A Novel Covariance Matrix based Approach for Detecting Network Anomalies

Mahbod Tavallaee, Wei Lu, Shah Arif Iqbal, Ali A. Ghorbani
Faculty of Computer Science, University of New Brunswick
{m.tavallaee, wlu, arif.iqbal, ghorbani}@unb.ca

Abstract

During the last decade, anomaly detection has attracted the attention of many researchers to overcome the weakness of signature-based IDSs in detecting novel attacks. However, having a relatively high false alarm rate, anomaly detection has not been widely used in real networks. In this paper, we have proposed a novel anomaly detection scheme using the correlation information contained in groups of network traffic samples. Our experimental results show promising detection rates while maintaining false positives at very low rates.

1. Introduction

Intrusion detection has been extensively studied since the seminal report written by Anderson [2]. Traditionally, intrusion detection techniques are classified into two categories: misuse detection and anomaly detection. Misuse detection aims to encode knowledge about patterns in the data flow that are known to correspond to intrusive procedures in form of specific signatures that, for example, may be represented with rules. It is based on the assumption that most attacks leave a set of signatures in the stream of network packets or in audit trails, and thus attacks are detectable if these signatures can be identified by analyzing the audit trails or network traffic behaviors. A big advantage of signature detection is that it allows uncovering known attacks with very low false alarm rates. However, it is useless when faced with unknown or novel forms of attacks for which the signatures are not yet available and not included in the underlying model.

To address the weakness of misuse detection, the concept of anomaly detection was formalized in the seminal report of Denning [7]. Anomaly detection is dedicated to establish normal activity profiles by computing various metrics and an intrusion is detected when the actual system behavior deviates from the normal profiles. According to the characteristics of the monitored sources, anomaly detection can be classified into host based and network based. Typically, a host-based anomaly detection system runs on a local monitored host and uses its log files or audit trail data as information sources. The major limitation of host based anomaly detection is its capability to detect distributed and coordinated attacks that show patterns in the network traffic. In contrast, network based anomaly detection aims at protecting the entire networks against intrusions by monitoring the network traffic either on designed hosts or specific sensors and thus can protect simultaneously a large number of computers running different operating systems against remote attacks such as port scans, distributed denial of service attacks, propagation of computer worms, which stand for a major threat to current Internet infrastructure. As a result, we restrict our focus to network anomaly detection in this paper.

Machine learning techniques have been widely used on detecting network anomalies recently since machine learning can construct the required models automatically based on some given training data. With the growing complexity and number of different attacks, machine learning techniques achieved good results on building and maintaining network anomaly detection system with less human effort. Some typical machine learning techniques used in network anomaly detection include nearest-neighbor methods [8], decision trees [14], support vector machines [18], artificial neural networks [19], genetic computation [15], Bayesian networks [4], outlier detection [16], Y-means clustering algorithm [9], to name a few (see [12] for a general survey on these techniques). Even though machine learning techniques obtained good performance on detecting network anomalies, they are still faced with some major challenges. For instance, Barreno et al. claim that machine learning might not be secure in [5] and Sabhnani et al. also conclude that behavioral non-similarity in training and testing data will totally fail learning algorithms on anomaly detection in [20].

As a result, signal processing techniques have been successfully applied to the network anomaly detection as an alternative to the traditional self-learning based network anomaly detection approaches [3]. The most typical examples consist of Principal Component Analysis (PCA) [22],
Cumulative Sum (CUSUM) [23] and Covariance Matrix [10]. Although covariance matrix has been applied for detecting network anomalies in [10], we use it in a very different way.

Covariance matrix has been applied in [10] and used as a method to transform the original data to a new feature space, called covariance space, where the correlation differences among samples are evaluated. A threshold based detection approach and a traditional decision tree approach are, then, applied to find the anomalies. However, in our proposed method, called Covariance Matrix Sign (CMS), we use the covariance matrix as an anomaly detection approach. Toward this aim, we compare the signs in the covariance matrix of a group of sequential samples with the signs in the covariance matrix of the normal data obtained during the training process. If the number of differences exceeds a specified threshold, all the samples in that group will be labeled as attacks; otherwise they will be labeled as normal.

To the best of our knowledge, this is the first time that the covariance matrix method is used directly as an anomaly detector. Using a sign comparison instead of complicated classification and clustering methods, we take the advantage of less computational complexities compared to the approach used in [10]. Furthermore, our approach seems to have very good detection rates while maintaining the false positives at very low rates. In other words, it is capable of detecting a considerable amount of anomalies without misclassifying any normal sample as an attack.

The rest of the paper is organized as follows. In Section 2, we briefly overview principal component analysis (PCA), which is wildly used as an effective and computationally inexpensive feature reduction method. Our proposed detection scheme based on covariance matrix will be explained in Section 3. Section 4 presents the experimental evaluation of our approach and discusses the obtained results. Finally, in Section 5, we draw conclusions.

2. Principal Component Analysis

Principal Component Analysis (PCA) is a popular and effective method of lossy data compression which during the last decade has been widely used in data mining as a technique to reduce multidimensional data sets to lower dimensions for analysis. Compared to similar data reduction techniques such as wavelet transform, PCA tends to be better at handling sparse data. As a result, since data found in anomaly detection are usually high dimensional and sparse, it has attracted a lot of interest in Intrusion Detection Systems. Moreover, PCA is fairly computationally inexpensive, and that makes it even more beneficial for anomaly detection. PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on [11].

Assume $A$ to be an $n \times p$ matrix of $n$ observations $x_i$ in a p-dimensional space in which $n > p$.

$$A^T = \begin{bmatrix} x_{1,1} & x_{2,1} & \cdots & x_{n,1} \\ x_{1,2} & x_{2,2} & \cdots & x_{n,2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1,p} & x_{2,p} & \cdots & x_{n,p} \end{bmatrix}_{p \times n}$$

where

$$x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,p}) \quad 1 \leq i \leq n$$

Using singular value decomposition, we can find three matrices of $U$, $\Sigma$, and $V$ such that

$$A = U \Sigma V^T \quad (1)$$

where $U$ is an $n \times p$ matrix in which the columns are the left singular vectors, $\Sigma$ is a $p \times p$ diagonal matrix whose diagonal contains the singular values (eigenvalues) $\lambda_i$ where $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$, and $V$ is a $p \times p$ matrix in which the columns are the right singular vector (eigenvectors).

In the next step, we should mean center the data set $A$, i.e., to produce a new data set $X$ whose mean is zero.

$$X = A^T - \begin{bmatrix} \mu_1 & \mu_1 & \cdots & \mu_1 \\ \mu_2 & \mu_2 & \cdots & \mu_2 \\ \vdots & \vdots & \ddots & \vdots \\ \mu_p & \mu_p & \cdots & \mu_p \end{bmatrix}_{p \times n} \quad (2)$$

where

$$\mu_k = \frac{1}{n} \sum_{i=1}^{n} x_{i,k} \quad 1 \leq k \leq p$$

Having the matrices $V$ and $X$, we can simply find the corresponding value of the observations in the new coordinate system of principal components using the following formula.

$$Y^T = V^T X \quad (3)$$

The matrix $Y$ is the final data set in the principal components feature space with data items in rows and dimensions along columns.

3. The Proposed Detection Scheme

In [10], Jin et al. showed that group detection methods, generally, achieve a much higher detection rate than
the traditional sample-by-sample detection methods. Based on this theory, we used a window of $n$ observations to be considered as a group. With the assumption of having $n \times m$ samples, we have $m$ windows of $w_1, \ldots, w_m$, each of which will be considered as a point in the detection phase. If the window is recognized as an outlier, all the observations in that window will be labeled as attacks; otherwise they will be labeled as normal.

### 3.1. PCA Pre-processing

As explained in the previous section, the main objective of principal component analysis (PCA) is to reduce the dimensionality of the data set but retain most of the original variability in the data. This characteristic makes it possible for distance-based anomaly detection methods to keep their detection rates almost fixed while reducing the number of samples. In other words, although we can use PCA to reduce the dimensions, our method will not lead to good results without having a PCA pre-processing.

### 3.2. Covariance Space

Assume $p$ physical features $y_1, \ldots, y_p$ are provided as a result of PCA pre-processing. Having a window of observations $w_m$, containing $x_1^m, x_2^m, \ldots, x_n^m$, the variance matrix $\Sigma_m$ can be computed using the following formula:

$$
\Sigma_m = \begin{bmatrix}
\sigma_{y_1^m y_1^m} & \sigma_{y_1^m y_2^m} & \cdots & \sigma_{y_1^m y_p^m} \\
\sigma_{y_2^m y_1^m} & \sigma_{y_2^m y_2^m} & \cdots & \sigma_{y_2^m y_p^m} \\
\vdots & \vdots & \ddots & \vdots \\
\sigma_{y_p^m y_1^m} & \sigma_{y_p^m y_2^m} & \cdots & \sigma_{y_p^m y_p^m}
\end{bmatrix}
$$

where

$$
x_i^m = (y_1^m(i), \ldots, y_p^m(i)) \quad 1 \leq i \leq n
$$

$$
\sigma_{y_i^m y_j^m} = \text{cov}(y_i^m, y_j^m)
$$

$$
= \frac{1}{n} \sum_{k=1}^{n} (y_i^m(k) - \mu_{y_i^m})(y_j^m(k) - \mu_{y_j^m})
$$

$$
\mu_{y_i^m} = E[y_i^m] = \frac{1}{n} \sum_{k=1}^{n} y_i^m(k)
$$

Having the covariance matrix constructed, we only consider those covariance values over the matrix diagonal. The reason is that for the covariance values on the diagonal, they are not covariance but variance of one feature which has no information about the correlation of features. In addition, for the values below the diagonal, they are exactly the same as those over the diagonal since covariance operation is symmetric, i.e., $\text{cov}(X, Y) = \text{cov}(Y, X)$.

Omitting all the values on and below the matrix diagonal, we will be left with $q$ features where $q = \frac{m(m-1)}{2}$.

### 3.3. Training Phase

Being categorized as supervised anomaly detection approach, our method has an advantage of requiring only normal samples in the training data set.

Having a training data set of normal data, the $q$ covariance values will be computed for each window $w_1, \ldots, w_m$, and at the end covariance values for each pair of features will be calculated considering the values from all the windows.

$$
\sigma_N^i = \frac{1}{m} \sum_{k=1}^{m} \sigma_{i,k}
$$

where $1 \leq i \leq q$ and $\sigma_{i,k}$ is the $\sigma_i$ for the $k$th window.

### 3.4. Detection Approach

The main idea of our detection approach, called Covariance Matrix Sign (CMS), is to compare the sign of the covariance values of the sample groups $\sigma_{1,i}, \ldots, \sigma_{q,i}$ ($1 \leq i \leq q$) with the values computed in the training period $\sigma_{1,N}, \ldots, \sigma_{q,N}$, and count the number of differences.

If the total number of differences exceeds a threshold, all the samples in that window will be labeled as normal. Symbolically, the detection approach may be represented as follows.

All the samples in window $i$ can be classified as attacks if

$$
\sum_{k=1}^{q} \text{cmp_sign}(\sigma_{k,i}, \sigma_{k,N}^i) > \alpha
$$

otherwise, they are all classified as normal.

where

$$
\text{cmp_sign}(x, y) = \begin{cases} 
0 ; & (x \geq 0 \land y \geq 0) \\
0 ; & (x < 0 \land y < 0) \\
1 ; & \text{otherwise}
\end{cases}
$$

and $\alpha$ is a threshold which can be relaxed to gain different detection and false-positive rates.

### 4. Experiments

In order to have a better evaluation of our method, two other successful approaches are applied to our data set, and the results are compared.
The first one is K-means, which in [21] is shown to have a better result compared to other unsupervised machine learning and datamining techniques. K-means algorithm is one of the best-known squared errors based clustering algorithm [17]. It is usually based on the Euclidean measure and hence it is easy to generate hyper-spherical clusters. The general process of K-means is described as follows:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid distance.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move.

The second anomaly detection technique we use in our paper for comparison is local outlier factor (LOF), which in [13] is shown to have a better performance compared to other density- and distance-base outlier detectors. LOF is a well known density-based outlier detection algorithm [6]. An outlier is something that does not belong to any existing group or class. This algorithm computes the outlierness or outlier factor of every point in the data set based on the nearest neighborhood it has. A point that belongs to a dense or sparse group of points has smaller outlierness then a point that is outside a group. In Figure 1, \( C_1 \) and \( C_2 \) are two classes in the data set and \( o_1 \) and \( o_2 \) are shown as outliers. Because it is a local outlier detector \( o_2 \) is considered as outlier for the reason it is close to \( C_2 \) which is very dense. On the other hand, \( o_1 \) is clearly an outlier based on any group.

![Figure 1. LOF approach mechanism](image)

### 4.1. Applied Data Sets and Performance Measures

The KDD CUP 1999 data set [1] is used for our experiments. This data set is built based on the data captured in DARPA98 IDS evaluation program. Data packets are encoded in a single connection vector which contains 41 features and labeled as either normal or an attack, with exactly one specific attack type. Among the 41 features, 34 are numeric and 7 are symbolic. However, in this study we have used only the 34 numeric features.

To analyze the performance of our method, we prepared two data sets with different relative population of normal and anomalous records. Table 1 shows the statistics of the two prepared data sets.

Data set 1 is a small data set created randomly by selecting a part of the complete KDDCup’99 data set, while the second one (Data Set 2) is the 10 percent KDDCup’99 data set with the last 21 connections omitted to make it a multiple of our selected window size.

<table>
<thead>
<tr>
<th>Number of Connections</th>
<th>Data Set 1</th>
<th>Data Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2L</td>
<td>87</td>
<td>1,126</td>
</tr>
<tr>
<td>U2R</td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>Probe</td>
<td>70</td>
<td>4,107</td>
</tr>
<tr>
<td>DoS</td>
<td>19,907</td>
<td>391,458</td>
</tr>
<tr>
<td>Attacks</td>
<td>20,064</td>
<td>396,493</td>
</tr>
<tr>
<td>Normal</td>
<td>60,536</td>
<td>97,257</td>
</tr>
<tr>
<td>Connections</td>
<td>80,600</td>
<td>494,000</td>
</tr>
</tbody>
</table>

For the evaluation phase, we have used the common way of presenting the classification result called confusion matrix (Table 2).

The standard metrics we have used for comparison are detection rate and false positive rate. Detection rate is computed as the ratio between the number of correctly detected attacks (TP) and the total number of attacks (TP + FN), while false positive rate is computed as the ratio between the number of normal connections that are incorrectly classified as attacks (FP) and the total number of normal connections (FP + TN).

<table>
<thead>
<tr>
<th>Actual</th>
<th>Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack</td>
<td>True Positive (TP)</td>
</tr>
<tr>
<td>Normal</td>
<td>False Positive (FP)</td>
</tr>
</tbody>
</table>

### 4.2. Experimental Results

As mentioned in Section 3, CMS needs a PCA pre-processing to have better results. To run PCA on each data
set, we have considered all the samples to compute the corresponding eigenvalue-eigenvector pairs. Having the pairs, we computed the samples’ coordinates in the principal component (PC) feature space.

In the next step, we randomly selected 300,000 normal connections from the KDDCup’99 data set as the training set. We, then, ran the PCA transform on the training set twice. First, with the eigenvalue-eigenvector pairs obtained from data set 1, and secondly, with the pairs obtained from data set 2.

After transforming the training set to the new PC feature space, we selected groups of 20 samples (window size is 20), and for each group we computed the covariance values for all the binary combinations of features. Having done with all the 15,000 groups of samples, we finally found the mean value for the calculated covariance values.

Since PCA is wildly used as a feature reduction method, we considered three different groups of features. Consequently, comparing the results of three groups together, we can find how much effective the feature reduction is for our detection approach (CMS). The three groups are: all the features (1-34), first twenty features (1-20), and last twelve features (23-34).

Not that these groups of features are not selected randomly, but based on sudden changes in the eigenvalues. In other words, the reason of selecting first twenty features is the existence of a sudden change between the corresponding eigenvalues of feature 20 and feature 21.

Figure 2 illustrates the ROC curves of three different groups of feature for both data sets. Generally speaking using all the features, we may get better detection rates. However, it is apparent from Figure 2 that in data set 2, the 1-20 group outperforms 1-34 group for the false positive rates between 4% and 20%.

As expected, PCA works effectively and the detection rates are so similar considering first twenty features and all the features. However, since our method is not based on the variance of features, it does not perform well by considering only the first 4 or 5 features. On the contrary, distance- and density-based approaches are shown to have very promising results using only the first 4 or 5 features.

To have a better understanding of the performance of the CMS, in Table 3 we have provided the detection rates corresponding to various false positive rates from 0% to 90%. Considering Table 3, it is apparent that CMS has a very promising detection result without having any false positives.

To the best of our knowledge, most of the classification and clustering methods used in anomaly detection, do not have a considerable detection rate without any false positive. Consequently, our approach seems to be a good solution for real-world networks dealing with millions of packets per minute where a false positive rate of 1% may lead to thousands of false alarms.

At the end, we compare our method with K-means and LOF, which are shown to have very hopeful results compared to other anomaly detections. Figure 3 illustrates the ROC curves of CMS, K-means, and LOF methods on both data sets. Considering the results in data set 1, we find that the detection rates are very similar for the false positives more than 3%. However, for false positive rates of less that 1%, CMS outperforms the others. In data set 2, which is a large data set containing huge number of attacks, CMS outperforms the two other methods, while K-means has a very poor performance.

5. Conclusions

Covariance Matrix Sign (CMS) is a new anomaly detection scheme proposed in this paper. In contrast to traditional detection approaches, this method uses the correlation information contained in groups of network traffic samples. The main idea of CMS is to compare the signs in the covariance matrix of a group of sequential samples with the signs in the covariance matrix of the normal data obtained
Table 3. Detection rates of CMS for different false positive rates

<table>
<thead>
<tr>
<th>False Alarm rate</th>
<th>Data Set 1</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-20</td>
<td>1-34</td>
<td>23-34</td>
<td>1-20</td>
<td>1-34</td>
<td>23-34</td>
</tr>
<tr>
<td>0%</td>
<td>12.06%</td>
<td>12.06%</td>
<td>12.06%</td>
<td>29.19%</td>
<td>29.19%</td>
<td>29.15%</td>
</tr>
<tr>
<td>10%</td>
<td>91.63%</td>
<td>93.07%</td>
<td>12.11%</td>
<td>90.39%</td>
<td>53.34%</td>
<td>29.76%</td>
</tr>
<tr>
<td>20%</td>
<td>97.57%</td>
<td>98.08%</td>
<td>12.41%</td>
<td>94.52%</td>
<td>97.70%</td>
<td>33.17%</td>
</tr>
<tr>
<td>30%</td>
<td>98.63%</td>
<td>98.83%</td>
<td>13.87%</td>
<td>99.68%</td>
<td>99.28%</td>
<td>35.69%</td>
</tr>
<tr>
<td>40%</td>
<td>98.91%</td>
<td>99.21%</td>
<td>16.02%</td>
<td>99.83%</td>
<td>99.63%</td>
<td>39.69%</td>
</tr>
<tr>
<td>50%</td>
<td>99.51%</td>
<td>99.76%</td>
<td>99.27%</td>
<td>99.93%</td>
<td>99.90%</td>
<td>83.77%</td>
</tr>
<tr>
<td>60%</td>
<td>99.82%</td>
<td>99.93%</td>
<td>99.41%</td>
<td>99.97%</td>
<td>99.95%</td>
<td>87.95%</td>
</tr>
<tr>
<td>70%</td>
<td>99.89%</td>
<td>99.99%</td>
<td>99.62%</td>
<td>99.98%</td>
<td>99.97%</td>
<td>90.47%</td>
</tr>
<tr>
<td>80%</td>
<td>99.96%</td>
<td>99.99%</td>
<td>99.78%</td>
<td>99.99%</td>
<td>99.98%</td>
<td>94.29%</td>
</tr>
<tr>
<td>90%</td>
<td>99.99%</td>
<td>99.99%</td>
<td>99.95%</td>
<td>99.99%</td>
<td>99.99%</td>
<td>98.83%</td>
</tr>
</tbody>
</table>

Figure 3. ROC curves of CMS, K-means, and LOF
during the training process. If the number of differences exceeds a specified threshold, all the samples in that group will be labeled as attacks; otherwise they will be labeled as normal.

Using a sign comparison instead of complicated classification and clustering methods, CMS has the advantage of less computational complexity compared to other anomaly detection schemes such as K-means and LOF. On the other hand, our experiments show that CMS outperforms K-means and LOF applying on large-scale data sets with a huge amount of anomalies, while having relatively similar results for small data sets with few attacks. Furthermore, our approach seems to have very good detection rates while maintaining the false positive at very low rates. In other words, it is capable of detecting a considerable amount of anomalies without misclassifying any normal sample as an attack.

References