COMPUTERS & SECURITY 26 (2007) 459–467

available at www.sciencedirect.com

ScienceDirect

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

Computers & Security

An active learning based TCM-KNN algorithm for supervised network intrusion detection

*Yang Li**, *Li Guo*

Institute of Computing Technology, Chinese Academy of Sciences, No. 6 Kexueyuan South Road, Zhongguancun, Haidian District, Beijing 100080, PR China

ARTICLE INFO

Article history:
Received 19 April 2007
Accepted 10 October 2007

Keywords:
Network security
Intrusion detection
TCM-KNN (Transductive Confidence Machines for K-Nearest Neighbors) Algorithm
Machine learning
Active learning

ABSTRACT

As network attacks have increased in number and severity over the past few years, intrusion detection is increasingly becoming a critical component of secure information systems and supervised network intrusion detection has been an active and difficult research topic in the field of intrusion detection for many years. However, it hasn't been widely applied in practice due to some inherent issues. The most important reason is the difficulties in obtaining adequate attack data for the supervised classifiers to model the attack patterns, and the data acquisition task is always time-consuming and greatly relies on the domain experts. In this paper, we propose a novel supervised network intrusion detection method based on TCM-KNN (Transductive Confidence Machines for K-Nearest Neighbors) machine learning algorithm and active learning based training data selection method. It can effectively detect anomalies with high detection rate, low false positives under the circumstance of using much fewer selected data as well as selected features for training in comparison with the traditional supervised intrusion detection methods. A series of experimental results on the well-known KDD Cup 1999 data set demonstrate that the proposed method is more robust and effective than the state-of-the-art intrusion detection methods, as well as can be further optimized as discussed in this paper for real applications.

1. Introduction

With the development of network technologies and applications, network attacks are greatly increasing both in number and severity. As a key technique in network security domain, Intrusion Detection System (IDS) plays vital role of detecting various kinds of attacks and secures the network security and information infrastructures. The main purpose of IDS is to find out intrusions among normal audit data and this can be considered as classification problem.

The two basic methods of detection are signature-based and anomaly-based (Bykova et al., 2001). The signature-based method, also known as misuse detection, looks for a specific signature to match, signaling an intrusion. Provided with the signatures or patterns, they can detect many or all known attack patterns, but they are of little use for as yet unknown attack methods. Most popular intrusion detection systems fall into this category.

Another approach to intrusion detection is called anomaly detection. Anomaly detection applied to intrusion detection and computer security has been an active area of research since it was originally proposed by Denning (1987). Anomaly detection algorithms have the advantage that they can detect new types of intrusions as deviations from normal usage. In

* *Corresponding author.*
E-mail address: samsunglinux@163.com (Y. Li).
0167-4048/$ – see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2007.10.002

this problem, given a set of normal data to train from, and given a new piece of test data, the goal of the intrusion detection algorithm is to determine whether the test data belong to ''normal'' or to an anomalous behavior. However, anomaly detection schemes suffer from a high rate of false alarms. This occurs primarily because previously unseen (yet legitimate) system behaviors are also recognized as anomalies, and hence flagged as potential intrusions.

In this paper, we propose a new supervised intrusion detection method based on TCM-KNN (Transductive Confidence Machines for *K*-Nearest Neighbors) algorithm for intrusion detection. TCM-KNN algorithm is commonly used machine learning and data mining method, thus effective in fraud detection, pattern recognition and outlier detection. To our best knowledge, it is the first time that TCM-KNN algorithm is applied to intrusion detection introduced by us. Contrast experimental results demonstrate that it has good detection performance (high detection rate and low false positives) even when provided with ''small'' data set for training than the state-of-the-art intrusion detection techniques. Most importantly, we further optimize it for intrusion detection in two aspects: (a) introduce active learning method to select much fewer good quality data for training than traditional random sampling, thus alleviate the large amounts of labeling workload for domain experts and reduce the scale of training data set, and consequently reduce the computational cost of TCM-KNN, and (b) feature selection method is proposed to select the most necessary and important features for TCM-KNN, therefore, greatly reduce the computational cost and avoid the ''curse of dimensionality'' effectively. Relevant experiments also suggest that the above optimization is reasonable and effective for TCM-KNN, hence demonstrate that the proposed method could be adopted in realistic network environment.

This remainder of this paper is organized as follows. We outline the related work in Section 2 and introduce TCM (Transductive Confidence Machines) and TCM-KNN (Transductive Confidence Machines for *K*-Nearest Neighbors) algorithm in Section 3. Section 4 details our proposed active learning method for TCM-KNN algorithm aiming at alleviating the annotation workload of training data and reducing the scale of training data set. Section 5 illustrates contrast experiment, active learning based experiment, feature selection based experiment and the evaluations. We conclude our work in Section 6.

2. Related work

In the past decades, a lot of intrusion detection systems have been proposed to detect intrusions. MADAM ID (Mining Audit Data for Automated Models for Intrusion Detection) (Lee and Stolfo, 2000) is one of the best known data mining projects in intrusion detection. It is an off-line IDS to produce anomaly and misuse intrusion models. Association rules and frequent episodes are applied in MADAM ID to replace hand-coded intrusion patterns and profiles with the learned rules. ADAM (Audit Data Analysis and Mining) (Barbarra et al., 2001b) is the second most widely known and well published project in the field. It is an on-line network based IDS. ADAM can detect known attacks as well as unknown attacks. Association rules and classification, two data mining techniques, are used in ADAM. IDDM (Intrusion Detection using Data Mining Techniques) (Abraham, 2001) is a real-time NIDS for misuse and anomaly detection. It applied association rules, meta rules, and characteristic rules. IDDM employs data mining to produce description of network data and uses this information for deviation analysis.

Also, various machine learning and data mining methods have been proposed for intrusion detection and made great success (Lee and Stolfo, 1998; Ghosh and Schwartzbard, 1999; Mahoney and Chan, 2002; Barbara et al., 2001a; Ye, 2000). For example, decision tree and fuzzy association rules are employed in intrusion detection (Sinclair et al., 1999; Luo and Susan, 2000). Neural network is used to improve the performance of intrusion detection (Lippmann and Cunningham, 2000). Support Vector Machine (SVM) is used for unsupervised anomaly detection in Eskin (2002) and for supervised intrusion detection in Mukkamala and Janoski (2002).

All in all, with the appearing of various deliberately designed machine learning and data mining methods, the detection efficiencies based on them are becoming better and better than ever before. However, the detection performance when employing the above traditional data mining methods for supervised intrusion detection is still not satisfactory in practice. The main reason is that the training data set, especially attack training data for learning, is very difficult to acquire in real network environment. Therefore, it has great negative impact on the performances of intrusion detection techniques, i.e., results in low true positives and high false positives. Hence, how to boost the detection performance of current supervised intrusion detection techniques under the environment of lacking adequate training set for modeling is a formidable and promising job.

3. Background of TCM-KNN algorithm

Transduction has been previously used to offer confidence measures for the decision of labeling a point as belonging to a set of pre-defined classes (Gammerman and Vovk, 2002). Transductive Confidence Machines (TCM) introduced the computation of the confidence using Algorithmic Randomness Theory. The confidence measure used in TCM is based upon universal tests for randomness or their approximation (Li and Vitanyi, 1998). The transductive reliability estimation process has its theoretical foundations in the algorithmic theory of randomness developed by Kolmogorov. Unlike traditional methods in machine learning, transduction can offer measures of reliability to individual points, and uses very broad assumptions except for the iid assumption (the training as well as new (unlabeled) points are independently and identically distributed). These properties make transduction an ideal mechanism to the application filed of pattern recognition, fraud detection, outlier detection and so forth.

Martin-Lof proved that there exists a universal method of finding regularities in data sequences. Unfortunately, universal tests are not computable, and have to be approximated using non-universal tests called *p*-values (Proedru et al., 2002). In the literature of significance testing, the *p*-value is defined as the probability of observing a point in the sample space that can be considered more extreme than a sample

of data. This p-value serves as a measure of how well the data support or not a null hypothesis (the point belongs to a certain class). The smaller the p-value, the greater the evidence against the null hypothesis (i.e., the point is an outlier). Users of transduction as a test of confidence have approximated a universal test for randomness (which is in its general form, non-computable) by using a p-value function called strangeness measure (Proedru et al., 2002). The general idea is that the strangeness measure corresponds to the uncertainty of the point being measured with respect to all the other labeled points of a class: the higher the strangeness measure, the higher the uncertainty.

Now, we will give the formal description of TCM-KNN problem for the application field of network intrusion detection. Imagine we have a intrusion detection training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, of n elements, where $X_i = \{x_i^1, x_i^2, \ldots, x_i^n\}$ is the set of feature values (such as the connection duration time, the SYN error numbers, etc.) extracted from the raw network packet (or network flow such as TCP flow) for point i and y_i is the classification for point i, taking values from a finite set of possible classifications (such as normal, DoS attack, Probe attack, etc.), which we identify as $\{1, 2, 3, \ldots, c\}$. We also have a test set of s points similar to the ones in the training set, our goal is to assign to every test point one of the possible classifications. For every classification we also want to give some confidence measures.

In the process of adopting K-Nearest Neighbors (KNN) algorithm, we denote the sorted sequence (in ascending order) of the distances (in this paper, we use the Euclidean distance to compute the distance between pairs of points) of point i from the other points with the same classification y as D_i^y. Also, D_{ij}^y stands for the jth shortest distance in this sequence and D_i^{-y} for the sorted sequence of distances containing points with classification different from y. We assign to every point a measure called the individual strangeness measure. This measure defines the strangeness of the point in relation to the rest of the points. In our case the strangeness measure for a point i with label y is defined as

$$\alpha_{iy} = \frac{\sum_{j=1}^{k} D_{ij}^{y}}{\sum_{j=1}^{k} D_{ij}^{-y}} \tag{1}$$

where k is the number of neighbors used. Thus, our measure for strangeness is the ratio of the sum of the k nearest distances from the same class to the sum of the k nearest distances from all other classes. This is a natural measure to use, as the strangeness of a point increases when the distance from the points of the same class becomes bigger or when the distance from the other classes becomes smaller (Barbara et al., 2006).

Provided with the definition of strangeness, we could use Eq. (2) to compute the p-value as follows:

$$p(\alpha_{\text{new}}) = \frac{\#\{i : \alpha_i \geq \alpha_{\text{new}}\}}{n+1} \tag{2}$$

In Eq. (2), # denotes the cardinality of the set, which is computed as the number of elements in finite set. α_{new} is the strangeness value for the test point (assuming there is only one test point or that the test points are processed one at a time) is a valid randomness test in the iid case. The proof takes advantage of the fact that since our distribution is iid, all permutations of a sequence have the same probability of occurring. If we have a sequence $\{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ and a new element α_{new} is introduced then α_{new} can take any place in the new (sorted) sequence with the same probability, as all permutations of the new sequence are equiprobable. Thus, the probability that α_{new} is among the j largest occurs with probability of at most $j/n+1$. Fig. 1 is the classic TCM-KNN algorithm.

It seems not very difficult for us to catch from Fig. 1 that TCM-KNN algorithm might encounter high computational cost and "curse of dimensionality" since it needs a large amount of distance calculations. In more detail, if the scale of training data set and the dimensions of vectors for points are not well limited, the corresponding distance calculations related to them will result in high computational cost, thus have negative impact on the availability of TCM-KNN. Therefore, accordingly optimization measures (data selection and feature selection) should be employed, in the following sections, we will address them in detail.

4. Active learning for TCM-KNN algorithm

It is a common sense that the performance of machine learning methods for intrusion detection unavoidably depends on security experts in exchange for a greater dependence on collected data. Given good quality labeled data, it is possible that machine learning algorithms can further eliminate the need of an expert and create more autonomous security systems. However, good quality data are very expensive to come by, therefore, how to select good quality data for machine learning methods is a critical problem. The same problem arises in our TCM-KNN algorithm, it also needs such a mechanism to effectively limit the scale of training set, thus reduce the computational cost and the workload needed to label a large amount of data by domain experts, under the condition of ensuring the detection performance without obvious loss. In this section, we employ active learning methods to reach our goals.

4.1. *Introduction to active learning*

The primary motivation for active learning comes from the time or expense of obtaining labeled training examples. In intrusion detection domains, a single training example may require several days and cost thousands of dollars to generate. In the past decades, the most well-known and prominent training data set for this domain is KDD Cup 1999 data set. It has been found that in many cases, if the examples to be labeled are selected carefully and properly, the data requirements for some tasks decrease drastically. Thus, it is expected that the amount of training data needed to train a supervised learning method can be reduced significantly. The motivation behind it is that the cost of manual annotation for producing training material is high, because human annotators are normally involved in the process.

In general, the typical active learning setting consists of the following components, as described in Tong and Koller (2001). The data are divided into (typically few) labeled instances TR and pool of unlabeled instances U. There is also a learner L, which is trained on the labeled data and a query module q. The module q decides which instances of U will be selected

Let k as the number of nearest neighbors to be used; m as the number of training points; c as the classes; r as the points to be classified

for i = 1 to m do

calculate D_i^y, D_i^{-y} and store

end for
calculate α for all training points and store
for i = 1 to r do
Calculate the dist vector as the distances of the new point from all training points
for j = 1 to c do
for every training point t classified as j do

if D_{tk}^j > dist(t) recalculate the alpha value of point t

end for
for every training point t classified as non- j do

if D_{tk}^{-j} > dist(t) recalculate the alpha value of point t

end for
Calculate alpha value for the new point classified as j
Calculate p-value for the new point classified as j
end for
predict the class with the largest p-value
output as confidence one minus the 2nd largest p-value
output as credibility the largest p-value
end for

Fig. 1 – TCM-KNN algorithm.

to be labeled and added in TR, which in turn will be used to train L. In a passive learning setting, q selects instances randomly, as opposed to active learning where the most informative instances are chosen.

The efficiency of active learning methods is measured in two ways. The more popular one is the reduction in the training data needed in order to achieve a certain level of performance. The second is the increase in performance for a certain amount of training data. A common baseline for active learning is random selection of data for annotation and incorporation in the training data.

Active learning is very promising in reducing the amount of training data needed and has been applied to various tasks. Baldridge and Osborne (2003) applied it to parse selection and report savings in annotation costs up to 73%. Tong and Koller (2001) presented impressive results in text classification using active learning. Sassano (2002) used it to reduce the training material needed for Japanese word segmentation, reporting that active learning achieved equal performance with random selection using only 17.4%. In the following sections, the two most widely used active learning methods are described, namely uncertainty based sampling and query by committee.

4.2. *Query function for TCM-KNN algorithm*

Uncertainty based sampling (Tong, 2001) is based on measuring the confidence of the classifier on unseen instances. It is expected that the classifier would benefit more from being trained on instances on which it is more uncertain when attempting to classify them. Uncertainty sampling requires a probabilistic classifier that assigns to unlabeled instances each possible label with a certain probability. Then it computes the entropy for the distribution of each instance and selects the instance or the instances with the highest entropy to be manually annotated and incorporated in the training data. High entropy for an instance suggests that the learner is highly uncertain for the classification it makes. Therefore, the learner would benefit from having such instances annotated as training examples.

Query by committee (Tong, 2001) is a method that is based on measuring the agreement among a committee of classifiers. The committee of classifiers is trained on the labeled material available and then it is presented with the unlabeled instances. The instances presenting the higher disagreement among the classifiers are manually annotated and incorporated in the training data. One way of measuring the disagreement is the vote entropy metric. The intuition behind it is that if a committee of classifiers cannot agree on the label of an instance it can be attributed to the hypothesis that their training set does not include enough or any similar instances, thus giving rise to conflicting decisions by the classifiers. It must be said that query by committee is benefitted by classifiers that work in a different manner so that their decisions are uncorrelated.

Considering the essences of TCM-KNN algorithm, we adopt the uncertainty based sampling method as query function to employ active learning algorithm. Let p be the p-values obtained for a particular example of the possible classification ($i = 1, \ldots, n$). Sort the sequence of p-values in descending order so that the first two p-values, say p_j and p_k are the two highest p-values with classifications j and k, respectively. We assume p_j to be the higher p-value between the two p-values. The predicted classification for the example is j with p-value p_j. This value defines the credibility of the predicted classification. If p_j is not high enough, the prediction is rejected. The lower p-value p_k is used to calculate a confidence value on the predicted classification. In principle, we would want p_j close to 1 and p_k close to 0. Note that the smaller the confidence the larger the ambiguity regarding the top choice. We consider four possible cases of p-values as follows:

(a) p_j high and p_k low: denotes prediction has high credibility and high confidence value.
(b) p_j high and p_k high: denotes prediction has high credibility but low confidence value.
(c) p_j low and p_k low: denotes prediction has low credibility but high confidence value.
(d) p_j low and p_k high: denotes prediction has low credibility and low confidence value.

Seeing from the above four cases, case (a) is the most ideal result and uncertainty in prediction occurs in cases (b)–(d). Note also that uncertainty of prediction occurs if $p_j \approx p_k$. We define "closeness" consistent with the definition of Ho and Wechsler (2003) as below:

$$C(i) = \left| p_j - p_k \right| \tag{3}$$

which indicates the quality of information possessed by the testing example. As $C(i)$ approaches 0, the more uncertain we are about classifying the testing example. The addition of this example to the training data thus provides new information about the structure of the data set. During active learning, one specifies a threshold value ε for $C(i)$, and if $C(i) < \varepsilon$, a decision is made to include example i in the training set. The threshold value in our experiment is empirical and we set it to 0.1.

4.3. *Active learning method for TCM-KNN*

In general, we find it relatively easy to collect unlabeled data sets in intrusion detection. Moreover, the assumption that a domain expert can separately label each example in such a set is also relatively mild. In most cases, the domain expert can directly judge whether the event is malicious. Therefore, by interactive communication with the domain experts with providing the most uncertain data for labeling, the active learner, i.e., the active learning based TCM-KNN algorithm, will improve step by step, until reaching a relatively stable detection performance.

Provided with the uncertain based sampling query function, we could give the active learning method for TCM-KNN in this section. Imagine we have a pool p of unlabeled examples being independent and identically distributed (iid assumption) from some underlying distribution, where each individual example can be labeled separately by a domain expert. The learning algorithm uses this pool to suggest which examples the expert next should label. After the examples have been labeled, they are added to the training set l, which is used to retrain the learning system.

The next section will give the relevant experiments that demonstrate the effectiveness of active learning for TCM-KNN, that is, it would greatly reduce the efforts for labeling the training data set, therefore reduce the scale of the training data set, and consequently result in the reduction of computational cost of TCM-KNN without deteriorating the detection performance. Fig. 2 gives the active learning method for TCM-KNN.

As described in Fig. 2, we run the interactive active learning procedure until the uncertain examples in the unlabeled data pool are exhausted. It is worth noting here that we could select different stopping criteria for us to finish the active learning method, we found our approach is simple and effective in TCM-KNN, and we will detail the relevant results in the next section.

5. Experiments and discussions

In order to verify the effectiveness of our TCM-KNN algorithm for the field of intrusion detection, we make use of the

```
Let l as the labeled training set, p as the unlabeled data pool,
ε as the threshold for uncertainty calculation

Initiate the training set as l ;
While ( p is not empty)
{
 Choose one instance i from p and compute its p-values;
 If ( C(i) < ε )
        Add i to l and remove i from p;
}
Output classier c on training set l .
```

Fig. 2 – Active learning method for TCM-KNN algorithm.

well-known KDD Cup 1999 Data (KDD 99) to make relevant experiments step by step. Firstly, we make contrast experiments between TCM-KNN algorithm and the classical algorithms commonly effectively used in intrusion detection, including SVM algorithm, neural networks, and KNN (K-Nearest Neighbors) algorithm. Secondly, we test the effectiveness of active learning methods employed in our TCM-KNN algorithm. Finally, we make experiments in order to validate the performance of TCM-KNN algorithm when we selected a feature subset from the KDD 99 data set in case of the ''curse of dimensionality''.

5.1. *Experimental data set and preprocess*

All experiments were performed in a Windows machine having configurations Intel (R) Pentium (R) 4, 1.73 GHz, 1 GB RAM, and the operation system platform is Microsoft Windows XP Professional (SP2). We have used an open source machine learning framework – Weka (the latest Windows version: Weka 3.5). Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a data set or called from your own Java code. Weka contains tools for data preprocessing, classification, regression, clustering, association rules, and visualization. It includes the machine learning algorithms (SVM, neural networks, and KNN) to be compared with the proposed method in this paper for our next experiments. We have used the KDD 99 labeled data set so as to evaluate our method.

The main reason we use the data set is that we need relevant data that can easily be shared with other researchers, allowing all kinds of methods developed by authors all over the world to be easily compared and improved in the same baseline. The common practice in intrusion detection to claim good performance with ''live data'' makes it difficult to verify and improve previous research results, as the traffic is never quantified or released for privacy concerns. The KDD 99 data set might have been criticized for its problems (Lippmann et al., 2000), but it is among the few comprehensive data sets that can be shared in intrusion detection nowadays.

As our test data set, the KDD 99 data set contains one type of normal data and 24 different types of attacks that are broadly categorized in four groups such as Probes, DoS (Denial of Service), U2R (User to Root) and R2L (Remote to Local). The packet information in the original TCP dump files were summarized into connections. This process is completed using the Bro IDS, resulting in 41 features for each connection. Therefore, each instance of data consists of 41 features and each instance of them can be directly mapped into the point discussed in TCM-KNN algorithm.

We sampled twice from KDD 99 data set. For the first time, we extracted 49,402 instances as training set for our experiments. They include 9472 normal instances, 39,286 DoS instances, 127 U2R instances, 112 R2L instances and 405 instances for Probe. Secondly, we extracted 12,350 instances as the independent testing set. By using these two data sets, we thus can effectively evaluate the performances of our method.

Before beginning our experiments, we preprocessed the data set. First, we normalized the data set. For the numerical data, they were normalized by replacing each attribute value with its distance to the mean of all the values for that attribute in the instance space, so as to avoid one attribute will dominate another. In order to do this, the mean and standard deviation vectors must be calculated:

$$\text{mean}[j] = \frac{1}{n}\sum_{i=1}^{n} \text{instance}_i[j] \tag{4}$$

$$\text{standard}[j] = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\text{instance}_i[j] - \text{mean}[j])^2} \tag{5}$$

From this, the new instances can be calculated by dividing the difference of the instances with the mean vector by the standard deviation vector:

$$\text{new instance}[j] = \frac{\text{instance}_i[j] - \text{mean}[j]}{\text{standard}[j]} \tag{6}$$

This results in rendering all numerical attributes comparable to each other in terms of their deviation from the norm. For discrete or categorical data, we represent a discrete value by its frequency. That is, discrete values of similar frequency are close to each other, but values of very different frequencies are far apart. As a result, discrete attributes are transformed to continuous attributes.

Moreover, the experiments employed Euclidean distance metric to evaluate the distance between two points. The metric is defined as follows:

$$\text{distance}(Y_1, Y_2) = \sqrt{\sum_{j=1}^{|Y_1|}(Y_{1j} - Y_{2j})^2} \tag{7}$$

where Y_1 and Y_2 are two feature vectors, Y_{ij} denotes the jth component of Y_i and $|Y_i|$ denotes the length of vector Y_i.

To evaluate our method we used two major indices of performance: the detection rate (also named true positive rate, TP) and the false positive rate (FP). TP is defined as the number of intrusion instances detected by the system divided by the total number of intrusion instances present in the test set. FP is defined as the total number of normal instances that were incorrectly classified as intrusions divided by the total number of normal instances.

5.2. *Contrast experimental results*

In the contrast experiments, we first used the independent extracted training and testing data set for training and test. Moreover, since the ''attack'' training data are very difficult to obtain and usually scarce, we resampled a smaller data set (4940 instances) that is 10 times smaller than that discussed in Section 5.1, and the distributions of instances for normal, DoS, U2R, R2L, Probe are 922, 25, 11, 3954 and 28, respectively. Hence, we use it to test whether our method is still robust and effective when provided with ''small'' data set.

The experimental parameters for SVM, neural networks, KNN algorithms as well as TCM-KNN algorithm were set, respectively. We use C-SVC SVM algorithm, select radius basis function as kernel type and set other relevant parameters as their defaults in Weka. For KNN algorithm, we set k 50 and use linear nearest neighbors search algorithm. As for neural networks, we take back propagation algorithm, use one layer for input, one for output and one for hidden layer. Dimension

for the hidden layer is set as (attribute + class)/2. The other parameters are set as their defaults also. We set the parameter K of our TCM-KNN algorithm 50. It is worth noting that in these experiments we will not adjust the parameters of each algorithm for optimization, in order to compare them in the same reasonable baseline.

Tables 1 and 2 show the detail running results of various supervised intrusion detection methods both when provided with adequate training data set and with "small" training data set, respectively. It is clear that although our method demonstrates just a little higher TP and lower FP than SVM and KNN methods in common cases when provided with adequate attack data, its detection performance is amazingly good than the other methods when lacking adequate attack data for training, since the false positive rate of them sharply increased while TCM-KNN not.

5.3. *Experimental results for active learning based TCM-KNN*

To verify the effectiveness of employing active learning in TCM-KNN to actively select and reduce the training set, we initiated the training set TR, of size 12, which consists of 12 instances randomly drawn from each class (one for normal data, and another for abnormal data that include all the four attack types) in KDD 99. Also, we provided a pool containing 500 unlabeled instances. In such a way, we compared the performance of our TCM-KNN employed with active learning method with that of TCM-KNN employed with random sampling method.

From Fig. 3, it seems that the performance of active learning is much more effective than that of random sampling, for the detection accuracy of TCM-KNN based detection method sharply increases when providing less than 40 active selecting instances from the unlabeled pool data set. The number of instances needed to reach such a good accuracy is far less than that needed by random sampling. In more accurate way, it is 40 or so for active learning and about 2000 for random sampling to gain the same accuracy of 99.7%. Therefore, it is evident and reasonable that our TCM-KNN based detection method can be greatly boosted to gain high accuracy with a few deliberately selected instances, consequently reducing the scale of training set for TCM-KNN without loss of detection performance.

5.4. *Experimental results using selected features*

Feature selection is one of the important and frequently used techniques in data preprocessing. It can reduce the number of features, remove irrelevant, redundant features and bring the immediate effects for intrusion detection. Therefore, for the next experiment, we have performed both Chi-Square method and SVM attribute evaluation method on KDD 99 to acquire the most relevant and necessary features from the 41 features. It is natural and necessary because as discussed in Section 3, the performance of TCM-KNN algorithm may deteriorate when meeting the "curse of dimensionality" and large scale training set, thus, by doing this, we can validate if our algorithm is robust and effective under the circumstance of adopting feature selection for reducing the training set to alleviate the computational cost.

The selected eight features and the experimental results are listed in Tables 3 and 4, respectively. Table 4 shows that the performance of our method is good both on original KDD 99 (TP = 99.7%, FP = 0) and on the data set after employing feature selection (TP = 99.6%, FP = 0.1%). Although the FP increased a little, but it is still very manageable, thus we can argue that it is possible to use a reduced-dimension data set to detect anomalies without significant loss of performance.

Table 1 – Experimental results on common data set

	TP (%)	FP (%)
SVM	99.5	1.0
Neural network	99.8	0.8
KNN	99.2	1.5
TCM-KNN	99.7	0

Table 2 – Experimental results on smaller data set

	TP (%)	FP (%)
SVM	98.7	2.7
Neural network	98.3	2.2
KNN	97.7	4.8
TCM-KNN	99.6	0.1

5.5. *Discussions*

From the above experimental results, we can clearly catch that our method based on TCM-KNN algorithm prevails over the state-of-the-art intrusion detection techniques. Experimental results show it can more effectively detect intrusions with low false positives.

Intuitively, our method fulfills intrusion detection tasks using all the available points already existing in training set to measure. Therefore, it could make correct detection decision by fully exploiting the strangeness discussed in Sections 3 and 4. The experimental results both on the "smaller" data

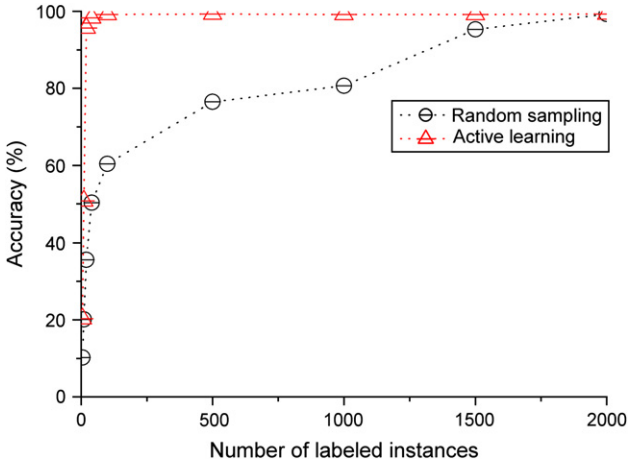

Fig. 3 – Active learning method for TCM-KNN algorithm.

Table 3 – Feature selection results based on Chi-Square approach

Rank	Feature
1	dst_host_same_srv_rate
2	dst_host_diff_srv_rate
3	dst_host_rerror_rate
4	src_bytes
5	dst_bytes
6	count
7	hot
8	num_compromised

set and on the data set being employed feature reduction evident the computational cost of our method could be effectively reduced without any obvious deterioration of detection performance, and it can remain good detection performance even training with small data set than the state-of-the-art supervised intrusion detection methods. Therefore, in this sense, we may claim our method can be optimized to a good candidate for intrusion detection in the realistic network environment by using the active learning and feature selection methods discussed in this paper.

In addition, the method does not make assumptions about the data distributions and only requires the number of nearest neighbors utilized in the distance calculation. We claim that the parameter does not need careful tuning and would not affect the detection performance seriously, which is consistent with the arguments in Barbara et al. (2006). We employed an extended experiment to support that the conclusion and the experimental results are depicted in Table 5. The TP and FP measures are little sensitive to the selection of K as the results listed in Table 5. Therefore, in the real applications, we could empirically select it without much consideration of its positive or negative effect on the detection performance of our TCM-KNN algorithm.

6. Conclusions and future work

In this paper, we proposed a novel supervised intrusion detection method based on TCM-KNN algorithm and active learning method. A series of experimental results demonstrate its effectiveness and advantages over the traditional intrusion detection methods.

In the near future, we will deploy the methods discussed in this paper in realistic network environment to verify its availability and performance. Therefore, feature selection and mapping classical attack patterns of specific application to limited points (vectors those are equivalent to the instances from KDD Cup 1999) for our methods are the most important problems to be deliberately resolved in the concrete implementation. Meanwhile, in terms of the good detection performance and strong theory foundation of TCM-KNN, we are currently embarking on applying the improved TCM-KNN algorithm to unsupervised anomaly detection domain and we have made a progressive success (Li et al., 2007; Li and Guo, 2007), it is a promising job that might improve the detection performance compared to the current anomaly detection methods. Moreover, we will also attempt to combine TCM-KNN and other data mining methods such as fuzzy logic to fulfill intrusion detection task aiming at further optimizing the detection performance of our methods in the real network environment.

Table 4 – Experimental results on total and selected features

	Without feature selection	After feature selection
TP (%)	99.7	99.6
FP (%)	0	0.1

Table 5 – Experimental results on various K-nearest neighbors

	$K=10$	$K=20$	$K=50$	$K=100$	$K=200$
TP (%)	98.8	99.1	99.7	99.6	99.3
FP (%)	0.1	0.2	0	0.1	0.2

REFERENCES

Abraham T. IDDM: intrusion detection using data mining techniques. Salisbury, Australia: DSTO Electronics and Surveillance Research Laboratory; May 2001.

Baldridge J, Osborne M. Active learning for HPSG parse selection. In: Proceedings of the seventh conference on natural language learning, Edmonton, Canada; 2003. p. 23–31.

Bykova M, Ostermann S, Tjaden B. Detecting network intrusions via a statistical analysis of network packet characteristics. In: Proceedings of the 33rd southeastern symposium on system theory, Athens, Greece; 2001. p. 309–14.

Barbara D, Wu N, Jajodia S. Detecting novel network intrusions using Bayes estimators. In: Proceedings of the first SIAM conference on data mining, Chicago, USA; 2001. p. 1–17.

Barbarra D, Julia C, Sushil J, Leonard P, Wu NN. ADAM: detecting intrusions by data mining. In: Proceedings of the 2001 IEEE workshop on information assurance and security, NY, USA; 2001. p. 310–18.

Barbara D, Carlotta D, James PR. Detecting outliers using transduction and statistical testing. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, New York, USA; 2006. p. 55–64.

Denning DE. An intrusion detection model. IEEE Transactions on Software Engineering 1987;SE-13:222–32.

Eskin E, Arnold A, Prerau M, Portnoy L, Stolfo S. A geometric framework for unsupervised anomaly detection: detecting intrusions in unlabeled data. In: Proceedings of applications of data mining in computer security, Boston, USA; 2002. p. 78–99.

Ghosh A, Schwartzbard A. A study in using neural networks for anomaly and misuse detection. In: Proceedings of the eighth USENIX security symposium, Washington, USA; 1999. p. 141–51.

Gammerman A, Vovk V. Prediction algorithms and confidence measure based on algorithmic randomness theory. Theoretical Computer Science 2002;287(1):209–17.

Ho SS, Wechsler H. Transductive confidence machine for active learning. In: Proceedings of the IEEE joint conference on neural networks, Portland, USA; 2003. p. 20–4.

Knowledge discovery in databases DARPA archive. Task description, <http://www.kdd.ics.uci.edu/databases/kddcup99/task.html>.
Lee W, Stolfo SJ. Data mining approaches for intrusion detection. In: Proceedings of the 1998 USENIX security symposium, Colorado, USA; 1998. p. 79–94.
Li M, Vitanyi P. Introduction to Kolmogorov complexity and its applications. 2nd ed. Springer Verlag; 1998.
Lee W, Stolfo SJ. A framework for constructing features and models for intrusion detection systems. ACM Transactions on Information and System Security 2000;3(4):227–61.
Luo JX, Susan MB. Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection. International Journal of Intelligent Systems 2000;15(8):687–704.
Lippmann RP, Cunningham PK. Improving intrusion detection performance using keyword selection and neural networks. International Journal of Computer and Telecommunications Networking 2000;34(4):597–603.
Lippmann RP, Fried DJ, Graf I, Haines JW, Kendall KR, McClung D, et al. Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation. In: Proceedings of DARPA information survivability conference and exposition, South Carolina, USA, vol. 2; 2000. p. 12–26.
Li Y, Guo L. An efficient network anomaly detection scheme based on TCM-KNN algorithm and data reduction mechanism. In: Proceedings of the 2007 information assurance and security workshop, West Point, NY, USA; 2007. p. 221–7.
Li Y, Fang BX, Guo L, Chen Y. Network anomaly detection based on TCM-KNN algorithm. In: Proceedings of the 2007 ACM symposium on information, computer and communications security, Singapore; 2007. p. 13–9.
Mahoney M, Chan P. Learning nonstationary models of normal network traffic for detecting novel attacks. In: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining, Edmonton, Canada; 2002. p. 376–85.
Mukkamala S, Janoski GH. Intrusion detection: support vector machines and neural networks. In: Proceedings of the IEEE international joint conference on neural networks, Honolulu, USA; 2002. p. 1702–07.
Proedru K, Nouretdinov I, Vovk V, Gammerman A. Transductive confidence machine for pattern recognition. In: Proceedings of the 13th European conference on machine learning, Heidelberg, Germany; 2002. p. 381–90.
Sassano M. An empirical study of active learning with support vector machines for Japanese word segmentation. In: Proceedings of the 40th annual meeting of the association for computational linguistics, Philadelphia, USA; 2002. p. 505–12.
Sinclair, C, Pierce L, Matzner S. An application of machine learning to network intrusion detection. In: Proceedings of the 15th annual computer security applications conference, Phoenix, USA; 1999. p. 371–7.
Tong S, Active learning: theory and applications. PhD thesis, Stanford University, California; August 2001.
Tong S, Koller D. Support vector machine active learning with applications to text classification. Machine Learning Research 2001;2(11):45–66.
Weka Software. Machine learning. The University of Waikato, Hamilton, New Zealand. Available from: <http://www.cs.waikato.ac.nz/ml/weka/>.
Ye N. A Markov chain model of temporal behavior for anomaly detection. In: Proceedings of the 2000 IEEE systems, man, and cybernetics information assurance and security workshop, West Point, USA; 2000. p. 171–4.

Yang Li received the BS and MS degrees in computer science and technology from the National University of Defense Technology, China in 2001 and 2004, respectively. He is currently pursuing the Ph.D. degree in Institute of Computing Technology, Chinese Academy of Sciences. His research interests include network security, spam fighting, network anomaly detection based on data mining and machine learning methods, etc.

Li Guo received the MS degree from Xiangtan University, China in 1994. She is currently the professor in Institute of Computing Technology, Chinese Academy of Sciences. Her current research interests include network and security.