An Effective TCM-KNN Scheme for High-Speed Network Anomaly Detection

Yang Li
Chinese Academy of Sciences, Beijing China, 100080
liyang@software.ict.ac.cn

Abstract.

Network anomaly detection has been a hot topic in the past years. However, high false alarm rate, difficulties in obtaining exact clean data for the modeling of normal patterns and the deterioration of detection rate because of "unclean" training set always make it not as good as we expect. Therefore, we propose a novel data mining method for network anomaly detection in this paper. Experimental results on the well-known KDD Cup 1999 dataset demonstrate it can effectively detect anomalies with high true positives, low false positives as well as with high confidence than the state-of-the-art anomaly detection methods. Furthermore, even provided with not purely "clean" data (unclean data), the proposed method is still robust and effective.

***Keywords**: network security, anomaly detection, data mining, TCM-KNN*

1 Introduction

As an important branch of intrusion detection field, anomaly detection has been an active area of research in network security since it was originally proposed by Denning [1]. A lot of data mining methods have been proposed for this hotspot [2]. Anomaly detection algorithms have the advantage over misuse detection that they can detect new types of intrusions as deviations from normal usage. In 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 methods 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. Moreover, if the training set is contaminated by the "noisy" data, the detection performance of anomaly detection methods would deteriorate sharply.

In this paper, we present a novel data mining method for network anomaly detection. It is based on TCM-KNN (Transductive Confidence Machines for K-Nearest Neighbors) algorithm, which is successfully applied to pattern recognition, fraud detection and outlier detection [6], [7]. The most distinguished characteristic for it is that it need not construct a classifier as the traditional data mining methods and is immune to the effect of "noisy" data in training dataset, therefore, it has better detection performance than the traditional anomaly detection methods in practice. A series of experiments on the well-known KDD Cup 1999 dataset demonstrate our method has higher detection rate (also named true positive rates) and lower false alarm (also named false positive rates) than the state-of-the-art anomaly detection methods. Furthermore, it holds good performance even inferred by the "noisy" data in training set.

2 TCM-KNN Algorithm

Transductive Confidence Machines (TCM) introduced the computation of the confidence using algorithmic randomness theory [5]. Unlike traditional methods in data mining, 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 (unlabelled) points are independently and identically distributed). There exists a universal method of finding regularities in data sequences. This p-value serves as a measure of how well the data supports 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 with respect to the current available classes). 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 [4]. 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.

Imagine we have a intrusion detection training set $\{(x_{1,}y_1),...,(x_{n,}y_n)\}$, of n elements, where $X_i = \{x_i^1, x_i^2, ..., x_i^n\}$ is the set of feature values (such as the connection duration time, the packet length, 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 $\{1,2,3,...,c\}$. 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 this paper, we combine K-Nearest Neighbors (KNN) algorithm with TCM for TCM-KNN algorithm and it is noted that TCM can be combined with any other data mining methods such as SVM. We denote the sorted sequence (in ascending order) of the distances of point i from the other points with the same classification y as D_i^y. Also, D_{ij}^y will stand 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

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

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.

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

$$p(\alpha_t) = \frac{\#\{i : \alpha_i \geq \alpha_t\}}{n+1} \qquad (2)$$

where # denotes the cardinality of the set, which is computed as the number of elements in finite set. α_t 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 occuring. If we have a sequence $\{\alpha_1, \alpha_2, ..., \alpha_m\}$ and a new element α_t is introduced then α_t 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_t is among the j largest occurs with probability of at most $\frac{j}{n+1}$.

3 Anomaly Detection Framework Based on TCM-KNN Algorithm

In standard TCM-KNN, we are always sure that the point we are examining belongs to one of the classes. However, in anomaly detection, we need not assign a point constructed from the network packets to a certain class, we only attempt to pinpoint the point in question is normal or abnormal.

Therefore, we propose to use a modified definition of α as follows:

$$\alpha_{iy} = \sum{}_{j=1}^{k} D_{ij}^{y} \tag{3}$$

This new definition will make the strangeness value of a point far away from the class considerably larger than the strangeness of points already inside the class. With respect to our anomaly detection task, there are no classes available, then the above test can be administered to the data as a whole (it all belonged to one class - normal). Therefore, it only requires a single α_i per point (as opposed to computing one per class), and the τ used directly reflects the confidence level ($1-\tau$) is required.

The process of our new simplified TCM-KNN algorithm for anomaly detection is depicted in Figure 1:

```
Parameters: $k$ (the nearest neighbors to be used), $m$ (size of training
dataset), $\tau$ (preset threshold), $r$ (instance to be determined)
for $i$ = 1 to $m$ {
calculate $D_i^y$ according to equation (1) for each one in training dataset
and store;
calculate strangeness $\alpha$ according to equation (3) for each one in
training dataset and store;
          }
calculate the strangeness for $r$ according to equation (3);
calculate the p-values for $r$ according to equation (2);
if ($p \le \tau$)
determine $r$ as anomaly with confidence $1-\tau$ and return;
else
claim $r$ is normal with confidence $1-\tau$ and return;
```

Fig. 1. Pseudocode of the TCM-KNN algorithm for anomaly detection

Figure 2 shows us an anomaly detection framework based on TCM-KNN algorithm and it clearly illustrates how to apply the proposed method to the realistic anomaly detection scenario.

The framework includes two phases: training phase and detection phase. In the first phase, three important jobs should be considered:

a) Data collection for modeling: representative data for normal network behaviors should be collected for our method to modeling. It is worth noting here that as anomaly detection, attack data is no need for us to collect.

b) Feature selection & vectorlization: to meet the requirement of TCM-KNN which mainly depends on the distance calculation based on vectors, feature selection and vectorlization work should be employed. For instances, the duration time of a TCP connection, the ratio between the number of SYN packets, etc. might be selected for the features. They are mostly the same as those in KDD Cup 1999 dataset whose connections meta information have been extracted as 41 features.

c) Modeling by TCM-KNN algorithm: for the last step, TCM-KNN algorithm introduced in this paper then calculates the strangeness and p-value for each instance in the training dataset as discussed in Figure 1, thus to construct the anomaly detection engine.

For the detection phase, all the real-time data collected from the network also should be preprocessed to vectors according to the selected features having been acquired in training phase, then would be directed to the anomaly detection engine based on TCM-KNN, benign or malicious traffic would be determined.

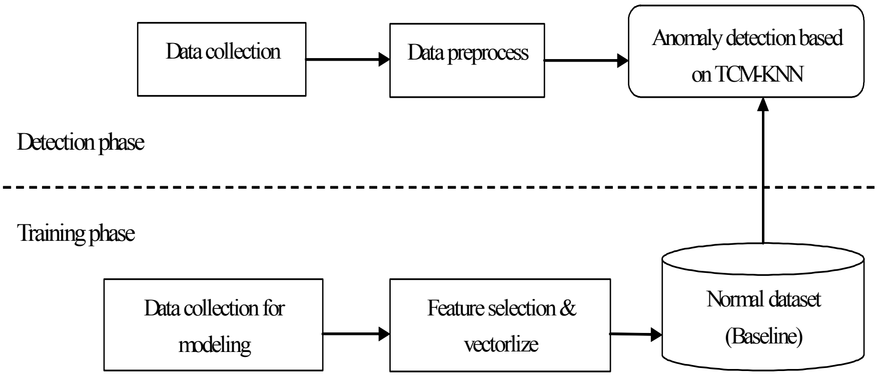
Data collection
Data preprocess
Anomaly detection based on TCM-KNN
Detection phase
Training phase
Data collection for modeling
Feature selection & vectorlize
Normal dataset (Baseline)

Fig. 2. Anomaly Detection Framework Based on TCM-KNN

4 Experimental Results

4.1 Dataset and Preprocess

In our experiments, we select the well-known KDD Cup 1999 dataset (KDD 99) [8] as our test dataset. It includes connections information summarized from the original TCP dump files. A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type. Each connection record consists of about 100 bytes. The attacks contain 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).

Before beginning our experiments, we preprocessed the dataset. First, we normalized the dataset. For the numerical data, in order to avoid one attribute will dominate another attribute, 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. 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 frequency are far apart. As a result, discrete attributes are transformed to continuous attributes.

4.2 Experimental Results

In the contrast experiments between TCM-KNN and the most distinguished anomaly detection methods proposed by authors in [3], we used the sampled "noisy" dataset for training and test (it includes 2048 normal instances and 1870 attack instances). We adopted tenfold cross-validation approach to make the experiment. For the unsupervised anomaly detection algorithms, we set their parameters as the same in [3] for the convenience of comparison. For our TCM-KNN, k is set 50 and τ 0.05 (therefore, the confidence level is 0.95). Figure 3 shows the comparison results of them. It is clear that our method demonstrates higher TP and especially the lower FP than the other three methods.

Moreover, we also use both "clean" dataset and "unclean" dataset for training, to test the adaptive performance of our TCM-KNN algorithm. The result is depicted in Table 1. It clearly shows that a little difference can be observed when we use the two types of training dataset. It strongly demonstrates the proposed TCM-KNN method can be a good candidate for anomaly detection in realistic network environment than the other three methods, because acquiring purely "clean" dataset for training is often impossible and the relatively "unclean" dataset is reasonable. Therefore, a robust detection performance in such a "noisy" network environment is a necessity for anomaly detection method. The results demonstrate our TCM-KNN method has such a good performance.

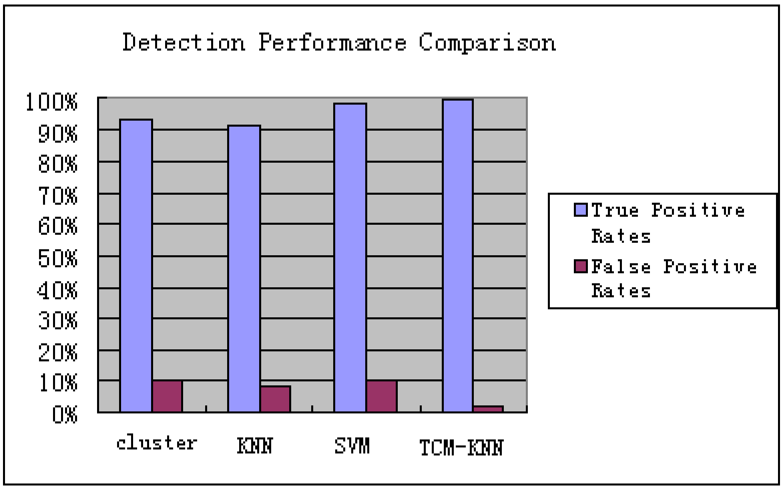

Figure 3. Detection performance comparison results between TCM-KNN and the other three distinguished anomaly detection methods. The left bar for each method denotes TP (true positive rates) and the right one denotes FP (false positive rates).

Table 1. Running results using both “clean” and “unclean” training dataset

	clean dataset	unclean dataset
TP	99.44%	99.42%
FP	1.74%	2.37%

5 Conclusions and Future Work

In this paper, we propose a novel anomaly detection method based on TCM-KNN data mining algorithm. Experimental results demonstrate its effectiveness and advantages over traditional unsupervised anomaly detection methods. As our preliminary work, a lot of work should be improved in the future. Among them, how to reduce the computational cost of TCM-KNN is the most important one. Data reduction and feature selection will be focused around and thereafter the real application of TCM-KNN for anomaly detection would be carried out.

References

[1] Denning, D.E.: An Intrusion Detection Model. IEEE Transactions on Software Engineering. (1987) 222-232

[2] Lee, W., Stolfo, S. J.: Data Mining Approaches for Intrusion Detection. Proceedings of the 1998 USENIX Security Symposium. (1998)

[3] Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S. J.: A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data. In D. Barbara and S. Jajodia (editors), Applications of Data Mining in Computer Security, Kluwer (2002)

[4] Gammerman, A., Vovk, V.: Prediction algorithms and confidence measure based on algorithmic randomness theory. Theoretical Computer Science. (2002) 209-217

[5] Li, M., Vitanyi, P.: Introduction to Kolmogorov Complexity and its Applications. 2nd Edition, Springer Verlag. (1997)

[6] Proedru, K., Nouretdinov, I., Vovk, V., Gammerman, A.: Transductive confidence machine for pattern recognition. Proc. 13th European conference on Machine Learning. (2002) 381-390

[7] Daniel Barbará, Carlotta Domeniconi, James P. Rogers: Detecting outliers using transduction and statistical testing. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. USA (2006) 55-64

[8] Knowledge discovery in databases DARPA archive. Task Description. 2010.

[9] http://www.kdd.ics.uci.edu/databases/kddcup99/task.html