Kernel Based Intrusion Detection System

Byung-joo Kim

Dept. of Info. and Comm.
Youngsan University
Yangsan, Korea
bjkim@ysu.ac.kr

Il-kon Kim

Dept. of Computer Science
Kyungpook National University
Daegu, Korea
ikkim@knu.ac.kr

Abstract

Recently applying artificial intelligence, machine learning and data mining techniques to intrusion detection system are increasing. But most of researches are focused on improving the performance of classifier. Selecting important features from input data lead to a simplification of the problem, faster and more accurate detection rates. Thus selecting important features is an important issue in intrusion detection. Another issue in intrusion detection is that most of the intrusion detection systems are performed by off-line and it is not proper method for realtime intrusion detection system. In this paper, we develop the realtime intrusion detection system which combining on-line feature extraction method with Least Squares Support Vector Machine classifier. Applying proposed system to KDD CUP 99 data, experimental results show that it have remarkable performance compared to off-line intrusion detection system.

1. Introduction

Computer security has become a critical issue with the rapid development of business and other transaction systems over the internet. Intrusion detection is to detect intrusive activities while they are acting on computer network systems. Most intrusion detection systems(IDSs) are based on hand-crafted signatures that are developed by manual coding of expert knowledge. These systems match activity on the system being monitored to known signatures of attack. The major problem with this approach is that these IDSs fail to generalize to detect new attacks or attacks without known signatures. Recently, there has been an increased interest in data mining based approaches to building detection models for IDSs. These models generalize from both known attacks and normal behavior in order to detect unknown attacks. They can also be generated in a quicker and more automated method than manually encoded models that require difficult analysis of audit data by domain experts. Several effective data mining techniques for detecting intrusions have been developed[1][2][3], many of which perform close to or better than systems engineered by domain experts. However, successful data mining techniques are themselves not enough to create effective IDSs. Despite the promise of better detection performance and generalization ability of data mining based IDSs, there are some difficulties in the implementation of the system. We can group these difficulties into three general categories: accuracy(i.e., detection performance), efficiency, and usability. In this paper, we discuss accuracy problem in developing a real-time two-tier based IDS. Another issue in IDS is that it should operate in real-time. In typical applications of data mining to intrusion detection, detection models are produced off-line because the learning algorithms must process tremendous amounts of archived audit data. These models can naturally be used for off-line intrusion detection. Effective IDS should work in real-time, as intrusions take place, to minimize security compromises. Elimination of the insignificant and/or useless inputs leads to a simplification of the problem, faster and more accurate detection result. Feature selection therefore, is an important issue in intrusion detection. In this paper we present an on-line feature extraction method form audit data which help discriminate attacks form normal data. These features can then be used by any of classification algorithm. Principal Component Analysis(PCA)[4] is a powerful technique for extracting features from data sets. For reviews of the existing literature is described in [5][6][7]. Traditional PCA, however, has several problems. First PCA requires a batch computation

[0]This study was supported by a grant of the Korea Health 21 R&D Project, Ministry of Health & Welfare, Republic of Korea (02-PJ1-PG6-HI03-0004)

step and it causes a serious problem when the data set is large i.e., the PCA computation becomes very expensive. Second problem is that, in order to update the subspace of eigenvectors with another data, we have to recompute the whole eigenspace. Finial problem is that PCA only defines a linear projection of the data, the scope of its application is necessarily somewhat limited. It has been shown that most of the data in the real world are inherently non-symmetric and therefore contain higher-order correlation information that could be useful[8]. PCA is incapable of representing such data. For such cases, nonlinear transforms is necessary. Recently kernel trick has been applied to PCA and is based on a formulation of PCA in terms of the dot product matrix instead of the covariance matrix[9]. Kernel PCA(KPCA), however, requires storing and finding the eigenvectors of a $N \times N$ kernel matrix where N is a number of patterns. It is infeasible method when N is large. This fact has motivated the development of on-line way of KPCA method which does not store the kernel matrix. It is hoped that the distribution of the extracted features in the feature space has a simple distribution so that a classifier could do a proper task. But it is point out that extracted features by KPCA are global features for all input data and thus may not be optimal for discriminating one class from others[9]. In order to solve this problem, we developed the two-tier intrusion detection system. Proposed real time IDS is composed of two parts. First part is used for on-line feature extraction. To extract on-line nonlinear features, we propose a new feature extraction method which overcomes the problem of memory requirement of KPCA by on-line eigenspace update method incorporating with an adaptation of kernel function. Second part is used for classification. Extracted features are used as input for classification. We take Least Squares Support Vector Machines(LS-SVM)[10] as a classifier. LS-SVM is reformulations to the standard Support Vector Machines(SVM)[11]. SVM typically solving problems by quadratic programming(QP). Solving QP problem requires complicated computational effort and needs more memory requirement. LS-SVM overcomes this problem by solving a set of linear equations in the problem formulation. Paper is composed of as follows. In Section 2 we will briefly explain the on-line feature extraction method. In Section 3 KPCA is introduced and to make KPCA on-line, empirical kernel map method is is explained. Proposed classifier combining LS-SVM with proposed feature extraction method is described in Section 4. Experimental results to evaluate the performance of proposed system is shown in Section 5. Discussion of proposed IDS and future work is described in Section 6.

2. On-line Feature Extraction

In this section, we will give a brief introduction to the method of on-line PCA algorithm which overcomes the computational complexity and memory requirement of standard PCA. Before continuing, a note on notation is in order. Vectors are columns, and the size of a vector, or matrix, where it is important, is denoted with subscripts. Particular column vectors within a matrix are denoted with a superscript, while a superscript on a vector denotes a particular observation from a set of observations, so we treat observations as column vectors of a matrix. As an example, A^i_{mn} is the ith column vector in an $m \times n$ matrix. We denote a column extension to a matrix using square brackets. Thus $[A_{mn}b]$ is an $(m \times (n+1))$ matrix, with vector b appended to A_{mn} as a last column. To explain the on-line PCA, we assume that we have already built a set of eigenvectors $U = [u_j], j = 1, \cdots, k$ after having trained the input data $\mathbf{x}_i, i = 1, \cdots, N$. The corresponding eigenvalues are Λ and $\bar{\mathbf{x}}$ is the mean of input vector. On-line building of eigenspace requires to update these eigenspace to take into account of a new input data. Here we give a brief summarization of the method which is described in [12]. First, we update the mean:

$$\overline{x}' = \frac{1}{N+1}(N\overline{x} + x_{N+1}) \tag{1}$$

We then update the set of eigenvectors to reflect the new input vector and to apply a rotational transformation to U. For doing this, it is necessary to compute the orthogonal residual vector $\hat{h} = (Ua_{N+1} + \overline{x}) - x_{N+1}$ and normalize it to obtain $h_{N+1} = \frac{h_{N+1}}{\|h_{N+1}\|_2}$ for $\| h_{N+1} \|_2 > 0$ and $h_{N+1} = 0$ otherwise. We obtain the new matrix of Eigenvectors U' by appending h_{N+1} to the eigenvectors U and rotating them :

$$U' = [U, h_{N+1}]R \tag{2}$$

where $R \in \mathbf{R}_{(\mathbf{k+1})\times(\mathbf{k+1})}$ is a rotation matrix. R is the solution of the eigenproblem of the following form:

$$DR = R\Lambda' \tag{3}$$

where Λ' is a diagonal matrix of new Eigenvalues. We compose $D \in \mathbf{R}_{(\mathbf{k+1})\times(\mathbf{k+1})}$ as:

$$D = \frac{N}{N+1}\begin{bmatrix} \Lambda & 0 \\ 0^T & 0 \end{bmatrix} + \frac{N}{(N+1)^2}\begin{bmatrix} aa^T & \gamma a \\ \gamma a^T & \gamma^2 \end{bmatrix} \tag{4}$$

where $\gamma = h_{N+1}^T(x_{N+1} - \bar{x})$ and $a = U^T(x_{N+1} - \bar{x})$. Though there are other ways to construct matrix D[13][14], the only method ,however, described in [12] allows for the updating of mean.

3. Eigenspace Updating Rule

The on-line PCA represents the input data with principal components $a_{i(N)}$ and it can be approximated as follows:

$$\widehat{x}_{i(N)} = U a_{i(N)} + \bar{x} \tag{5}$$

To update the principal components $a_{i(N)}$ for a new input x_{N+1} , computing an auxiliary vector η is necessary. η is calculated as follows:

$$\eta = \left[U\widehat{h}_{N+1}\right]^T (\overline{x} - \overline{x}') \tag{6}$$

then the computation of all principal components is

$$a_{i(N+1)} = (R')^T \begin{bmatrix} a_{i(N)} \\ 0 \end{bmatrix} + \eta, \quad i = 1, \cdots, N+1 \tag{7}$$

The above transformation produces a representation with $k+1$ dimensions. Due to the increase of the dimensionality by one, however, more storage is required to represent the data. If we try to keep a k-dimensional eigenspace, we lose a certain amount of information. It is needed for us to set the criterion on retaining the number of eigenvectors. There is no explicit guideline for retaining a number of eigenvectors. Here we introduce some general criteria to deal with the model's dimensionality:

- Adding a new vector whenever the size of the residual vector exceeds an absolute threshold;
- Adding a new vector when the percentage of energy carried by the last eigenvalue in the total energy of the system exceeds an absolute threshold, or equivalently, defining a percentage of the total energy of the system that will be kept in each update;
- Discarding Eigenvectors whose Eigenvalues are smaller than a percentage of the first Eigenvalue;
- Keeping the dimensionality constant.

In this paper we take a rule described in second. We set our criterion on adding an Eigenvector as $\lambda'_{k+1} > 0.7\bar{\lambda}$ where $\bar{\lambda}$ is a mean of the λ. Based on this rule, we decide whether adding u'_{k+1} or not.

4. On-line KPCA

A prerequisite of the on-line eigenspace update method is that it has to be applied on the data set. Furthermore on-line PCA builds the subspace of eigenvectors on-line, it is restricted to apply the linear data. But in the case of KPCA this data set $\Phi(x^N)$ is high dimensional and can most of the time not even be calculated explicitly. For the case of nonlinear data set, applying feature mapping function method to on-line PCA may be one of the solutions. This is performed by so-called *kernel-trick*, which means an implicit embedding to an infinite dimensional Hilbert space[11](i.e. feature space) F.

$$K(x, y) = \Phi(x) \cdot \Phi(y) \tag{8}$$

Where K is a given kernel function in an input space. When K is semi positive definite, the existence of Φ is proven[8]. Most of the case ,however, the mapping Φ is high-dimensional and cannot be obtained explicitly. The vector in the feature space is not observable and only the inner product between vectors can be observed via a kernel function. However, for a given data set, it is possible to approximate Φ by empirical kernel map proposed by Scholkopf[15] and Tsuda[16] which is defined as $\Psi_N : \mathbf{R^d} \rightarrow \mathbf{R^N}$

$$\begin{aligned} \Psi_N(x) &= [\Phi(x_1) \cdot \Phi(x), \cdots, \Phi(x_N) \cdot \Phi(x)]^T \\ &= [K(x_1, x), \cdots, K(x_N, x)]^T \end{aligned} \tag{9}$$

A performance evaluation of empirical kernel map was shown by Tsuda. He shows that support vector machine with an empirical kernel map is identical with the conventional kernel map[11]. The empirical kernel map $\Psi_N(x_N)$,however, do not form an orthonormal basis in $\mathbf{R^N}$, the dot product in this space is not the ordinary dot product. In the case of KPCA ,however, we can be ignored as the following argument. The idea is that we have to perform linear PCA on the $\Psi_N(x_N)$ from the empirical kernel map and thus diagonalize its covariance matrix. Let the $N \times N$ matrix $\Psi = [\Psi_N(x_1), \Psi_N(x_2), \ldots, \Psi_N(x_N)]$, then from equation (9) and definition of the kernel matrix we can construct $\Psi = NK$. The covariance matrix of the empirically mapped data is:

$$C_\Psi = \frac{1}{N}\Psi\Psi^T = NKK^T = NK^2 \tag{10}$$

In case of empirical kernel map, we diagonalize NK^2 instead of K as in KPCA. Mika shows that the two matrices have the same eigenvectors $\{u_k\}$[17]. The eigenvalues $\{\lambda_k\}$ of K are related to the eigenvalues $\{k_k\}$ of NK^2 by

$$\lambda_k = \sqrt{\frac{k_k}{N}} \tag{11}$$

and as before we can normalize the eigenvectors $\{v_k\}$ for the covariance matrix C of the data by dividing each $\{u_k\}$ by $\sqrt{\lambda_k N}$. Instead of actually diagonalize the covariance matrix C_Ψ, the IKPCA is applied directly on the mapped data $\Psi = NK$. This makes it easy for us to adapt the on-line eigenspace update method to KPCA such that it is also correctly takes into account the centering of the mapped data in an on-line way. By this result, we only need to apply the empirical map to one data point at a time and do not need to store the $N \times N$ kernel matrix.

5. Proposed System

In earlier Section 3 we proposed an on-line KPCA method for nonlinear feature extraction. Feature extraction by on-line KPCA effectively acts a nonlinear mapping from the input space to an implicit high dimensional feature space. It is hoped that the distribution of the mapped data in the feature space has a simple distribution so that a classifier can classify them properly. But it is point out that extracted features by KPCA are global features for all input data and thus may not be optimal for discriminating one class from others. For classification purpose, after global features are extracted using they must be used as input data for classification. There are many famous classifier in machine learning field. Among them neural network is popular method for classification and prediction purpose. Traditional neural network approaches, however have suffered difficulties with generalization, producing models that can overfit the data. To overcome the problem of classical neural network technique, support vector machines(SVM) have been introduced. The foundations of SVM have been developed by Vapnik and it is a powerful methodology for solving problems in nonlinear classification. Originally, it has been introduced within the context of statistical learning theory and structural risk minimization. In the methods one solves convex optimization problems, typically by quadratic programming(QP). Solving QP problem requires complicated computational effort and need more memory requirement. LS-SVM overcomes this problem by solving a set of linear equations in the problem formulation. LS-SVM method is computationally attractive and easier to extend than SVM.

6. Experiment

To evaluate the performance of proposed realtime IDS system, we use KDD CUP 99 data[17]. The raw training data(kddcup.data.gz) was about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic. This was processed into about five million connection records. Similarly, the two weeks of test data yielded around two million connection records. 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. Attacks fall into four main categories:

- DOS: denial-of-service, e.g. syn flood;
- R2L: unauthorized access from a remote machine, e.g. guessing password;
- U2R: unauthorized access to local superuser (root) privileges, e.g., various "buffer overflow" attacks;
- Probing: surveillance and other probing, e.g., port scanning.

It is important to note that the test data(corrected.gz) is not from the same probability distribution as the training data, and it includes specific attack types not in the training data. This makes the task more realistic. The datasets contain a total of 24 training attack types, with an additional 14 types in the test data only. We randomly split the the training data as 80% and remaining as validation data. A RBF kernel has been taken and optimal hyperparameter of LS-SVM was obtained by 10-fold cross-validation procedure. In experiment we will evaluate the generalization ability of proposed IDS on test data set since there are 14 additional attack types in the test data which are not included int the training set. To do this, extracted features by on-line KPCA will be used as input for LS-SVM. Our results are summarized in the following tables. Table 1 gives the result of extracted features for each class by on-line KPCA method.

Table 2 shows the results of the classification performance and computing time for training and testing data by proposed system using all features. Table 3 shows the results of the classification performance and computing time for training and testing data by proposed system using extracted features. Comparing Table 2 with Table 3, we obtain following results.

Table 1: **Extracted features on each class by on-line KPCA.**

Class	Extracted features
Normal	1,2,3,5,6,7,8,9,10,11,12,13,14,16,17,18, 20,21,22,23,25,27,29,30,31,32,34,37,38,39,41
Probe	3,5,6,23,24,32,33,38
DOS	1,3,6,8,19,23,28,32,33,35,36,38,39,41
U2R	5,6,15,16,18,25,32,33,38,39
R2L	3,5,6,24,32,33,34,35,38

Table 2: **Performance of proposed system using all features.**

Class	Accuracy(%)	Train(Sec)	Test(Sec)
Normal	98.55	5.83	1.45
Probe	98.59	28.0	1.96
DOS	98.10	16.62	1.74
U2R	98.64	2.7	1.34
R2L	98.69	7.8	1.27

- The performance of using the extracted features do not show the significant differences to that of using all features. This means that proposed on-line feature extraction method has good performance in extracting features.

- Using the important features for each class gives remarkable performance . The accuracy increase slightly for one class 'Normal', decreases slightly for three class 'Probe' , 'DOS' and 'R2L', and remains the same for the one class' U2R'.

- Using extracted features decreases the training and testing time. This makes proposed IDS suitable for realtime IDS.

Table 3: **Performance of proposed system using extracted features.**

Class	Accuracy(%)	Train(Sec)	Test(Sec)
Normal	98.43	5.25	1.42
Probe	98.63	25.52	1.55
DOS	98.14	15.92	1.48
U2R	98.64	2.17	1.32
R2L	98.70	7.2	1.08

7. Conclusion and Remarks

In this paper, we present the realtime two-tier based intrusion detection system. Applying artificial intelligence, machine learning and data mining techniques to intrusion detection system are increasing. But most of researches are focused on improving the performance of classifier. These classifiers are performed by batch way and it is not proper method for realtime IDS. Applying proposed system to KDD CUP 99 data, experimental result shows that it has remarkable performance in detection rate and computation time compared to off-line IDS. Our ongoing experiment is that applying proposed system to more realistic world data to evaluate the realtime detection performance.

References

[1] E. Eskin, "Anomaly detection over noisy data using learned probability distribution", *In Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp.443-482.

[2] A. Ghosh, and A. Schwartzbard, "A Study in using neural networks for anomaly and misuse detection", *In Proceedings of the Eighth USENIX Security Symposium*, 1999, pp.443-482.

[3] W. Lee, S.J. Stolfo, and K. Mok, "A Data mining in workflow environments : Experience in intrusion detection", *In Proceedings of the 1999 Conference on Knowledge Discovery and Data Mining*, 1999.

[4] M.E. Tipping, and C.M. Bishop, "Mixtures of probabilistic principal component analysers", *Neural Computation 11(2)*, 1998, pp.443-482.

[5] M.A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks", *AICHE Journal 37(2)*, 1991, pp.233-243.

[6] K.I. Diamantaras, and S.Y. Kung, "Principal Component Neural Networks: Theory and Applications", New York John Wiley & Sons, Inc. 1996.

[7] Kim, Byung Joo. Shim, Joo Yong. Hwang, Chang Ha. Kim, Il Kon, "On-line Feature Extraction Based on Emperical Feature Map", *Foundations of Intelligent Systems, volume 2871 of Lecture Notes in Artificial Intelligence*, 2003, pp440-444.

[8] W.S Softky, and D.M Kammen, "Correlation in high dimensional or asymmetric data set: Hebbian neuronal processing", *Neural Networks vol. 4*, Nov. 1991, pp.337-348.

[9] H. Gupta, A.K. Agrawal, T. Pruthi, Shekhar, C., and R. Chellappa, , "An Experimental Evaluation of Linear and Kernel-Based Methods for Face Recognition", *accessible at http://citeseer.nj.nec.com.*

[10] J.A.K. Suykens, and J. Vandewalle, "Least squares support vector machine classifiers", *Neural Processing Letters, vol.9*, 1999, pp.293-300.

[11] V. N. Vapnik, textitStatistical learning theory, John Wiley & Sons, New York 1998.

[12] P. Hall, D. Marshall, and R. Martin, "on-line eigenalysis for classification", *In British Machine Vision Conference, volume 1*, September 1998, pp.286-295.

[13] J. Winkeler, B.S. Manjunath, and S. Chandrasekaran, "Subset selection for active object recognition", *In CVPR, volume 2*, IEEE Computer Society Press, June 1999, pp.511-516.

[14] H. Murakami, B.V.K.V. Kumar, "Efficient calculation of primary images from a set of images", *IEEE PAMI, 4(5)*, 1982, pp.511-515.

[15] B. Scholkopf, A. Smola, and K.R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem", Neural Computation 10(5), 1998, pp.1299-1319.

[16] K. Tsuda, "Support vector classifier based on asymmetric kernel function", *Proc. ESANN*, 1999.

[17] S. Mika, "Kernel algorithms for nonlinear signal processing in feature spaces", Master's thesis, Technical University of Berlin, November 1998.

[18] *http://kdd.ics.uci.edu/databases/kddcup99.*