A Model for Network Traffic Anomaly Detection

Nguyen Ha Duong*, Hoang Dang Hai**

*Faculty of Information and Technology, National University of Civil Engineering,
55 Giai Phong, Hanoi, Vietnam

** Posts and Telecommunications Institute of Technology, Ministry of Information and Communication,
Nguyen Trai, Hanoi, Vietnam

duongnh@nuce.edu.vn, hdhai@mic.gov.vn

***Abstract*—Network traffic anomaly detection can find unusual events cause by hacker activity. Most research in this area focus on supervised and unsupervised model. In this work, we proposed a semi-supervised model based on combination of Mahalanobis distance and principal component analysis for network traffic anomaly detection. We also experiment clustering technique with suitable features to remove noise in training data along with some enhanced detection technique. With the approach of combining anomaly detection and signature-based detection system, we believe the quality of normal dataset will greatly improve.**

***Keyword*—Network traffic anomaly, anomaly detection, semi-supervised model, intrusion detection, network security**

I. INTRODUCTION

TODAY, network security is world-wide major concern of many countries. Organizations, companies and agencies are often facing with network attacks. The Intrusion Detection System (IDS) is implemented as an effective device to detect attacks outside or inside of a network. However, typical IDS often rely on signature database or pattern of known attack [1][2]. Therefore, intruders can change some parameters or characters that different from known patterns to make IDS unable to detect the new variances. Anomaly detection is the approach of recent IDS [3-6], since it does not require any prior knowledge about the attack signatures. Thus, it is capable to detect new attacks. Anomaly detection system (ADS) is used to detect the abnormal behaviour of a system. ADS can operate independently or as a component of IDS.

Manuscript received April 30, 2015. This work is a follow up of the accepted conference paper as an outstanding paper for the 17th International Conference on Advanced Communication Technology.

Nguyen Ha Duong is with the Faculty of Information and Technology, National University of Civil Engineering (corresponding author to provide phone: +84-98-756-7271; fax: +84-04-3869-1910; e-mail: duongnh@nuce.edu.vn).

Hoang Dang Hai was with Vietnam Computer Emergency Response Teams (VNCERT), Vietnam. He is now with the Posts and Telecommunications Institute of Technology, Ministry of Information and Communication, Vietnam (e-mail: hdhai@mic.gov.vn).

There are various network anomaly detection methods in recent years including machine learning techniques, statistical-based methods, principal component analysis (PCA) methods, etc. A review of different approaches for anomaly detection was given in [1-7]. Various methods and techniques proposed for anomaly detection indicate the difficulties of network traffic anomaly detection. There are several reasons: 1) the techniques of attackers become more sophisticated. There are many types of anomalies with different traffic data features. 2) No existing method is considered better than the others due to the complexity of the anomalies. Several issues remain unsolved regarding detection speed, accuracy, confidence, complexity, etc. Among various anomaly detection approaches, PCA has been proposed as an effective solution [8-10]. PCA is useful to reduce the complexity of the dataset while maintaining significant dataset features. From high level viewpoint, anomaly detection can be categorized into 3 models [7]:

1) Supervised: models both normality and abnormality. The entire area outside the normal class represents the outlier class. Supervised detection techniques fail to recognize behaviour that is not previously modelled, thus it lacks the ability to classify unknown anomalies.
2) Unsupervised: no prior knowledge of the data is needed. It processes the data as a static distribution, pinpoints the most remote points, flags them as potential outliers (anomaly).
3) Semi-Supervised: models only normality. It needs pre-classified data but only learns data marked normal. It is suitable for static or dynamic data as it only learns one class which provides the model of normality (baseline). If a point's distance exceeds the established threshold from the normal baseline, it is considered abnormal point.

In this paper, we propose a semi-supervised model using a modified Mahalanobis distance based on PCA (M-PCA) for network traffic anomaly detection. In order to reduce the noise of anomalies, we propose to use the K-means clustering algorithm to group similar data points and to build normal profile of traffic. This algorithm helps to improve the quality of the training dataset. The remainder of the paper is organized as follows: Section 2 presents related previous works. Section 3 proposes our research model. Section 4

proceeds with our experiment and results. Then, concluding remarks are provided in Section 5.

II. RELATED WORKS

The authors in [3-7] presented a review of anomaly based intrusion detection systems. A version of apriori algorithm was used with systolic arrays to build efficient pattern matching similar to a signature based method. In [5], the existence of irrelevant and redundant features has been studied that affect the performance of machine learning part of the detection system. This work showed that a good selection of the features will result in better classification performance. The authors in [6] demonstrated that the elimination of the unimportant features and irrelevant features did not reduce the performance of the detection systems.

Anomaly detection models based on PCA was proposed by Shyu in [8]. PCA was used together with outlier detection in assumption that the anomalies appear as outliers to the normal data. PCA can reduce the dimensionality of the dataset. The authors in [7-10] further improved PCA algorithm in combination with several algorithm such as sketch-based and signal-analysis based in a framework. The authors in [11] showed that an important advantage of combining redundant and complementary classifiers is to increase accuracy and better overall generalization. Several authors [9-12] also identified important input features in building IDS that are computationally efficient and effective. This work showed the performance of various feature selection algorithms including bayesian networks, classification and regression trees.

Several works provided experiments using KDD-CUP'99 dataset [13-16], which is a subset of the Intrusion Detection Evaluation dataset of DAPRA. Almost works proposed to use all features from the raw data of this dataset.

III. PROPOSED MODEL

A. PCA and Mahalanobis Distance

Principal Component Analysis (PCA) is a method for identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences [8]. PCA produces a set of principal components (PCs), based on eigenvalue/eigenvector pairs. Eigenvalues/eigenvectors can be built from covariance or correlation matrix

The covariance or correlation of any pair of PCs is equal to zero. PCA produces a set of independent variables so the total variance of a sample is the sum of all the variances accounted for by the PCs.

Outlier detection techniques are used to calculate the distance of captured live network data to the normal data projected by PCA procedure. We proposed using Mahalanobis distance for outlier detection, thus outliers measured are presumably anomalous network connections. Any network connection with a distance greater than an established threshold value is considered an outlier. The equation [8] of Mahalanobis distance d between observation x and the sample mean μ is:

$$d^2(x,\mu) = (x-\mu)'S^{-1}(x-\mu) \quad (1)$$

where: S^{-1} is the sample covariance matrix.

(1) takes into account the covariance matrix, thus, it can measure correlation between variables. In this paper, we use correlation matrix instead of covariance matrix since many variables in the training dataset were measured on different scales and ranges. The drawback of this method is the computationally demanding when calculating the inverse of the correlation matrix for feature vectors with a large number of dimensions. We need a method to calculate this distance more efficiently for each new connection. As in [8], the sum square of standardized PCs score is equivalent to the Mahalanobis distance of the observation x from the mean of the sample as follows:

$$d(x,\mu) = \sum_{i=1}^{p} \frac{y_i^2}{\lambda_i} \quad (2)$$

where: y_i is i^{th} the PC score, λ_i is the i^{th} eigenvalue, μ is the mean vector of the trained data set.

(2) is not the best choice equation for outlier detection. Some outlier in y_i^2/λ_i can be small and the total sum of all y_i^2/λ_i take little account for that outlier. One advantage of PCA is the PCs can be sorted in order of decrease eigenvalue $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p > 0$ then using only some PCs to calculate distance and find outlier. The number of retained PCs is the weight (W) in our work. We need experiment and choose the effective W for anomaly detection.

Intuitively, we assume that the last PCs (*minor PCs*) contain variances which are inconsistent with the data structure of the original variables as indication of outlier. In experiment, we found that using only the *minor PCs* (1-threshold method) with the weight W_2 can achieve good detection result. Any observation has distance greater than an established threshold is considered abnormal event or attack.

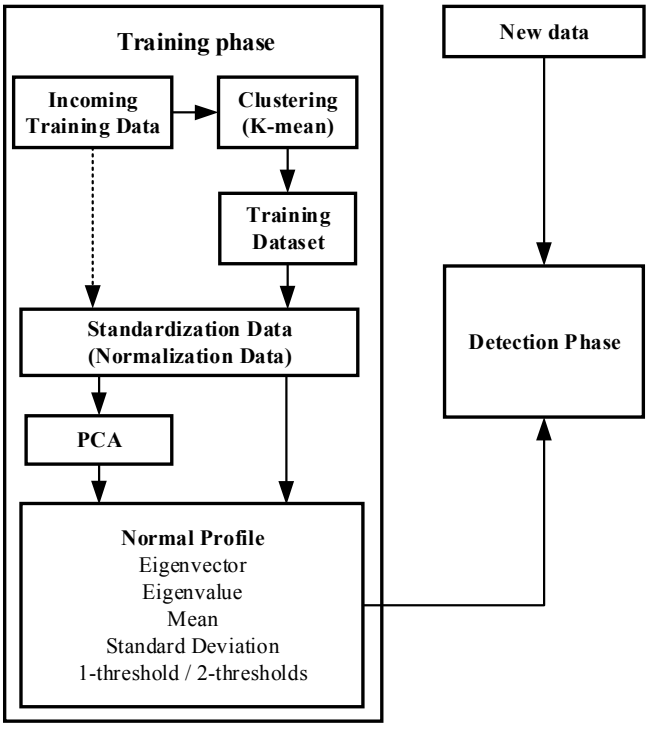

Fig. 1. The proposed model for anomaly detection.

Other method is using 2 set PCs: *major PCs* (the most significant PCs) with the weight W_1 and *minor PCs* (least significant PCs) with the weight W_2. Each set has separate distance calculation and upper threshold. Any observation has distance greater than corresponding threshold is

considered outlier. *Major PCs* often capture normal trend in variance of original variables. The use of both *major* and *minor PCs* is called 2-thresholds method. We call both 1-threshold and 2-thresholds methods with a common name: M-PCA method.

B. The Basic Model

Fig. 1 shows our proposed model for anomaly detection using M-PCA method and a method to reduce the noise in training data.

The core principle of anomaly detection is calculating the distance and building normal traffic profile. Distance describes how far a point compared to a centre of the known distribution. Our anomaly detection scheme require 2 phases: Training phase and detection phase.

C. Training Phase

The purpose of training phase is building normal traffic profile from normal data pattern. It has following steps:

Step 1: Choose the features $X_1, X_2,..., X_p$ which affect normal profile (p is the number of the features used in training and detection phase). Build the normal profile on selected features will reduce the number of dimensions needed to process. PCA is used to analyse the contribution of each feature to PCs of normal data.

Step 2: The network traffic needs to be free of attacks at training time in order to get a snapshot of captured network traffic for training dataset. In reality, this traffic can contain some small attacks considered as noise. Thus, we need to clean it beforehand. We propose using K-means clustering algorithm to remove outliers (noise) of the input data. We assume that the noise is much lesser than the normal data and recommend accepted noise level approximately at 10% of all incoming training data.

Step 3: The cleaned training data needs to standardize:

$$z_k = \frac{x_k - \overline{x}_k}{\sqrt{s_k}} \tag{3}$$

where: $\overline{x_k}$ and s_k is the sample mean and sample variance of the feature X_k in trained data set respectively; z is the standardized vector of training data set, and $z = (z_1, z_2, ..., z_p)'$

Step 4: Calculate the correlation matrix then pairs of eigenvector and eigenvalue.

Step 5: Compute the PC score of each sample in training data with z and eigenvector:

$$y_i = e'_i z \tag{4}$$

where: y_i is i^{th} the PC score, e_i is the i^{th} eigenvector

Step 6: Sort the PCs by eigenvalues in descending order: $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p > 0$

Step 7: Compute distance for each observation of training dataset with 1-threshold or 2-thresholds method. 1-threshold method use only minor PCs while 2-thresholds method use both major PCs and minor PCs:

$$d(major\ PCs) = \sum_{i=1}^{q} \frac{y_i^2}{\lambda_i} \tag{5}$$

$$d(minor\ PCs) = \sum_{i=r}^{p} \frac{y_i^2}{\lambda_i} \tag{6}$$

where: $0 < q < r < p$

Step 8: Build the empirical cumulative distribution function (ECDF) of distances. Choose the thresholds corresponding to estimated false positive ratio.

D. Remove noise of input data by K-means

K-means clustering help to clean the data when it have more noise than it should be. The noise can be recognized as attack and normal connection which are outlier with other normal connection. K-means is a clustering analysis algorithm that groups objects based on their feature values into K disjoint clusters. Objects classified into the same cluster have similar feature values. K is a positive integer number specifying the number of clusters, it has to be given in advance. Here are the steps of the K-means clustering algorithm:

Step 1: Define the number of clusters K and initialize K cluster centroids. This can be done by arbitrarily dividing all objects into K clusters, computing their centroids, and verifying that all centroids are different from each other. The centroids can be initialized to various objects chosen arbitrarily.

Step 2: Iterate over all objects and compute the distances to the centroids of all clusters. Assign each object to the cluster with the nearest centroid.

Step 3: Recalculate the centroids of both modified clusters.

Step 4: Repeat step 2 until the centroids do not change any more.

The distance we used in K-means algorithm is Pearson correlation distance [15]. Pearson correlation measures the similarity in shape between two profiles. The formula for the Pearson Correlation distance is:

$$d = 1 - c \tag{7}$$

where: $c = z(u) \bullet z(v)/n$ is the dot product of the z-scores of the vectors u and v. The z-score of u is constructed by subtracting from u its mean and dividing by its standard deviation. Each centroid is the component-wise mean of the points in that cluster, after centering and normalizing those points to zero mean and unit standard deviation.

E. Detection Phase

In detection phase we use the sub-score of M-PCA method to detect each new point from the distribution of the trained data point.

This phase match each new observation with established normal profile to detect anomaly. This include following steps:

Step 1: Standardize data with means and variances from sample training dataset.

Step 2: Compute PC score of each observation with trained eigenvectors which map observed data to subspace.

Step 3: Compute distances of each observation as in (5), (6). A new connection will have 1 or 2 distance values depend on 1-threshold method or 2-thresholds method.

Step 4: Compare thresholds and detection decision: If new connection's distance is greater than any of the established threshold, it marks as anomaly connection. Otherwise, it is normal connection.

1-threshold method:

If $d(minor\ PCs) > d_2$, classify new connection as abnormal

Else $d(minor\ PCs) \leq d_2$ classify new connection as normal

2-thresholds method:

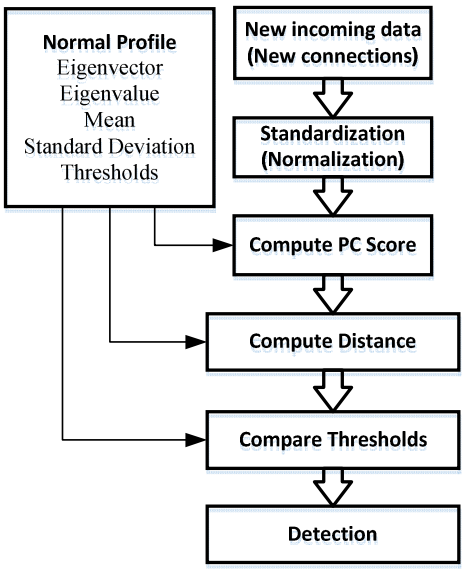

Fig. 2. Detection phase of the model.

If $d(major\ PCs) > d_1$ OR $d(minor\ PCs) > d_2$, classify new connection as abnormal

Else $d(major\ PCs) \leq d_1$ AND $d(minor\ PCs) \leq d_2$ classify new connection as normal

Where: d_1 and d_2 are thresholds of *major PCs* and *minor PCs* respectively.

F. Proposal Enhancement of the Model

In our approach, Anomaly Detection System (ADS) works as an inherent component with signature-based detection system (other name is misuse IDS). ADS alone is a system of suspicious events detection. ADS will collect suspicious events and these events will be validated by signature database, administrator (human) (or supervised classification modules). Misuse IDS can detect intrusion base on packets while ADS often work with connections or flows which limit its response time. For this reason, ADS appropriate for the role of informer for network signature-based IDS generally. In this way, all components of detection system work in circle of spy and detective network manner.

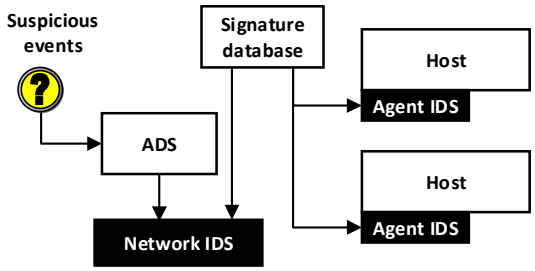

Fig. 3. ADS work as informer for network signature-based IDS

Misuse system rely on signature which have many rules and need to update regularly. Some IDSes work as agents (sensors) for hosts (computers) which only detect intrusion related to those hosts. Using all rules with agent-IDSes will overload the resource available in these hosts. Normally, only a subset of rules are active for some specific services in each host. If we use ADS to check anomaly traffic and then validate by a network signature-based IDS, the detection system will concentrate on more important events from ADS. That's why the precision of ADS system is important because too many false alarm will make ADS becomes unreliable. The number of output alarm from ADS can be large. There should be a high performance network signature-based IDS to validate the result from ADS frequently. Validation can happen at packet, content, connection, flow level. Misuse detection cannot detect unknown attack types. In theory, ADS can detect novel attacks and then validate by human or supervised machine learning. From post-processing, new signature can be generated.

Next section describe an enhancement model which integrate our ADS with misuse IDS. The model focus on the training phase because the quality of normal dataset is very important. Fig. 4 shows the enhancement of the model.

At first, incoming training data will go directly to training dataset pool if the administrator can guarantee there are no attacks. Otherwise, if incoming data contains some noise (attacks), clustering module (K-means or other algorithm) will be used to filter noise. For filter noise at more depth, input training data can go through the signature-based detection module to check and remove known attacks. All rule in signature database must be used because using only subset of rules may let some known attack connections pass as negative. Connection pass all the signature-checking as negative will be considered data for clustering. However, clustering rely on selection of some specific features or variables. In case that feature set is not available, clustering module need to disable and data pass signature detection can be added to training dataset before PCA step.

New data after pass of the anomaly detection module will go through validation checking. This step validates the correct of detection result and identify attack types. If the system only detect and cannot classify type of abnormal events, the detection result is almost nonsense. Signature database is used again to automatic check for attack types with all positive records from anomaly module. Normal data is often has good detection rate and outnumber attack data. For negative records, to avoid overwhelming with large number of data, signature database only need to check randomly chosen records. After validation, these normal records use as feedback to the normal dataset. This way, system doesn't waste resource for many unnecessary negative records and saves resource for identify attacks in positive records.

If anomaly module has too many false alarm records which discover by signature module, all records have to be checked by signature module whether they are positive or negative. And administrator must check the anomaly module, compare score with result from signature module to find what is wrong.

Positive records cannot identified with signature module can go through manual check. Identify attacks by human is challenged task and should only implement for novel attacks.

If the system already built pre-classified attack classes in machine learning database, positive records can be checked with this database to classify attack types. In this manner, system works as supervised model.

Data in training set need to have aging time. New cleaned training data will replace oldest data. Some old attack records still in the normal dataset will be gradually removed. This will help regulate the training data more efficiently.

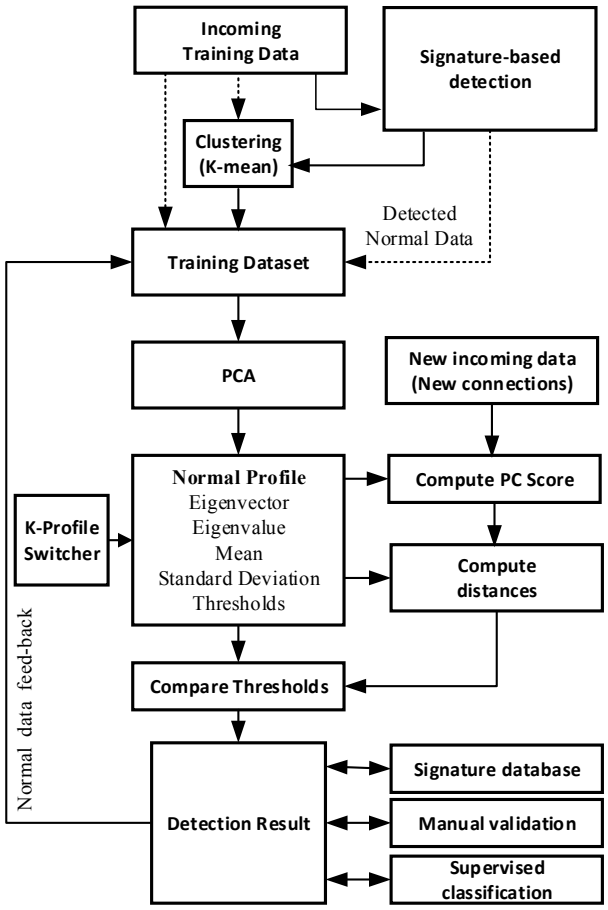

Fig. 4. The Enhancement model

Profile of network can change due to some reasons. Administrators build multiple profiles based on daily usage of network traffic characteristics. A profile switcher can be programmed to change suitable profile accordingly.

IV. EXPERIMENT AND RESULTS

A. NSL-KDD Dataset

In [8], Shyu's test results use KDD CUP 99, the old data set contains 75% of redundant records. KDD CUP'99 is the mostly widely used data set for anomaly detection. The inherent problem of KDD dataset leads to new version of NSL KDD dataset that are mentioned in [14-16]. In [16], the authors conducted a statistical analysis on this data set and found two important issues which highly affect the performance of evaluated system, and results in very poor evaluation of anomaly detection approaches. To solve these issues, they proposed a new data set, NSL-KDD, which consists of selected records of the complete KDD data set [14] and does not suffer from any of the mentioned shortcomings.

We don't use the KDDCUP'99 but the NSL-KDD instead, since the advantages of NSL KDD dataset are as follows:

1) No redundant records in the train set, so the classifier will not produce any biased result.
2) No duplicated records in the test set which have better reduction rates.
3) The number of selected records from each difficult level group is inversely proportional to the percentage of records in the original KDD data set.

The NSL-KDD data includes 41 features and 5 classes that are normal and 4 categories of attacks: Denial of Service Attack (DoS), Probe, Remote to Local Attack (R2L), and User to Root Attack (U2R). Attack categories and types in NSL-KDD data set were given in [14].

B. Performance Measures

We use the following performance measures:

True Positive (TP): the event when an attack connection correctly detected.

True Negative (TN): the event when a normal connection correctly detected.

False Positive (FP): the event when a normal connection detected falsely as attack connection.

False Negative (FN): the event when an attack connection detected falsely as normal connection.

Precision: The ratio of true positive and the number of detected connection as attack.

True Positive Rate (Recall): The ratio of true positive and the number of real attack in the sample data set.

False Positive Rate (FPR): The ratio of false positive and the real number of normal connection in sample data set.

Total Accuracy: The overall successful prediction of both attack and normal connection.

C. Removing Noise by K-means

We implemented experiments in Matlab R2013a. We use the KDDTrain+ dataset [14] for both training and detection phase.

Through experiments we found some features in NSL-KDD are significantly affects the normal profile (Table I). This will make the processing data faster. The K-means clustering algorithm was used with input training data to derive a more cleaned data. The goal of clustering is to remove attack points that not follow the baseline of major normal data. Attacks should be much lesser than normal traffic, otherwise malicious connection will dominate the normal traffic. We recommend malicious volume below 10% of the total incoming training data. In daily condition, many networks have that upper bound limit. Our experiments use 7000 connections for input data, which are chosen randomly from KDDTrain+ data and include above 900 attack connections.

TABLE I
6-FEATURES USED IN CLUSTERING

Features	Meaning
protocol_type	Protocol types (tcp, udp, …)
src_bytes	Number of bytes from source
count	Number of connections to the same host as the current connection in the past two seconds
diff_srv_rate	% of connections to different services
dst_host_same_srv_rate	% of connections to the same service for destination host
dst_host_serror_rate	% of connections that have SYN errors for destination host

The distance used of K-means algorithm is correlation distance. We use the number of clustering $K=2$ for K-means. The step of clustering input training data are:

1) Choose target cluster for training set.
2) Run K-means several times until target cluster have adequate data point. With 7000 records, the training data should have above 5000 records.

Table II depicted the number of attack connection before and after clustering input data. The result showed that clustering the input data significantly reduced the noise which includes attacks in the background traffic.

TABLE II
REDUCED NOISE OF INPUT DATA

Test	Number of attacks before clustering	Number of attacks after clustering	Attack Reduced Ratio (%)
Test1	928	152	83.60
Test2	937	157	83.20
Test3	935	164	82.50

D. Experiment with 2-thresholds Method

After clustering, the targeted cluster becomes the data for training. The system will derive the training parameters needed. For measuring the test accuracy, we use random 50,000 connections in KDDTrain+ data with the same features in table I. Through experiments we found that it is better to keep major PCs $W_1 = 3$ and minor PCs $W_2=3$. Increase or decrease PCs more or less than 3 will make the total accuracy decrease.

The results before clustering input data are shown in table III. Table IV depicts the same tests after clustering.

TABLE III
RESULT BEFORE CLUSTERING INPUT DATA

Test	Precision (%)	Recall (%)	FPR (%)	Total Accuracy (%)
Test1	86.6	22.7	3	62.6
Test2	87	22	2.83	62.4
Test3	83.6	22.2	3.7	62

Recall rate is sensitive with outlier in the data. Result in table III showed that a small of approximately 900 attack records in total 7000 training records still make the recall result very low accuracy. That make the total accuracy only achieve above 60%. Table IV shows the effective of removing noise with K-means in detection result.

TABLE IV
RESULT AFTER CLUSTERING INPUT DATA

Test	Precision (%)	Recall (%)	FPR (%)	Total Accuracy (%)
Test1	92	84.7	6.3	89.5
Test2	92	81.2	6.1	88
Test3	92	82.1	5.6	88.7

E. Experiment with 1-threshold Method

In the next experiment, we use 1-threshold method for detection using 13 features in Table V. We choose these features by experience.

We use some small sets of training data which select from pure normal connections, the first and second set have 1000 connections, the third and fourth set has 500 connections. The detection test with random 60000 connections. At first, we use all PCs ($W= 13$) for training phase and detection phase. Then we only use minor PCs ($W_2 = 3, W_1=0$). The detection result in Table VI and Table VII showed that 1-threshold method has good accuracy even with small training dataset. Using only minor PCs for 1-threshold give better recall rate and overall accuracy. Other advantage is the reduction dimension from 13 to 3 PCs.

TABLE V
13-FEATURES USED IN 1-THRESHOLD EXPERIMENT

Features	Meaning
interval	Interval of the connection
protocol_type	Protocol types (tcp, udp, …)
service	Destination service (e.g. telnet, ftp)
flag	Status flag of the connection
source_bytes	Bytes sent from source to destination
destination bytes	Bytes sent from destination to source
count	Number of connections to the same host as the current connection in the past two seconds
srv_count	Number of connections to the same service as the current connection in the past two seconds
serror_rate	% of connections that have Synchronization errors
rerror_rate	% of connections that have Rejection errors
diff_srv_rate	% of connections to different services
dst_host_count	Count of connections having the same destination host
dst_host_srv_count	Count of connections having the same destination host and using the same service

TABLE VI
1-THRESHOLD METHOD USING ALL PCS

Test	Precision (%)	Recall (%)	FPR (%)	Total Accuracy (%)
Test1	95.2	80	3.5	88.7
Test2	94.7	79.6	3.9	88.3
Test3	94	80.9	4.5	88.6
Test4	93.3	81	5.11	88.4

TABLE VII
1-THRESHOLD METHOD USING 3 MINOR PCS

Test	Precision (%)	Recall (%)	FPR (%)	Total Accuracy (%)
Test1	94.2	88	4.6	92
Test2	93.3	86.4	5.32	91
Test3	92.8	87.1	5.8	91
Test4	92.6	88.7	6.1	91.5

Next we test the detection (Table VIII) with training data after remove noise by K-means. The detection result is still good with 1-threshold method. We believe that clustering step remove some normal data with higher distance that make the false positive ratio above 10%.

TABLE VIII
1-THRESHOLD RESULT AFTER CLUSTERING

Test	Precision (%)	Recall (%)	FPR (%)	Total Accuracy (%)
Test1	86.2	84.7	11.6	86.7
Test2	86	83.8	11.7	86.2
Test3	86	81	11.2	85.2

In this case, using PCs with higher variance will take more normal data of higher distance. Table IX depicts the result when using all PCs to compute the distance ($W=13$). With more normal data detected, true negative ratio (TNR) which is the ratio between normal data and total normal data increase above 90% (and decrease FPR= 1−TNR).

TABLE IX
USING ALL PCS TO TAKE MORE NORMAL DATA

Test	Precision (%)	Recall (%)	FPR (%)	Total Accuracy (%)
Test1	91.8	83.4	6.5	88.7
Test2	91.9	83.3	6.4	88.7
Test3	91.2	83.5	7.1	88.5

We don't need to use all PCs for this purpose. Table X depicts the result when using 7 *major PCs* (W_1=7, W_2=0). As we expected, major PCs have more variance which detect normal data of high distance better. The advantage here is smaller number of necessary dimensions.

TABLE X
USING 7 MAJOR PCS TO TAKE MORE NORMAL DATA

Test	Precision (%)	Recall (%)	FPR (%)	Total Accuracy (%)
Test1	91.7	83.7	6.6	88.8
Test2	91.3	83.8	7.1	88.6
Test3	90.8	84	7.5	88.5

Because noise filter step by K-means often remove high distance data especially in *minor PCs*, K-means should only use with new incoming training data. Normal data feedback after validation will go directly to normal data pool (Fig. 4) awaiting for principal component analysis next time. With the feedback-regulation mechanism of normal data, more high distance data will add to the training dataset and improve the quality of normal profile. When a new profile is created from better quality normal dataset, detection system can use 1-threshold with small minor PCs. 1-threshold method can reduce the computation overhead and delay when analysis large amount of data.

V. CONCLUSION

In this work, we proposed M-PCA method for network traffic anomaly detection. Our approach concentrated on building normal traffic profile of the anomaly detection model. Through experiments we also showed that some features of NSL-KDD dataset are efficient with the normal profile. We propose a K-means clustering algorithm to reduce noise with input training data. The experiments showed that even with small training dataset (less than 1000 points), our approach has good performance including detection accuracy. We also proposed a new model integrates anomaly detection system with signature-based detection system along with some enhancements of building quality normal profile. In our future plan, we will develop and experiment the proposed model with an open source IDS in real network.

REFERENCES

[1] S. Axelsson, Intrusion Detection Systems: A Survey and Taxonomy, Chalmers University of Technology, Goteborg Sweden, 2000. http://www.cs.chalmers.se/~sax/ pub/

[2] S.S. Rajan, V.K. Cherukur, An Overview of Intrusion Detection Systems, 2010. http://www.idt.mdh.se/

[3] V. Jyothana, V. V. R. Prasad, K. M. Prasad, A Review of Anomaly based Intrusion Detection Systems, *Interl. Journal of Computer Applications*, Vol. 28–No.7, August 2011, pp. 28-34.

[4] A. Jain, B.Verma, J. L. Rana, Anomaly Intrusion Detection Techniques: A Brief Review, *Interl. Journal of Scientific & Engineering Research*, Vol. 5, Iss. 7, July 2014, pp.1372-1383.

[5] S. Myers, J. Musacchio, N. Bao, Intrusion Detection Systems: A Feature and Capability Analysis. Tech.Report UCSC-SOE-10-12. Jack Baskin School of Engineering, 2010.

[6] C. Kacha, K. A. Shevade, Comparison of Different Intrusion Detection and Prevention Systems, *Intl. Journal of Emerging Technology and Advanced Engineering*, Vol.2, Iss.12, Dec.2012, pp.243-245.

[7] M.H.Bhuyan, D.K.Bhattacharyya, J.K.Kalita, Survey on Incremental Approaches for Network Anomaly Detection, *Journal of Communication Nteworks and Information Security (IJCNIS)*, Vol.3, No.3, pp. 226-239, Dec.2011.

[8] M.L. Shyu, S.C.Chen, K.Sarinnapakorn, L.W.Chang, A Novel Anomaly Detection Scheme Based on Principle Component Classifier, *Proc. of the IEEE foundation and New Directions of Data Mining Workshop in 3rd IEEE Intl. Conference on Data Mining (ICDM03),* pp. 172-179, 2003.

[9] D. Brauckhoff, K. Salamatian, and M. May: Applying PCA for Traffic Anomaly Detection: Problems and Solutions, *IEEE INFOCOM*, pp.2866-2870, April, 2009.

[10] H.Huang, H.Al-Azzawi, H.Brani, Network Traffic Anomaly Detection, ArXiv:1402.0856v1, 2014.

[11] C.A.P.Boyce, A.N.Zincir-Heywood, A Comparison of Four Intrusion Detection Systems for Secure E-Business, 2003.

[12] V.J.Hodge, J.Austin, A survey of outlier detection methodologies, 2004.

[13] The KDD Archive, KDD99 cup dataset, 1999, http://kdd.ics.uci.edu /databases/kddcup99/kddcup99.html

[14] NSL-KDD data set for network-based intrusion detection systems, http://nsl.cs.unb.ca/KDD/NSLKDD.html, March 2009.

[15] M.K. Siddiqui, S.Naahid, Analysis of KDD CUP 99 Dataset using Clustering based Data Mining, 2013.

[16] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A Detailed Analysis of the KDD CUP 99 Data Set, *In the Proc. of the IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009)*, pp. 1-6, 2009.

Nguyen Ha Duong received the B.E. degree in electronic and telecommunication engineering from the Ha Noi University of Technology, Vietnam, in 2001, and the Msc. degree in electronic and telecommunication engineering from the Ha Noi University of Technology, Vietnam, in 2003. In 2001, he joined the Department of Network and System Engineering, IT Faculty, National University of Civil Engineering as a lecturer. His current research interests include network security, network protocol, routing, data mining and machine learning.

Hoang Dang Hai received the Diplom-Ing. degree in Technical Cybernetics from the Technical University Ilmenau (Germany) in 1984, Dr.-Ing. degree in Telematics and Dr.-Ing.habil. degree from the Technical University Ilmenau (Germany) in 1999 and 2003, respectively. He is currently an Associate Professor at the Post and Telecommunication Institute of Technology (PTIT), Ministry of Information and Communications of Vietnam. His current research interests include information security, wireless sensor networks, network security and network traffic management.