ANOMALY NETWORK INTRUSION DETECTION METHOD IN NETWORK SECURITY BASED ON PRINCIPLE COMPONENT ANALYSIS

[1]Witcha Chimphlee, [2]Mohd Noor Md Sap, [3]Abdul Hanan Abdullah, and **[4]Siriporn Chimphlee**

[2,3]Faculty of Computer Science & information System, Universiti Teknologi Malaysia,
81310 UTM Skudai, Johor, Malaysia. Tel: (607) 5532070, Fax: (607) 5565044
{[2]mohdnoor, [3]hanan}@fsksm.utm.my

[1,4]Faculty of Science and Technology, Suan Dusit Rajabhat University
295 Rajasima Road, Dusit, Bangkok, Thailand. Tel: (662) 2445225, Fax: (662) 6687136
{[1]witcha_chi, [4]siriporn_chi}@dusit.ac.th

Abstract

Most current intrusion detection methods cannot process large amounts of audit data for real-time operation. In this paper, anomaly network intrusion detection method based on Principle Component Analysis (PCA) for data reduction and classifier in presented. Each network connection is transformed into an input data vector. Moreover, PCA is applied to reduce the high dimensional data vectors and distance between a vector, and its projection onto the subspace. Based on the preliminary analysis using a set of benchmark data from KDD (Knowledge Discovery and Data Mining) Competition designed by DARPA, PCA demonstrates the ability to reduce huge dimensional data into a lower dimensional subspace without losing important information. This finding can be used to further enhance the detection accuracy in detecting new types of intrusion by taking PCA as the preprocessing requirement in reducing high dimensional data.

Keywords: Anomaly detection, principal component analysis, high dimensional data, vector, covariance matrix.

1 Introduction

Currently most organizations protect sensitive data by using security software consisting of user IDs, passwords, and biometrics or secure cards to confirm that the user attempting to access a computer application or data is indeed who they claim to be. Computer network's security becomes a critical issue and it is important to develop mechanisms to defense against the intrusions. These systems are vulnerable to attacks from both non-authorized users (outsiders attacks) as well as attacks from authorized users who abuse their privileges (insider attacks). As a result to this situation the need for user accountability is very important, both as a deterrent and for terminating abusive computer usage once it is discovered [1]. An Intrusion Detection System (IDS) is a second line important component of the defense-in-depth security mechanisms. Attack alarms from IDSs are usually reported to

auto response systems or security staff for automatic or manual appropriated response actions according to the specific attacks [2].

The intrusion detection problem has received great interest from researchers. In 1999, the Third International Knowledge Discovery and Data Mining Tools Competition were held in conjunction with the Fifth International Conference on Knowledge Discovery and Data Mining (KDD-99). The contest tasks was to build a network intrusion detector from the KDD Cup 1999 data, which is a predictive model capable of distinguishing between "bad" connections (called attacks) and "good" normal connections. Three winning entries in the KDD'99 Classifier Learning contest were B. Pfahringer, I. Levin , and M. Vladimir [3].

Early studies on anomaly detection mainly focus on learning normal system or user behaviors from monitored system log or accounting log data. Examples of the information derived from these logs are: CPU usage, time of login, duration of user session, etc. In recent years, many research in anomaly detection focus on learning normal program behavior.

Data found in intrusion detection problem are high dimensional in nature. It is desirable to reduce the dimensionality of the data for easy exploration and further analysis [3]. We cannot expect that the training data will always consist of only normal instances. Some suspicious data or intrusions may be buried in the data set.

Principle Component Analysis (PCA, also called Karhunen-Loeve transform) is one of the most widely used dimension reduction techniques for data analysis and compression in practice. Its many application areas include data compression, image analysis, visualization, pattern recognition, and time series prediction. The principal component based approach to intrusion detection has some advantages [3]. First, it does not have any distributional assumption. Secondly, it is typical for the data of this type of problem to be high dimensional that PCA is applied to reduce the dimension of data without sacrificing valuable information. Thirdly, PCA can be computed in less amount of time during the detection stage, which makes it possible to use the method in real time. In this paper, we propose anomaly intrusion detection method based on principal component analysis.

This paper is organized as follows. Section 2 provides some background PCA. The proposed scheme is described in Section 3. Section 4 gives the details of the experiments followed by the results and the discussions in Section 5. We conclude our study in Section 6.

2 Intrusion Detection Method Based on PCA

Principal Component Analysis approach functions by projecting users' profiles onto a feature space. PCA is the method that explores the correlations between each feature and finds the most important axis to express the scattering of data. The most important axis denotes the normal state of network activity, and when a attack takes place, it generally

deviates from this axis[4]. When the projection distance with maximum value is extracted as T_h, (which means it is out of range as normal traffic) it will be imply as attack (Eq 1).

$$\begin{cases} d(x) > T_h : attack \\ d(x) \le T_h : normal \end{cases} \quad (1)$$

2.1 PCA using covariance

The significant features are known as *eigenprofiles* because they are the eigenvectors (principal components) of the set of user profiles. The projection operation characterizes a user profile by a weighted sum of the eigenprofile features, so as to detect whether a user profile is anomalous, it is sufficient to compare its weights to those of known user profiles. Figure 1 show projection distance of the normal or abnormal state.

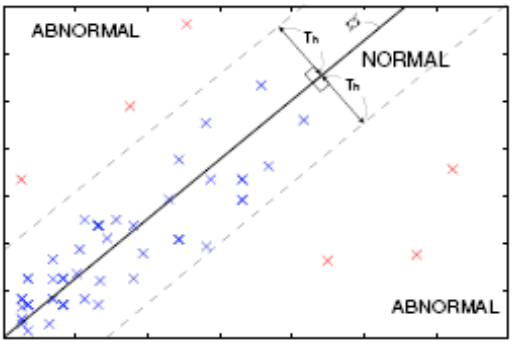

Figure 1. Dividing projection distance into two states [4]

Principal components are particular linear combinations of the *m* random variables $x_1, x_2,\ldots, x_m$ with three important properties[3]: (1) the principal components are uncorrelated, (2) the first principal component has the highest variance, the second principal component has the second highest variance, and so on, and (3) the total variation in all the principal components combined equal to the total variation in the original variables $x_1, x_2,\ldots, x_m$. The new variables with such properties are easily obtained from eigenanalysis of the covariance matrix or the correlation matrix of $x_1, x_2,\ldots, x_m$[5].

In many data sets, the first several principal components contribute most of the variance in the original data set, so that the rest can be disregarded with minimal loss of the variance for dimension reduction of the data [2, 5, 6]. The transformation works as follows.

Given a set of observations of $x_1, x_2,\ldots, x_n$, where each observation is represented by a vector of length *m*, the data set is represented by a matrix X_{nxm}

$$\mathrm{X}_{\mathrm{n\times m}} = \begin{bmatrix} x_{11} & \dots & x_{1m} \\ x_{21} & \dots & x_{2m} \\ \dots & \dots & \dots \\ x_{n1} & \dots & x_{nm} \end{bmatrix} = [x_1, \dots, x_n] \tag{2}$$

The average observation is defined as

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3}$$

The deviation from the average is defined as

$$\Phi_i = x_i - \mu \tag{4}$$

The sample covariance matrix of the data set is defined as

$$C = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T = \frac{1}{n}AA^T \tag{5}$$

where $A = [\Phi_1, \Phi_2, \dots, \Phi_n]$.

To apply PCA, the eigenvalues and corresponding eigenvectors of the sample covariance matrix *C* are usually computed by the Singular Value Decomposition (SVD). Suppose $(\lambda_1, u_1), (\lambda_2, u_2), \dots, (\lambda_m, u_m)$ are *m* eigenvalue-eigenvector pairs of the sample covariance matrix *C*. The dimensionality of the subspace *k* can be determined by [5]

$$\frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{m}\lambda_i} \geq \alpha \tag{6}$$

where α is the ratio of variation in the subspace to the total variation in the original space. We form a *m x k* matrix *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 [6].

$$y_i = U^T(x_i - \mu) = U^T\Phi_i \quad (7)$$

2.2 PCA using Singular Value Decomposition (SVD)

Another algebraic solution for PCA and in the process, find that PCA is closely related to singular value decomposition (SVD).

Definition 1. Singular Value Decomposition. Let A be a general real *M x N* matrix. The singular value decomposition (SVD) of A is the factorization

$$A = U\sum V^T \quad (8)$$

where U is a column-orthonormal N x r matrix, r is the rank of the matrix A, $\sum$ is a diagonal r x r matrix of the eigenvalues λ_i of A, where $\lambda_1 \geq ... \geq \lambda_r \geq 0$ and V is a column-orthonormal M x r matrix [7].

The eigenvalues and the corresponding eigenvectors are sorted in non-increasing order. V is called the right eigenvector matrix, and U the left eigenvector matrix. Note that the eigenvectors obtained by applying SVD to covariance matrix are the same as the principal components.

3 Experiments

We study the performance of our proposed scheme for data reduction and classifier by PCA. The method based on the Euclidean distance. The experiments are conducted under the following framework:

1) The training and testing data are from KDD'99 training data set.
2) Each training data set consists of 5,000 normal connections randomly selected.
3) To assess the accuracy of the classifiers, we carry out 5 independent experiments with 5 different training samples.
4) In each experiment, the classifiers are test with a test set of 65,000 normal connections and 77,291 attack connections from KDD'99 data set.

We want to extract the relevant information in a connection profile and to find principal components of the distribution of the behaviors, or the eigenvectors of the covariance matrix of the set of user profiles, treating a behavior as a point (or vector) in a space of a dimension equal to the number of the different metrics used. The eigenvectors are ordered, each one accounting for a different amount of the variation among the user behaviors [1]. Following the anomaly detection approach, we assume that the anomalies are qualitatively different from the normal instances.

3.1 The KDD'99 Data

KDD Cup 1999 data [8] was the data set used for the *Third International Knowledge Discovery and Data Mining Tools Competition.* The training data set contains 494,021 connection records, and the test data contains 311,029 records that were not from the same probability distribution as the training data. A connection is a sequence of TCP packets containing values of 41 features and labeled as either normal or an attack, with exactly one specific attack type. There are 22 attack types in the training data. The 41 features can be divided into three groups; the first group is the basic features of individual TCP connections, the second group is the content features within a connection suggested by domain knowledge, and the third group is the traffic features computed using a two-second time window. Among the 41 features, 34 are numeric and 7 are symbolic. Table 1 shows features in data set.

Table 1: List of features (KDD-Cup99 task description)

Feature Name	Description	Type
Duration	Length(number of seconds) of the connection	Continuous
Protocol type	Type of the protocol, e.g. tcp, udp, etc.	Discrete
Service	Network service on the destination, e.g., http, telnet, etc.	Discrete
Src_bytes	Number of data bytes from source to destination	Continuous
Dst_bytes	Number of data bytes from destination to source	Continuous
Flag	Normal or error status of the connection	Discrete
Land	1 if connection is from/to the same host/port; 0 otherwise	Discrete
Wrong_fragment	Number of "wrong" fragments	Continuous
Urgent	Number of urgent packets	Continuous
Hot	Number of "hot" indicators	Continuous
Num_failed_logins	Number of failed login attempts	Continuous
Logged in	1 if successfully logged in; 0 otherwise	Discrete
Num_compromised	Number of "compromised" conditions	Continuous
Root_shell	1 if root shell is obtained; 0 otherwise	Discrete
SU_attempted	1 if "su root" command attempted; 0 otherwise	Discrete
Num_root	Number of "root" accesses	Continuous
Num_file_creations	Number of file creation operations	Continuous
Num_shells	Number of shell prompts	Continuous
Num_access_files	Number of operations on access control files	Continuous
Num_outbound_cmds	Number of outbound commands in an ftp session	Continuous
Is_hot_login	1 if the login belongs to the "hot" list; 0 otherwise	Discrete
Is_guest_login	1 if the login belongs is a "guest" login; 0 otherwise	Discrete

Table 1: List of features (KDD-Cup99 task description) (cont.')

Count	Number of connections to the same host as the current connection in the past two seconds	Continuous
Serror_rate	% of connections that have "SYN" errors	Continuous
Rerror_rate	% of connections that have "REJ" errors	Continuous
Same_srv_rate	% of connections to the same service	Continuous
Diff_srv_rate	% of connections to different services	Continuous
Srv_count	Number of connections to the same service as the current connection in the past two seconds	Continuous
Srv_serror_rate	% of connections that have "SYN" errors	Continuous
Srv_rerror_rate	% of connections that have "REJ" errors	Continuous
Srv_diff_host_rate	% of connections to different hosts	Continuous

3.2 Performance Measures

The result of classification is typically presented in matrix called confusion matrix. The accuracy of a classifier is measured by its misclassification rate, or alternatively, the percentage of correct classification.

Table 2: Confusion metrics for evaluations of attack

Type	**Predicted Connection**	
	Attack	**Normal**
Attack	Correctly detected (TP)	False negative (FN)
Normal	False alarm (FP)	True negative (TN)

Two other performance measures, precision and recall are also of interest [9].

$$\text{Precision} = \frac{\text{TP}}{(\text{TP}+\text{FP})}$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP}+\text{FN})}$$

Another valuable tool for evaluating an anomaly detection scheme is the receiver operating characteristic (ROC) curve, which is the plot of the detection rate against the false

alarm rate. The nearer the ROC curve of a scheme is to the upper-left corner, the better the performance of the scheme is.

4 Experimental Results and Discussion

In KDD'99 training data, there are 24 attack types that fall into 4 big categories: Does – denial-of-service, Probe – surveillance and other probing, U2R – unauthorized access to local super user (root) privileges, and R2L – unauthorized access from a remote machine. The task was to predict the value of each connection (normal or one of the above attack categories) for each of the connection record of the test dataset. It is important to test data is not from the same probability distribution as the training data and the test data includes some specific attack types not in the training data.

5 Conclusions

In this paper, anomaly network intrusion detection method based on PCA is proposed. By using the proposed method, the huge dimensional data can be greatly reduced by projecting them onto a lower dimensional subspace for intrusion detection so that the complexity of the detecting algorithm is significantly reduced. We have presented PCA to detect anomalies with less data. This model is now at implement stage of development. Our long term goal is to implement in a real time environment and compare with KDDCup1999 data set. More results could be obtained once we finish implementing the system.

References

[1] Y. Boozed and S. Gimbals, "Intrusion Detection Using Principal Component Analysis," in *Proceedings of the 7th World Multiconference on Systemics, Cybermetrics and Informatics*. Orlando, Florida, 2003.

[2] W. Wang and R. Battiti, "Identifying Intrusions in Computer Networks based on Principal Component Analysis," 2005.

[3] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, "A Novel Anomaly Detection Scheme Based on Principal Component Classifier," presented at Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with Third IEEE International Conference on Data Mining (ICDM'03), Melbourne, Florida, USA, 2003.

[4] S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour, and Y. Nemoto, "A Self-adaptive Intrusion Detection Method for AODV-based Mobile Ad Hoc Networks," presented at Mobile Adhoc and Sensor Systems Conference, 2005, IEEE International Conference on, 2005.

[5] J. I.T., *Principal Component Analysis*. New York: Springer-Verlag, 2002.

[6] D. R.O., H. P.E., and S. D.G., *Pattern Classification*. Beijing, 2004.

[7] F. Korn, H. V. Jagadish, and C. Faloutsos, "Efficiently supporting ad hoc queries in large datasets of time sequences.," presented at ACM-SIGMOD International Conference on Management of Data, Tucson, 1997.

[8] KDD, "The 3rd international knowledge discovery and data mining tools competition (KDDCup1999)." California, Irvine: University of California :http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 2005.

[9] Y. Yang, *An Evaluation of Statistical Approaches to Text Categorization.* Netherlands, 1999.

Appendix A: Code

This code is written for Matlab by examine the covariance of the data set.

```
function [signals,PC,V] = pca1(data)
%   PCA1: Perform PCA using covariance.
%   data - MxN matrix of input data
%          (M dimensions, N trials)
%   signals - MxN matrix of projected data
%   PC - each column is a PC
%   V - Mx1 matrix of variances

[M,N] = size(data);

%------------------------------   subtract off the mean for each dimension
mn = mean(data,2);
data = data - repmat(mn,1,N);

% ------------------------------  calculate the covariance matrix
covariance = 1/(N-1) * data * data';

%------------------------------ finds the eigenvectors and eigenvalues
[PC,V] = eig(covariance);

% ------------------------------ extract diagonal of matrix as vector
V=diag(V);

% ------------------------------ sort the variances in decreasing order
[junk, rindices] = sort(-1*V);
V=V(rindices);
PC=PC(:,rindices);

% ------------------------------ project the original data set
signals = PC' * data;
```

This second version computing PCA through SVD.

```
function [signals,PC,V] = pca2(data)
%   PCA1: Perform PCA using SVD.
%   data - MxN matrix of input data
%          (M dimensions, N trials)
%   signals - MxN matrix of projected data
%   PC - each column is a PC
%   V - Mx1 matrix of variances

[M,N] = size(data);

% ------------------------------------- subtract off the mean for each dimension
mn = mean(data,2);
data = data - repmat(mn,1,N);

% ------------------------------------- Construct the matrix Y
Y = data' / sqrt(N-1)

% ------------------------------------   SVD does it all
[u,S,PC] = svd(Y);

%------------------------------------    calculate the variances
S = diag(S);
V = S .* S;

% ------------------------------------- project the original data set
signals = PC' * data;
```