Combining Principal Component Analysis, Decision Tree and Naïve Bayesian Algorithm for Adaptive Intrusion Detection

Zhi-Guo Chen
Department of Internet Multimedia
Konkuk University, South Korea
+82-10-2807-0614
chenzhiguo520@gmail.com

Sung-Ryul Kim*
Department of Internet Multimedia
Konkuk University, South Korea
+82-2-450-4134
kimsr@konkuk.ac.kr

ABSTRACT
In this paper, a new learning algorithm for adaptive network intrusion detection using principal component analysis, decision tree and Na ïve Bayesian classifier is presented. First we use PCA (Principal Component Analysis) to remove unimportant information like the noise in the data sets, to reduce the dimension, and to retain the important information as much as possible. Then we use the Decision tree and Naive Bayesian algorithm to make Intrusion Detection Model. We have tested the performance of our proposed algorithm on the KDD99 benchmark intrusion detection dataset. The experimental result prove that the proposed algorithm achieved high detection rates (DR), low false positive (FP) and low false negative (FN) for different types of network intrusions.

Categories and Subject Descriptors
C.2.0 [**Computer Communication Network**]: General-Security and protection.

General Terms
Algorithms, Security

Keywords
Intrusion Detection, Principal Component Analysis (PCA), Decision Tree, and Naive Bayesian

1. INTRODUCTION
With the rapid development and application of computing and communication technologies, computer network security has become more and more important. Since Intrusion detection system (IDS) are being put in environments with ever more traffic loads, we need more efficient Intrusion Detection System that have high detection rate (DR), low false positive (FP) and false negative (FN).

Intrusion Detection System [1] is broadly classified into two categories: Misuse-based IDS and Anomaly-based IDS [2]. Misuse-based IDS usually performs pattern matching techniques to detect known attacks. Anomaly-based IDS is able to detect the unknown data by build profiles of normal behaviors. If the new behavior is far from normal behaviors of the profiles, then the behavior will be treated as an attack. In this paper, a new learning algorithm for adaptive network intrusion detection using principal Component analysis, Decision Tree [3] and Na ïve Bayesian [4] classifier is presented which has high detection rate (DR) on KDD Cup 99 Dataset. PCA (Principal Component Analysis) [5, 6] is used to reduce the dimension of original high dimensional data and remove noise effectively. Then we use the new dataset to make Intrusion detection Model by Decision Tree and Naive Bayesian algorithms. The experimental results show that the proposed method is efficient for Anomaly-based intrusion detection.

The rest of this paper is organized as follow: in section 2, we will introduce the KDD Dataset, in section 3, introduce performances estimation of IDS. In section 4, we will introduce related work. Section 5 describes our proposed method. In section 6, we introduce Experiment Results and Evaluations. Finally, in section 7, we conclude this paper.

2. DATASET
The KDD Cup 1999 dataset [7] is used in this experiment. The dataset is network connection data from the U.S Air Force LAN in nine weeks. Each record has 42 attributes. Among them, 41 are characteristic attributes and 1 is class identity, and 34 attributes are numeric (continuous data type) and 7 (discrete data type) attributes are symbolic in the characteristic attributes. The data contains attack types (marked at class identity) that can be classified into four main categories:

1. Denial of Service (DOS): Denial of Service (DOS) attack makes some computing or memory resources too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.

2. Remote to User (R2L): Remote to User (R2L) is an attack that a remote user gains access of a local user account by sending packets to a machine over a network communication.

3. User to Root (U2R): User to Root (U2R) is an attack that an intruder begins with the access of a normal user account and then becomes a root-user by exploiting various vulnerabilities of the system.

RACS'13, October 1–4, 2013, Montreal, QC, Canada.

*Corresponding Author

4. Probing: Probing (Probe) is an attack that scans a network to gather information or find known vulnerabilities. An intruder with a map of machines and services that are available on a network can use the information to look for exploits.

So, all of the records are classified into 5 main classes: normal, DOS, R2L, U2R and Probing.

3. PERFORMANCES ESTIMATION

Detection rate (DR), True Positive Rate (TP) and False Positive Rate (FP) are important parameters that are used for performance estimation [8] of Intrusion Detection Models. They are defined as follows:

Table 1. Parameters for performances estimation of IDS

Parameters	Definition
True Positive (TP) or Detection Rate (DR)	Attack occur and alarm raised
False Positive (FP)	No attack but alarm raised
True Negative (TN)	No attack and no alarm
False Negative (FN)	Attack occur but no alarm

$$Detection\ \ Rate = \frac{Number\ of\ classified\ pattons}{Total\ number\ of\ patterns} * 100\% \qquad (1)$$

$$True\ Positive\ Rate = TP / TP+FN \qquad (2)$$

$$False\ Positive\ Rate = FP / FP+TN \qquad (3)$$

4. RELATED WORK

4.1 PCA Algorithm

The performance of the Intrusion Detection Model depends on the quality of dataset. So Noise in the dataset is one of the challenges in data mining. The reason for dealing with noisy data is that it will avoid over fitting the dataset. The irrelevant and redundant attributes of dataset may lead to complex classification model and reduce the classification accuracy. So we need to reduce noises and irrelevant and redundant attributes firstly.

The advantages of PCA algorithm are that it can reduce unimportant information like noise in dataset and also reduce the dimensionality in the dataset retain the variation presents in the original dataset [6]. So we choose PCA algorithm to reduce the dimensionality of dataset to make our Intrusion detection Model.

The following describe the PCA algorithm we use to deal with dataset S $(D \times N)$ where N is the number of data example and D is the number of dimension of original dataset S. Each column represents one data that have dimensionality D.

$$S = \{S_1, S_2, ..., S_N\} = \begin{pmatrix} S_{11} & \cdots & S_{m1} & \cdots & S_{N1} \\ S_{12} & \cdots & S_{m2} & \cdots & S_{N2} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ S_{1D} & \cdots & S_{mD} & \cdots & S_{ND} \end{pmatrix}$$

1. Map discrete attributes to continuous attributes.

2. Calculate the mean value of all data examples $\overline{S}$:

$$\overline{S_i} = \frac{1}{N}\sum_{m=1}^{N} S_i (1 < i < D)$$

$$S_m = (S_{m1}, S_{m2}, ..., S_{mi}, ..., S_{mD})^T (1 < m < N)$$

$$\overline{S} = (\overline{S_1}, \overline{S_2}, ..., \overline{S_i}, ..., \overline{S_D})^T$$

3. Subtract mean vector for each data example:

$$S' = \begin{pmatrix} S'_{11} & \cdots & S'_{m1} & \cdots & S'_{N1} \\ S'_{12} & \cdots & S'_{m2} & \cdots & S'_{N2} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ S'_{1D} & \cdots & S'_{mD} & \cdots & S'_{ND} \end{pmatrix}$$

$$S'_{\mathrm{m}} = S_m - \overline{S} = (S_{m1} - \overline{S_1}, S_{m2} - \overline{S_2}, ..., S_{mD} - \overline{S_D})^T$$

$$S'_{\mathrm{m}} = (S'_{m1}, S'_{m2}, ..., S'_{mD})^T (1 < m < N)$$

4. Find covariance matrix $\sum$ of dataset S, and then calculate all eigenvectors and eigenvalues of $\sum$;

5. According the eigenvalues to select the dimension number d of biggest eigenvectors to make the new dataset. The eigenvectors are represented as $u_1, u_2, \ldots, u_d$ and then find its matrix transpose U, represented as follows:

$$U = \begin{pmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_d^T \end{pmatrix} = \begin{pmatrix} u_{11} & \cdots & u_{1m} & \cdots & u_{1D} \\ u_{21} & \cdots & u_{2m} & \cdots & u_{2D} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ u_{d1} & \cdots & u_{dm} & \cdots & u_{dD} \end{pmatrix}$$

$$D' = US' = \begin{pmatrix} u_{11} & \cdots & u_{1m} & \cdots & u_{1D} \\ u_{21} & \cdots & u_{2m} & \cdots & u_{2D} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ u_{d1} & \cdots & u_{dm} & \cdots & u_{dD} \end{pmatrix} \begin{pmatrix} S'_{11} & \cdots & S'_{m1} & \cdots & S'_{N1} \\ S'_{12} & \cdots & S'_{m2} & \cdots & S'_{N2} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ S'_{1D} & \cdots & S'_{mD} & \cdots & S'_{ND} \end{pmatrix}$$

$$= \begin{pmatrix} D'_{11} & \cdots & D'_{m1} & \cdots & D'_{N1} \\ D'_{12} & \cdots & D'_{m2} & \cdots & D'_{N2} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ D'_{1d} & \cdots & D'_{md} & \cdots & D'_{Nd} \end{pmatrix}$$

6. Get new dataset $D = (D')^T$ $(N \times d)$, where N is the number of the data examples and d is the dimension of the new dataset D. Add the original dataset's class identity in new Dataset D and we have completed the transformed dataset D and with every record included with d attributes and one class identity.

4.2 Decision Tree Algorithm

Decision tree has been widely used in classification and decision making. ID3 algorithm is proposed by R. Quinlan in [9]. The ID3 decision tree divides data items into subsets, based on the attributes. The basic idea is to find the one attribute that maximizes information gain and divide the data using that attribute.

Let us assume that D is a dataset and that A_i is one of attribute of the dataset. Let us also assume that the dataset has a set of classes $\{C_1, C_2, ..., C_m\}$.

Information Gain (D, A) is an impurity-based criterion that uses the entropy measure (originally from information theory) as the impurity measure [9]. The information gain of an attribute is defined by how much the entropy is reduced with attribute A. It is calculated as the follows:

$$Gain(D, A) = Entropy(D) - \sum_{i \in Values(A)} \frac{|D_i|}{|D|} Entropy(D_i) \qquad (4)$$

Where,

$$Entropy\ (D) = -\sum_{i=1}^{m} \frac{freq(C_i, D)}{|D|} \log_2 \frac{freq(C_i, D)}{|D|} \quad (5)$$

Where $freq(C_i, D)$ denotes the number of examples in the dataset D belonging to class C_i and D_i is the subset of dataset D divided by the attribute A's value A_i.

1. If A_i is a discrete attribute [13] then divide the dataset by discrete values of A_i (a_1, a_2,…, a_m).

2. If A_i is a continuous attribute then use information gain to efficiently find the best split point and disperse the values of the continuous attribute [10]. If attribute A_i has values $\{A_{i1}, A_{i2}, \ldots, A_{im}, \ldots, A_{ih}\}$ then firstly sort the continuous values (called candidate cut point) from small to large and then divide dataset with these candidate cut points with $> A_{im}$ *and* $\leq A_{im}$ $(i \leq m < h)$. Then we calculate the information gain for each possible cut point. The cut point for which the information gain is maximized amongst all the candidate cut points is taken as the best cut point. We then divide the current dataset (subset dataset) according this point. Now we build the tree by dividing the current dataset (subset dataset) by the selected attribute (discrete attributes according the attribute values, continuous attributes according to best cut point).

The stop condition is as follows:

1. The Entropy of one dataset (included subset dataset) is 0. We are sure that the Class of this dataset (subset dataset) is same. In this case we stop dividing the dataset.

2. If more than one attribute have same maximum information gain in the subset dataset D'''. We do not know which attribute to select here. We will use $\{freq(C_1, D'''), freq(C_2, D'''), \ldots, freq(C_m, D''')\}$ to determine the classes in dataset D'''.

4.3 Naive Bayes Algorithm

The Naive Bayesian classifier is based on Bayes' theorem with independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for our dataset.

Every example is d dimension vector described as $\{A_1, A_2, \ldots, A_d\}$.where d is the number of attributes. Assume that the dataset have a set of Class $\{C_1, C_2, \ldots, C_m\}$. Given an unknown example X $\{x_1, x_2, \ldots, x_d\}$. Naive Bayesian classifier will allocate the example X to class C_i, if and only if a posterior probability $P(C_i / X) > P(C_j / X), (i \neq j)$.

According to Bayes' theorem:

$$P(C_i / X) = \frac{P(X / C_i) P(C_i)}{P(X)} \quad (6)$$

For any one category, $P(X)$ is a constant, so we just need to calculate $P(X / C_i)P(C_i)$.

A priori probability $P(C_i)$ can be calculated by $freq(C_i, D)$. $freq(C_i, D)$ denotes that the number of examples in the training dataset D that belong to the Class C_i.

$P(X / C_i)$ is the likelihood which is calculated by training dataset according to follow function:

$$P(X / C_i) = \prod_{k=1}^{d} p(x_k / C_i) \quad (7)$$

1. If A_k is discrete attribute [12]. Then $p(x_k / C_i) = S_{ik} / S_i$ where, S_{ik} is the number of examples that have x_k value in attribute A_k that belong the class C_i in dataset. S_i is the number of data that belong to C_i in training dataset.

If $S_{ik} = 0$, $p(x_k / C_i)$ will be 0, A zero probability cancels the effects of all of the other a posteriori probabilities on C_i. So, we can assume that our training set is so large that adding one to each count that we need would only make a negligible difference in the estimated probabilities, yet would avoid the case of zero probability values. This technique is known as Laplacian correction (or Laplace estimator) [15].

$$p(x_k / C_i) = \frac{S_{ik} + 1}{S_i + q} \quad (8)$$

Where, q is the number of discrete values in attribute A_k.

2. If A_k is continuous attribute [14]. It is often assumed that the attribute obedience to Gaussian distribution [11, 12].

$$P(x_k / C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) = \frac{1}{\sqrt{2\pi}\sigma_{C_i}} e^{-\frac{(x_k - \mu_{C_i})^2}{2\sigma_{C_i}^2}} \quad (9)$$

Where, $\mu_{C_i}, \sigma_{C_i}^2$ is expectation and variance belonging to Class Ci in dataset.

If we assume that the attribute A_k have $A_{k1}, A_{k2,\ldots}, A_{km,\ldots}, A_{kh}$ continuous values belong to Class C_i.(h is the number of continuous values in attribute A_k belong to Class C_i), $\mu_{C_i}, \sigma_{C_i}^2$ can be calculated by follow function:

$$\mu_{C_i} = (A_{k1} + A_{k2} + \ldots + A_{km} \ldots + A_{kh}) / h \quad (10)$$

$$\sigma_{C_i}^2 = \{(A_{k1} - \mu_{C_i})^2 + (A_{k2} - \mu_{C_i})^2 + \ldots + (A_{kh} - \mu_{C_i})^2\} / (h - 1) \quad (11)$$

5. OUR PROPOSED METHOD

5.1 Using PCA Algorithm

We first randomly select data in file "corrected" which is downloaded from KDD Cup 99 site to make training dataset and test dataset. The dataset is $S\ (N \times D)$, where N is the number of the data records and D is the dimension of the KDD data (41 dimension except for class identity). And then we use the PCA algorithm to reduce the dimension of the KDD data to get new Dataset $D\ (N \times d)$. Where N is the number of the new dataset and d is the dimension (the number of attributes) of new dataset D. We then add the original dataset's class identity in the new Dataset D. After reducing the dimension by PCA we use the Decision Tree and Naive Bayesian algorithms to build the Intrusion Detection Model.

5.2 Decision Tree and Naïve Bayesian

After reducing the dimension by PCA, we get new dataset D where all of the attributes are continuous ones. So we should use the continuous attributes handling method that has been described earlier.

Dataset D contains attributes $\{A_1, A_2, \ldots, A_d\}$ and each attribute A_i contains the following continuous attribute values $\{A_{i1}, A_{i2}, \ldots, A_{im}, \ldots, A_{ih}\}$. The training dataset also has a set of Class $\{C_1, C_2, \ldots, C_m\}$. Each data in the training data D have particular class C_j. We follow the next procedure.

1. Find the best cut point of every attributes and calculate information gain in training dataset D with decision tree continuous attribute handling method. Compare the information gain of every attribute. Select A_i among the attributes $\{A_1, A_2, \ldots, A_d\}$ which maximizes information gain to make the root attribute node, and then divide the training dataset D into subsets $\{D_1, D_2\}$ depending on the best cut point A_{im}.

2. Find the best cut point of every attributes and calculate information gain in subset dataset D_i. Compare the information gain of every attribute. Select A_i among the attributes $\{A_1, A_2, \ldots, A_d\}$ which have maximize information to make the attribute node, Then divide the subset dataset D_i into subsets $\{D_{i1}, D_{i2}\}$ depending on the best cut point A_{im}.

3. Continue this process until subset dataset's entropy is zero or more than one attribute have same maximizes information gain in the subset dataset D_{ijk}. If subset dataset's entropy is zero and we know that all of the Class C_i in subset dataset is the same. In this case, a leaf node will be set up. And if the chosen attribute A_i 's information gain is equal to other attributes, we save the subset dataset in that node.

When we test an example data X, we can use the tree structure to find the Class (leaf node). If the class is not unique, it's leading-out the subset dataset in that node. And then use Naive Bayesian algorithm continuous attribute handling method to calculate a posterior probability $P(C_i / X)$ to label the Class C_i to the example data.

6. EXPERIMENTAL ANALYSIS

In order to evaluate the performance of the proposed algorithm for intrusion detection, we performed 5-Class (Normal, Probing, DOS, U2R, and R2L) classification using KDD99 dataset. We compared the result with ID3 algorithm, ID3 and Naive Bayesian algorithm.

Table 2. Number of training and test examples

Types	Training Examples	Test Examples
Normal	9766	9784
Probing	711	670
Denial of Service	36884	36905
User to Root	41	45
Remote to User	2598	2596
Total	50000	50000

Table 3. Performance of each algorithm

Method	Normal	Probe	DOS	U2R	R2L
Proposed Algorithm(DR %)	94.06	95.52	99.97	77.78	77.20
Proposed Algorithm(FP %)	5.93	0.26	0.11	0.01	5.55
Proposed Algorithm(FN %)	6.37	3.43	0.020	13.3	22.61
ID3 (DR %)	95.93	92.39	99.74	68.88	64.21
ID3 (FP %)	4.07	0.47	0.36	0	3.24
ID3 (FN %)	10.14	0.15	0.15	22.22	35.71
ID3, Naive Bayesian (DR%)	86.66	94.77	99.97	31.11	97.77
ID3, Naive Bayesian (FP %)	13.38	0.26	0.12	0	13
ID3, Naive Bayesian (FN %)	0.87	3.73	0.016	17.78	1.78

Our experimental results show that the proposed method give better detection rate, low false positive and low false negative than ID3, ID3 and Naive Bayesian method. From these result, we conclude that combining PCA, ID3 and Naive Bayesian method outperforms ID3 and combining ID3 and Naive Bayesian method. So, we can see that PCA can be effective to reduce noise and retain the variation present in the original dataset as much as possible. Since we randomly select data in file "corrected" which is downloaded from KDD Cup 99 site to make training examples and test example, there is not many examples of U2R, Probe and R2L, so we just get detection rate 77.78%, 95.52%, 77.20% for these type. We use the ID3 algorithm to make the tree and use Gaussian distribution function to deal with the data at every node that has not unique Class. But a large number of attribute values in original dataset S is 0 or 1. So if the attribute A_k's values at one node is same, the variance will be 0, and Gaussian distribution function will have no meaning. A posterior probability of Naive Bayesian algorithm will no longer be accurate. It is the reason of ID3 and Naive Bayesian method have low detection rate at U2R and normal than ID3 method. We use PCA algorithm to deal with the original dataset S to new dataset D can avoid this situation, so our proposed method outperforms other method.

As a result our proposed method is effective to improve the performances of Anomaly-based intrusion detection.

7. CONCLUSION

This paper introduced a new hybrid learning algorithm for network intrusion detection using PCA (Principal Component Analysis), decision tree and naive Bayesian algorithm. The results show that it has high detection rate, low false positive and low false negative to detect the 5 type of the KDD dataset. But the false negative of remote to user (R2L) is high. The reason is there have not many examples of R2L and the intrusion detection model is classified the data as normal. So the future work will focus on reducing the false positive and false negative of remote to user (R2L) and improve the detection rate of R2L and user to root (U2R). We also will focus on check the overhead and the benefit such as size or time about this new learning algorithm.

8. ACKNOWLEDGMENTS

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2011-0029924).

9. REFERENCES

[1] Intrusion detection system, http://www.sans.org/reading_room/whitepapers/detection/understanding- intrusion-detection-systems_337.

[2] DOUGLAS J. BROWN, BILL SUCKOW, and TIANQIU WANG: A Survey of Intrusion Detection Systems.

[3] N. Ben Amor, S. Benferhat, and Z. Elouedi. Naive bayes vs decision trees in intrusion detection systems. In ACM symp osium on Applied computing (SAC2004), pages 420–424, Nicosia, Cyprus, 2004.

[4] Yang-Xia Luo: The Research of Bayesian Classifier Algorithms in Intrusion Detection System. 2010 International Conf-erence on E-Business and E-Government.

[5] Lindsay I. Smith: A tutorial on Principal Components Analysis New York (2002).

[6] Lu Zhao, Ho-Seok Kang, Sung-Ryul Kim: Improved Clustering for Intrusion Detection by Principal Component Analysis with Effective Noise Reduction. Information and Communication Technology Lecture Notes in Computer Volume 7804, 2013, pp 490-495.

[7] The third international knowledge discovery and data mining tools competition dataset\KDD99-Cup. Available at http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (1999).

[8] Thuzar Hlaing: Feature Selection and Fuzzy Decision Tree for Network Intrusion Detection. International Journal of

Informatics and Communication Technology (IJ-ICT) Vol.1, No.2, December 2012, pp. 109~118 ISSN: 2252-8776.

[9] J. R. Quinlan. Induction of Decision Trees. Machine Learning, 1:81-106, 1986

[10] Usama M. Fayyad, Keki B. Irani: On the handling of continous-valued attributes in decision tree generation. Machine Learning January 1992, Volume 8, Issue 1, pp 87-102.

[11] https://en.wikipedia.org/wiki/Variance#Population_variance_and_sample_variance.

[12] https://en.wikipedia.org/wiki/Naive_Bayes_classifier.

[13] Yogendra Kumar Jain, Upendra: An Efficient Intrusion Detection Based on Decision Tree Classifier Using Feature Reduction. International Journal of Scientific and Research Publications, Volume 2, Issue 1, January 2012 ISSN 2250-3153.

[14] George H. John, Pat Langley: Estimating Continuous Distributions in Bayesian Classifiers. UAI'95 Proceedings of the Eleventh conference on Uncertainty in artificial intelligence. Pages 338-345.

[15] http://cis.poly.edu/~mleung/FRE7851/f07/naiveBayesianClassifier.pdf