows that our proposed work overcomes the existing work with variations of differences.

REFERENCES

[1] J.P. Anderson, Computer Security Threat Monitoring and Surveillance, http://csrc.nist.gov/publications/histo ry/ande80.pdf, 2010.

[2] R. Bace and P. Mell, Intrusion Detection Systems, Computer Security Division, Information Technology Laboratory, Nat'l Inst. of Standards and Technology, 2001.

[3] N.B. Amor, S. Benferhat, and Z. Elouedi, "Naïve Bayes vs. Decision Trees in Intrusion Detection Systems," Proc. ACM Symp. Applied Computing (SAC '04), pp. 420-424, 2004.

[4] Kapil Kumar Gupta, Baikunth Nath and Ramamohanaroo kotagiri, "A layered approach using conditional random fields for intrusion detection", IEEE Tranc. on Dependence and secure computing, Vol.7, 2010.

[5] Jeff Markey. Using Decision Tree Analysis for Intrusion Detection: A How-to Guide. SANS Institute.

[6] K.K. Gupta, B. Nath, and R. Kotagiri, "Network Security Framework," Int'l J. Computer Science and Network Security, vol. 6, no. 7B, pp. 151-157, 2006.

[7] K.K. Gupta, B. Nath, and R. Kotagiri, "Conditional Random Fields for Intrusion Detection," Proc. 21st Int'l Conf. Advanced Information Networking and Applications Workshops (AINAW '07), pp. 203-208, 2007

[8] Herv´e Debar, Monique Becke, Didier Siboni, "A Neural Network Component for an Intrusion detection system," Proceedings of the IEEE Symposium on Research in Security and Privacy, pp. 240–250, 1992.