International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-6, Issue-2)

Research Article | **February 2017**

Hybrid Approch for Intrusion Detection Using ANN Technique

Pradhnya Kamble, R. C. Roychaudhary
Department of Computer Science & Engineering, St Vincent Pallotti College of Engineering & Technology, Nagpur, Maharashtra, India

Abstract—

In recent years, internet and computers have been utilized by many people all over the world in several fields. In order to come up with efficiency and up to date issues, most organizations rest their applications and service items on internet. On the other hand, network intrusion and information safety problems are ramifications of using internet. The growing network intrusions have put companies and organizations at a much greater risk of loss. In this paper, we propose two a new learning methodology towards developing a novel intrusion detection system (IDS) by
1) Back propagation neural networks (BPN)
2) Extreme Learning Machine(ELM).
The main function of Intrusion Detection System is to protect the resources from threats. It analyzes and predicts the behaviors of users, and then these behaviors will be considered an attack or a normal behavior. There are several techniques which exist at present to provide more security to the network, but most of these techniques are static.

Keywords— Intrusion Detection, ELM, BPN, Neural Network

I. INTRODUCTION

The ubiquity of the Internet poses serious concerns on the security of computer infrastructures and the integrity of sensitive data. With the rapid expansion of computer networks during the past decade, security has become a crucial issue for computer systems. Different soft-computing based methods have been proposed in recent years for the development of intrusion detection systems.

The problem of protecting information has existed since information has been managed. However, as technology advances and information management systems become more and more powerful, the problem of enforcing information security also becomes more critical . The enlargement of this electronic environment comes with a corresponding growth of electronic crime where the computer is used either as a tool to commit the crime or as a target of the crime.

Tradinal neural networks have been extensively used in many fields due to their ability: (1) to approximate complex nonlinear mappings directly from the input samples; and (2) to provide models for a large class of natural and artificial phenomena that are difficult to handle using classical parametric techniques. On the other hand, there lack faster learning algorithms for neural networks.

The traditional learning algorithms are usually far slower than required. It is not surprising to see that it may take several hours, several days, and even more time to train neural networks by using traditional methods. Proposed learning algorithm can be easily implemented, tends to reach the smallest training error, obtains the smallest norm of weights and the good generalization performance, and runs extremely fast, in order to differentiate it from the other popular SLFN (single-hidden layer feedforward neural networks)learning algorithms, it is called the extreme learning machine in the context of this paper.

In past years, numerous computers are hacked because they do not consider the necessary of precautions to protect against network attacks. The failure to secure their systems puts many companies and organizations at a much greater risk of loss.

Usually, a single attack can cost millions of dollars in potential revenue. Moreover, that's just the beginning. The damages of attacks include not only loss of intellectual property and liability for compromised customer data (the time/money spent to recover from the attack) but also customer confidence and market advantage. There is a need to enhance the security of computers and networks for protecting the critical infrastructure from threats. Accompanied by the rise of electronic crime, the design of **safe-guarding** information infrastructure such as the intrusion detection system (IDS) for preventing and detecting incidents becomes increasingly challenging. Figure 1 illustrates the intrusion detection system and external/internal network intrusion attacks.

The intrusion detector learning task is to build a predictive model (i.e. a classifier) capable of distinguishing between bad intrusions and normal connections. Recently, an increasing amount of research has been conducted on applying neural networks to detect intrusions. An neural network consists of a collection of processing elements that are highly interconnected. Give a set of inputs and a set of desired outputs, the transformation from input to output is determined by the weights associated with the interconnections among processing elements. By modifying these interconnections, the network is able to adapt to the desired outputs. The ability of high tolerance for learning-by-example makes neural networks flexible and powerful in IDS. However, the time required to induce the model from a large dataset is long. It can accurately predict probable attack behavior in IDS.

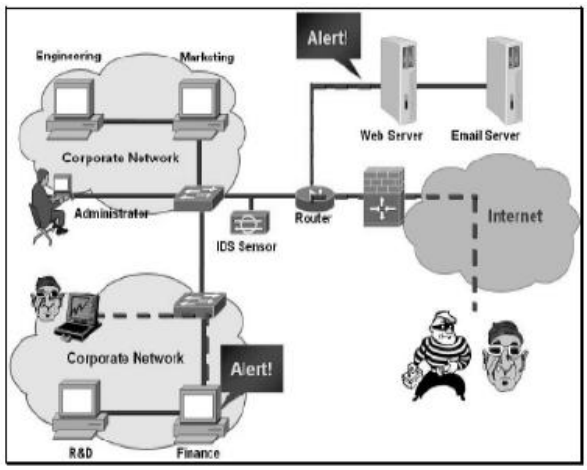

Figure 1: The Intrusion Detection System and External/Internal Network Intrusion Attacks.

This objective of system is to presents a new approach of intrusion detection system based on Back propagation neural networks (BPN) and Extreme Learning Machine(ELM). Multi Layer Percepron (MLP) architecture is used for Intrusion Detection System. The performance and evaluations are performed by using the set of benchmark data from a KDD (Knowledge discovery in Database) dataset.

Different from traditional learning algorithms the proposed learning ELM algorithm not only tends to reach the smallest training error but also the smallest norm of weights. Bartlett's theory on the generalization performance of neural networks states for neural networks reaching smaller training error, the smaller the norm of weights is, the better generalization performance the networks tend to have. Therefore, the proposed learning algorithm tends to have good generalization performance for neural networks.

II. LITERATURE SURVEY

Paper: Intrusion detection using neural networks and support vector machines
Author: S. Mukkamala, G. Janoski and A. Sung

Description: At first the concept of intrusion detection system was suggested by Anderson (1980) . He applied statistic method to analyze user's behavior and to detect those attackers who accessed system in an illegal manner.

Paper: Extreme learning machine: Theory and applications
Author: Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew

Description: This paper proposed a simple and efficient learning algorithm for single-hidden layer feedforward neural networks (SLFNs) called extreme learning machine (ELM), which has also been rigorously proved in this paper. The proposed ELM has several interesting and significant features different from traditional popular gradient-based learning algorithms for feedforward neural networks.

Paper: The application of extreme learning machines to the network intrusion detection problem
Author: Gideon Creech and Frank Jiang

Description: The Extreme Learning Machine (ELM) algorithm conventionally suffers from the inferior batch training performance. In this paper, a new approach to combine ELM outputs is proposed with a view to further develop a persistent IDS. Specifically, this paper proposes the application of an Extreme Learning Machine based approach to the network-based intrusion detection system (IDSs). Good performance is achieved and preliminary results are reported in this paper.

Paper: Research in intrusion detection systems: a survey
Author: S. Axelsson

Description: In the author proposed a data mining framework for constructing intrusion detection models. The key idea is to apply data mining programs namely, classification, meta-learning, association rules, and frequent episodes to audit data for computing misuse and anomaly detection models that accurately capture the actual behavior (i.e., patterns) of intrusions and normal activities.

Although, proposed detection model can detect a high percentage of old and new PROBING and U2R attacks, it missed a large number of new DOS and R2L attacks.

Paper: Host-based intrusion detection using user signatures
Author: S. Freeman, A. Bivens, J. Branch and B. Szymanski

Description: Reference is mostly focused on data mining techniques that are being used for such purposes, and then presented a new idea on how data mining can aid IDSs by utilizing biclustering as a tool to analyze network traffic and enhance IDSs.

Paper: State transition analysis: A rule-based intrusion detection approach
Author: K. Ilgun, R.A. Kemmerer and P.A. Porras

Description :Reference [6] proposed a new weighted support vector clustering algorithm and applied it to the anomaly detection problem. Experimental results show that mentioned method achieves high detection rate with low false alarm rate.

Intrusion detection attacks are segmented into two groups,

- Host-based attacks and
- Network-based attacks .

In case of host-based attacks, the intruders aim at a particular machine and attempt to get access to privileged services or resources on that particular machine. Detection of these kind of attacks typically uses routines that acquire system call data from an audit-process which monitors all system calls made with the support of each user. In case of network-based attacks, it is extremely complicated for legitimate users to use various network services by purposely occupying or disrupting network resources and services.

Paper: A. Research in intrusion detection and response - a survey
Author: Kabiri P, Ghorbani

Description: A few number of taxonomies of IDS proposed by different researchers at different time have been describe in this paper like 1) Host-Based Intrusion Detection 2) Network-Based Intrusion Detection 3) Misuse based detection 4) Anomaly based detection.

III. PROPOSED METHODOLGY

Back Propagation Working:

The back propagation algorithm is a quite essential one of the neural network. The algorithm is the training or learning algorithm rather than the network itself. A Back Propagation network learns by example.

You give the algorithm examples of what you want the network to do and it changes the network's weights so that, when training is finished, it will give you the required output for a particular input. Back Propagation networks are ideal for simple Pattern Recognition and Mapping Tasks4. As just mentioned, to train the network you need to give it examples of what you want the output you want (called the Target) for a particular input as shown in Figure 2

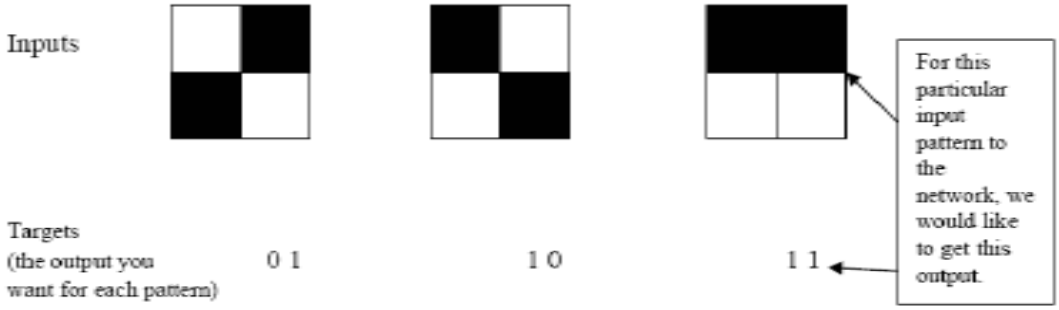

Figure 2 A Back Propagation Training Set.

So, if the first pattern to the network, we would like the output to be 0 1 as shown in figure 2 (a black pixel is represented by 1 and a white by 0 as in the previous examples). The input and its corresponding target are called a Training Pair.

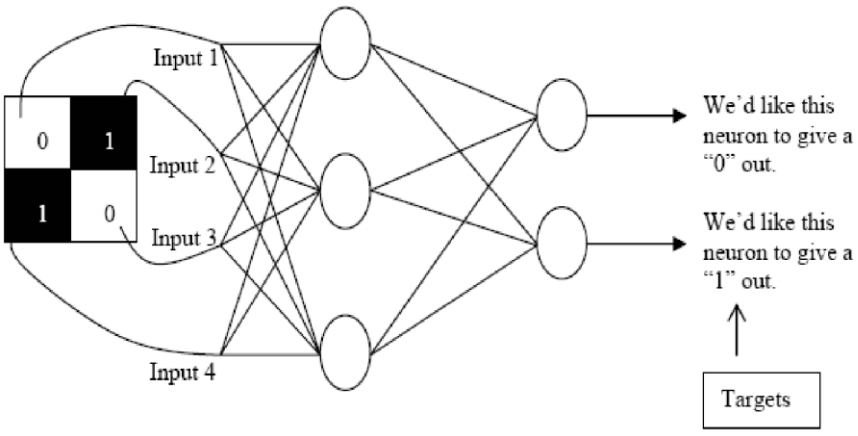

Fig 3:Applying a Training Pair to a Network

Once the network is trained, it will provide the desired output for any of the input patterns. Let's now look at how the training works. The network is first initialized by setting up all its weights to be small random numbers – say between –1 and +1. Next, the input pattern is applied and the output calculated (this is called the forward pass).

The calculation gives an output which is completely different to what you want (the Target), since all the weights are random. Then calculate the Error of each neuron, which is essentially: Target – Actual Output (i.e. what you want – What you actually get). This error is then used mathematically to change the weights in such a way that the error will get smaller. In other words, the Output of each neuron will get closer to its Target (this part is called the reverse pass). The process is repeated again and again until the error is minimal. Let's do an example with an actual network to see how the process works. Just look at one connection initially, between a neuron in the output layer and one in the hidden layer, figure 4.

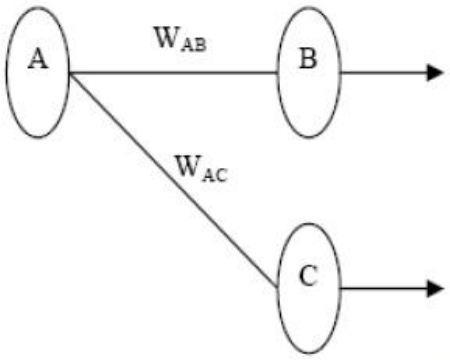

Figure 4: A Single Connection Learning in a Back Propagation network.

The connection interested in is between neuron A (a hidden layer neuron) and neuron B (an output neuron) and has the weight WAB. The diagram also shows another connection, between neuron A and C, but we'll return to that later.

The algorithm works like this:

1. First apply the inputs to the network and work out the output – remember this initial output could be anything, as the initial weights were random numbers.
2. Next work out the error for neuron B. The error is what you want – What you actually get, in other words:
   ErrorB = OutputB (1-OutputB) (TargetB – OutputB)
   The "Output (1-Output)" term is necessary in the equation because of the Sigmoid Function – if we were only using a threshold neuron it would just be (Target –Output).
3. Change the weight. Let W+AB be the new (trained) weight and WAB be the initial weight.
   W+AB = WAB + (ErrorB x OutputA)
   Notice that it is the output of the connecting neuron (neuronA) we use (not B). We update all the weights in the output layer in this way.
4. Calculate the Errors for the hidden layer neurons. Unlike the output layer we can't calculate these directly (because we don't have a Target), so we Back Propagate them from the output layer (hence the name of the algorithm). This is done by taking the Errors from the output neurons and running them back through the weights to get the hidden layer errors. For example if neuron A is connected as shown to B and C then we take the errors from B and C to generate an error for A.
   ErrorA = Output A (1 - Output A) (ErrorB WAB + ErrorC WAC) Again, the factor "Output (1 - Output)" is present because of the sigmoid squashing function.
5. Having obtained the Error for the hidden layer neurons now proceed as in stage 3 to change the hidden layer weights. By repeating this method we can train a network of any number of layers. This may well have left some doubt in your mind about the operation, so let's clear that up by explicitly showing all the calculations for a full sized network with 2 inputs, 3 hidden layer neurons and 2 output neurons as shown in figure 3.4. W+ represents the new, recalculated weight, where as W(without the superscript) represents the old weight. Reverse process is done in the same way. Hence the attackers are calculated.

ELM Working:

(1) When the learning rate Z is too small, the learning algorithm converges very slowly. However, when Z is too large, the algorithm becomes unstable and diverges.
(2) Another peculiarity of the error surface that impacts the performance of the BP learning algorithm is the presence of local minima . It is undesirable that the learning algorithm stops at a local minima if it is located far above a global minima.
(3) Neural network may be over-trained by using BP algorithms and obtain worse generalization performance. Thus, validation and suitable stopping methods are required in the cost function minimization procedure.
(4) Gradient-based learning is very time-consuming in most applications.

Aim of this paper is to resolve the above issues related with gradient-based algorithms and propose an efficient learning algorithm for feedforward neural networks.

Unlike the traditional function approximation theories which require to adjust input weights and hidden layer biases, input weights and hidden layer biases can be randomly assigned if only the activation function is infinitely

differentiable. It is very interesting and surprising that unlike the most common understanding that all the parameters of BP need to be adjusted, the input weights wi and the hidden layer biases bi are in fact not necessarily tuned and the hidden layer output matrix H can actually remain unchanged once random values have been assigned to these parameters in the beginning of learning. For fixed input weights wi and the hidden layer biases bi,, to train an BP is simply equivalent to finding a least squares solution B of the linear system.

ELM algorithm introduced in this paper, shows that BP with N sigmoidal hidden nodes and with input weights randomly generated but hidden biases appropriately tuned can exactly learn N distinct observations. Hidden nodes are not randomly generated, although the input weights are randomly generated, the hidden biases need to be determined based on the input weights and input training data.

IV. CONCLUSION

Test the proposed method by a benchmark intrusion dataset to verify its feasibility and effectiveness. Results show that choosing good attributes and samples will not only have impact on the performance, but also on the overall execution efficiency. The proposed method can significantly reduce the training time required. Additionally, the training results are good. It provides a powerful tool to help supervisors analyze, model and understand the complex attack behavior of electronic crime.

It is clear that the learning speed of feed forward neural networks is in general far slower than required and it has been a major bottleneck in their applications for past decades. Two key reasons behind may be: (1) the slow gradient-based learning algorithms are extensively used to train neural networks, and (2) all the parameters of the networks are tuned iteratively by using such learning algorithms. Unlike these conventional implementations, this paper proposes a new learning algorithm called extreme learning machine (ELM) for single-hidden layer feed forward neural networks (SLFNs) which randomly chooses hidden nodes and analytically determines the output weights of SLFNs. In theory, this algorithm tends to provide good generalization performance at extremely fast learning speed.

REFERENCES

[1] Chi Cheng, "Extreme learning machines for intrusion detection", Neural Networks (IJCNN), The 2016 International Joint Conference on 10-15 June 2016.
[2] Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew, "Extreme learning machine: Theory and applications", NeuroComputing, December 2015.
[3] P.L. Bartlett, The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network, IEEE Trans. Inf. Theory 44 (2) (1998) 525–536.
[4] D. Anderson, T. Frivold and A. Valdes, "Nextgeneration intrusion detection expert system (NIDES): a summary", Technical Report SRI-CSL-95-07. Computer Science Laboratory, SRI International, Menlo Park, CA, 1995.
[5] S. Axelsson, "Research in intrusion detection systems: a survey", Technical Report TR 98-17 (revised in 1999). Chalmers University of Technology, Goteborg, Sweden, 1999.
[6] S. Freeman, A. Bivens, J. Branch and B. Szymanski, "Host-based intrusion detection using user signatures", Proceedings of the Research Conference. RPI, Troy, NY, 2002.
[7] K. Ilgun, R.A. Kemmerer and P.A. Porras, "State transition analysis: A rule-based intrusion detection approach", IEEE Trans. Software Eng, Vol. 21, No. 3, Pp. 181–199, 1995.
[8] D. Marchette, "A statistical method for profiling network traffic", Proceedings of the First USENIX Workshop on Intrusion Detection and Network Monitoring, Santa Clara, CA, Pp. 119–128, 1999.
[9] Anderson, D., Frivoid, T. & Valdes Next-generation Intrusion Detection Expert System (NIDES): A Summary. SRI International Technical Report SRI-CSL-95-07, 1995.
[10] Sammany, M, Sharawi, M, El-Beltagy, M &Saroit, I. (2007). Artificial Neural Network Architecture for Intrusion Detection Systems and Classification of Attacks. Faculty of Computers and Information Cairo University. Retrieved October 18, 2011, from http://infos2007.fci.cu.edu.eg/Computational%20Intelligence/071777.pdf.
[11] Tavallaee, M, Bagheri, E, Lu, W & Ghorbani, A. (2009). A Detailed Analysis of Computational Intelligence in Security and Defence applications (CISDA 2009). Retreived October 25, 2011, from http:www.tavallaee.com/publications/CISDA.pdf.
[12] Fahlman, Scott E. and Lebiere, Christian: The Cascade- Correlation Learning Architecture, Neural Information Processing Systems 2, page 524-532, 1990
[13] Witten, I. H., and Frank E. (1999) Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, San Francisco.
[14] [KDD Cup network intrusion dataset, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[15] A. Pelc. Detecting errors in searching games. Journal of Combinatorial Theory Series A, 51(1):43–54, 1989...

