Intrusion Detection Using Artificial And Hierarchical Neural Networks

Sudhakar Hallur[1], Roopa Kulkarni[2], Dr. Prashant. P. Patavardhan[3]
[1, 2]Assistant Professor, Dept of Electronics and Communication Engineering
[3] Professor, Dept of Electronics and Communication Engineering
[1, 2, 3] KLSGIT, Belagavi

Abstract- *Intrusion Detection Systems (IDSs) provide an important layer of security for computer systems and networks, and are becoming more and more necessary as reliance on Internet services increases and systems with sensitive data are more commonly open to Internet access. An IDS's responsibility is to detect suspicious or unacceptable system and network activity and to alert a systems administrator to this activity. Classification algorithms are used to discriminate between normal and different types of attacks. In this paper, a comparative study between the performances of recent nine artificial neural networks (ANNs) based classifiers is evaluated, based on a selected set of features. The results showed that; the Multilayer perceptrons (MLPS) based classifier provides the best results; about 99.63% true positive attacks are detected. It is an Artificial Neural Network that supports an ideal specification of an Intrusion Detection System and is a solution to the problems of traditional IDSs. Therefore, An Artificial Neural Network inspired by nervous system has become an interesting tool in the applications of Intrusion Detection Systems due to its promising features. Intrusion detection by Artificial Neural Networks is an ongoing area. This paper describes results concerning the robustness and generalization capabilities of artificial neural networks in detecting intrusions using network audit trails. Through a variety of comparative experiments, it is found that neural network performs the best for intrusion detection.*

Keywords- Component; artificial neural networks; Multilayer perceptrons Back Propagation algorithm; Radius basis functions; Hierarchical neural network

I. INTRODUCTION

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or a network. There are two general categories of attacks which intrusion detection technologies attempt to identify - anomaly detection and misuse detection. Anomaly detection identifies activities that vary from established patterns for users, or groups of users. Anomaly detection typically involves the creation of knowledge bases that contain the profiles of the monitored activities.

The second general approach to intrusion detection is misuse detection. This technique involves the comparison of a user's activities with the known behaviors of attackers attempting to penetrate a system. While anomaly detection typically utilizes threshold monitoring to indicate when a certain established metric has been reached, misuse detection techniques frequently. One of the main problems with IDSs is the overhead, which can become unacceptably high. To analyze system logs, the operating system must keep information regarding all the actions performed, which invariably results in huge amounts of data, requiring disk space and CPU resource. A well designed intrusion detection system should have the ability to detect both misuse and anomaly attacks. There are three drawbacks in a single neural network structure. First, it lacks the understanding of a system. The same neural network structure may be used to classify different subjects as long as it is retrained and its input and output node number is the same. Secondly, all nodes of the network depend on each other. If its input data have any changes, the whole system has to be retrained. Last drawback is that the neural network will become increasingly complex if more variables and hidden layers are introduced. A modular neural network architecture can overcome these drawbacks.

In the paper, two modular neural network frameworks, serial hierarchical framework and parallel hierarchical framework, are proposed for intrusion detection. Both of them use Radial Basis Functions (RBF) learning algorithm. The two proposed frameworks have the abilities of adjusting their structure automatically and adaptively to detect time. They work in a way of on-line detecting novel intrusions, classifying them into different classes according to a given criterion, real-time training new neural network classifiers for novel intrusions, and automatically changing their structures by adding the new neural network classifiers into the existing IDS. Due to the complexity of the hierarchical structure, the algorithm in a single classifier of the

hierarchical structure needs to have high detection rate and short training time. In order to find the best suitable learning algorithms for the two hierarchical structures, two popular neural network learning algorithms, Back Propagation learning (BPL) and RBF, are introduced and compared.

II. INTRUSION DATA

The LAN was operated like a real environment, but being blasted with multiple attacks. For each TCP/IP connection, 41 various quantitative and qualitative features were extracted for intrusion analysis. Attacks are classified into the following types. Attack types fall into four main categories:

1. DOS: denial of service
2. R2L: unauthorized access from a remote machine
3. U2Su: unauthorized access to local super user (root) privileges
4. Probing: surveillance and other probing.

In experiments, performed 5-class classification. The (training and testing) data set contains 11982 randomly generated points from the data set representing the five classes, with the number of data from each class proportional to its size, except that the smallest class is completely included. The set of 5092 training data and 6890 testing data are divided in to five classes: normal, probe, denial of service attacks, user to super user and remote to local attacks.

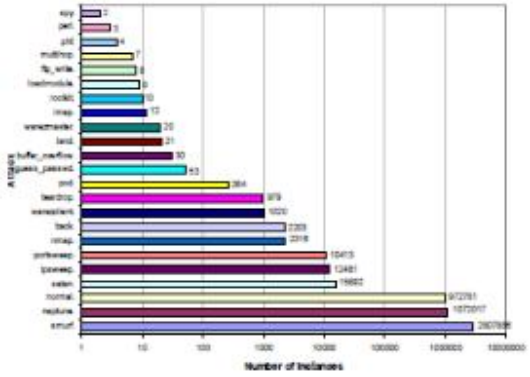

Fig 1. Data Distribution of Intrusion Detection

III. STATISTICAL ANALYSIS

A. Chi-Square Analysis

Chi square is a non-parametric test of statistical significance for bivariate tabular analysis. Any appropriately performed test of statistical significance lets you know the degree of confidence you can have in accepting or rejecting a hypothesis. Typically, the hypothesis tested with chi square is whether or not two different samples are different enough in some characteristic or aspect of their behavior that we can generalize from our samples that the populations from which our samples are drawn are also different in the behavior or characteristic. Consider a set of k measurements of size: {*x*1, *x*2... *xk*} where *x*1 is the size of the first measurement etc. They are supposed to be "normally" distributed with mean μ and standard deviation . The quantity chi-square is given by the equation

$$x^2 = \frac{(x_1-\mu)^2}{\sigma^2} + \frac{(x_2-\mu)^2}{\sigma^2} + \ldots\ldots + \frac{(x_k-\mu)^2}{\sigma^2} = \sum_{i=1}^{k} \frac{(x_1-\mu)^2}{\sigma^2}$$

B. Logistic Regression

Logistic regression is part of a category of statistical models called generalized linear models. Logistic regression allows one to predict a discrete outcome, such as group membership, from a set of variables that may be continuous, discrete, dichotomous, or a mix of any of these. Generally, the dependent or response variable is dichotomous, such as presence/absence or success/failure. Logistic Regression method is used for bivariate analysis of data i.e., either 0 or 1 .

$$\theta = \frac{e^{(\alpha+\beta_1 x_1+\beta_2 x_2+\ldots\ldots+\beta_i x_i)}}{1+e^{(\alpha+\beta_1 x_1+\beta_2 x_2+\ldots\ldots+\beta_i x_i)}}$$

Where α = the constant of the equation and, β = the coefficient of the predictor variables.

C. Statistical Results

The data set considered for this experiment involves bivariate responses. The data contains 41 features with total of 5092 samples. Logistic regression was used to rank the features based on the chi-square values for different subsets selected using best subset selection model. Higher the chi-square value higher is the ranking. The 41 features were ranked for different subsets with the subset size ranging from 1 to 41.

D. Normal distribution

Normal distribution is a continuous distribution which gives a bell-shaped curve and is particularly good for modeling situations in which the uncertain quantity is subject to different sources of uncertainty like the height, weight and length etc. As we are considering the length of the data, it is

appropriate to use this distribution. The experiment is carried out and a graph as shown in fig b is obtained in which the upper part of the bell-shaped curve has slightly deviated from 0. This implies a risk in the data i.e., the target data in normal conditions is different from the data in attacked conditions.

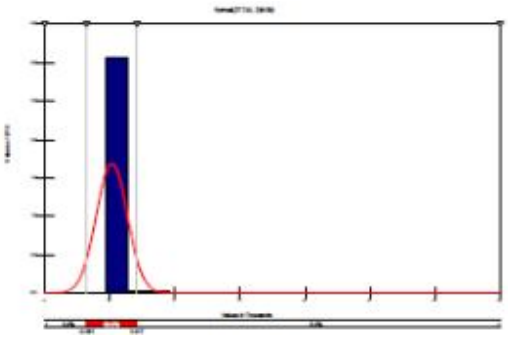

Fig 2. Data distribution

From Fig (2), the following values are obtained.

Mean = **27.743** Median = **27.743**
Mode = **27.743** St. Dev = **236.59**

E. Beta Distribution

Beta distribution is also a continuous distribution. It is more appropriate than the normal distribution when considering the proportion that take only values from 0 to 1. From Fig (3), the following values are obtained

Mean = **27.6355** Median = **26.9689**
Mode = **25.6355** St. Dev = **236.4741**

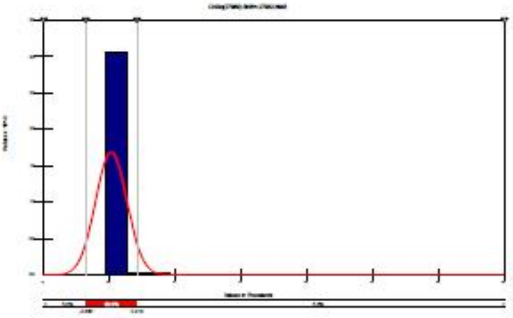

Fig 3. Beta Distribution

The target data in the attack conditions is different from the target data in the normal conditions with mean 27.743 and standard deviation 236.59 as per normal distribution. And we can also observe the difference between the normal and beta distribution values i.e., mean mode, median and standard deviation.

IV. ALGORITHMS IN NEURAL NETWORKS

The neural networks have been studied for many decades. Frank Rosenblatt s research is significant in neural network history. He was the first to apply single-layer perceptrons, a generalization of the 1943 McCulloch –Pitts concept of the functioning of the brain, to pattern classification learning in the late 1950s. Since then, a few neural network models and learning algorithms have been proposed and studied. BPL and RBF are two important learning algorithms used in neural networks. A multi-layer perceptron (MLP) neural network trained by BPL has one input layer, one output layer, and one or more hidden layers. There are no recurrent connections in the network. Every node except for those in the input layer has its own activation function. The activation functions are used to introduce nonlinearity into the network. Unipolar Sigmoidal and logistic functions are commonly used as activation functions. The BPL training procedure consists of two stages: feed forward and back propagation. In the feed forward stage, the input data are fed into the input nodes, and then every node of the hidden layers and output layer calculates its activation value sequentially. The differences between the output of the end layer and the desired target are used to generate the error. In the back propagation stage, the error is propagated back from the output layer to input layer. The error is used to adjust weights between the output nodes and the hidden layer nodes first. Usually, the gradient descentmethod is used to update weights. After the weightsare updated, the new error at the hidden nodes is calculated and used to update hidden layers_ weights again. The neural network continuously updates its weights until the error of the network or the training epoch reaches a threshold. A RBF neural network always consists of three layers: input layer, hidden layer, and output layer. It is fully connected, but only the weights between the output layer and the hidden layer are trained. The structure of a RBF network is shown in Fig. 4. The hidden nodes compute their activation using radial basis functions. Gaussian function is one of the most popular radial basis functions. These radial basis functions divide the pattern space into some local spaces with hyperspheres. The training procedure of RBF can also be divided into two stages: unsupervised learning and supervised learning. In the unsupervised learning stage, RBF uses clustering methods to determine the parameters of the network, such as the number of hidden nodes, the centers and the covariance matrices of these nodes. In the supervised learning stage, after the parameters of hidden nodes are frozen, the weights between the hidden layer and the output layer can be calculated by feed-forward calculation.

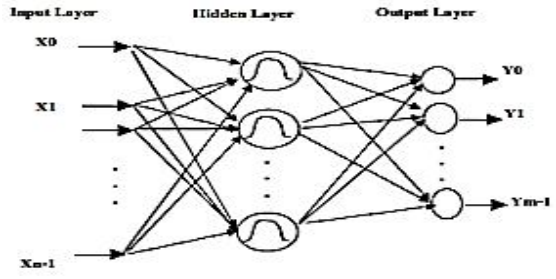

Fig 4. RBF Structure

Mainly because of the diversity of their activation functions, BPL and RBF have different performance in their applications. Compared with RBF, BPL has the drawbacks of reaching local minima, slow convergence, determining the number of hidden layers and nodes, and initializing weights. Furthermore it is inflexible to tune the network by analyzing the input data, because there is no intuitive relationship between the data and the network. RBF has some advantages over BPL. RBF can model any nonlinear function using a single hidden layer, which eliminates considerations of determining the number of hidden layers and nodes. The simple linear transformation in the output layer can be optimized fully by using traditional linear modelling techniques, which are fast and less susceptible to the local minima problem. Because the weights only exist between the output layer and the hidden layer, RBF requires less computation. The number of hidden nodes and function parameters of RBF network can be preset in accordance with the prior understanding of the training data or requirements of the output. On the other hand, BPL has its own advantages. The training procedure of BPL is quite simple. It is not necessary to normalize the training and testing data, and it simplifies data pre-preparation.

V. PROPOSED HIERARCHICAL NEURAL NETWORK

Considering the advantages of using the hierarchical structures mentioned in the introduction section, we proposed two types of neural networks based hierarchical frameworks in IDS. The goal is to detect attacks with both misuse and anomaly techniques in real-time without human interruption. There are two prerequisites to use hierarchical neural networks in IDS. First, each individual classifier should have an acceptable performance, otherwise, the errors of upper levels will be accumulated to influence the performance of lower levels. Detection rate and a false positive rate are two main performance indicators. False positive rate especially is critical to the performance of an intrusion detection system. Small difference of the false positive rate may translate into a prohibitively high number false alarms compared to the actual number of real alarms. In most of situations, it is not the ability of identifying attacks but rather its ability of suppressing false alarms that limits the performance of an intrusion detection system. Axelsson demonstrates that the false alarm rate is the limiting factor for the performance of an intrusion detection system because of the base-rate fallacy phenomenon. Secondly, the classification subjects basically can be divided into several groups according to some criteria. Each group can be assigned to its own classifier, then the classifiers or their output can be combined together. This way reduces the computation required by the system, and facilitates fine tuning and control. The two prerequisites are completely satisfied in our IDS applications. To the first prerequisite, experiments show that RBF neural network based IDS has a 98% detection rate and 1.6% false positive rate in misuse detection, and it has an overall 99.2% detection rate and 1.2% false positive rate in anomaly detection. The performance is good enough to adapt RBF to hierarchical frameworks. The second prerequisite is also satisfied, because the security threats usually can be divided into different main categories according to the purpose of the attacks and their consequences.

There are many methods to classify intrusion data into categories. In this paper, intrusion packets of the experiment data are classified into four categories by their features. They are Denial of Service (DoS), unauthorized access from a remote machine (R2L), unauthorized access to local super-user privileges (U2R) and surveillance and other probing.

a) Serial hierarchical IDS (SHIDS):

A serial hierarchical IDS was proposed mainly based on the fact that each individual classifier has good performance in misuse and anomaly detection. The central idea of this framework is to update the structure automatically and adaptively according to novel intrusions identified by a clustering program.

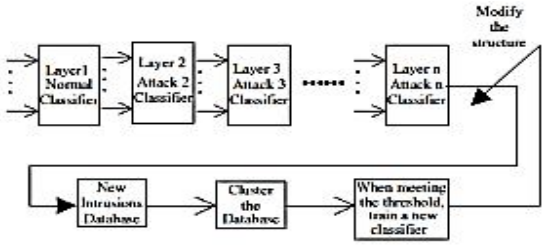

Fig 5. Structure of SHIDS

The working procedure of SHIDS is as follows: first, an anomaly classifier is trained based on pure NORMAL training data. This initial classifier of the IDS is an anomaly classifier and can only identify whether a packet is normal or not. The normal packets pass through the classifier, but the

intrusion packets are detected and stored into a database. When more attacks are detected and saved into the database, a clustering algorithm is used to cluster these attacks into different groups based on their statistical distributions. When the number of the attack records in the largest group of the database reaches a preset threshold, namely number-threshold, the system will automatically trigger a training program, which uses the attack records of this group to train a new RBF-based classifier.

Since the classifier is trained by certain group data, it is used to detect the corresponding attacks. After the training, the new classifier is added to the last level of SHIDS. This architecture will be updated continually whenever the database collect enough novel attack data. There are three advantages of using SHIDS: detecting new intrusions on-line; training new classifiers in real-time; automatic update of its structure.

b) Parallel hierarchical IDS (PHIDS)

Though SHIDS expands the functions of singlelevel IDS, it has its own disadvantages. For example, all of the upstream detection errors are accumulated to influence the downstream classifiers. The more levels a SHIDS has, the great errors it accumulates, and the more detection time it needs. Furthermore, if any upstream classifier collapses, all of the downstream classifiers will have no chance to identify further attacks. In other word, SHIDS has the problem of ‘‘a single point failure’’.

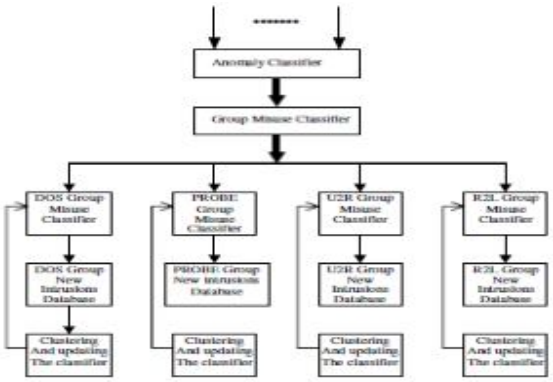

Fig 6. The Structure of PHIDS

```
Function Clustering (records) returns classes
    1. Let R be a detected record.
    2. If: R is a known type record
        Classify it and Return
   Else:
       Send R to a database
      Cluster the database with C-means algorithm
      Let Mj be the largest number of the records of
different          classes
      If: Mj > the given number-threshold,
          Move the records out of the database and use
them to      train a new classifier
      Add the new classifier to the SHIDS to update
the HIDS
 automatically.
       Update the database
   End
   Return
End
```

Fig 7. Clustering Algorithm

In PHIDS, an anomaly detection classifier is trained and used as the first level. The second level is a misuse detection classifier, which identifies the main groups of intrusion packets. In the paper, there are actually four main groups based on the experimental data. The second level is the key in

PHIDS and is trained using as much training data as possible. The third level of PHIDS initially has four classifiers separately connected to each output of the second level to represent four kinds of typical intrusions: Dos, R2L, U2R, and PROBE. These classifiers are used to identify well-known attacks and will modify their structures with the increasing of novel intrusions. For example, when a novel intrusion occurs in the input data, it will be classified as attack at the first level, and then it will be classified as the one of the four groups at the second level because of feature similarity of the same group. In the third level, because the classifier has no knowledge about this novel attack, the novel intrusion will be saved into a database. In the database, the intrusion packet will be clustered by the clustering algorithm mentioned in Fig. 7. If the number of one kind of the saved novel intrusions reaches the preset number-threshold, the third level classifier will be retrained and updated. PHIDS will update its classifiers in the third level continually according to novel intrusions. Hence, the PHIDS can identify the novel intrusion with the updated third level classifier. However, PHIDS will keep the three-level structure no matter how many kinds of intrusions there are. Therefore, it reduces the error accumulation problem,
which occurs in the SHIDS structure. Compared with SHIDS, PHIDS has two main advantages. Firstly, PHIDS has only three levels, so the problems of error accumulation and ‘‘a single point failure’’ can be ignored in PHIDS. Secondly, the processing of PHIDS classification is much quicker than SHIDS. On the other hand, there is a challenge in PHIDS. It is harder to choose a suitable decision threshold to identify novel

intrusions in the third level classifier, and the problem will become more serious when similar intrusions increase.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

1. Resilient back propagation achieved the best performance among the neural networks in terms of accuracy (97.04 %) and training (67 epochs).
2. Chi-squared analysis produces largely consistent results. Features ranked as important by Chi-squared analysis heavily overlap for all the attacks classes.
3. Using the important features for each class gives the most remarkable performance: the testing time decreases in each class, the accuracy increases slightly for normal, probe, DOS and remains the same for the two most serious attack classes.
4. Forward selection and backward elimination also produces largely consistent results. We note, however, that the difference in accuracy figures tend to be very small and may not be statistically significant, especially in view of the fact that the 5 classes of patterns differ in their sizes tremendously.

More definitive conclusions can only be made after analyzing larger sets of representative intrusion data, the collection of which is itself a significant problem. There are two main objectives of the work reported in this paper. The first objective is to find a suitable method, which can be applied to intrusion detection with less training time, high detection rates and less false positive rates. Because of the many advantages of neural networks, BPL and RBF algorithms are applied to train neural network based intrusion detectors (classifiers) for IDSs. Considering the advantages of RBF over BPL mainly because of the difference in their activation functions, we initially believed RBF had better performance in IDS from the training time and detection rate aspects.

The experimental results in successfully showed that RBF network based IDS has a good performance in misuse detection with a 98% detection rate and a 1.6% false positive detection rate. It is further showed in that RBF has an excellent result in anomaly detection. The second objective of the paper is to design an IDS with the abilities of detecting both misuse and anomaly attacks, and adaptively training new modules, and updating its structure for novel attacks. Two types of hierarchical neural network frameworks were proposed. Some possible directions for future work include considering other types of classifiers such as support vector machines; dealing with time dependent data; and online learning techniques.

REFERENCES

[1] G. Vigna, R. A. Kemmerer, “NetSTAT: a network-based Intrusion Detection Approach”, Proceedings of 14th Annual Computer Security Applications Conference, 1998, pp. 25 –34.

[2] W. Lee, S. J. Stolfo, K. Mok, “A Data Mining Framework for Building Intrusion Detection Models”, Proceedings of 1999 IEEE Symposium of Security and Privacy, pp. 120-132

[3] M. Moradi and M. Zulkhenine, “A Neural Network Based System for Intrusion Detection and Classification of Attacks”, in Proc. IEEE Advances in Intelligent Systems - Theory and Applications, pp. 148:1-6, Luxembourg, November 2004

[4] J. Shum and H.A. Malki, “Network intrusion detection system using neural network”, in Proc. IEEE Fourth Int. Conference on Natural Computation, pp. 242-246, 2008.

[5] J. M. Bonifacio, et al., Neural Networks Applied in Intrusion Detection System, IEEE, 1998, pp. 205-210

[6] Chaivat Jirapummin, Naruemon, Wattanapongsakorn and Prasert Kanthamanon (2000), ”Hybrid Neural Networks for Intrusion Detection System”, Department of Computer Engineering, Faculty of Engineering, King Mongkut’s University of Technology Thonburi, Bangkok , Thailand.

[7] John Zhong, Lei and Ali Ghorbani (2004) “Network Intrusion Detection Using an Improved Competitive Learning Neural Network” in Proceedings of the 2nd Annual Conference on Communication Networks and Services Research (CNSR 2004), Canada. IEEE Computer Society, ISBN 0-7695-2096-0 pp. 190-197.

[8] Northcut S and Novak J, “Network Intrusion Detection”, 3rd ed. Indianapolis, IN: New Riders Publishing, 2002.

[9] Devikrishna K. S. and Ramakrishna B. B. (2013), An Artificial Neural Network based Intrusion Detection System and Classification of Attacks, International Journal of Engineering Research and Applications (IJERA) ,Vol. 3, Issue 4, pp. 1959-1964 1959, ISSN: 2248-9622

[10] Bhavin S. and Bhushan H. T. (2012), “Artificial Neural Network based Intrusion Detection System: A Survey”, International Journal of Computer Applications 39(6):13-18.