Expert Systems with Applications 37 (2010) 6225–6232

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering

Gang Wang [a,b,*], Jinxing Hao [b,c], Jian Ma [b], Lihua Huang [a]

[a] School of Management, Fudan University, Shanghai 200433, PR China
[b] Department of Information Systems, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong
[c] School of Economics and Management, Beihang University, Beijing 100083, PR China

ARTICLE INFO

Keywords:
Intrusion detection systems
Artificial Neural Networks
Fuzzy clustering

ABSTRACT

Many researches have argued that Artificial Neural Networks (ANNs) can improve the performance of intrusion detection systems (IDS) when compared with traditional methods. However for ANN-based IDS, detection precision, especially for low-frequent attacks, and detection stability are still needed to be enhanced. In this paper, we propose a new approach, called FC-ANN, based on ANN and fuzzy clustering, to solve the problem and help IDS achieve higher detection rate, less false positive rate and stronger stability. The general procedure of FC-ANN is as follows: firstly fuzzy clustering technique is used to generate different training subsets. Subsequently, based on different training subsets, different ANN models are trained to formulate different base models. Finally, a meta-learner, fuzzy aggregation module, is employed to aggregate these results. Experimental results on the KDD CUP 1999 dataset show that our proposed new approach, FC-ANN, outperforms BPNN and other well-known methods such as decision tree, the naïve Bayes in terms of detection precision and detection stability.

1. Introduction

With the coming of Internet age, network security has become the key foundation to web applications, such as online retail sales, online auctions, etc. Intrusion detection attempts to detect computer attacks by examining various data records observed in processes on the network (Anderson, 1980; Endorf, Schultz, & Mellander, 2004). It is one of the important ways to solve network security problems. Detection precision and detection stability are two key indicators to evaluate intrusion detection systems (IDS) (Silva, Santos, Mancilha, Silva, & Montes, 2008). In order to enhance the detection precision and detection stability, many researches have been done (e.g., Patcha & Park, 2007). In the early stage, the research focus lies in using rule-based expert systems and statistical approaches (Manikopoulos & Papavassiliou, 2002). But when encountering larger datasets, the results of rule-based expert systems and statistical approaches become worse. Thus a lot of data mining techniques have been introduced to solve the problem (e.g., Dokas, Ertoz, Lazarevic, Srivastava, & Tan, 2002; Wu & Yen, 2009). Among these techniques, Artificial Neural Network (ANN) is one of the widely used techniques and has been successful in solving many complex practical problems. And ANN has been successfully applied into IDS (Endorf et al., 2004; Ryan, Lin, & Miikkulainen, 1998).

However, the main drawbacks of ANN-based IDS exist in two aspects: (1) lower detection precision, especially for low-frequent attacks, e.g., Remote to Local (R2L), User to Root (U2R), and (2) weaker detection stability (Beghdad, 2008). For the above two aspects, the main reason is that the distribution of different types of attacks is imbalanced. For low-frequent attacks, the leaning sample size is too small compared to high-frequent attacks. It makes ANN not easy to learn the characters of these attacks and therefore detection precision is much lower. In practice, low-frequent attacks do not mean they are unimportant. Instead, serious consequence will be caused if these attacks succeeded. For example, if the U2R attacks succeeded, the attacker can get the authority of root user and do everything he likes to the targeted computer systems or network device. Furthermore in IDS the low-frequent attacks are often outliers. Thus ANN is unstable as it often converges to the local minimum (Haykin, 1999). Although prior research has proposed some approaches, when encountering large datasets, these approaches become not effective (Joo, Hong, & Han, 2003; Patcha & Park, 2007).

To solve the above two problems, we propose a novel approach for ANN-based IDS, FC-ANN, to enhance the detection precision for low-frequent attacks and detection stability. The general procedure of FC-ANN approach has the following three stages. In the first stage, a fuzzy clustering technique is used to generate different training subsets. Based on different training sets, different ANNs

* Corresponding author. Address: Department of Information Systems, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong. Tel.: +852 9799 0955; fax: +852 2788 8694.
E-mail address: wgedison@gmail.com (G. Wang).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.02.102

are trained in the second stage. In the third stage, in order to eliminate the errors of different ANNs, a meta-learner, fuzzy aggregation module, is introduced to learn again and combine the different ANN's results. The whole approach reflects the famous philosophy "divide and conquer". By fuzzy clustering, the whole training set is divided into subsets which have less number and lower complexity. Thus the ANN can learn each subset more quickly, robustly and precisely, especially for low-frequent attacks, such as U2R and R2L attacks. To illustrate the applicability and capability of the new approach, the results of experiments on KDD CUP 1999 dataset demonstrated better performance compared to BPNN and other well-known methods such as decision tree, the naïve Bayes in terms of detection precision and detection stability.

The rest of this paper is organized as follows. In Section 2, we discuss the related work on IDS. In Section 3, we elaborate the framework of FC-ANN approach, and explain its principles and working procedures. To evaluate the FC-ANN approach, Section 4 illustrates the data preparation, evaluation criteria, results and discussions of experiments. Finally, Section 5 draws conclusions and future research directions.

2. Related work on IDS

IDS is split into two categories: misuse detection systems and anomaly detection systems (Anderson, 1980; Endorf et al., 2004). Misuse detection is used to identify intrusions that match known attack scenarios. However, anomaly detection is an attempt to search for malicious behavior that deviates from established normal patterns. In this paper our interesting is in anomaly detection.

In order to detect the intrusion, various approaches have been developed and proposed over the last decade (Depren, Topallar, Anarim, & Ciliz, 2005; Patcha & Park, 2007). In the early stage, rule-based expert systems and statistical approaches are two typical ways to detect intrusion. A rule-based expert IDS can detect some well-known intrusions with high detection rate, but it is difficult to detect new intrusions, and its signature database needs to be updated manually and frequently (Lindqvist & Porras, 1999). Statistical-based IDS, employs various statistical methods including principal component analysis (Shyu, Chen, Sarinnapakorn, & Chang, 2003), cluster and multivariate analysis (Taylor & Alves-Foss, 2001), Bayesian analysis (Barbard, Wu, & Jajodia, 2001), and frequency and simple significance tests (Qin & Hwang, 2004). But this type of IDS needs to collect enough data to build a complicated mathematical model, which is impractical in the case of complicated network traffic (Gordeev, 2000).

To solve the limitations of above methods, a number of data mining techniques have been introduced (Dokas et al., 2002; Wu & Yen, 2009). Among these techniques, ANN is one of the most used techniques and has been successfully applied to intrusion detection (Horeis, 2003; Joo et al., 2003; Kevin, Rhonda, & Jonathan, 1990; Tan, 1995). According to different types of ANN, these techniques can be classified into the following three categories: supervised ANN-based intrusion detection, unsupervised ANN-based intrusion detection, and hybrid ANN-based intrusion detection.

Supervised ANN applied to IDS mainly includes multi-layer feed-forward (MLFF) neural networks and recurrent neural networks (Mukkamala, Janoski, & Sung, 2002). Ryan et al. (1998) and Tan (1995) used MLFF neural networks for anomaly detection based on user behaviors. But in practice the number of training set is very large and the distribution of training set is imbalanced, the MLFF neural networks is easy to reach the local minimum and thus stability is lower. Especially, for low-frequent attacks, the detection precision is very low. Some researchers have compared the effectiveness of supervised ANN with other methods such as support vector machine (SVM) and multivariate adaptive regression splines (MARS) (Mukkamala, Sung, Abraham, & Ramos, 2004; Mukkamala et al., 2002). Supervised ANN had been shown to have lower detection performance than SVM and MARS.

The second category uses unsupervised ANN to classify input data and separate normal behaviors from abnormal or intrusive ones (Endorf et al., 2004). Using unsupervised ANN in intrusion detection has many advantages. The main advantage is that unsupervised ANN can improve their analysis of new data without retraining. Fox (Kevin et al., 1990) was the first to apply a self-organizing map (SOM) to learn the characteristics of normal system activity and identify statistical variations from the normal trends. Just like using supervised learning ANN, the performance of unsupervised ANN is also lower. Especially for low-frequent attacks, unsupervised ANN also gets lower detection precision (Beghdad, 2008).

The third category is hybrid ANN which combines supervised ANN and unsupervised ANN, or combine ANN with other data mining techniques to detect intrusion (Han & Cho, 2005; Jirapummin, Wattanapongsakorn, & Kanthamanon, 2002). The motivation for using the hybrid ANN is to overcome the limitations of individual ANN. Jirapummin et al. (2002) proposed employing a hybrid ANN for both visualizing intrusions using Kohenen's SOM and classifying intrusions using a resilient propagation neural networks. Horeis (2003) used a combination of SOM and radial basis function (RBF) networks. The system offers generally better results than IDS based on RBF networks alone. Han and Cho (2005) proposed an intrusion detection technique based on evolutionary neural networks in order to determine the structure and weights of the call sequences. Chen, Abraham, and Yang (2007) proposed hybrid flexible neural-tree-based IDS based on flexible neural tree, evolutionary algorithm and particle swarm optimization (PSO). Empirical results indicated that the proposed method is efficient. For ANN-based intrusion detection, hybrid ANN has been the trend (Chen et al., 2007). But different ways to construct hybrid ANN will highly influence the performance of intrusion detection. Different hybrid ANN models should be properly constructed in order to serve different aims.

Following this stream, we propose a hybrid ANN, called FC-ANN, to solve the two drawbacks of current ANN-based IDS mentioned in Section 1, i.e., lower detection precision for low-frequent attacks and weaker detection stability. FC-ANN approach introduces fuzzy clustering technique into ordinary ANN. By using fuzzy clustering technique, the whole training set can be divided into subsets which have less size and lower complexity. Therefore based on these sub sets, the stability of individual ANN can be improved, the detection precision, especially for low-frequent attacks, can also be enhanced. The detailed framework of FC-ANN is shown in Section 3.

3. Framework of FC-ANN

In this section, we elaborate our new approach, FC-ANN. We firstly present the whole framework of the new approach. Then we discuss the three main modules, i.e., fuzzy clustering module, ANN module, and fuzzy aggregation module.

3.1. Framework of IDS based on ANN and fuzzy clustering

FC-ANN firstly divides the training data into several subsets using fuzzy clustering technique. Subsequently, it trains the different ANN using different subsets. Then it determines membership grades of these subsets and combines them via a new ANN to get

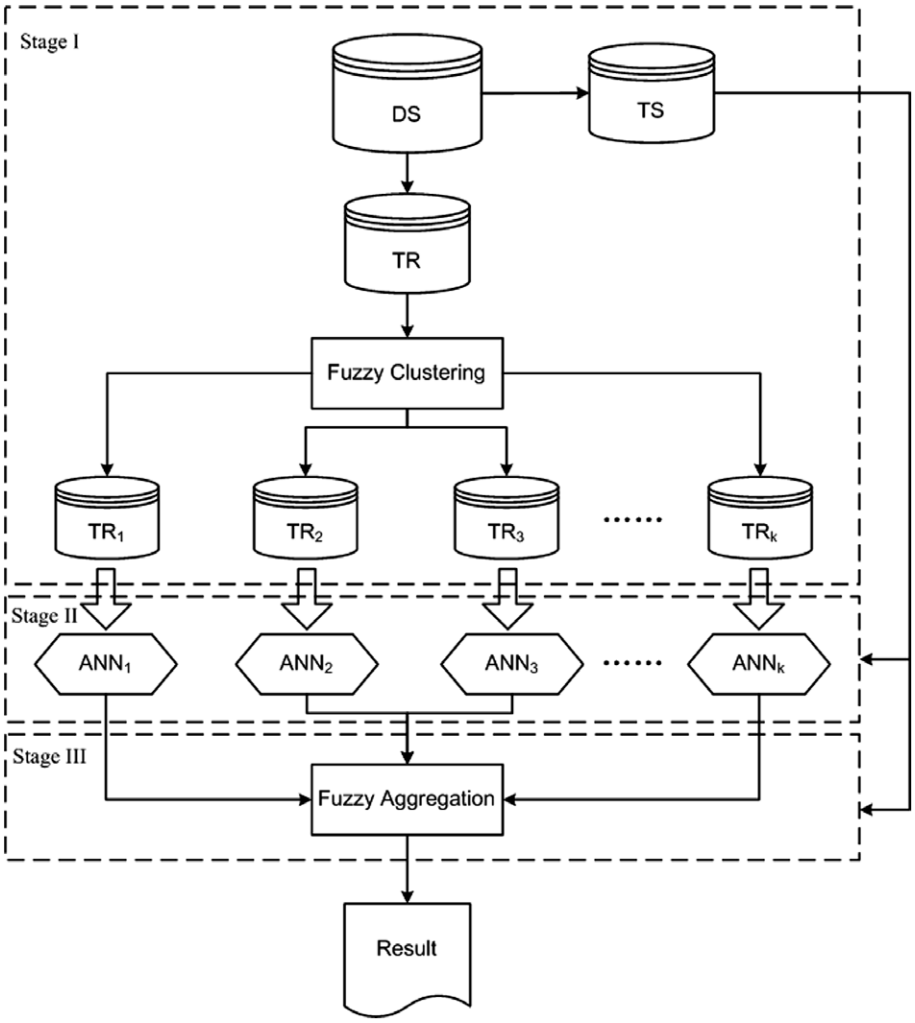

Fig. 1. Framework of FC-ANN for IDS.

final results. The whole framework of FC-ANN is illustrated in Fig. 1.

As typical machine learning framework, FC-ANN incorporates both the training phase and testing phase. The training phase includes the following three major stages:

Stage I: For an arbitrary data set *DS*, it is firstly divided into training set *TR* and testing set *TS*. Then the different training subsets $TR_1, TR_2, \ldots, TR_k$ are created from *TR* with fuzzy clustering module.

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

Stage III: In order to reduce the error for every ANN_i, we simulate the ANN_i using the whole training set *TR* and get the results. Then we use the membership grades, which were generated by fuzzy clustering module, to combine the results. Subsequently, we train another new ANN using the combined results.

In the testing phase, we directly input the testing set data into the k different ANN_i and get outputs. Based on these outputs, the final results can then be achieved by the last fuzzy aggregation module.

The three stages of FC-ANN framework raise three important issues: (1) how to create k different training subsets from the original training dataset *TR*; (2) how to create different base model ANN_i with different training subsets; (3) how to aggregate the different results produced by different base model ANN_i. These issues will be addressed by the following sections, respectively.

3.2. Fuzzy clustering module

The aim of fuzzy cluster module is to partition a given set of data into clusters, and it should have the following properties: homogeneity within the clusters, concerning data in same cluster, and heterogeneity between clusters, where data belonging to different clusters should be as different as possible. Through fuzzy clustering module, the training set is clustered into several subsets. Due to the fact that the size and complexity of every training subset is reduced, the efficiency and effectiveness of subsequent ANN module can be improved.

The clustering techniques can be divided into hard clustering techniques and soft clustering techniques (Bezdek, 1973). Beside partition of training set, we also need to aggregate the results for fuzzy aggregation module. Therefore, we choose one of the popular soft clustering techniques, fuzzy c-means clustering, for fuzzy clustering module (Chiu, 1994; Yager & Filev, 1994).

Fuzzy c-means is a data clustering algorithm in which each data point belongs to a clustering to a degree specified by a membership grade (Chiu, 1994; Yager & Filev, 1994). In fuzzy clustering module, it is based on the minimization of the following objective function:

$$J_m^{TR} = \sum_{j=1}^{k}\sum_{i=1}^{n} u_{ij}^{TR^m}\left\|x_i^{TR} - c_j^{TR}\right\|^2, \quad 1 \le m \prec \infty \tag{1}$$

where m is any real number greater than 1, u_{ij}^{TR} is the degree of membership of x_i^{TR} in the cluster j, x_i^{TR} is the ith of d-dimensional measured data, c_j^{TR} is the d-dimensional center of cluster, and $\| * \|$ is any norm expressing the similarity between any measured data and center.

Fuzzy partitioning is carried out through an iterative optimization of the object function shown above, with the update of membership u_{ij}^{TR} and the cluster centers c_j^{TR} by

$$u_{ij}^{TR} = \frac{1}{\sum_{p=1}^{k}\left(\frac{\left\|x_i^{TR} - c_j^{TR}\right\|}{\left\|x_i^{TR} - c_p^{TR}\right\|}\right)^{\frac{2}{m-1}}}, \quad c_j^{TR} = \frac{\sum_{i=1}^{n} u_{ij}^{TR} x_i^{TR}}{\sum_{i=1}^{n} u_{ij}^{TR}} \tag{2}$$

This iteration will stop when:

$$\max_{ij}\left\{\left|u_{ij}^{TR(q+1)} - u_{ij}^{TR(q)}\right|\right\} < \varepsilon \tag{3}$$

where ε is a termination criterion between 0 and 1 and q is the iteration steps. Based on the above analysis, the fuzzy cluster module is composed of the following steps:

Step 1: Initialize $U^{TR} = \left[u_{ij}^{TR}\right]$ matrix: $U^{TR}(0)$ and $q = 1$.

Step 2: At q-step: calculate the centers vectors $C^{TR}(q) = \left[c_j^{TR}\right]$ with $U^{TR}(q)$

$$c_j^{TR} = \frac{\sum_{i=1}^{n} u_{ij}^{TRm} \cdot x_i^{TR}}{\sum_{i=1}^{n} u_{ij}^{TR^m}} \tag{4}$$

Step 3: Update $U(q+1)$

$$u_{ij}^{TR} = \frac{1}{\sum_{p=1}^{k}\left(\frac{\left\|x_i^{TR} - c_j^{TR}\right\|}{\left\|x_i^{TR} - c_p^{TR}\right\|}\right)^{\frac{2}{m-1}}} \tag{5}$$

Step 3: If $\left\|U^{TR}(q+1) - U^{TR}(q)\right\| < \varepsilon$ then Step 5; otherwise return to Step 2.

Step 4: Based on $argmax\left(u_{ij}^{TR}\right)$, every individual sample of TR can be allocated into subsets TR_k.

After the above five steps, the training set TR can be divided into k subsets TR_k. Subsequently, ANN_i is needed to train using these subsets TR_k. Next section, we will discuss how to create different base model ANN_i with different training subset TR_k.

3.3. ANN module

ANN module aims to learn the pattern of every subset. ANN is a biologically inspired form of distributed computation (Anderson, 1995; Haykin, 1999). It is composed of simple processing units, and connections between them. In this study, we will employ classic feed-forward neural networks trained with the back-propagation algorithm to predict intrusion.

A feed-forward neural networks has an input layer, an output layer, with one or more hidden layers in between the input and output layer. The ANN functions as follows: each node i in the input layer has a signal x_i as network's input, multiplied by a weight value between the input layer and the hidden layer. Each node j in the hidden layer receives the signal $In(j)$ according to:

$$In(j) = \theta_j + \sum_{i=1}^{n} x_i w_{ij} \tag{6}$$

Then passed through the bipolar sigmoid activation function:

$$f(x) = \frac{2}{(1+\exp(-x))} - 1 \tag{7}$$

The output of the activation function $f(In(j))$ is then broadcast all of the neurons to the output layer:

$$y_k = \theta_k + \sum_{j=1}^{m} w_{jk} f(In(j)) \tag{8}$$

where θ_j and θ_k are the biases in the hidden layer and the output layer.

The output value will be compared with the target; in this study, we used the mean absolute error as error function:

$$E_m = \frac{1}{2n}\sum_{k}\sqrt{(T_k - Y_k)^2} \tag{9}$$

when n is the number of training patterns, Y_k and T_k are the output value and the target value, respectively.

The gradient descent method searches for the global optimum of the network weights, and partial derivatives $\partial E/\partial w$ are computed for each weight in the network. And the weight will adjust according to the expression:

$$w(t+1) = w(t) - \eta \partial E(t)/\partial w(t) \tag{10}$$

where t is the number of epochs, η is the learning rate.

To accelerate the convergence of the error in the learning procedure, the momentum with the momentum gain, α, is include into Eq. (10) (Anderson, 1995):

$$w(t+1) = w(t) - \eta \partial E(t)/\partial w(t) + \alpha \Delta w(t) \tag{11}$$

in which the value for α is within 0 and 1.

Based on the feed-forward neural networks trained with the back-propagation algorithm, every ANN_i can complete training using different subsets TR_k. However, next question is how to aggregate the different results produced by different base model ANN_i.

3.4. Fuzzy aggregation module

The aim of fuzzy aggregation module is to aggregate different ANN's result and reduce the detection errors as every ANN_i in ANN module only learns from the subset TR_i. Because the errors are nonlinear, in order to achieve the objective, we use anther new ANN to learn the errors as follows:

Step 1: Let the whole training set TR as data to input the every trained ANN_i and get the outputs:

$$Y_j^{TR} = \left[y_{j1}^{TR}, y_{j2}^{TR}, \ldots, y_{jk}^{TR}\right], \quad j = 1, 2, \ldots, n \tag{12}$$

where n is the number of training set: TR, y_{jk}^{TR} is the output of ANN_k.

Step 2: Form the input for new ANN:

$$Y_{input} = \left[Y_1^{TR} \cdot U_1^{TR}, Y_2^{TR} \cdot U_1^{TR}, \ldots, Y_n^{TR} \cdot U_n^{TR}\right] \tag{13}$$

where U_n^{TR} is the membership grade of TR_n belonging to C^{TR}.

Step 3: Train the new ANN. We can use Y_{input} as input and use the whole training set TR's class label as output to train the new ANN.

Through above three steps, the new ANN can learn the errors which caused by the individual ANN_i in ANN module.

During the stage of testing, work procedure of ANN module and fuzzy aggregation module is similar to the above. Firstly we calculate the membership grade, based on the cluster centers C^{TR}. For a new input x_i^{TS} is coming, firstly based on C^{TR}, the membership U^{TS} can be calculated by:

$$u_{ij}^{TS} = \frac{1}{\sum_{p=1}^{k}\left(\frac{\left\|x_i^{TS}-c_j^{TR}\right\|}{\left\|x_i^{TS}-c_p^{TR}\right\|}\right)^{\frac{2}{m-1}}} \tag{14}$$

Then, respectively, using ANN module and fuzzy aggregation module, the output, Y_{output}^{TS}, can be gotten.

4. Experiments and results

To evaluate the performance of FC-ANN approach, a series of experiments on KDD CUP 1999 dataset were conducted. In these experiments, we implemented and evaluated the proposed methods in Matlab 2007b on a Windows PC with Duro-Core 1.83 GHz CPU and 2 GB RAM.

4.1. Data preparation

In the experiments, KDD CUP 1999 dataset is used (KDD data set, 1999). The KDD CUP 1999 dataset is a version of the original 1998 DARPA intrusion detection evaluation program, which is prepared and managed by the MIT Lincoln Laboratory.

The dataset contains about five million connection records as training data and about two million connection records as test data. And the dataset includes a set of 41 features derived from each connection and a label which specifies the status of connection records as either normal or specific attack type. These features have all forms of continuous, discrete, and symbolic variables, with significantly varying ranges falling in four categories: (1) the first category consists of the intrinsic features of a connection, which include the basic features of individual TCP connections. The duration of the connection, the type of the protocol (TCP, UDP, etc.), and network service (http, telnet, etc.) are some of the features. (2) the content features within a connection suggested by domain knowledge are used to assess the payload of the original TCP packets, such as the number of failed login attempts. (3) the same host features examine established connections in the past two seconds that have the same destination host as the current connection, and calculate the statistics related to the protocol behavior, service, etc. (4) the similar same service features inspect the connections in the past two seconds that have the same service as the current connection.

Likewise, attacks fall into four categories: (1) *Denial of Service (DoS):* making some computing or memory resources too busy to accept legitimate users access these resources. (2) *Probe (PRB):* host and port scans to gather information or find known vulnerabilities. (3) *Remote to Local (R2L):* unauthorized access from a remote machine in order to exploit machine's vulnerabilities. (4) *User to Root (U2R):* unauthorized access to local super user (root) privileges using system's susceptibility.

Random selection has been used in many applications to reduce the size of the dataset. In this study, we randomly select 18,285 records, similar to prior research (Beghdad, 2008). The PRB, R2L, and U2R attack classes were totally selected because of their low portion in the KDD dataset. Three-thousand normal connections (records) and 10,000 DoS connections were randomly selected. For the testing step, the KDD testing set was used. Table 1 shows detailed information about the number of all records. It is important to note that the test data includes specific attack types not present in the training data. This makes the intrusion detection task more realistic.

Table 1
Number and distribution of training and test dataset.

Connection type	Training dataset		Testing dataset	
Normal	3000	16.41%	60,593	19.48%
DoS	10,000	54.69%	229,853	73.89%
PRB	4107	22.46%	4166	1.34%
R2L	1126	6.16%	16,189	5.2%
U2R	52	0.28%	288	0.09%

4.2. Evaluation criteria

The following measurements are often proposed to evaluate the detection precision of IDS (Axelsson, 2003): true positives, true negatives, false positives, and false negatives. A true positive indicates that the intrusion detection system detects precisely a particular attack having occurred. A true negative indicates that the intrusion detection system has not made a mistake in detecting a normal condition. A false positive indicates that a particular attack has been detected by the intrusion detection system but that such an attack did not actually occur. A false positive is often produced due to loose recognition conditions, a limitation on detection methods in the intrusion detection system or phenomena caused by particular environmental factors. It represents the accuracy of the detection system. If it is consistently high, this will lead to administrators intentionally ignoring system warnings, and thus allow the system to remain in a dangerous status. A false negative indicates that the intrusion detection system is unable to detect the intrusion after a particular attack has occurred. This is probably caused by a shortage of information about an intrusion type or by the recognition information about such an intrusion event having been excluded from the intrusion detection system. This reveals the completeness of the detection system.

However as the number of instance for the U2R, PRB, and R2L attacks in the training set and test set is every low, these quantities is not sufficient as a standard performance measure (Dokas et al., 2002). Hence, if we use these quantities as a measure for testing the performance of the systems, it could be biased. For these reasons, we give the precision, recall, and *F*-value which are not dependent on the size of the training and the testing samples. They are defined as follows:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{15}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{16}$$

$$F\text{-value} = \frac{(1+\beta^2)^*\text{Recall}^*\text{Precision}}{\beta^{2*}(\text{Recall}+\text{Precision})} \tag{17}$$

where *TP*, *FP*, and *FN* are the number of true positives, false positives, and false negatives, respectively, and β corresponds to the relative importance of precision versus recall and is usually set to 1.

ANN is unstable as it often converges to the local minimum and fails to train. This is one of important factors that significantly influence detection stability of IDS (Beghdad, 2008; Patcha & Park, 2007). Thus besides above evaluation criteria, we also calculate percentage of training successfully to measure detection stability for ANN-based IDS.

$$\text{Percentage of training successfully} = \frac{\text{The number of training successfully}}{\text{The number of training}} \quad (18)$$

4.3. Results and discussions

In our experiments, each item is described by 41 features which form a vector. Note that some features are continuous and some are nominal. Since the clustering and classification algorithms require continuous values, these nominal values will be first converted to continuous values. For the fuzzy clustering module, we divide the training set into six subsets using fuzzy clustering module. And a stand three-layer networks is used for ANN module and fuzzy aggregation module in the experiments. For ANN module, in the input layer, there were 41 nodes. For fuzzy aggregation module, there were five nodes, equal to the number of attacks. The number of output nodes in ANN module and fuzzy aggregation module were all five, equal to the number of attacks, i.e., Normal, DoS, PRB, R2L and U2R. The number of hidden nodes was determined by empirical formula $\sqrt{I+O}+\alpha$ ($\alpha = 1$–10), where I is the number of input node, O is the number of output node and α is random number (Haykin, 1999). Considering the complexity of intrusion detection, in our experiment α is equal to 10. Thus the structure of ANN in ANN module and fuzzy aggregation module are referred as [41; 18; 5] and [5; 13; 5], respectively. The input and hidden nodes used the sigmoid transfer function and the output node used the linear transfer function. The mean square error (MSE) in the training step is 0.001. The learning rate was set at 0.01, and a momentum factor of 0.2 was applied.

We perform 10 experiments by randomly selecting data according to the sampling rules in Section 3.1. And we also compare the results with BPNN, and other well-known methods such as decision tree, naïve Bayes. These three techniques were run with the help of the WEKA Data Mining tool (Witten & Frank, 2005). The average results of experiments are shown in Tables 2–6 and Figs. 2–4.

As shown by above tables, we can clearly see the difference of every evaluation criteria under different attacks, i.e., normal, DoS, PRB, R2L and U2R. While FC-ANN gets similar results for high-frequent attacks normal, DoS and PRB, FC-ANN gets the highest precision, recall and *F*-value than decision tree, naïve Bayes and BPNN for low-frequent attacks, i.e., R2L and U2R.

As is illustrated by Figs. 2–4, we can see that for normal attack, DoS attack and PRB attack, the above four methods get the similar results. To the R2L attack and U2R attack, decision tree, naïve Bayes and BPNN get the similar results. However FC-ANN gets the highest precision, recall and *F*-value than other three methods.

In Table 7, we can observe that FC-ANN gets the average accuracy 96.71, greater than BPNN and naïve Bayes. These results reveal that through introducing fuzzy clustering technique, the detection precision of ANN can be enhanced. Especially to R2L and U2R attacks, the detection precision enhanced greatly.

And we can find the percentage of training successfully of FC-ANN is 100%, much higher than BPNN. The result reveals that the stability of FC-ANN is also improved by clustering the training set.

Table 2
Performance comparison of various methods (normal).

	Decision tree	Naïve Bayes	BPNN	FC-ANN
Precision (%)	91.22	89.22	89.75	91.32
Recall (%)	99.41	97.70	98.20	99.08
F-value (%)	95.14	93.27	93.79	95.04

Table 3
Performance comparison of various methods (DoS).

	Decision tree	Naïve Bayes	BPNN	FC-ANN
Precision (%)	99.84	99.69	99.79	99.91
Recall (%)	97.24	96.65	97.20	96.70
F-value (%)	98.52	98.15	98.48	98.28

Table 4
Performance comparison of various methods (PRB).

	Decision tree	Naïve Bayes	BPNN	FC-ANN
Precision (%)	50.00	52.61	60.94	48.12
Recall (%)	78.13	88.13	88.75	80.00
F-value (%)	60.98	65.89	72.26	60.09

Table 5
Performance comparison of various methods (R2L).

	Decision tree	Naïve Bayes	BPNN	FC-ANN
Precision (%)	33.33	46.15	57.14	93.18
Recall (%)	1.43	8.57	5.71	58.57
F-value (%)	2.74	14.58	10.39	71.93

Table 6
Performance comparison of various methods (U2L).

	Decision tree	Naïve Bayes	BPNN	FC-ANN
Precision (%)	50.00	25.00	50.00	83.33
Recall (%)	15.38	7.69	23.08	76.92
F-value (%)	23.53	11.76	31.58	80.00

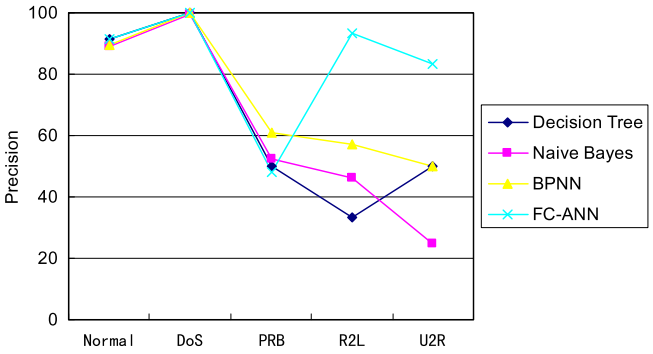

Fig. 2. Precision (%) of different methods.

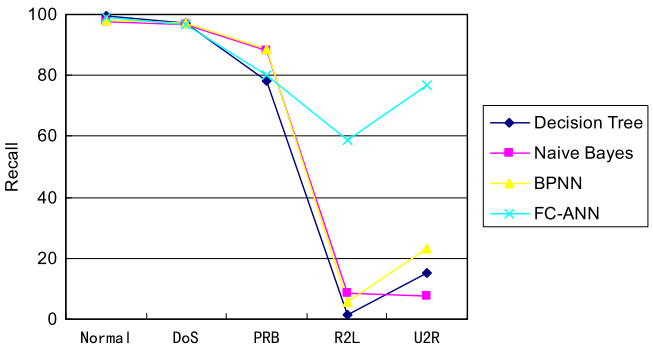

Fig. 3. Recall (%) of different methods.

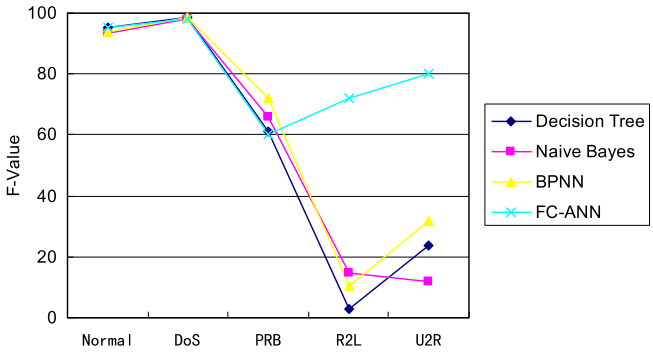

Fig. 4. *F*-value (%) of different methods.

Table 7
Average accuracy, percentage of training successfully, and training time of various methods.

	Decision tree	Na Bayes	BPNN	FC-ANN
Average accuracy (%)	96.75	96.11	96.65	96.71
Percentage of training successfully (%)	100	100	60	100
Training time (s)	2.68	1.93	1538.17	2125.4

For training time, decision tree, naïve Bayes, BPNN and FC-ANN observe 2.68, 1.93, 1538.17 and 2125.4 s, respectively. It is obvious that ANN spent more time in training than decision tree and naïve Bayes. Note that the average training time of FC-ANN is more than BPNN because FC-ANN training time includes fuzzy clustering time and every ANN training time. For the ANN module if the every ANN is training in parallel, then the training time will decrease. In our experiments, we add the training time of fuzzy clustering module, fuzzy aggregation module and maximum training time of parallel individual ANN_i in ANN module. The training time of FC-ANN is 972.08 s.

Therefore we can draw the conclusion that FC-ANN approach can get higher detection precision, especially for low-frequent attacks, and stronger detection stability. Moreover, if the ANN_i in ANN module can operate in parallel, less training time can also be achieved. Such improvement may be largely attributed to the fuzzy clustering module. It makes a heterogeneous training set divided into several homogeneous subsets. During our experiment, we also find the number of clusters, k, also influence the detection precision and recall. Figs. 5–7 show the results.

For high-frequent attacks, i.e., normal, DoS, and PRB attacks, the precision, recall and *F*-value is relatively stable, For low-frequent attacks, i.e., R2L and U2R attacks, the precision, recall and *F*-value are generally increasing with k increasing. This is the reason why FC-ANN can get more detection precision and stability. Subsequently, we can see that for normal, DoS and PRB attacks, the precision, recall and *F*-value will decrease with k increasing further. As the size of these categories is larger than R2L and U2R attacks, the

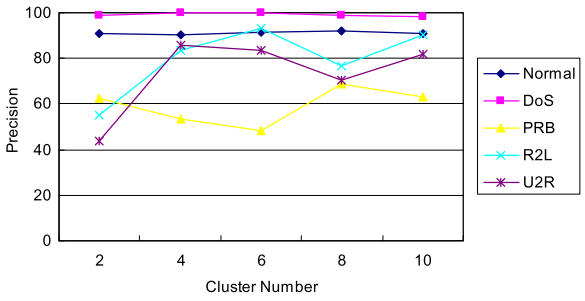

Fig. 5. Precision (%) of different clustering number.

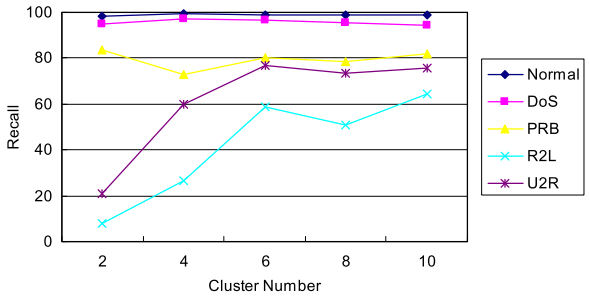

Fig. 6. Recall (%) of different clustering number.

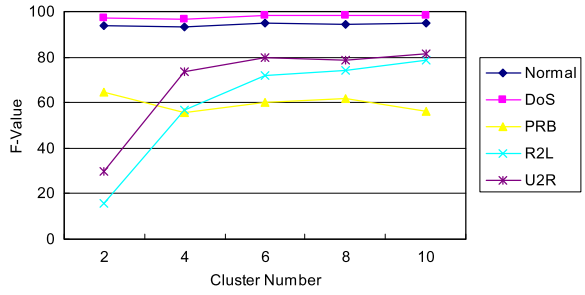

Fig. 7. *F*-value (%) of different clustering number.

average accuracy will decrease although the effectiveness of R2L and U2R detections can be enhanced. For this reason we choose the best average precision, 96.71, to report when the cluster number $k = 6$. The reason underlying the phenomenon might be that there are different distributions of different categories. However, this is only conjecture that should be justified by rigorous theoretical analysis and far more experiments.

5. Conclusions and future directions

Prevention of security breaches completely using the existing security technologies is unrealistic. As a result, intrusion detection is an important component in network security. IDS offers the potential advantages of reducing the manpower needed in monitoring, increasing detection efficiency, providing data that would otherwise not be available, helping the information security community learn about new vulnerabilities and providing legal evidence.

In this paper, we propose a new intrusion detection approach, called FC-ANN, based on ANN and fuzzy clustering. Through fuzzy clustering technique, the heterogeneous training set is divided to several homogenous subsets. Thus complexity of each sub training set is reduced and consequently the detection performance is increased. The experimental results using the KDD CUP 1999 dataset demonstrates the effectiveness of our new approach especially for low-frequent attacks, i.e., R2L and U2R attacks in terms of detection precision and detection stability. In future research, how to determine the appropriate number of clustering remains an open problem. Moreover, other data mining techniques, such as support vector machine, evolutionary computing, outlier detection, may be introduced into IDS. Comparisons of various data mining techniques will provide clues for constructing more effective hybrid ANN for detection intrusions.

Acknowledgements

The authors would like to thank the Editor-in-Chief and reviewers for their recommendation and comments. This work is partially supported by the grants from the Innovation and Technology Fund (ITF) of HK (GHP/006/07, InP/007/08).

References

Anderson, J. P. (1980). *Computer security threat monitoring and surveillance*. Technical Report, Fort Washington, PA, USA.

Anderson, J. (1995). *An introduction to neural networks*. Cambridge: MIT Press.

Axelsson, S. (2003). The base-rate fallacy and the difficulty of intrusion detection. *ACM Transaction on Information and System Security, 3*, 186–205.

Barbard, D., Wu, N., & Jajodia, S. (2001). Detecting novel network intrusions using Bayes estimators. In: *Proceedings of the first SIAM international conference on data mining* (pp. 1–17).

Beghdad, R. (2008). Critical study of neural networks in detecting intrusions. *Computers and Security, 27*(5-6), 168–175.

Bezdek, J. C. (1973). *Fuzzy mathematics in pattern classification*. PhD thesis, Applied Math. Center, Cornell University Ithaca.

Chen, Y. H., Abraham, A., & Yang, B. (2007). Hybrid flexible neural-tree-based intrusion detection systems. *International Journal of Intelligent Systems, 22*(4), 337–352.

Chiu, S. L. (1994). Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems, 2*, 267–278.

Depren, O., Topallar, M., Anarim, E., & Ciliz, M. K. (2005). An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. *Expert Systems with Applications, 29*(4), 713–722.

Dokas, P., Ertoz, L., Lazarevic, A., Srivastava, J., & Tan, P. N. (2002). Data mining for network intrusion detection. *Proceeding of NGDM*, 21–30.

Endorf, C., Schultz, E., & Mellander, J. (2004). *Intrusion detection and prevention*. California: McGraw-Hill.

Gordeev, M. (2000). *Intrusion detection: Techniques and approaches*. <http://www.gosecure.ca/SecInfo/library/IDS/ids2.pdf> (accessed March 2009).

Han, S. J., & Cho, S. B. (2005). Evolutionary neural networks for anomaly detection based on the behavior of a program. *IEEE Transactions on Systems, Man and Cybernetics (Part B), 36*(3), 559–570.

Haykin, S. (1999). *Neural networks: A comprehensive foundation*. Prentice Hall.

Horeis, T. (2003). *Intrusion detection with neural network – Combination of self-organizing maps and redial basis function networks for human expert integration*. <http://ieee-cis.org/_files/EAC_Research_2003_Report_Horeis.pdf> (accessed March 2009).

Jirapummin, C., Wattanapongsakorn, N., & Kanthamanon, P. (2002). Hybrid neural networks for intrusion detection system. *Proceedings of ITC–CSCC*, 928–931.

Joo, D., Hong, T., & Han, I. (2003). The neural network models for IDS based on the asymmetric costs of false negative errors and false positive errors. *Expert Systems with Applications, 25*(1), 69–75.

KDD CUP 1999 dataset (1999). <http://archive.ics.uci.edu/ml/datasets/KDD+Cup+1999+Data> (accessed March 2009).

Kevin, L. F., Rhonda, R. H., & Jonathan, H. R. (1990). A neural network approach towards intrusion detection. In: *Proceedings of the 13th national computer security conference* (pp. 125–134).

Lindqvist, U., & Porras, P. A. (1999). Detecting computer and network misuse through the production-based expert system toolset. In: *IEEE symposium on security and privacy* (pp. 146–161).

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

Mukkamala, S., Janoski, G., & Sung, A. (2002). Intrusion detection using neural networks and support vector machines. In: *Proceedings of the IEEE international joint conference on neural networks* (pp. 1702–1707).

Mukkamala, S., Sung, A. H., Abraham, A., & Ramos, V. (2004). Intrusion detection systems using adaptive regression splines. In: *Proceedings of ICEIS-04* (pp. 26–33).

Patcha, A., & Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks, 51*(12), 3448–3470.

Qin, M., & Hwang, K. (2004). Frequent rules for intrusive anomaly detection with Internet data mining. In: *Proceedings of the 13th USENIX security symposium* (pp. 456–462).

Ryan, J., Lin, M., & Miikkulainen, R. (1998). *Intrusion detection with neural networks. Advances in neural information processing systems* (Vol. 10). Cambridge, MA: Springer.

Shyu, M. L., Chen, S. C., Sarinnapakorn, K., & Chang, L. (2003). A novel anomaly detection scheme based on principal component classifier. In: *Proceedings of the IEEE foundations and new directions of data mining workshop* (pp. 172–179).

Silva, L. D. S., Santos, A. C., Mancilha, T. D., Silva, J. D., & Montes, A. (2008). Detecting attack signatures in the real network traffic with ANNIDA. *Expert Systems with Applications, 34*(4), 2326–2333.

Tan, K. (1995). The application of neural networks to UNIX computer security. In: *Proceedings of the IEEE international conference on neural networks* (Vol. 7, pp. 476–481).

Taylor, C., & Alves-Foss, J. (2001). "Low cost" network intrusion detection. In: *Proceedings of the new security paradigms workshop* (pp. 1–15).

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques*. Boston: Morgan Kaufmann Publishers.

Wu, S., & Yen, E. (2009). Data mining-based intrusion detectors. *Expert Systems with Applications, 36*(3), 5605–5612.

Yager, R. R., & Filev, D. P. (1994). Approximate clustering via the mountain method. *IEEE Transactions on Systems, Man and Cybernetics, 24*(8), 1279–1284.