The Journal of Systems and Software 82 (2009) 1974–1981

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

# Constructing attribute weights from computer audit data for effective intrusion detection

Wei Wang [a,*], Xiangliang Zhang [b], Sylvain Gombault [c]

[a] Center for Quantifiable Quality of Service (Q2S) in Communication Systems, Norwegian University of Science and Technology (NTNU), O.S. Bragstads Plass 2E, 7491 Trondheim, Norway
[b] Laboratoire de Recherche en Informatique, Université Paris-Sud 11, 91405 Orsay Cedex, France
[c] Institut Telecom; Télécom Bretagne; RSM Université européenne de Bretagne, France, 2 rue de la Châtaigneraie, CS 17607, 35576 Cesson-Sévigné Cedex, France

ARTICLE INFO

*Article history:*
Received 8 December 2008
Received in revised form 26 May 2009
Accepted 23 June 2009
Available online 28 June 2009

*Keywords:*
Intrusion detection
Masquerade detection
Distance measures
*k*-Nearest neighbor
Principal component analysis
Chi-square

ABSTRACT

Attributes construction and selection from audit data is the first and very important step for anomaly intrusion detection. In this paper, we present several cross frequency attribute weights to model user and program behaviors for anomaly intrusion detection. The frequency attribute weights include plain term frequency (TF) and various forms of term frequency-inverse document frequency (tfidf), referred to as Ltfidf, Mtfidf and LOGtfidf. Nearest Neighbor (NN) and *k*-NN methods with Euclidean and Cosine distance measures as well as principal component analysis (PCA) and Chi-square test method based on these frequency attribute weights are used for anomaly detection. Extensive experiments are performed based on command data from Schonlau et al. The testing results show that the LOGtfidf weight gives better detection performance compared with plain frequency and other types of weights. By using the LOGtfidf weight, the simple NN method and PCA method achieve the better masquerade detection results than the other 7 methods in the literature while the Chi-square test consistently returns the worst results. The PCA method is suitable for fast intrusion detection because of its capability of reducing data dimensionality while NN and *k*-NN methods are suitable for detection of a small data set because of its no need of training process. A HTTP log data set collected in a real environment and the sendmail system call data from University of New Mexico (UNM) are used as well and the results also demonstrate the effectiveness of the LOGtfidf weight for anomaly intrusion detection.



## 1. Introduction

Intrusion detection is an important technique in the defense-in-depth network security framework and has become a widely studied topic in computer security in recent years (Lee and Xiang, 2001). In general, the techniques for intrusion detection are categorized as signature-based detection and anomaly detection. Anomaly detection has been an active research area because of its capability of detecting new attacks (Denning, 1987). In most computing environments, the behavior of a subject (e.g., a program, a user or a network element) is observed and analyzed via the available computer audit data (Lee and Xiang, 2001). It is always a big challenge to construct important and suitable attributes from audit data to best characterize behavioral patterns of a subject so that abnormality can be clearly distinguished from normal activities.

Most previous work in anomaly detection considered two probabilistic attributes of activities in computer systems and networks, namely, the transition attributes and the frequency attributes of audit data. One can also call these two attributes as dynamic models and static models (Yeung and Ding, 2003), or time series and non-time series (Axelsson et al., 2000). In 1996, Forrest et al. (1996) introduced an anomaly detection method called *stide* (sequence time-delay embedding) by using a fixed length of system calls invoked by active and privileged processes. Profiles of normal program behavior were built by enumerating all fixed length of distinct and contiguous system calls that occur in the training data sets and unmatched sequences in actual detection are considered as anomalous. This method can be considered as using the transition attributes of audit data because each sequence contains some order information between system calls. This method is also called as *n-gram*. In subsequent research, a lot of work in detecting anomalous program behavior has used fixed length sequences of system calls as observable (*n-gram*). For example, Lee and Xiang (1998) used a data mining approach (*Ripper*) to study a sample of system call data and characterize sequences of normal data by a small set of rules. The sequences violating those rules were then treated as anomalies. Warrender et al. (1999) proposed a Hidden Markov Model (HMM) based method for modeling and evaluating invisible events. This method was further studied by many other research-

* Corresponding author. Tel.: +47 73592618.
*E-mail address:* wei.wang.email@gmail.com (W. Wang).

0164-1212/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2009.06.040

ers (Yeung and Ding, 2003; Cho and Park, 2003; Wang et al., 2004). Lee and Xiang (2001) used information-theoretic measures for anomaly detection. Masquerade detection is as important as anomalous program behavior detection. Masquerades are people who impersonate other people on the computer (Schonlau and Theus, 2000) and relatively difficult to be detected. Schonlau and Theus (2000) and Schonlau et al. (2001) attempted to detect masquerades by building normal user behavioral models using truncated command sequences. Experiments with six masquerade detection techniques (Schonlau and Theus, 2000; Schonlau et al., 2001): Bayes one-step Markov, Hybrid multi-step Markov, IPAM, sequence-match, compression and uniqueness, were performed and compared. The first five methods are mainly based on the transition attributes of user command data.

The frequency attributes of audit data have also been widely used for intrusion detection. Liao and Vemuri (2002) developed an intrusion detection method by using the text categorization techniques based on the frequency attributes of system calls. In subsequence research, Hu et al. (2003) applied robust support vector machines (SVM) for intrusion detection based on the frequency attributes of system call data. Zhang and Shen (2005) modified the conventional SVM, Robust SVM and one-class SVM, respectively for intrusion detection also based on the frequency property of system call data. Chen et al. (2005) developed artificial neural networks (ANNs) and SVM based methods for detecting potential system intrusions with the frequency attribute of system call data. Yeung and Ding (2003) used information measure for detecting anomalous user and program behavior based on the frequency attributes of computer audit data. In our previous work, we employed non-negative matrix factorization (NMF) (Guan et al., 2009) and self organizing maps (SOM) (Wang et al., 2006), to reduce high dimensional data for intrusion detection with high efficiency and low use of system resources. These techniques are also based on the frequency attributes of computer audit data.

In general, intrusion detection methods based on the transition information model temporal variation of the audit data. The intrusion detection methods using the frequency information, on the other hand, convert the temporal sequences into some non-temporal representation typically in the form of multidimensional feature vectors with no time dimension (Yeung and Ding, 2003). Our previous work (Wang et al., 2006, 2008) is consistent with Ye et al.'s work (2001) and indicates that considering the transition information of audit data can improve some detection accuracy but have to sacrifice some real-time performance compared to using the frequency information. In practice, audit data in intrusion detection problem is typically very large. For example, in colleting system calls of *sendmail* on a host machine, only 112 messages produced a combined trace with the length of over 1.5 million system calls (Forrest et al., 1996). Fast processing of massive audit data in real-time is therefore essential for a practical intrusion detection system (IDS) so that actions for response can be taken as soon as possible. However, intrusion detection methods considering the transition information of audit data usually require much time to train the models and to detect intrusions by processing a large amount of data. For example, it took Hidden Markov Models (HMM) approximately two months to train an anomaly detection model with a large data set (Warrender et al., 1999). This is clearly not adequate for real-time intrusion detection (Wang et al., 2008). On the other hand, intrusion detection methods only taking account of frequency information usually cannot achieve good detection accuracy (Wang et al., 2006, 2008; Ye et al., 2001).

In this paper, we propose a novel intrusion detection method by constructing frequency attribute weights that not only consider the frequency information of events in each sequence of audit data, but also consider the distribution of the event in the whole data (Wang and Gombault, 2007a). We may call this kind of data preprocessing method as considering cross frequency information of audit data. The weights are originally from information retrieval and from text mining and were known as term frequency – inverse document frequency (*tfidf*). These weights almost do not increase the computation expense and are thus suitable for real-time detection. Plain term frequency (TF) and various types of frequency weights defined as *Ltfidf*, *Mtfidf* and *LOGtfidf* are used in this paper for attribute construction. Several distance measures, namely, nearest neighbor (NN) and $k$-NN with Euclidean distance and with Cosine distance as well as Chi-square test are used for masquerades detection based on the four weight schemes. In this paper, we also use principal component analysis (PCA) to discover the interrelationships and dependencies among the attributes of audit data by using the covariance matrix. PCA is effective for real-time masquerade and intrusion detection due to its capacity of dimensionality reduction (Wang et al., 2008).

Extensive experiments are conducted with user command data from Schonlau and Theus (2000) and Schonlau et al. (2001) and testing results show that based on the *LOGtfidf* frequency weight of audit data, even simple NN method can achieve the better masquerade detection results than the other 6 methods in Schonlau and Theus (2000), Schonlau et al. (2001) and also than our previous results with NMF (Guan et al., 2009). The *LOGtfidf* weight improves the detection accuracy with 27.9% than plain frequency TF and improves with 30.8% than *Ltfidf* based on the same NN method. A real world HTTP data set and *Sendmail* system call data from University of New Mexico (UNM) are used as well for further validating the *LOGtfidf* weights and the testing results demonstrate its effectiveness for anomaly intrusion detection. The experimental results also show that PCA method improves a lot the effectiveness as well as efficiency of intrusion detection with frequency weights.

The main contributions of our work lie in the following three aspects. First, instead of using the plain frequency of audit data, we construct frequency attribute weights for masquerade and intrusion detection. The frequency weights not only consider the frequency of each event in its sequence, but also take into account of how important (or how unique) the event is to the whole data set and this helps a lot to improve the detection accuracy. The frequency weights are very effective, very simple and thus are very easy to implement onto a practical IDS. Second, we employ various distance measures (e.g., NN and $k$-NN method with Euclidean distance and Cosine distance) for intrusion detection and compare their detection results as well as discuss the advantages and disadvantages of these measures. Third, instead of only using the first-order statistic of audit data (e.g., Chi-square test), we use PCA to discover the relations and dependencies among attributes in audit data for intrusion detection. Also, instead of having an assumption that most audit data sets are small or low dimensional, PCA method has the capability of processing massive audit data for fast intrusion detection based on dimensionality reduction and on a simple classifier.

The remainder of this paper is organized as follows. The next Section introduces the frequency attribute weights. Section 3 describes the intrusion detection methods based on various distance measures as well as the PCA method. Experiments are described and the results are summarized in Section 4. Discussion is given in Section 5. Conclusion and future work is summarized in Section 6.

## 2. Constructing attributes with various types of frequency weights

Attributes construction is usually the first step for anomaly detection. To facilitate comparison, we construct various types of frequency weights from audit data. The first one is the plain frequency of events in audit data and we call it as term frequency (TF). We call the second types as Liao's term frequency – inverse

**Table 1**
The notation and terminology.

| | |
|---|---|
| $n$ | Total number of sequences in the observation data set |
| $m$ | Total number of distinct events in the observation data set |
| $f_{ij}$ | Frequency of event $i$ in sequence $j$ |
| $n_i$ | Number of times that event $i$ appears in the observation data set |
| $s_i$ | Number of sequences containing event $i$ |
| $X$ | Test sequence |
| $T$ | Training sequences in the observation data set |

document frequency (*Ltfidf*) as it was first used by Liao and Vemuri (2002). The other two types have never been used for intrusion detection by now and we called them as Mean tfidf (*Mtfidf*) and LOG tfidf (*LOGtfidf*), respectively. These four types of frequency attribute weights are described below and the notation and terminology used in this paper are listed in Table 1.

### 2.1. Plain term frequency (TF)

Plain term frequency (TF) may itself be used as the basis for attributes construction in intrusion detection. It is very straightforward. Nearly all the current frequency based intrusion detection methods used this kind of measures for attribute construction (Liao and Vemuri, 2002; Hu et al., 2003; Zhang and Shen, 2005; Chen et al., 2005; Guan et al., 2009; Wang et al., 2006, 2008; Ye et al., 2001). It is defined as

$$tf_{ij} = f_{ij} \tag{1}$$

### 2.2. Liao's term frequency – inverse document frequency (Ltfidf)

Liao and Vemuri (2002) first used this kind of measure for intrusion detection based on system call data. We then call this measure as *Ltfidf*. In subsequent research, Zhang and Shen (2005) and Chen et al. (2005) also used the same measure for intrusion detection based on system call data. It is defined as

$$Ltfidf_{ij} = \frac{f_{ij}}{\sqrt{\sum_{l=1}^{m} f_{lj}^2}} \times \log\left(\frac{n}{n_i}\right) \tag{2}$$

### 2.3. Mean term frequency – inverse document frequency (Mtfidf)

The *Mtfidf* has been used in information retrieval and text mining (Tang et al., 2005) and we propose to use this scheme for intrusion detection in this paper. It is defined as

$$Mtfidf_{ij} = f_{ij} \times \log\left(\frac{n}{s_i}\right) \tag{3}$$

### 2.4. LOG term frequency - inverse document frequency (LOGtfidf)

*LOGtfidf* is a revised scheme for attributes construction. The logarithm of the TF is to amend unfavorable linearity. It is defined as

$$LOGtfidf_{ij} = \log(0.5 + f_{ij}) \times \log\left(\frac{n}{s_i}\right) \tag{4}$$

## 3. Distance measures as well as PCA method for anomaly intrusion detection

### 3.1. *k*-Nearest neighbor (*k*-NN)

*k*-Nearest neighbor (*k*-NN) is a method for classifying objects based on closest training examples in the feature space. It is easily accessible but has been demonstrated effective for many classification tasks (Duda et al., 2004). For a given *k*, *k*-NN ranks the neighbors of a test sequence *X* among the training sample, and uses the class labels of the *k* most nearest neighbors to predict the class of the test vector. Euclidean distance and Cosine distance are usually used for measuring the similarity between two vectors. The Euclidean distance measure and cosine distance measure are respectively defined as follows:

$$dis_{eu}(X, T_j) = \|X - T_j\| = \sqrt{\sum_{i=1}^{m}(x_i - t_{ij})^2} \tag{5}$$

$$dis_{\cos}(X, T_j) = \frac{X'T_j}{\|X\|\|T_j\|} = \frac{\sum_{i=1}^{m} x_i \times t_{ij}}{\sqrt{\sum_{i=1}^{m} x_i^2}\sqrt{\sum_{i=1}^{m} t_{ij}^2}} \tag{6}$$

where $x_i$ is the $i$th variable in the test vector $\mathbf{x}$ (here we use vector $\mathbf{x}$ to represent test sequence $X$); $T_j$ is the sequence $j$ in the training data set and $t_{ij}$ is the $i$th variable in sequences $T_j$.

In anomaly detection, each sequence of the observation data set is first transformed into a data vector, respectively based on the plain TF or various types of frequency weights defined in Eqs. (1)–(4). Suppose there are $m$ distinct events in total in the observation data set, each sequence can then be expressed as a vector with $m$ dimensions. The distance between a new test vector and each vector in the training data set is calculated by using Euclidean distance and Cosine distance defined in Eqs. (5) and (6). The distance scores are sorted and the $k$ nearest neighbors are chosen to determine whether the test vector is normal or not. In anomaly detection, we average the $k$ closest distance scores as the *anomaly index*. If the *anomaly index* of a test sequence vector is above a threshold $\varepsilon$, the test sequence is then classified as abnormal. Otherwise it is considered as normal (Wang and Gombault, 2007a).

### 3.2. Nearest neighbor (NN)

Nearest neighbor (NN) is a slight modification of *k*-nearest neighbor (*k*-NN) presented in Section 3.1, when $k = 1$. NN is simpler than *k*-NN but usually is as effective as *k*-NN for some classification tasks. Similarly, the closest distance between a test vector $\mathbf{x}$ and each vector in the training set is found and used as *anomaly index* for anomaly detection. The test vector is classified as abnormal if its *anomaly index* is above a pre-defined threshold $\varepsilon$.

### 3.3. Chi-square test

Chi-square distance test (also called as $X^2$ test) is a multivariate statistical technique. For a given test vector $\mathbf{x}$, the $X^2$ test statistic is given by the equation (Wang and Gombault, 2007b):

$$X^2 = \sum_{i=1}^{m} \frac{(x_i - \bar{t}_i)^2}{\bar{t}_i} \tag{7}$$

where $x_i$ is the $i$th variable in the test vector $\mathbf{x}$ and $\bar{t}_i$ is the average $i$th variable of all the training vectors. The distance of a test vector $\mathbf{x}$ from the center of the normal data population can be measured by $X^2$ test and are considered as *anomaly index* for the test vector. When the $m$ variables are independent and $m$ is large (e.g., greater than 30), the $X^2$ statistic follows approximately a normal distribution according to the central limit theorem (Ye et al., 2001; Johnson and Wichern, 2002). We compute the mean and standard deviation of the $X^2$ population as $\bar{X}^2$ and $\sigma_{\bar{X}^2}$ and set a threshold based on a zone of some combinations of $\bar{X}^2$ and $\sigma_{\bar{X}^2}$, e.g., $[\bar{X}^2 - \alpha\sigma_{\bar{X}^2}, \bar{X}^2 + \alpha\sigma_{\bar{X}^2}]$, where $\alpha$ is a parameter. For a test vector $\mathbf{x}$, if its *anomaly index* is outside of the zone, it is then classified as abnormal.

### 3.4. Principal component analysis (PCA)

Given a set of observations (sequences) be $\mathbf{x}_1, ,\mathbf{x}_2, \cdots, \mathbf{x}_n$, suppose each observation is represented by a row vector of length $m$. The data set is thus represented by a matrix $X_{n\times m}$

$$X_{n\times m} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} = [\mathbf{x}_1, \quad \mathbf{x}_2, \quad \ldots \quad ,\mathbf{x_n}] \tag{8}$$

The average observation is defined as

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \tag{9}$$

Observation deviation from the average is defined as

$$\boldsymbol{\Phi}_i = \mathbf{x}_i - \boldsymbol{\mu} \tag{10}$$

The sample covariance matrix of the data set is defined as

$$C = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \tag{11}$$

The covariance matrix $C$ considers the first and second-order statistic of variables in audit data. Suppose $(\lambda_1, \mathbf{u}_1), (\lambda_2, \mathbf{u}_2), \ldots, (\lambda_m, \mathbf{u}_m)$ are $m$ eigenvalue-eigenvector pairs of the sample covariance matrix $C$. We choose $k$ eigenvectors having the largest eigenvalues. Often there will be only a few large eigenvalues, and this implies that $k$ is the inherent dimensionality of the subspace governing the "signal" while the remaining $(m-k)$ dimensions generally contain noise (Duda et al., 2004). The dimensionality of the subspace $k$ can be determined by Duda et al. (2004)

$$\frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{m}\lambda_i} \geqslant \alpha \tag{12}$$

where $\alpha$ is the ratio of variation in the subspace to the total variation in the original space. We form a $m \times k$ matrix $\boldsymbol{U}$ whose columns consist of the $k$ eigenvectors. The representation of the data by principal components consists of projecting the data onto the $k$-dimensional subspace according to the following rules (Duda et al., 2004)

$$\mathbf{y}_i = (\mathbf{x}_i - \boldsymbol{\mu})U = \boldsymbol{\Phi}_i U \tag{13}$$

A test data vector $\mathbf{x}$ which represents a test sequence of data can be projected onto the $k$-dimensional subspace according to the rules defined by (13). The distance between the test data vector and its reconstruction in the subspace is simply the distance between the mean-adjusted input data vector $\boldsymbol{\Phi} = \mathbf{x} - \boldsymbol{\mu}$ and

$$\boldsymbol{\Phi}_f = (\mathbf{x} - \boldsymbol{\mu})UU^T = \mathbf{y}U^T \tag{14}$$

If the test data vector $\mathbf{x}$ is normal, that is, if the test data vector is very similar to the training vectors corresponding to normal behavior, the test data vector and its reconstruction will be very similar and the distance between them will be very small (Duda et al., 2004). Based on this property, normal user behavior can be profiled for masquerade and intrusion detection. As PCA seeks a projection that best represents the data in a least-square sense, we use the squared Euclidean distance to measure the distance between the two vectors (Wang and Gombault, 2007b; Wang et al., 2008):

$$\varepsilon = \|\boldsymbol{\Phi} - \boldsymbol{\Phi}_f\|^2 \tag{15}$$

In anomaly detection, $\varepsilon$ are characterized as *anomaly indexes*. If is above a predetermined threshold, the test data $\mathbf{x}$ is then classified as normal. Otherwise it is treated as anomalous.

## 4. Experiments

### 4.1. Masquerade detection by profiling user behavior based on command data

#### 4.1.1. Data set

The command data sets collected by Schonlau and Theus (2000) and Schonlau et al. (2001) are used in our experiments for masquerade detection. The command data consists of user names and the associated command sequences (without arguments). Fifty users are included with 15000 consecutive commands for each user, divided into 150 blocks of 100 commands. The first 50 blocks are uncontaminated and used as training data. Starting at block 51 and onward, some masquerading command blocks, randomly drawn from outside of the 50 users, are inserted into the command sequences of the 50 users. The goal is to correctly detect the masquerading blocks in the user community. The data used in the experiments are available for downloading at http://www.schonlau.net/intrusion.html.

#### 4.1.2. Experimental results

In the experiments, we first convert each block of data into a feature vector based on the four frequency weights. NN, $k$-NN with Euclidean distance and Cosine distance as well as Chi-square test and PCA are then used, respectively for masquerade detection. We use the same threshold for all the users for NN, $k$-NN and PCA methods and use different thresholds for different users for Chi-square distance test based on the zone defined in Section 3.3. There is no updating during the training and detection steps in our experiments. For PCA method, $\alpha$ was set as 99.999% in the experiments based on our previous experiments (Wang et al., 2008). Receiver operating characteristic (ROC) curves are used to evaluate the masquerade detection performance based on different methods with various frequency weights. The ROC curve is the plot of detection rates (DR), calculated as the percentage of masquerades detected, against False Alarm Rates (FAR), calculated as the percentage of normal blocks falsely classified as masquerades. There is a tradeoff between the DR and FAR and the ROC curve is obtained by setting different thresholds. Points nearer to the upper left corner of the ROC curve are the most desirable, as they indicate high DR with correspondingly low FAR.

*4.1.2.1. Evaluating the intrusion detection performance with different attribute weights.* To evaluate the intrusion detection performance with different attribute weights, we plot ROC curves of the results shown in Fig. 1 base on NN method by using Euclidean distance measure with four weights. It is observed from the figure that *LOGtfidf* and *Mtfidf* are much better than TF and *Ltfidf* in terms of detection accuracy. In detail, *LOGtfidf* is slightly better than *Mtfidf* and TF is better than *Ltfidf*. We also plot the ROC curves based on the results of $k$-NN ($k$ = 10) with Cosine distance measure and of PCA methods by using all the four attribute weights and the results are shown in Figs. 2 and 3, respectively. It is seen that the results are consistent with those of Fig. 1.

*4.1.2.2. Evaluating the intrusion detection performance with different distance measures as well as with PCA method.* For comparing the detection performance with different methods, we also plot ROC curves shown in Fig. 4 by using the five distance measures as well as PCA method with *LOGtfidf* as it has been demonstrated as the most effective weight.

From (Fig. 4), it is observed that the PCA, NN and $k$-NN methods outperform Chi-square test method for masquerade detection. The Euclidean distances are slightly better than Cosine distance in terms of detection accuracy. The PCA method is comparable to

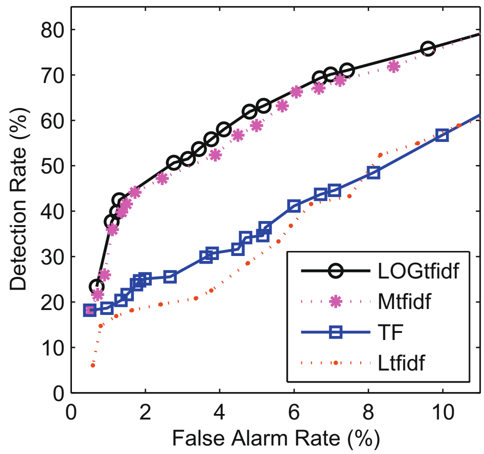


**Fig. 1.** ROC curves for the NN method by using Euclidean distance with four different attribute weights.

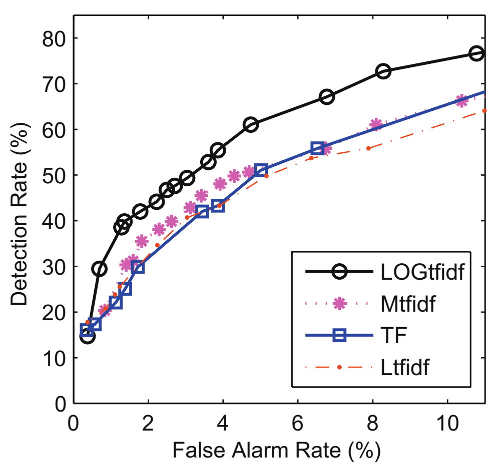


**Fig. 2.** ROC curves for the $k$-NN method ($k = 10$) by using Cosine distance measure with four different attribute weights.

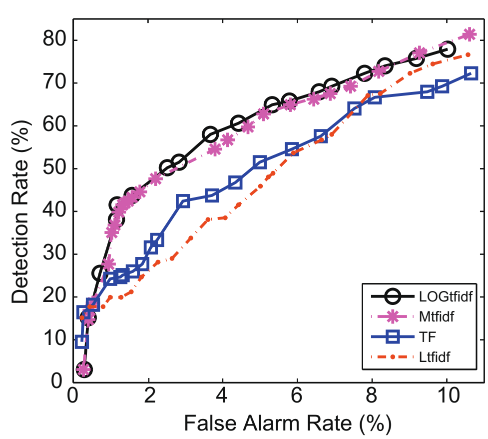


**Fig. 3.** ROC curves for the PCA method with four different attribute weights.

the $k$-NN method when $k = 10$. (Fig. 4) also indicates that the simple NN method with Euclidean distance measure can give a good performance for the detection, using *LOGtfidf* attribute weight.

It is seen from Fig. 4 that the PCA method, considering the first and second-order statistic of audit data, outperforms the Chi-

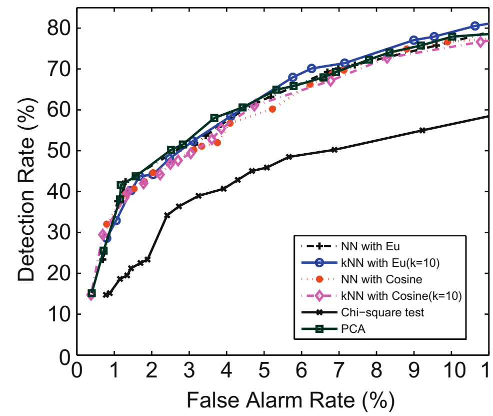


**Fig. 4.** ROC curves with different distance measures as well as with PCA method by using the *LOGtfidf* weight.

square test that only considers the first-order statistic of audit data. PCA method improves the detection performance with 29.3% than Chi-square test based on the same command data.

Based on the same data set, Schonlau and Theus (2000) and Schonlau et al. (2001) has used Bayes one-step Markov, Hybrid multi-step Markov, IPAM, Sequence-Match, Compression and Uniqueness for masquerade detection. We have also used NMF for masquerade detection based on the plain TF attribute of the same data (Guan et al., 2009). (Table 2) summarizes the results obtained with NN method (Euclidean distance) as well as with PCA based on the *LOGtfidf* weight along with the results from another 7 methods reported in Schonlau and Theus (2000), Schonlau et al. (2001), and Guan et al. (2009). Fig. 5 shows the ROC curves of the results with NN method using the *LOGtfidf* weight as well as with other 7 methods.

From (Table 2) and Fig. 5, it is observed that based on the *LOGtfidf* weight, even the simple NN method achieves better results than the other 7 methods reported in Schonlau and Theus (2000), Schonlau et al. (2001), and Guan et al. (2009). The PCA method is comparable to NN method in terms of intrusion detection performance. By using the same NN method, the *LOGtifidf* weight improves the detection rate with 27.9% than plain frequency TF and improves with 30.8% than *Ltfidf* based on the same data set.

### 4.2. Anomaly detection by profiling user behavior based on HTTP log data

#### 4.2.1. Data set

Web-based attack detection is a very important issue in computer network security. To further validate the method, in this

**Table 2**
The false alarm rate and missing alarm rate with comparison.

| Method | | False alarm (%) | Missing alarm (%) |
|---|---|---|---|
| Compression | | 5.0 | 65.8 |
| Sequence-match | | 3.7 | 63.2 |
| IPAM | | 2.7 | 58.9 |
| Hybrid multi-step Markov | | 3.2 | 50.7 |
| Bayes one-step Markov | | 6.7 | 30.7 |
| Uniqueness | | 1.4 | 60.6 |
| NMF | | 1.9 | 57.5 |
| NN with *LOGtfidf* | $\varepsilon = 1.38$ | **1.3** | **57.5** |
| NN with *TF* | $\varepsilon = 0.60$ | 1.3 | 79.7 |
| NN with *Ltfidf* | $\varepsilon = 1.88$ | 1.3 | 83.1 |
| PCA with *LOGtfidf* | $\varepsilon = 1.81$ | **1.1** | **57.5** |
| PCA with *TF* | $\varepsilon = 0.18$ | 1.1 | 75.3 |

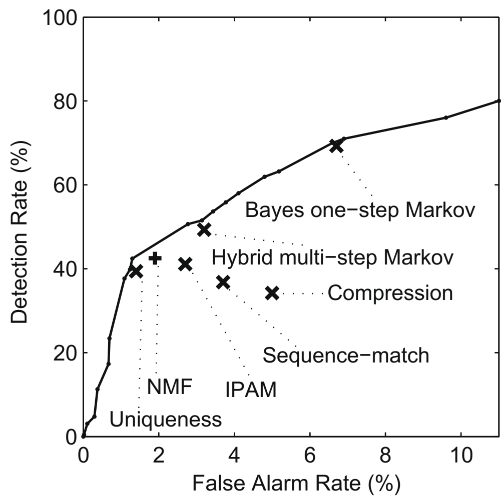


**Fig. 5.** ROC curves of the results obtained with the simple NN method based on the *LOGtfidf* weight along with the results from other 7 methods reported in Schonlau and Theus (2000), Schonlau et al. (2001), and Guan et al. (2009).

paper, we also used a real world HTTP log data set to evaluate the performance of *LOGtfidf* and plain frequency with NN classifier. The HTTP log data was collected on a HTTP server (Apache). In order to reduce a large amount of noise contained in the data set, we filtered out nearly all the static requests (e.g., .html, .jpg, .wav, .wmv, .pdf, .swf) as well as most widely known search engine robots (e.g., googlebot, Msnbot, Spider) before the detection, because static requests cannot be attacks to the server. The data we used in the experiments includes 33967 HTTP requests in which 33856 requests are normal and 111 requests are attacks.

In the experiments, we use character distribution of each path source in each HTTP request as the features. There are 256 types of ASCII codes in total and only 95 types of ASCII codes appear in the path source. These ASCII codes are between 33 and 127. The *LOGtfidf* weights as well as the plain frequency of each ASCII code in each path source are computed. In this way, each HTTP request is thus represented by a 95-dimensional vector and our goal is to identify whether each vector is normal or anomalous.

### 4.2.2. Experiment results

We used the first normal 800 HTTP requests for training and used all the other data for test. The comparison results with NN method are shown in Table 3 and the ROC curves are presented in Fig. 6.

From (Table 3) and Fig. 6, it is observed that even using the NN method, *LOGtfidf* improves the detection accuracy comparing to using plain TF.

**Table 3**
The detection rates false alarm rates with LOGtfidf and plain frequency based on NN classifier.

| *Logtfidf* | | TF | |
|---|---|---|---|
| Detection rate (%) | False alarm rate (%) | Detection rate (%) | False alarm rate (%) |
| 21.6 | 0.2 | 14.4 | 0.5 |
| 26.1 | 0.3 | 27.9 | 0.6 |
| 63.1 | 1.7 | 63.9 | 1.5 |
| 85.6 | 2.5 | 72.1 | 3.1 |
| 95.5 | 3.7 | 86.4 | 5.5 |
| 100 | 4.6 | 100 | 58.4 |

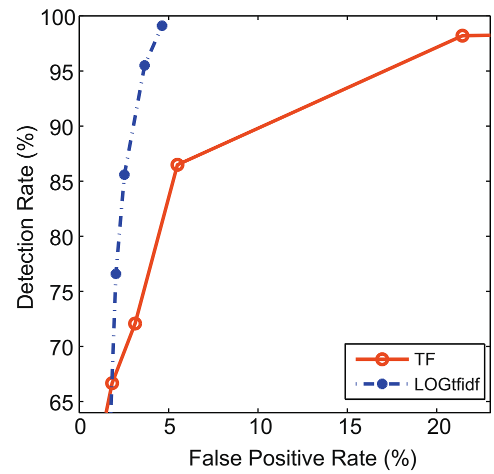


**Fig. 6.** ROC curves of the results obtained with the simple NN method based on the *LOGtfidf* weight and plain TF using a real world HTTP log data set.

### 4.3. Anomaly detection by profiling program behavior based on system call data

### 4.3.1. Data set

We used another data set to test the robustness of the *LOGtfidf* weight with NN. The data set is *sendmail* system call sequences, collected in a UNIX-based host at UNM by Forrest et al. (1996), since they were widely used for testing many other intrusion detection models. In the experiments, we used CERT synthetic *sendmail* data in which one sequence of normal data named "sendmail.int.gz" and 12 sequences of abnormal data including 4 syslog attacks and 2 unsuccessful intrusions (sm5x and sm565a) are used for test. The data sets are available at http://www.cs.unm.edu/immsec/, and the procedures of generating the data are also described on the website.

### 4.3.2. Experiment results

There are 147 normal traces and 34 abnormal traces of system calls in total in the data set. Each trace of the data corresponds a single process. In the experiments, all the system calls associated to the same process is grouped together. We then used *LOGtfidf* weight to convert each process into a vector. Intrusion detection is to identify whether a vector is normal or anomalous. Twenty normal processes are randomly selected for training and the other

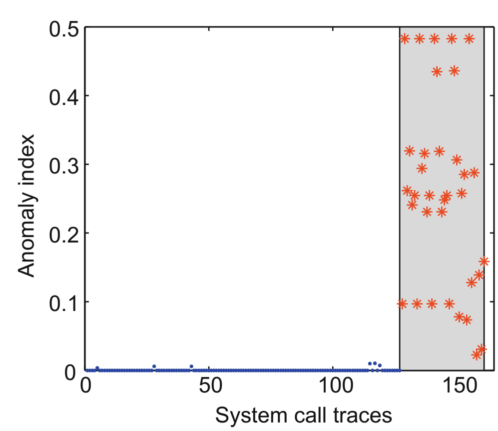


**Fig. 7.** Testing results for the NN method by using Euclidean distance with *LOGtfidf* weight based on *sendmail* system call data. The *y*-axis represents the *anomaly index* and *x*-axis represents system call trace (process) number. The stars (*) in the gray shading indicate attacks and dots (•) with no shading stand for normal data.

127 normal processes as well as 34 abnormal processes are used for test. (Fig. 7) shows the experiment results for the *sendmail* system call data based on NN method by using Euclidean distance with *LOGtfidf* weight.

From (Fig. 7), it is observed that the abnormal data can be 100% distinguished from the normal data without any false alarms by using NN distance with *LOGtfidf* weight. This shows that the *LOGtfidf* have some robustness for anomaly intrusion detection.

## 5. Discussion

Most current anomaly intrusion detection methods consider the transition or the frequency information of audit data. However, the methods considering the transition attributes of audit data normally need high computation and this may not be suitable for real-time detection. The methods taking account of the frequency attributes of audit data have the capability of processing massive data for real-time detection but have to sacrifice some detection accuracy (Warrender et al., 1999; Cho and Park, 2003; Wang et al., 2004; Schonlau and Theus, 2000; Schonlau et al., 2001; Liao and Vemuri, 2002; Hu et al., 2003; Zhang and Shen, 2005; Chen et al., 2005; Guan et al., 2009; Wang et al., 2006; Wang et al., 2008; Ye et al., 2001) and this may reduce the effectiveness of an IDS.

In this paper, we propose several anomaly detection methods based on frequency attribute weights, e.g., the *LOGtfidf* and *Mtfidf* weights, which not only consider the frequency of each event in its sequence, but also takes into account how important the event is to the whole data set. The importance increases proportionally to the number of times an event appears in a sequence but is offset by the frequency of the event in the whole data set. For example, if an event appears in a sequence with a high frequency but seldom appears in the whole data, the *LOGtfidf* and *Mtfidf* score of the event becomes bigger and this helps a lot for detection of abnormal sequence of audit data. As these kinds of weight consider the frequencies of each event in its sequence as well as in the whole data set, we may call these weights as cross frequency attribute weights. Using the cross frequency weights is essentially more effective than only using the plain frequency attributes of audit data for anomaly intrusion.

The additional computation for frequency weights is $\log\left(\frac{n}{s_i}\right)$ (see Eqs. (3) and (4)). Computing $s_i$ only needs to count how many non-zero events in each dimension. Normally the dimensionality of a data set is not too high (e.g., less than 100 for most user models, 95 for HTTP Logs) and the computation is thus not costly. In the experiments for web attack detection, we use a machine (Intel dual core 2.53 GHz with memory 3.5G) to compute the plain frequencies as well as the *LOGtfidf* frequency weights of each distinct character in the HTTP log data (33967 requests in total). The average CPU time used for calculating plain frequency is 0.1404 s while that for calculating *LOGtfidf* frequency weights is 0.2146 s. The cost for computing the cross frequency weights is very close to the expense of computing the plain frequency and is low overhead, compared to use more than 10,000 s to consider transition information of the audit data for the anomaly detection in our previous report (Wang et al., 2008). In this way, by using the cross frequency weights, the detection accuracy can be improved a lot while the computational expense almost does not increase so that an effective IDS can be developed for real-time detection.

Although frequency weights proposed in this paper improve a lot the masquerade detection rates while suppress the missing alarms based on command data, we are aware that the current detection accuracy is not satisfactory in practical environments. Axelsson (2000) indicated that a very high standard of false alarming rates (e.g., less than 1/100,000 per "event" given the stated set of circumstances) is expected for a practical IDS. In real environments, however, false alarm rate with 1% normally can be considered as a good level for the detection. In anomaly intrusion detection, besides the features and statistical detection methods, the question of what kind of data source is used is also crucial. Although the detection results on command data is not very good, in the experiments on system call data, all the attacks are detected without any false alarms, using the frequency weights. In intrusion detection, we suggest that cross frequency weights should be tried, at least to see the comparison results with plain frequency. As shown in our experiments, frequency weights improve the detection rates in many computing circumstances.

In this paper, we use principal component analysis (PCA) to discover the interrelationships and dependencies among the attributes of audit data for intrusion detection. The testing results show that PCA, considering the first and second-order statistic of audit data, is better than Chi-square test method that only takes into account the first-order statistic of audit data for intrusion detection.

Based on *LOGtfidf* weight, *k*-NN and PCA are both effective and comparable for intrusion detection. *k*-NN is a simple algorithm and easily accessible. Compared to PCA, *k*-NN method doesn't need training process. However, it requires $O(m^2n)$ calculations during test step, where $m$ is the dimensionality of the vector and $n$ is the total number of training samples. Usually if the dimensionality of the data is very high and the training data set is very large, it needs a lot of computation. PCA, on the other hand, is relatively time consuming during training process, but only requires $O(mq)$ calculations in detection process, where $q$ is the number of principal components used in the model. Experimental results show that after the high dimensional data is reduced, the original data can be represented by a linear combination of only a small number of principle components without sacrificing valuable information. Based on PCA, normally the original data can be largely reduced for intrusion detection and $q$ is very small. Because the subspace is low dimensional and the classifier is simple, little computational effort is required for the detection. Moreover, system resources may be largely saved for low dimensional data which are conveniently stored and transmitted. It is suggested that PCA is more suitable for processing large of amount of network data for anomaly intrusion detection. However, as a very simple method, *k*-NN is appropriate for intrusion detection if the data is not so massive, because it is light-weight to periodically retrain the detection model by only collecting normal data.

## 6. Conclusion

In this paper, we proposed plain TF weight and several cross frequency weights, namely, *LOGtfidf, Mtfidf* and *Ltfidf* for feature construction from audit data. The nearest neighbor (NN) and *k*-NN with Euclidean distance and Cosine distance as well as PCA and Chi-square test are employed for anomaly intrusion detection. Command data from Schonlau and Theus (2000) and Schonlau et al. (2001) are used to validate the various weights and the different intrusion detection methods. Experimental results show that the *LOGtfidf* and *Mtfidf* weight are better than plain term frequency (TF) and *Ltfidf* in terms of detection accuracy. For detection algorithms, PCA and *k*-NN are effective and comparable for intrusion detection while Chi-square test consistently returns the worst results. Based on the *LOGtfidf* weight, even the simple NN method can achieve the better results than the other 7 methods in Schonlau and Theus (2000), Schonlau et al. (2001) and in Guan et al. (2009). The *LOGtifidf* weight improves the detection rates with 27.9% than plain TF and improves with 30.8% than *Ltfidf* based on the NN method. A real world HTTP log data set and the *sendmail* system call data from UNM are used as well in the experiments and the testing results also demonstrated the effectiveness of *LOGtfidf* weight for anomaly intrusion detection.

Research in progress is on finding more effective weights for constructing valuable attributes from audit data to improve the detection performance. The ways how to combine the frequency attributes with the transition information of audit data to achieve lower false alarm rates and missing alarm rates are also being investigated. Masquerade detection is a difficult task. We also plan to resolve this issue by using other data sources, e.g., key stoke behaviors or adding some arguments to the truncated commands.

## Acknowledgements

The authors thank Dr. Florent Masseglia, INRIA Sophia Antipolis, France, for his help to provide a real world HTTP log data set. The work of NTNU part was supported by the Centre for Quantifiable Quality of Service in Communication Systems (Q2S), Centre of Excellence, which is appointed by the Research Council of Norway and funded by the Research Council, NTNU and UNINETT. The work of TELECOM Bretagne part was supported by French Ministry of Research (CNRS ACI-SI), Dependable Anomaly Detection with Diagnosis (DADDi) project. The research of the first author is also supported by ERCIM "Alain Bensoussan" Fellowship Programme.

## References

Axelsson, S., 2000. The base-rate fallacy and the difficulty of intrusion detection. ACM Transactions on Information Systems Security 3 (3), 186–205.
Axelsson, S., 2000. Intrusion detection systems: A survey and taxonomy, Technical Report 99-15, Chalmers Univ.
Chen, W.H., Hsu, S.H., Shen, H.P., 2005. Application of SVM and ANN for intrusion detection. Computer Operation Research 32, 2617–2634.
Cho, S.B., Park, H.J., 2003. Efficient anomaly detection by modeling privilege flows using hidden markov model. Computer and Security 22 (1), 45–55.
Denning, D.E., 1987. An intrusion detection model. IEEE Transactions on Software Engineering 13 (2), 222–232.
Duda, R.O., Hart, P.E., Stork, D.G., 2004. Pattern Classification, second ed. China Machine Press, Beijing.
Forrest, S., Hofmeyr, S.A., Somayaji, A., Longstaff, T.A., 1996. A sense of self for Unix processes. In: Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, pp. 120–128.
Guan, X., Wang, W., Zhang, X., 2009. Fast intrusion detection based on a non-negative matrix factorization model. Journal of Network and Computer Applications, Elsevier 31 (1), 31–44.
Hu, W., Liao, Y., Vemuri, V.R., 2003. Robust support vector machines for anomaly detection in computer security. In: Proceeding of the 2003 International Conference on Machine Learning and Applications.
Johnson, R.A., Wichern, D.W., 2002. Applied Multivariate Statistical Analysis. Prentice-Hall.
Lee, W., Stolfo, S., 1998. Data mining approaches for intrusion detection. In: Proceedings of the Seventh USENIX Security Symposium, San Antonio, TX, pp. 79–94.
Lee, W., Xiang, D., 2001. Information-theoretic measures for anomaly detection. In: Proceedings of the 2001 IEEE Symposium on Security and Privacy, pp. 130–143.
Liao, Y., Vemuri, V.R., 2002. Using text categorization techniques for intrusion detection. In: 11th USENIX Security Symposium, pp. 51–59.
Schonlau, M., Theus, M., 2000. Detecting masquerades in intrusion detection based on unpopular commands. Information Processing Letters 76, 33–38.
Schonlau, M., Dumouchel, W., Ju, W.-H., Karr, A.F., Theus, M., Vardi, Y., 2001. Computer Intrusion: Detecting Masquerades. Statistical Science 16 (1), 58–74.
Tang, B., Shepherd, M., Milios, E., Heywood, M.I., 2005. Comparing and combining dimension reduction techniques for efficient text clustering. International Workshop on Feature Selection for Data Mining – Interfacing Machine Learning and Statistics in conjunction with 2005 SIAM International Conference on Data Mining, Newport Beach, CA.
Wang, W., Gombault, S., 2007a. Distance measures for anomaly intrusion detection. In: Proceedings of 2007 International Conference on Security and Management (SAM'07), Las Vegas, NV, pp. 17–23.
Wang, W., Gombault, S., 2007b. Detecting masquerades with principal component analysis based on cross frequency weights. In: Proceedings of 14th Anniversary HP-SUA Workshop, Munich, Germany, pp. 227–232.
Wang, W., Guan, X., Zhang, X., 2004. Modeling program behaviors by hidden Markov models for intrusion detection. In: Proceedings of the Third International Conference on Machine Learning and Cybernetics (ICMLC'2004), pp. 2830–2835.
Wang, W., Guan, X., Zhang, X., Yang, L., 2006. Profiling program behavior for anomaly intrusion detection based on the transition and frequency property of computer audit data. Computers and Security, Elsevier 25 (7), 539–550.
Wang, W., Guan, X., Zhang, X., 2008. Processing of massive audit data streams for real-time anomaly intrusion detection. Computer Communications Elsevier 31 (1), 58–72.
Warrender, C., Forrest, S., Pearlmutter, B., 1999. Detecting intrusions using system calls: Alternative data models. In: Proceedings of 1999 IEEE Symposium on Security and Privacy, pp. 133–145.
Ye, N., Li, X., Chen, Q., Emran, S.M., Xu, M., 2001. Probabilistic techniques for intrusion detection based on computer audit data. IEEE Transactions on Systems, Man, and Cybernetics 31 (4), 266–274.
Yeung, D.Y., Ding, Y., 2003. Host-based intrusion detection using dynamic and static behavioral models. Pattern Recognition 36 (1), 229–243.
Zhang, Z., Shen, H., 2005. Application of online-training SVMS for real-time intrusion detection with different considerations. Computer Communications 28, 1428–1442.

**Wei Wang** received his B.S. and M.S. degree from Xi'an Shiyou University, Xi'an, China, in 1997 and 2000, respectively, and his Ph.D. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2005. He was a research fellow from 2005 to 2006 in Department of Information and Communication Technology, University of Trento, Italy. He was a postdoctoral research fellow (research engineering expert) in TELECOM Bretagne, France, in 2007. He was a postdoctoral research fellow in IRISA/INRIA (French National Institute Research in Computer Science and Control), France in 2008. He is currently working at Q2S center of NTNU, Norway, as an ERCIM fellow from 2009. His research interests focus on computer network security.

**Xiangliang Zhang** received her B.S. degree in Information and Communication Engineering and M.S. degree in Electronic Engineering from Xi'an Jiaotong University, Xi'an, China, in 2003 and 2006, respectively. She was an internship student in Department of Information and Communication, University of Trento, Italy, from February 2006 to May 2006. She is currently a Ph.D. student in Laboratoire de Recherche en Informatique, mixed with French National Institute for Research in Computer Science and Control (INRIA), National Center for Scientific Research (CNRS) and University of Paris-sud 11, France. She has published over 20 papers in various journals and conferences, including SIGKDD, ECML/PKDD. Her research interests include machine learning, data mining and their applications, e.g., complex system modeling, grid management, bioinformatics, network systems and security.

**Sylvain Gombault** is Associate Professor at Télécom Bretagne, in the Networking, Security and Multimedia (RSM) department, where he belongs to the Serval (SEcuRity and VALidation of networks and services) research team. He holds an engineering degree in computer science from INSA Rennes and a post-degree in networking from Supélec. He first worked in industry and then joined Télécom Bretagne in 1993. His research interests are related to networking security: intrusion detection (policy based and anomaly based) and high-speed network filtering.