Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Pattern Recognition Letters 26 (2005) 779–791

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Intrusion detection using hierarchical neural networks

Chunlin Zhang, Ju Jiang, Mohamed Kamel *

Pattern Analysis and Machine Intelligence Research Group, Department of Electrical and Computer Engineering, University of Waterloo, Canada

Received 6 August 2003; received in revised form 16 February 2004

Abstract

Most intrusion detection system (IDS) with a single-level structure can only detect either misuse or anomaly attacks. Some IDSs with multi-level structure or multi-classifier are proposed to detect both attacks, but they are limited in adaptively learning. In this paper, two hierarchical IDS frameworks using Radial Basis Functions (RBF) are proposed. A serial hierarchical IDS (SHIDS) is proposed to identify misuse attack accurately and anomaly attacks adaptively. A parallel hierarchical IDS (PHIDS) is proposed to enhance the SHIDS's functionalities and performance. The experiments show that the two proposed IDSs can detect network intrusions in real-time, train new classifiers for novel intrusions automatically, and modify their structures adaptively after new classifiers are trained.

Keywords: Intrusion detection; Neural networks; BackPropagation algorithm; Radius basis functions; Hierarchical neural network; Neural network ensembles

1. Introduction

The possibilities of exposing sensitive information to intruders increase with the extensive use of the Internet. Certain techniques are used to secure important data, such as firewall, encryption, Virtual Private Network (VPN) etc. Firewall acts as a defense to protect sensitive data, but it merely reduces exposure rather than monitors or eliminates vulnerabilities in computer systems (Ghosh et al., 1998). Any encrypted message can be decrypted in theory, and encryption adds extra burden on hosts or applications. Moreover, any new security techniques themselves might have design flaws. Obviously, it is important to have a detecting and monitoring system to protect important data (Manikopoulos and Papavassiliou, 2002; Anderson, 1980).

IDS is a system of monitoring and detecting data traffic or user behavior to identify intruders. There are two primary approaches to detect intrusions:

* Corresponding author. Tel.: +1 519 885 1211; fax: +1 519 746 4791.
E-mail address: mkamel@uwaterloo.ca (M. Kamel).

0167-8655/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2004.09.045

misuse detection and anomaly detection (Jones and Sielken, 1999; Axelsson, 2000b). The misuse detection first attempts to model specific patterns of intrusions to a system, then systematically scans the system for their occurrences. Since the knowledge of the intrusions has to be known before the modelling, this method is mostly used to detect well-known intrusions. Anomaly detection creates a profile of typical normal traffic activities or user behaviors, then it compares the deviation between the profile and the input activity with a preset threshold to decide whether the input instance is normal or not (Jones and Sielken, 1999; Cannady, 2000). The preset threshold can be adjusted to meet desired performance. Anomaly detection addresses the problem of detecting novel intrusions. Usually, it cannot provide detail information about the attacks. A well designed intrusion detection system should have the ability to detect both misuse and anomaly attacks.

Rule-based expert system and statistical are two common approaches for IDSs. A rule-based expert IDS can detect some well-known intrusions with high detection rate, but it is difficult to detect novel intrusions, and its signature database needs to be updated manually and frequently (Lindqvist and Porras, 1999; Denning and Dorothy, 1987; Michael, 2002). A statistical-based IDS needs to collect enough data to build a complicated mathematical model, which is impractical in the case of complicated network traffic (Gordeev, 2000).

Unlike hard computing, whose prime desiderata are precision, certainty, and rigor, soft computing attempts to provide imprecise and uncertain answers when the cost is limited and the tolerance is acceptable (Zadeh, 1994). Artificial neural networks (ANN) is one of the main soft computing algorithms. It has been successful in solving many complex practical problems. More and more researchers have realized the advantages of applying neural networks to IDS, and they have made some progresses in certain areas (Ghosh et al., 1998; Giacinto et al., 2003; Cho, 2002; Cannady, 1998; Ryan et al., 1998; Fan et al., 2001; Lee, 2001; Endler, 1998). Most of these neural network applications use a single neural network structure, which has only one set of input, output, and hidden layers. 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 (Michael, 2002). A modular neural network architecture can overcome these drawbacks. It is proposed to break a complicated problem into some smaller sub-problems, each sub-problem can be resolved by a smaller neural network at different levels or different locations. Then, these smaller networks can be combined back to generate the solution for the original problem. Some popular modular architectures, such as serial, parallel, and parallel-serial architectures, are discussed in (Dasarathy, 1994). Modular neural network decreases the complexity of design and training especially for huge systems. When there is any change in a system, it only needs to partially train some networks related to the change. Modular neural network, on the other hand, has some drawbacks too. It requires that users have knowledge to divide a complex system into small ones and establish connections between them. The users must understand the system well in advance in order to remove redundant or useless connections.

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 both well-known and novel intrusions in real 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, BackPropagation learning (BPL) and RBF, are introduced and compared.

The rest of the paper is organized as follows. Section 2 introduces BPL and RBF algorithms and the MLP neural network used in the experiment, and then compares their differences. Section 3 gives details of two proposed hierarchical neural network structures and explains the principles and working procedures of these two structures. Section 4 presents the details of the experiment preparation, data preprocessing and results. The results show the evaluation of the single level structure; the serial hierarchical IDS framework; and the parallel hierarchical IDS frameworks. Based on the results and observations of these experiments, some conclusions and future directions are drawn and explained in Section 5.

2. 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 (Werbos, 1974; Tsoukalas and Uhrig, 1997; Rumelhart et al., 1986). 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 descent method is used to update weights. After the weights are 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 (Ghosh and Nag, 2000). 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. 1. 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

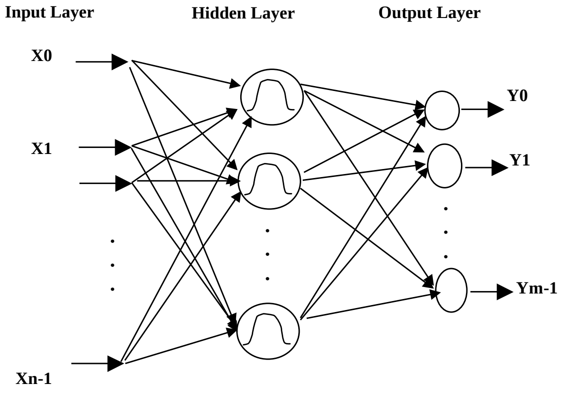

Fig. 1. RBF structure.

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.

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.

3. Proposed hierarchical neural network based IDS

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 (Axelsson, 2000a). 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 (refer to Section 4) 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 (Wenstrom, 2002). 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 (PROBE) (KDD, 1999).

3.1. Serial hierarchical IDS (SHIDS)

A serial hierarchical IDS was proposed mainly based on the fact that each individual classifier

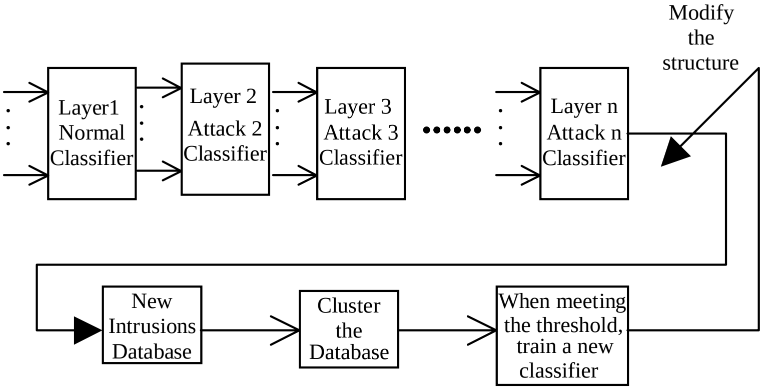
Modify the structure
Layer1 Normal Classifier
Layer 2 Attack 2 Classifier
Layer 3 Attack 3 Classifier
Layer n Attack n Classifier
New Intrusions Database
Cluster the Database
When meeting the threshold, train a new classifier

Fig. 2. Structure of SHIDS.

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. The framework of SHIDS is shown in Fig. 2.

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; automatical update of its structure. Fig. 3 shows detail of the clustering algorithm.

```
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 $M_j$ be the largest number of the records of different classes
         in the database.
         If:$M_j$ > 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. 3. A clustering algorithm.

3.2. Parallel hierarchical IDS (PHIDS)

Though SHIDS expands the functions of single-level 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". A parallel hierarchical IDS framework, shown in Fig. 4, is proposed to enhance the abilities of SHIDS.

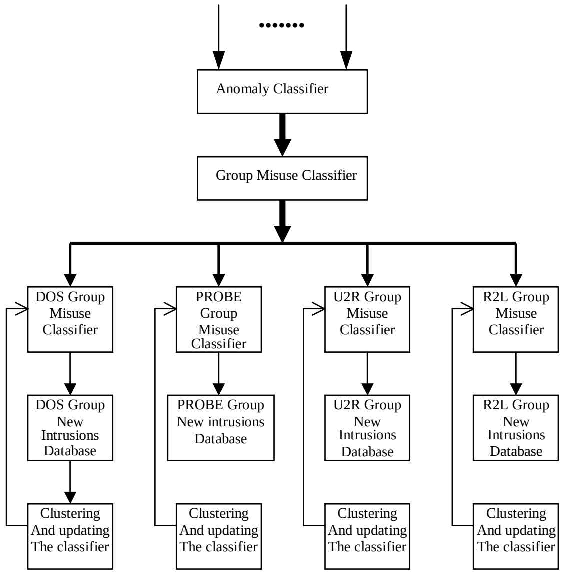
••••••
Anomaly Classifier
Group Misuse Classifier
DOS Group Misuse Classifier
PROBE Group Misuse Classifier
U2R Group Misuse Classifier
R2L Group Misuse Classifier
DOS Group New Intrusions Database
PROBE Group New intrusions Database
U2R Group New Intrusions Database
R2L Group New Intrusions Database
Clustering And updating The classifier
Clustering And updating The classifier
Clustering And updating The classifier
Clustering And updating The classifier

Fig. 4. The structure of PHIDS.

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. 3. 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.

4. Experiments and results

4.1. Data preparation

The data set used in the experiments is "KDD Cup 1999 Data", which is a subversion of DARPA 1998 dataset. The raw data includes a wide variety of intrusions simulated in the network. The attacks in the data set fall into four main categories: DoS, R2L, U2R, and PROBE. In order to demonstrate the abilities of detecting different kinds of intrusions, the training data and testing data cover all intrusion categories. Totally, 22,000 attack data and 10,000 normal data were prepared for training and another set of 22,000 attack instances and 10,000 normal data were selected as the testing data. The training and testing data were selected randomly, both of them have the same approximate distribution as the KDD data set. Each record is unique in the data set with 34 numerical features and 7 symbolic features. The symbolic features need to be converted to numeric values. A binary ASCII coding method is used to convert symbolic features to numerical features.

Data normalization is necessary in some neural network algorithms. In the experiments, the unnormalized data is used for BPL, and the normalized data is used for RBF. The data is normalized by a linear normalization method as in Eq. (1) because of its good performance and simplicity. It is important to notice that the statistical parameters of the experiments, such as the mean and standard deviation used in neural networks, are computed from the training data instead of testing data. The testing data is normalized using the statistics computed from the training data.

$$\bar{x}_i(j) = \frac{x_i(j) - x_{\min}(j)}{x_{\max}(j) - x_{\min}(j)}, \qquad (1)$$

where

$$x_{\max}(j) = \mathrm{Max}(x_i(j)), \quad i = 1, 2, \ldots, n,$$

$$x_{\min}(j) = \mathrm{Min}(x_i(j)), \quad i = 1, 2, \ldots, n.$$

4.2. Experiment 1: comparison of BPL and RBF in IDS

The objective of experiment 1 is to show that RBF has good performance in intrusion detection, and it can be applied to hierarchical framework. The experiment demonstrates RBF's performance by comparing BPL and RBF in both misuse and anomaly intrusion detection.

4.2.1. Misuse detection

In the experiment, three types of attack data (PROBE, DoS, R2L) and normal data were used, the outputs in the experiment were defined as [1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], and [0, 0, 0, 1], which represent NORMAL, PROBE, DoS, and R2L pattern in a binary-output form respectively.

Two methods were used to standardize the output since the outputs are not exactly 1's or 0's. The first method is to find the greatest element of the output, and then to let the element be 1, the other elements be 0. The second method is to use the nearest neighborhood method, which means that for an input vector x, the output is y, x belongs to the class C_i if Eq. (2) is satisfied. The first method was adopted because of the better performance in the experiment.

$$\|y - e_i\| < \|y - e_j\|, \quad j \neq i, \qquad (2)$$

where e_l is a vector whose lth element is 1 and all the other elements are 0's.

Table 1 shows that BPL has a slightly better performance than RBF. However, RBF takes less training time and is easier in terms of tuning the hidden layer structure and the threshold. When the classifier is used in a real-time classification situation, the training time is significant.

4.2.2. Anomaly detection

The MLP neural network trained by BPL has the ability of detecting anomaly intrusions, but its performance is still in need of improvement. The empirical studies showed that the pure anomaly detection model is capable of detecting more than 77% of all unknown intrusion classes with more than 50% accuracy per intrusion class (Fan et al., 2001). The objective of this experiment is to evaluate if RBF can overcome the limitations of BPL in anomaly detection. The training data consists of pure NORMAL instances. There is only one output node. In anomaly detection, a deviation threshold is a key to tune the performance.

Kolmogorov's theorem (Tsoukalas and Uhrig, 1997) shows that a three-layer neural network can perform any mapping from R^m to R^n exactly if the network has m nodes in the input layer, n nodes in the output layer and $2m + 1$ nodes in

Table 1
Results of BPL and RBF in misuse detection

	OVERALL		PROBE		DoS		R2L	
	DR (%)	FP (%)	DR (%)	FP (%)	DR (%)	FP (%)	DR (%)	FP (%)
BPL	99.2	1.2	99.2	1.2	99.6	1.2	98.8	1.2
RBF	98	1.6	98	1.6	98.8	1.6	97.2	1.6

DR: detection rate, FP: false positive rate.

hidden layer. Though it does not prove that it is the most efficient for the mapping, it guarantees to resolve all nonlinearly separable problems. Based on this theory, for the MLP neural network trained by BPL, the number of the hidden nodes is set to 83. The maximum training epoch is chosen as 200. It took two hours to finish the training. In RBF network, it has 41 input nodes, 1 hidden node and 1 output node. The range of deviation threshold, which separates 1's and 0's, of RBF is wider than the range in BPL. The detection rate and false positive rate do not change much when the threshold changes. The reason is the Gaussian activation function of RBF assures that the same classes are clustered together and magnifies the output difference if the instances belong to different classes, and then the boundary of the hypersphere of the RBF classifier can be found easily. This characteristic makes it easy to determine the threshold for RBF.

In the testing stage, overall performance was obtained when the testing date include NORMAL, PROBE, DoS, and R2L intrusions. Specific attack detection performances were obtained when testing data only includes certain intrusion data. Figs. 5 and 6 are the pairs of detection rate (DR) and false positive rate (FP) in the case of different threshold. Table 1 shows different results when we

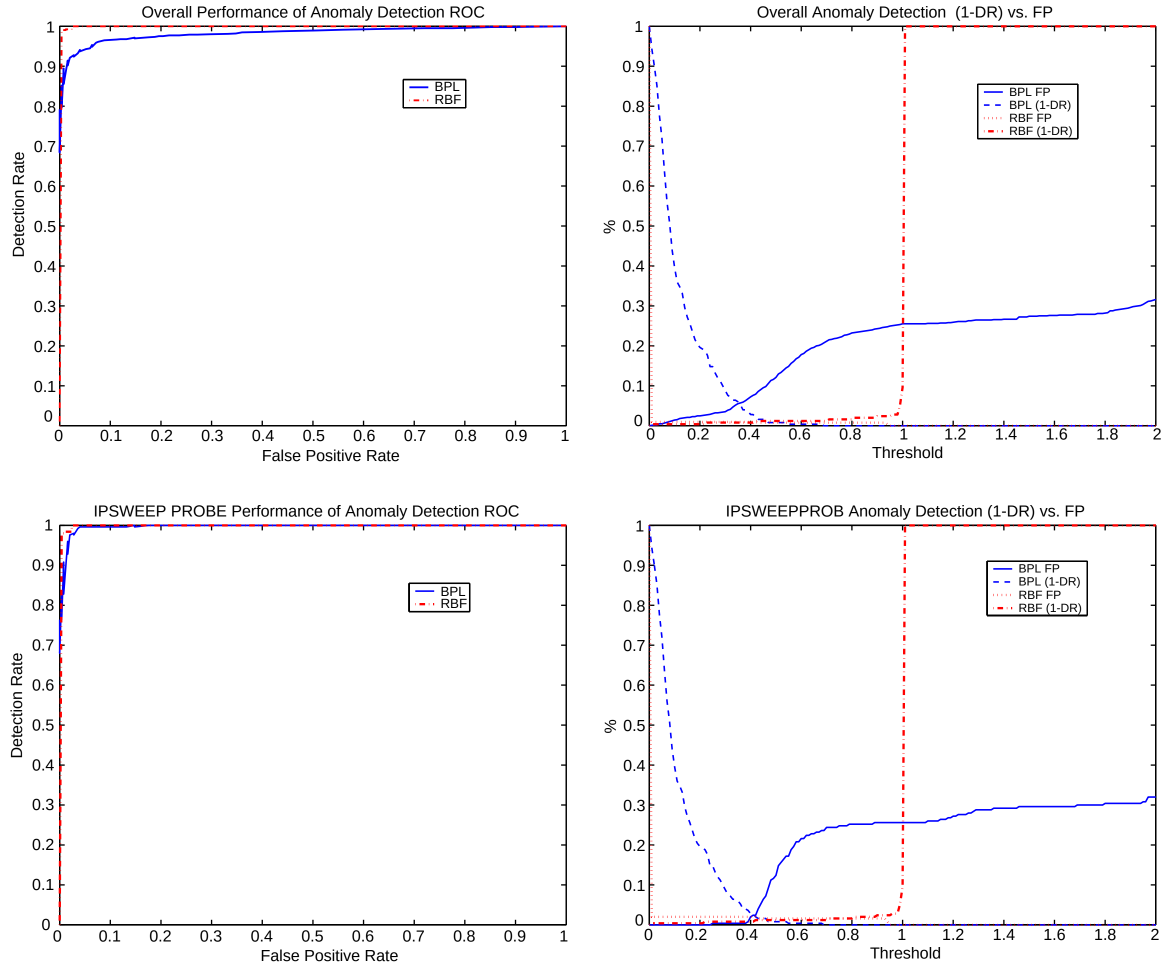

Fig. 5. Comparison of BP and RBF in anomaly detection (1).

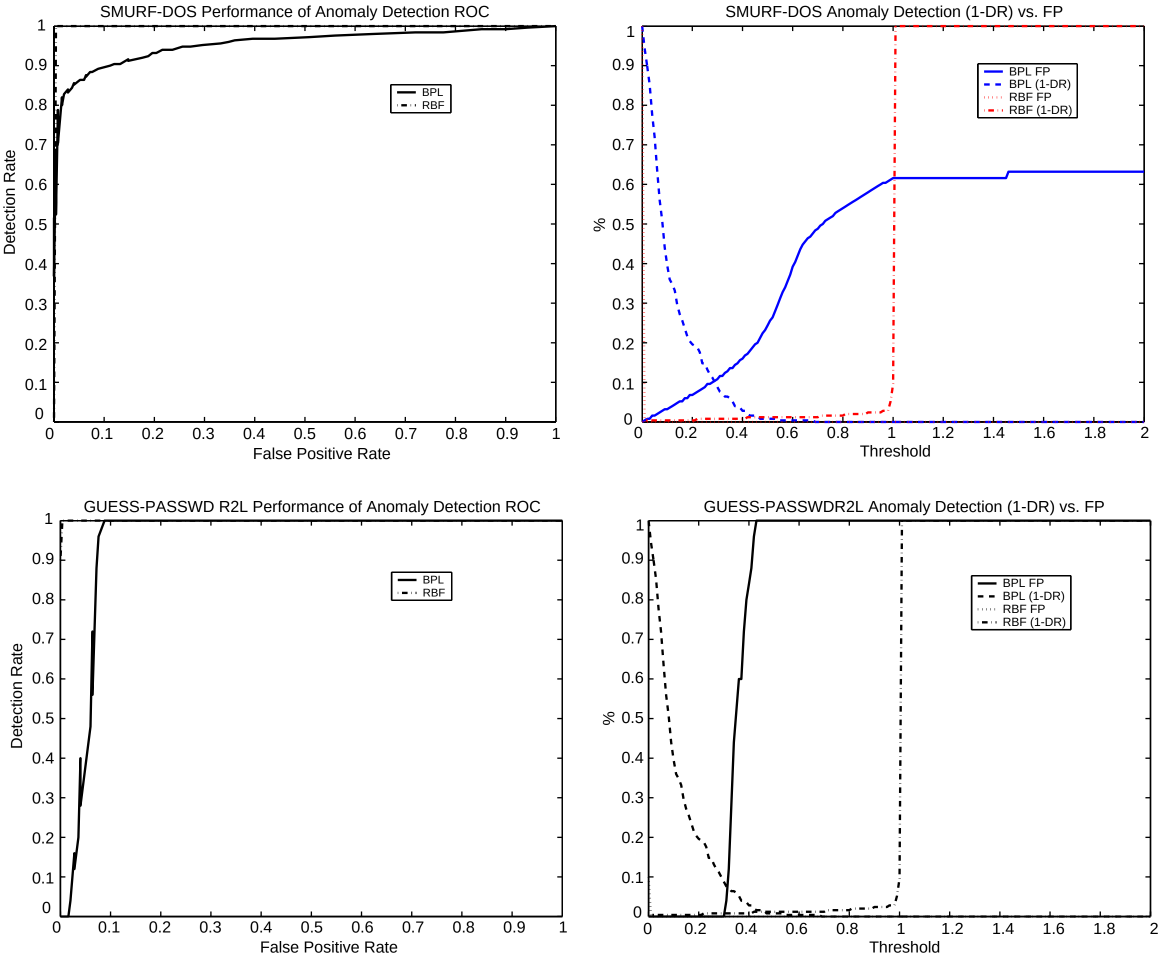

Fig. 6. Comparison of BPL and RBF in anomaly detection (2).

use the optimal threshold. From Figs. 5 and 6, it is obvious that BPL suffers from a low detection rate and a high false positive rate. For example, when the threshold is 0.31, the overall performance of BPL is 94% detection rate and 8.8% false positive rate. For R2L intrusion detection, BPL classifier has only 88% detection rate and 7.2% false positive rate. For DoS intrusion detection, 90% detection rate with 11% false positive rate are obtained. The results of BPL in anomy detection are unacceptable in a crucial network environment. In contrast to BPL, RBF shows an excellent performance.

Another observation is that RBF has a flat slope and broad threshold range as presented in Figs. 5 and 6. The slope of FP and DR determines the stability of the performance. A flatter slope means a stable performance. For example, the threshold of RBF can vary from 0.1 to 0.9 but the curves of FP and DR almost stay constant. Their performances change slightly when the threshold changes. In contrast to RBF, if the threshold of BP varies by even 0.1, its performance will change greatly.

Table 2 displays the performances of BPL and RBF when the best-optimized threshold is set for different kinds of intrusions. The overall detection rate of BPL is 93.7%. It is much lower than the 99.2% of RBF. The false positive of BPL is 7.2%. It is much higher than the 1.2% of RBF.

Table 2
Detail results of BPL and RBF in anomaly detection

	OVERALL		PROBE		DoS		R2L	
	BPL	RBF	BPL	RBF	BPL	RBF	BPL	RBF
TR[a]	0.2–0.4	0.1–0.9	0.3–0.5	0.1–0.9	0.2–0.4	0.1–0.9	0.2–0.4	0.1–0.9
TGP[b]	0.34	0.5	0.43	0.5	0.28	0.5	0.34	0.5
DR (%)	93.7	99.2	97	98.4	90.4	100	88	100
FP (%)	7.2	1.2	2.8	1.2	13.2	1.2	7.2	1.2

[a] TR—threshold range.

[b] TGP—threshold golden point.

RBF has a very broad threshold range from 0.1 to 0.9. An optimized threshold of 0.5, in RBF, can be chosen for all of the intrusion detections, including novel intrusion detections. It is impracticable for BPL to find a threshold suitable for all intrusions. For example, in BPL, when the threshold is set to 0.28, the intrusion detection system achieves the best performance only for SMURF intrusions. However, it has 12% FP in Guess_Passwd intrusion detection, and 13% FP in IPSWEEP detection.

4.2.3. Observation of experiment 1

Both BPL and RBF can be used to train neural networks used for IDSs. The use of these neural networks in IDS results in an acceptable detection and misclassification rates. Both have a good performance in identifying well-known intrusion patterns. The results of experiment 1 show that a neural network is an excellent tool in identifying well-known intrusions. Compared with the rule-based expert system, the use of a neural network has the advantage of detecting intrusions that are variants of well-known signatures.

RBF achieves a better performance than BFL in anomaly detection. The flat slope and broad threshold range make the intrusion detection system reliable, and the training time is cut to five minutes. The results demonstrate that RBF is an ideal neural network algorithm for anomaly detection. Furthermore, it shows that the performance of RBF neural networks can meet the first prerequisite of applying hierarchical neural networks in IDS as mentioned in the previous section. This helps us conduct further experiments in applying RBF in hierarchical frameworks.

4.3. Experiment 2: hierarchical IDS

The object of this experiment is to show that the proposed hierarchical IDS can meet the desired goal: a hybrid real-time intrusion detection system with the ability of adaptively and automatically training new classifiers and updating their structures for novel attacks.

4.3.1. Serial hierarchical IDS

This experiment used the structure in Fig. 2. The working procedures were described in the previous section. The first level classifier, which is an anomaly detector, is used to flag any novel attacks. The anomaly classifier used in the experiment is a RBF network based classifier, which is obtained from experiment 1. This anomaly classifier has the overall detection rate of 99.2% and a 1.2% false positive rate. The clustering is performed using the C-Means clustering algorithm presented in Fig. 3.

The distance threshold used in C-Means clustering algorithm is critical. A greater distance threshold will confuse similar types of intrusions together, but a smaller distance threshold will cluster the data into too many small groups. An empirical number was set in the experiment. Another threshold, namely number-threshold, needs to be preset before the experiment. This threshold is used to trigger training for a new classifier. the number-threshold "200" was obtained based on the experiment. Since the training data were fed into the structure in the sequence of IPSWEEP, SMURF, GUESS_PASSWORD and BUFFER_ OVERFLOW, the first group that reached the number-threshold was IPSWEEP group. Then, an IPSWEEP classifier was trained using the saved records, and the classifier was added to the existing structure. After that, the SHIDS extended to a two-level classifier structure. Later, SMURF intrusion reached the number-threshold and a three-level SHIDS was built. Repeating the same procedure, SHIDS modified its structure automatically according to the network intrusions. After a five-level RBF-based hierarchical intrusion detection system was generated adaptively during the training stage, the SHIDS was tested using the testing data, which included NORMAL, IPSWEEP, SMURF, GUESS_PASSWORD, and BUFFER_OVERFLOW data. The results are presented in Table 3. It shows that the SHIDS has both a high detection rate, low false positive rate, and can automatically detect and classify novel intrusions.

Compared with the single-level IDS discussed in experiment 1, the SHIDS can detect novel intrusions automatically and adaptively with good performance. Different testing data are used to verify the efficiency of SHIDS. Similar results were obtained.

4.3.2. Parallel hierarchical IDS

This experiment used the structure in Fig. 4. The training data and testing data are little bit different from SHIDS. The second level, a group misuse classifier, is the key of the PHIDS. The training data of the group misuse classifier include attacks from four different main groups mentioned above.

The first level of PHIDS was an anomaly detection classifier, which was trained in the anomaly detection experiment with a detection rate of 99.2% and 1.2% false positive. The second level classifier was trained to classify attacks into four groups. In the third level, first, few classifiers were trained as misuse classifiers to identify certain well-known intrusions, such as IPSWEEP intrusion. Then they were connected to the group misuse classifier to form a basic three-level PHIDS. The last step was to modify the misuse classifiers of the third level according to the increasing of novel intrusions based on the basic structure. In the experiment, 250 PORTSWEEP intrusion samples were added to the training data and were fed into the three-level IDS. Following the procedures mentioned in the previous section, PHIDS automatically retrained the PROBE group misuse classifier so that it can classify the PORTSWEEP intrusions. Currently an empirical estimation was adopted to set the decision threshold in this experiment. In the future, more research will be conducted to find an effective way to set the threshold adaptively. In the testing stage, PORTSWEEP records were added into the testing data to test the updated classifier. Table 4 gives the results of the experiment of the PHIDS. The results

Table 3
The experiment results of SHIDS

Detector	Data	NORMAL	A^1	A^2	A^3	A^4	DR	FP
Level1	Initial	250	250	250	250	150	896/900	3/250
NORMAL	Identified[b]	247	4	0	0	0	=99.5%	=1.2%
Level2	Test[a]	3	246	250	250	150	647/653	4/246
IPSWEEP	Identified[b]	0	242	6	0	0	=99.1%	=1.62%
Level3	Test[a]	3	4	244	250	150	404/407	11/244
SMURF	Identified[b]	0	1	233	2	0	=99.3%	=4.5%
Level4	Test[a]	3	3	11	248	150	164/167	0/23
GuessP	Identified[b]	0	0	0	248	3	=98.2%	=0%
Level5	Test[a]	3	3	11	0	147	16/17	8/147
BufferO	Identified[b]	1	0	0	0	139	=94.1%	=5.4%
	Remain	2	3	11	0	8		

A^1: IPSWEEP; A^2: SMURF; A^3: Guess Password; A^4: Buffer Overflow.

[a] Testing data left from previous level.

[b] Correctly identified intrusions in this level.

Table 4
The experiment results of PHIDS

Data	N	A^1	A^2	A^3	A^4	$A^{1.1}$	DR	FP
L1-T[a]	250	250	250	250	150	60	958/960	3/250
L1-I[b]	247	2	0	0	0	0	=99.8%	=1.2%
L2-T[a]	3	248	250	250	150	60	A^1:99.5%	0.8%
L2-I[b]		A^1:246	A^2:250	A^3:240	A^4:145		A^2:99%	0%
		N:3	A^1:2	$A^{1.1}$:2	A^3:10		A^3:99.7%	4%
		$A^{1.1}$:58	A^4:5				A^4:98.8%	3.3%
L3-T[a]		A^1:246					53/(58+3)	0/246
L3-Iℓ		N:3					=86.9%	=0%
		$A^{1.1}$:5						
		$A^{1.1}$:53						

ℓ: Identified as $A^{1.1}$: PortSweep attack; A^1: IPSWEEP; A^2: SMURF; A^3: Guess Password; A^4: Buffer Overflow; $A^{1.1}$: PortSweep; L1: level 1 anomaly detector; L2: level 2 misuse group detector; L3: level 3 IPSWEEP detector.

[a] Testing data left from previous level.

[b] Correctly identified intrusions in this level.

show that PHIDS has high DR (more than 98%) and low FP (less than 10%) in the first level and the second level. Even in the third level, the DR is still more than 86% for the novel intrusion PORTSWEEP by using a newly updated classifier. Different testing data were used to evaluate PHIDS and similar results were obtained.

5. Conclusion and future directions

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 Table 1 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 Table 2 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. The results presented in Tables 3 and 4 demonstrate that SHIDS and PHIDS can meet this objective.

Based on the experimental results, a number of conclusions can be drawn: the RBF-based IDS has the abilities to detect well-known and novel intrusions; the short training time, high DR, and low FP of RBF neural networks are more valuable than BPL neural networks in IDS; SHIDS can monitor network traffic in real-time, train new classifiers automatically for novel intrusions, and modify its structures adaptively after new classifiers are trained; PHIDS partly solves the problems of SHIDS by using the hybrid three-level structure. Moreover, PHIDS runs faster than SHIDS. These advantages make PHIDS more practicable than SHIDS. However, it is difficult to choose a suitable decision threshold to identify novel intrusions in PHIDS.

Some possible directions for future work include considering other types of classifiers such as support vector machines; dealing with time dependant data; and online learning techniques.

References

Anderson, J.P., 1980. Computer security threat monitoring and surveillance. Technical Report, James P. Anderson Company, Fort Washington, PA, April 1980.

Axelsson, S., 2000a. The base-rate fallacy and the difficulty of intrusion detection. ACM Trans. Inform. System Security 3 (3), 186–205.

Axelsson, S., 2000b. Intrusion detection systems: A survey and taxonomy. Technical Report 99-15, Chalmers University, March.

Cannady, J., 1998. Artificial neural networks for misuse detection. In: Proc. 1998 National Information Systems Security Conf. (NISSC'98) October 5–8, 1998. Arlington, VA, pp. 443–456.

Cannady, J., 2000. Next generation intrusion detection: Autonomous reinforcement learning of network attacks. In: Proc. 23rd National Information Systems Security Conf., NISSC 2000.

Cho, S.-B., 2002. Incorporating soft computing techniques into a probabilistic intrusion detection system. IEEE Trans. Systems Man Cybernet. 32 (2), 154.

Dasarathy, B.V., 1994. Decision Fusion. IEEE Computer Society Press.

Denning, E.A., Dorothy, E., 1987. Views for multilevel database security. IEEE Trans. Software Eng. SE-13, 129–140.

Endler, D., 1998. Intrusion detection: Applying machine learning to solaris audit data. In: Proc. 1998 Annual Computer Security Applications Conf., Los Alamitos, CA, pp. 268–279.

Fan, W., Miller, M., Stolfo, S., Lee, W., Chan, P., 2001. Using artificial anomalies to detect unknown and known network intrusions. In: IEEE Internat. Conf. on Data Mining, pp. 123–130.

Ghosh, A., Wanken, J., Charron, F., 1998. Detecting anomalous and unknown intrusions against programs. In: Proc. 1998 Annual Computer Security Applications Conf., ACSAC'98, Los Alamitos, CA, USA, December 1998. IEEE Comput. Soc. 1998, 259–267.

Ghosh, J., Nag, A., 2000. An overview of radial basis function networks. In: Howlett, R.J., Jain, L.C. (Eds.), Radial Basis Function Neural Network Theory and Applications. Physica-Verlag.

Giacinto, G., Roli, F., Didaci, L., 2003. Fusion of multiple classifiers for intrusion detection in computer networks. Pattern Recognition Lett. 24 (12), 1795.

Gordeev, M., 2000. Intrusion detection: Techniques and approaches. The Distributed Systems Group in the Information Systems Institute of Technical University of Vienna, October.

Jones, A.K., Sielken, R.S., 1999. Computer system intrusion detection: A survey. Technical Report, Computer Science Department, University of Virginia.

KDD, KDD Cup 1999 data. The Fifth International Conference on Knowledge Discovery and Data Mining.

Lee, H.D., S.C., 2001. Training a neural-network based intrusion detector to recognize novel attacks, systems, man and cybernetics, Part A IEEE Transactions on IEEE Computer Press 31, 294–299.

Lindqvist, U., Porras, P.A., 1999. Detecting computer and network misuse through the production-based expert system toolset (p-BEST). In: IEEE Symposium on Security and Privacy, pp. 146–161.

Manikopoulos, C., Papavassiliou, S., 2002. Network intrusion and fault detection: A statistical anomaly approach. Commun. Mag. IEEE 40 (10), 76–82.

Michael, N., 2002. Artificial Intelligence: A Guide to Intelligent Systems. Addison Wesley, Essex, England.

Rumelhart, D., Hinton, G., Williams, R., 1986. Learning internal representations by error propagationIn Parallel Distributed Processing: Explorations in the Microstructures of Cognition, vol. 1. MIT Press, Cambridge, MA.

Ryan, J., Lin, M.-J., Miikkulainen, R., 1998. Intrusion detection with neural net-works. In: Jordan, M.I., Kearns, M.J., Solla, S.A. (Eds.), Advances in Neural Information Processing Systems, vol. 10. The MIT Press.

Tsoukalas, L., Uhrig, R., 1997. Fuzzy and Neural Approaches in Engineering. Wiley, New York.

Wenstrom, M., 2002. Managing Cisco Network Security. Cisco Press.

Werbos, P., 1974. Beyond regression: New tools for prediction and analysis in the behavioral sciences. Ph.D. Thesis, Harvard University.

Zadeh, L.A., 1994. Fuzzy logic, neural networks, and soft computing. Commun. ACM 37, 77–84.