Intrusion Detection System Using PCA and Kernel PCA Methods

Zyad Elkhadir, Khalid Chougdali and Mohammed Benattou

Abstract The network traffic data used to build an intrusion detection system is frequently enormous with important useless information which decreases IDS's efficiency. In order to overcome this problem, we have to reduce as much as possible this meaningless information from the original high dimensional data. In this paper, we compare the performance of two features reduction techniques, the first one is Principal Component Analysis (PCA), and the second is Kernel Principal Component Analysis (KPCA). After the step of dimension reduction, samples are classified using k nearest neighbor (K-NN) algorithm in order to determine whether the test data are normal or anomalous corresponding to attacks against computer networks. Experimental results on KDDcup99 intrusion detection dataset shows that KPCA with the quadratic kernel performs better than many other types of kernels. Furthermore, KPCA outperforms PCA mainly when we vary the number of nearest neighbors from one to four. Finally, we have also noted that KPCA with the quadratic kernel overcomes PCA in detecting denial of service (DOS) and probing attacks.

Keywords Network security · Intrusion detection system (IDS) · PCA · KPCA

1 Introduction

The security of a computer network is compromised when an intrusion takes place. An Intrusion Detection System (IDS) is an important mechanism that attempts to identify any set of actions or malicious activities which can compromise network security policy.

Z. Elkhadir · M. Benattou
RLCST Research Laboratory, Ibn Tofail University, Kenitra, Morocco

K. Chougdali (✉)
National School of Applied Sciences (ENSA), Ibn Tofail University, Kenitra, Morocco
e-mail: chougdali@gmail.com

A. El Oualkadi et al. (eds.), *Proceedings of the Mediterranean Conference on Information & Communication Technologies 2015*, Lecture Notes in Electrical Engineering 381, DOI 10.1007/978-3-319-30298-0_50

Practically, there are two principal intrusion detection techniques: misuse detection and anomaly detection. Misuse detection recognizes a suspicious behavior by comparing it to a specific attack signature that has been already stored in a database of attacks signatures; unfortunately it can't detect new attacks. Examples of IDS using misuse detection techniques are STAT [1] and Snort [2]. On the other side, anomaly detection defines normal behavior as a model, and tries to check any deviation from the model and thus decides to generate or not the corresponding alert. Anomaly detection was originally introduced by Anderson [3] and Denning [4] and then implemented in some IDS like IDES [5] or EMERALD [6].

Many methods have been developed for anomaly based IDS, such as machine learning, data mining, neural networks and statistical methods. All of them have been applied directly on the rough high dimensional data without any dimension reduction technique. It can be considered as one of the principal factors contributing in IDS inefficiency.

The main idea behind our proposed work is to reduce original features of database connection records by extracting its relevant information. A simple technique to extract the information contained in network connection records is to capture the variance in a collection of connection records. The given information is used to classify these network connection records as normal or attack connection.

Mathematically speaking, we want to find the principal components of the connection records of a dataset. It can be seen as binary TCP/IP network connections records corresponding to a normal connection or to a specified attack. To do this, the approach extracts the relevant information using the eigenvectors of the covariance matrix of all connection records [7]. These eigenvectors can be defined as a set of features used to reduce the variation between record connections. Indeed, each connection is expressed using only the eigenvectors with the largest eigenvalues given by the most variance within the set of connection records. The new space generated is constructed using the Principal Component Analysis (PCA) [8] which has proven to be efficient in intrusion detection [9, 10] and in many application domains including data compression, image analysis, visualization and pattern recognition.

In this paper, we have used two reduction techniques PCA and KPCA (Kernel Principal Component Analysis) which extracts principal components by adopting a non-linear kernel method [11]. The simulation results show that a KPCA with quadratic kernel gives a higher detection rate than PCA when the number of nearest neighbors varies between one and four, particularly for detecting DOS and PROBE attacks.

This paper is organized as follows: Sect. 2 is dedicated to presents briefly the two dimensionality reduction methods PCA and KPCA. Section 3 describes and discusses the obtained results, and Sect. 4 gives the concluding remarks and outlines our future works.

2 PCA and Kernel PCA

In this section, we present a modeling concepts and theoretical analysis of PCA and KPCA.

2.1 PCA

Principal component analysis (PCA) is a mathematical technique that transforms a number of correlated variables into a number of uncorrelated variables called principal components (PCs). Generally, the number of these principal components is less than or equal to the number of original variables. The main goal of principal component analysis is to reduce dimensionality (number of variables) of the initial dataset, while retaining as much as possible the variance present in this dataset. This is achieved by taking only the first few PCs, sorted in decreasing order, so that they retain most of the variance present in all of the original variables [8].

Suppose we have a training set of M vectors $\omega_1, \omega_2, \ldots, \omega_M$ each vector contain n features. To $n'(n' \ll n)$ get principal components of the training set the procedure is based on the following steps:

1. Compute the average σ of this set:

$$\sigma = \left(\frac{1}{M}\right) \sum_{i=1}^{M} \omega_i. \tag{1}$$

2. Subtract the mean σ from ω_i and get ρ_i:

$$\rho_i = \omega_i - \sigma. \tag{2}$$

3. Compute the covariance matrix C where:

$$C_{(n\times n)} = \left(\frac{1}{M}\right) \sum_{i=1}^{M} \rho_i \, \rho_i^T = AA^T. \tag{3}$$

$$A_{(n\times M)} = \left(\frac{1}{\sqrt{M}}\right) \rho_i. \tag{4}$$

4. Let U_k be the kth eigenvector of C corresponding to the λ_k associated eigenvalue and $U_{(n\times n')} = [U_1 \ldots U_{n'}]$ the matrix of these eigenvectors, so we have:

$$CU_k = \lambda_k U_k. \tag{5}$$

5. Sort the eigenvalues (and the corresponding eigenvectors) in decreasing order and choose the first eigenvectors, those eigenvectors are principal components (PC_i). Practically, the number of the principal components chosen depends on the precision we wish to have. The inertia ratio defined by these axes is given by:

$$\tau = \frac{\sum_{i=1}^{n'} \lambda_i}{\sum_{i=1}^{n} \lambda_i}. \tag{6}$$

This ratio defines the information rate kept, from the whole rough input data, by the corresponding n eigenvalues. Finally, the projection of a new column vector sample x_{new} on the space constructed by principal components can be obtained by:

$$t_i = PC_i^T x_{new}. \tag{7}$$

2.2 Kernel PCA

PCA allows only a linear dimensionality reduction. However, if the data has more complicated structures, which cannot be well represented in a linear subspace, standard PCA will not be very helpful. Fortunately, kernel PCA (KPCA) allows us to generalize PCA to nonlinear dimensionality reduction. This can be done by a nonlinear mapping function φ, that transform all samples input into a higher-dimensional feature space F as follows:

$$\varphi : \omega \in R^n \rightarrow \varphi(\omega_i) \in F$$

where $\varphi(\omega_i)$ is sample of F and $\sum_{i=1}^{M} \varphi(\omega_i) = 0$. The mapping of ω_i is simply noted as $\varphi(\omega_i) = \varphi_i$ and the covariance matrix of this sample in the feature space F can be constructed by:

$$C = \left(\frac{1}{M}\right) \sum_{i=1}^{M} (\varphi_i - mean)(\varphi_i - mean)^T \tag{8}$$

where $mean = \sum_{i=1}^{M} (\frac{\varphi_i}{M})$. The covariance matrix C can be diagonalized with nonnegative eigenvalues λ satisfying:

$$Cv = \lambda_i v. \tag{9}$$

It's easy to see that every eigenvector v of C can be linearly expanded by:

$$v = \sum_{i=1}^{M} (\alpha_i \varphi_i). \tag{10}$$

To obtain the coefficients α_i, a kernel matrix K with size $M \times M$ is defined and its elements are determined as follows:

$$K_{ij} = \varphi_i^T \varphi_j = \varphi_i \cdot \varphi_j = k(\omega_i, \omega_j) \tag{11}$$

where $k(\omega_i, \omega_j) = <\varphi_i, \varphi_j>$ is the inner-product of two vectors in F. If the projected dataset $\{\varphi(\omega_i)\}$ does not have zero mean, we can use the Gram matrix K' to substitute the kernel matrix K using:

$$K' = K - 1_M K - K 1_M + 1_M K 1_M \tag{12}$$

with, $1_M = (1/M)_{M \times M}$.

In order to solve the eigenvalue problem in (9), we can reformulate this equation as [11]:

$$K'\alpha = M\lambda\alpha. \tag{13}$$

Let column vectors α_i be the orthonormal eigenvectors of K' corresponding to the p largest positive eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$, hence the orthonormal eigenvectors v_i of C can be expressed as:

$$v_i = \left(\frac{1}{\sqrt{\lambda_i}} \varphi_i \alpha_i\right). \tag{14}$$

For a new column vector sample x_{new}, the mapping to the feature space F is $\varphi(x_{new})$ and then the projection of x_{new} onto eigenvectors v_i is:

$$t = (v_1, v_2, \ldots, v_p)^T \varphi(x_{new}). \tag{15}$$

The ith KPCA transformed feature t_i can be obtained by:

$$t_i = v_i^T \varphi(x_{new}) = \left(\frac{1}{\sqrt{\lambda_i}}\right) \alpha_i^T k(\omega_i, x_{new}). \tag{16}$$

It can be noted that the kernel matrix can directly constructed from the training dataset. The commonly used kernels are:

- Gaussian kernel

$$k(x,y) = e^{\left(-\frac{\|x-y\|^2}{2\times sigma^2}\right)} \tag{17}$$

- Polynomial kernel

$$k(x,y) = (x^T y + 1)^d \quad where\ d \in \mathrm{N} \tag{18}$$

- Power kernel

$$k(x,y) = \|x-y\|^d \quad where\ d \geq 1 \tag{19}$$

- Rational Power kernel

$$k(x,y) = \|x-y\|^d \quad where\ 0 < d < 1. \tag{20}$$

3 Experiments and Discussion

This section is dedicated to present and evaluate the results obtained when applying the two dimensionality reduction methods PCA and KPCA with K-NN. In our simulation experiments, we have chosen from the 10 % of KDDcup99 [12] training dataset, a subset composed of 1000 normal data, 100 DOS data, 50 U2R data, 100 R2L data, and 100 PROBE data. Likewise, the test dataset is composed of 100 normal data, 100 DOS data, 50 U2R data, and 100 R2L data, all randomly selected. To evaluate the performance of the proposed system we have used detection rate (DR) defined as follows:

$$DR = \left(\frac{TP}{TP+FN}\right) * 100 \tag{21}$$

where TP are true positives (intrusions correctly classified), FN are false negatives (intrusions wrongly classified), FP are false positive (normal instances wrongly classified), and TN are true negatives (normal instances correctly classified).

After applying PCA on training dataset, we obtained principal components (PC) then we have computed the projection of the test dataset on different number of (PC).

Figure 1 shows that, only the first three principal components give a highest detection rate (%) with inertia ratio $\tau > 0.99$ (Eq. 6).

After that we have fixed the number of PCs at 3 and tried to find the optimal detection rate for every type of attacks (DOS, U2R, R2L, and PROBE). Table 1 shows this manipulation. It's clear that, the two categories of attacks DOS and

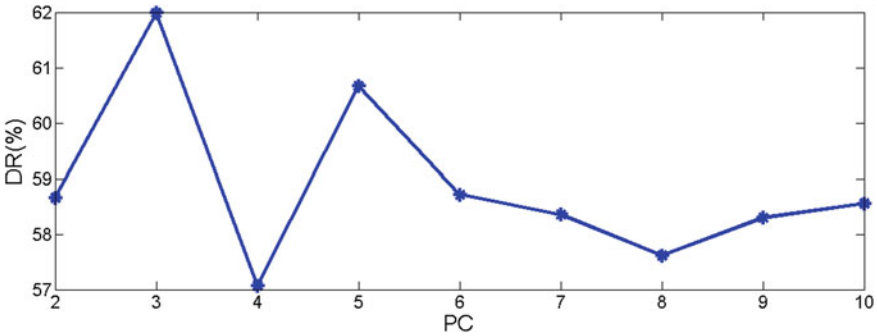

Fig. 1 Detection rate (%) versus number of principal components (PC)

Table 1 Attack's detection rate for PCA

DOS	U2R	R2L	PROBE
95,13 (%)	8,13 (%)	3,5 (%)	69,8 (%)

PROBE are detected with a rate of 95,13 % for DOS and 69,8 % for PROBE. In the other hand, U2R and R2L are not well detected with only 8,13 % for U2R and 3,5 % for R2L.

In the second part of our experiments, we have evaluated the effectiveness of Kernel PCA in intrusion detection system. Firstly, we implement four kernels, described by Eqs. (17)–(20). Secondly, we try to pick up the maximum detection rate varying the different kernels parameters. Indeed, to achieve this goal we set the number of principal components at three, and then we have changed kernel's parameters.

A next experience seeks to identify the best kernel for KPCA. For this reason, in Fig. 2 we have compared detection rates obtained by using the four kernels with the adequate parameters. We can observe that the power kernel gives higher detection

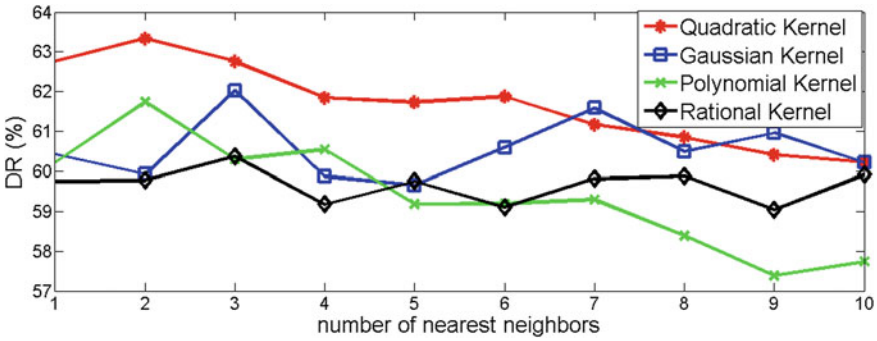

Fig. 2 Performance comparison of different kernels

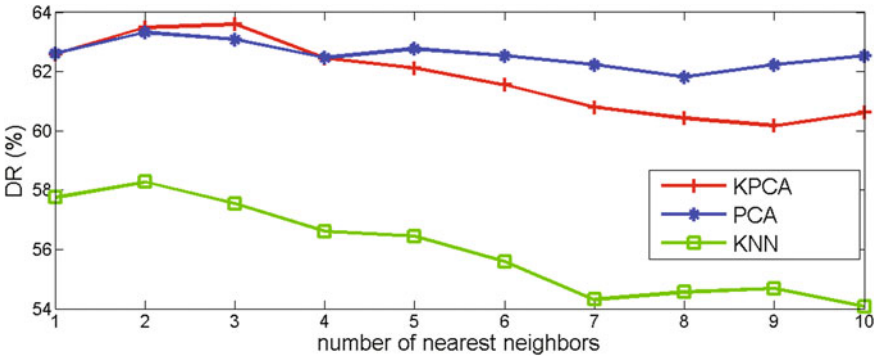

Fig. 3 Detection rate (%) of KPCA, PCA and KNN versus number of nearest neighbors

Table 2 Attack's detection rate for KPCA

DOS	U2R	R2L	PROBE
96,13 (%)	9.93 (%)	3,46 (%)	73,8 (%)

rate than the others kernels especially when K nearest neighbors is less than seven. Hereafter, we will call the power kernel with d = 2 as a quadratic kernel.

The results of comparison described in Fig. 3 of PCA and KPCA with quadratic kernel, show that KPCA outperforms PCA when number of nearest neighbors is between 1 and 4.

Finally, we visualize detection rate for every type of attacks using kernel PCA in Table 2. Indeed, the two categories of attacks DOS and PROBE are well detected with a rate of 96.13 % for DOS and 73.8 % for PROBE attacks. Furthermore, we can conclude that these detection rates are better than those founded with PCA (95.13 % and 69.8 %). In the other hand, U2R and R2L attacks are not well detected with only 9.93 % for U2R (slightly better than PCA which gives 8.13 %) and just 3.46 % for R2L.

4 Conclusion

The main idea behind the work presented in this paper is to reduce the original features that represent all connection records stored in a dataset for the purpose of intrusion detection. The proposed work shows, how we can extract relevant information using PCA and KPCA in order to build a robust network IDS with the maximum detection rate. The experimental results show that a KPCA with quadratic kernel gives a higher detection rate than PCA when the number of nearest neighbors varies between one and four, particularly for detecting DOS and PROBE

attacks. Our future works will be oriented towards advanced dimension reduction techniques in order to improve the performance of an IDS particularly for the CPU time consuming.

References

1. Kumar, S., Spafford, E.H.: A Software architecture to support misuse intrusion detection. In: Proceedings of the 18th National Information Security Conference, pp. 194–204 (1995)
2. Beale, J.: Snort 2.1 Intrusion Detection. Syngress (2004)
3. Anderson, J. P.: Computer Security Threat Monitoring and Surveillance. Technical report, James. P. Anderson Co., Fort Washington, Pennsylvania (1980)
4. Denning, D.: An intrusion detection model. IEEE Trans. Software Eng. **13**(2), 222–232 (1987)
5. Lunt, T., Tamaru, A., Gilham, F.: A Real-time Intrusion Detection Expert System (IDES) Final Technical Report. Computer Science Laboratory. SRI International, Menlo Park, California (1992)
6. Porras, P.A., Neumann, P.G., EMERALD: Event monitoring enabling responses to anomalous live disturbances. In: Proceedings of National Information Systems Security Conference, Baltimore MD (1997)
7. Jolliffe, I.T.: Principal Component Analysis. Springer, New York, NY (2002)
8. Kirby, M., Sirovich, L.: Application of the Karhunen Loeve procedure for the characterization of human faces. IEEE Trans. Pattern Anal. Mach. Intell. 103–107 (1990)
9. Bouzida, Y., Cuppens, F., Cuppens-Boulahia, N., Gombault, S.: Efficient intrusion detection using principal component analysis. In: 3éme Conférence sur la Sécurité et Architectures Réseaux (SAR), La Londe, France (2004)
10. Hashem, S.H.: Efficiency of Svm and Pca to enhance intrusion detection system. J. Asian Sci. Res. 381–395 (2013)
11. Scholkopf, B., Smola, A., Muller., K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. 1299–1319 (1998)
12. KDD 99 Task. Available at: http://kdd.ics.uci.edu/databases/kddcup99/task.html (1999)