A novel PCA-based Network Anomaly Detection

Christian Callegari, Loris Gazzarrini, Stefano Giordano, Michele Pagano, and Teresa Pepe
Dept. of Information Engineering, University of Pisa, ITALY
E-mail: {firstname.lastname}@iet.unipi.it

Abstract—The increasing number of network attacks causes growing problems for network operators and users. Thus, detecting anomalous traffic is of primary interest in IP networks management.

In this paper we address the problem considering a method based on PCA for detecting network anomalies. In more detail, we present a new technique that extends the state of the art in PCA based anomaly detection. Indeed, by means of the Kullback-Leibler divergence we are able to obtain great improvements with respect to the performance of the “classical” approach. Moreover we also introduce a method for identifying the flows responsible for an anomaly detected at the aggregated level.

The performance analysis, presented in this paper, demonstrates the effectiveness of the proposed method.

I. INTRODUCTION

The increasing number of network attacks causes growing problems for network operators and users. Thus, detecting anomalous traffic is of primary interest in IP networks management. In recent years, Principal Component Analysis (PCA) has emerged as a very promising technique for detecting a wide variety of network anomalies. PCA is a dimensionality-reduction technique that returns a compact representation of a multi-dimensional dataset, by projecting the data onto a lower dimensional subspace. In more detail, it allows the reduction of the dataset dimensionality (number of variables), while retaining most of the original variability in the data. The set of the original data is projected onto new axes, called Principal Components (PCs). Each PC has the property that it points in the direction of maximum variance remaining in the data, given the variance already accounted for in the preceding components.

In this work, we have focused on the development of an anomaly based Network Intrusion Detection System (IDS) based on PCA. The starting point for our work is represented by the work by Lakhina et al. [1], [2], [3]. Indeed, we have taken the main idea of using the PCA to decompose the traffic variations into their normal and anomalous components, thus revealing an anomaly if the anomalous components exceed an appropriate threshold. Nevertheless, our approach introduces several novelties in the method, allowing great improvements in the system performance. First of all we have introduced a novel method for identifying the anomalous flows inside the aggregates, once an anomaly has been detected. To be noted that previous works are only able to detect the anomalous aggregate, without providing any information at the flow level. Moreover, we have applied, together with the entropy, the Kullback-Leibler divergence for detecting anomalous behavior, showing that our choice results in better performance and more stability for the system.

The remainder of this paper is organized as follows: Section II presents some relevant related works. Then Section III provides a detailed description of the implemented system, while in section IV we analyze the experimental results, focusing on the improvements offered by our approach, and finally section V concludes the paper with some final remarks.

II. RELATED WORKS

PCA is the most commonly used technique to analyze high dimensional data structures. Originally applied in the framework of image compression, in the last years it has been widely used in the domain of intrusion detection to solve the problem of high dimension of typical IDS datasets. In more detail, recent papers in networking literature have applied PCA to the problem of traffic anomaly detection with promising initial results [4], [5], [1], [2], and [3].

In [4] the author proposed an anomaly detection scheme where PCA was used as an anomaly detector and was applied to reduce the dimensionality of the data. Lakhina et al. pioneered the use of NetFlow data sets from multiple sites and PCA as the basis for network-wide anomaly detection [1], [2], [3].

In [5] PCA is used to make a separation of the high dimensional space, occupied by a set of network traffic measurements, into disjoint subspaces corresponding to normal and anomalous network conditions.

A recent work by Ringberg et al. provides important insights on the difficulties in tuning network-wide PCA-based detectors in practice [6].

Other notable techniques that employ the principal component analysis methodology include the work done by Wang et al. [7], Bouzida et al. [8] and Wang et al. [9].

As far as the use of the sketch in the anomaly detection field is concerned, there exist several works. As an example, in [10] the authors improve the performance of their own PCA based system, by inserting the use of such structures, while in [11] the authors use the count-min sketch for evaluating the variations in the Hierarchical Heavy Hitters in the network traffic.

III. SYSTEM ARCHITECTURE

In this Section, we present the architecture of the system we have implemented to detect anomalies in the network traffic.

A. System Input

First of all the input data are processed by a module called Data Formatting. Indeed, this module is responsible of reading

In more detail, in our implementation we have in input Netflow data, measuring the traffic gone through a given router over given time-bins (in the following we have \( N \) distinct time-bins). To be noted that data are randomly aggregated by means of sketches that are data-structures used to map a given set of data onto a smaller set [14]. The output of this block is given by \( 4N \) distinct files, each file corresponding to a specific time-bin and a specific aggregate.

### B. Time Series Construction

After the data have been correctly formatted, they are passed as input to the module, responsible for the construction of the time series. Typically the distribution of packet features (packet header fields) observed in flow traces reveals both the presence and the “structure” of a wide range of anomalies. Indeed, traffic anomalies induce changes in the normal distribution. Thus, in each time-bin for each aggregate we have evaluated the histogram as follows:

\[
X^t = \{n^t_i, i = 1, \ldots, N\}
\]

where \( n^t_i \) is the number of bytes transmitted by the \( i \)-th IP address in the time-bin \( t \).

Unfortunately, the histogram is an high-dimensional object quite difficult to handle with low computational resources. For this reason we have tried to concentrate all the information taken by the histogram in a single value, able to hold the most of the useful information, that, in our case, is the “trend” of the distribution.

As already emphasized by previous works [1] [2] [3], the entropy provides a computationally efficient metric for estimating the degree of dispersion or concentration of a distribution. Given the empirical histogram, \( X^t \), we can evaluate the entropy value as follows:

\[
H^t = -\sum_{i=1}^{N} \frac{n^t_i}{S} \log_2 \frac{n^t_i}{S} \quad \text{with} \quad S = \sum_{i=1}^{N} n^t_i
\]

Nevertheless, the entropy is only able to capture the information related to a single time-bin, while from our point of view it would be much more important to capture the difference between packet feature distributions of two adjacent time-bins. For this reason, in this work we have also introduced another metric that is the Kullback-Leibler (K-L) divergence. Thus, given two histogram \( X^t \) (captured in time-bin \( t \)) and \( X^{t-1} \) (captured in time-bin \( t-1 \)), the K-L divergence is defined as follows:

\[
D_{KL}^t = \sum_{i=1}^{N} n^t_i \log \frac{n^t_i}{n^{t-1}_i}
\]

Despite of the method used, this module will output a matrix for each type of aggregation, in which, for all the aggregates, the values of the metric (entropy or K-L divergence) evaluated in each time-bin are reported.

### C. PCs computation

After the time-series have been constructed, they are passed to the module that applies the PCA. As stated in the introduction, typically there is a set of PCs (called dominant PCs) that contributes most of the variance in the original data set.

The idea is to select the dominant PCs and describe the normal behavior only using these ones.

It is worth highlighting that the number of dominant PCs is a very important parameter, and needs to be properly tuned when using PCA as a traffic anomaly detector.

In our approach the set of dominant PCs is selected by means of the scree-plot method. As a result, we separate the PCs into two sets, dominant and negligible PCs, that will be then used to distinguish between normal and anomalous variations in traffic.

![Scree Plot](image)

In more detail, a scree-plot is a plot of the percentage of variance captured by a given PC. Figure 1 reports a scree-plot of a particular data-set used in our study. From the graph we can observe that the first \( r = 6 \) PCs are able to correctly capture the majority of the variance. These PCs are the dominant PCs:

\[
P = (v_1, \ldots, v_r)
\]

The method is based on the assumption that these PCs are sufficient to describe the normal behavior of traffic.

### D. Detection Phase

Once the matrix \( P \) has been constructed, we can partition the space into a normal subspace (\( \hat{S} \)), spanned by the dominant PCs, and an anomalous subspace (\( \tilde{S} \)), spanned by the remaining PCs.

The normal and anomalous components of data can be obtained by projecting the aggregate traffic onto these two subspaces. Thus, the original data, in the time-bin \( t \), \( Y_t \) are decomposed into two parts as follows:

\[
Y_t = \hat{Y}_t + \tilde{Y}_t
\]
where \( \hat{Y}_t \) and \( \hat{Y}_i \) are the projection onto \( \hat{S} \) and \( \hat{S} \), respectively:

\[
\hat{Y}_t = PP^TY_t \tag{6}
\]

\[
\hat{Y}_i = (I - PP^T)Y_i \tag{7}
\]

To be noted that, when anomalous traffic crosses the network, a large change in the anomalous component (\( \hat{Y}_t \)) occurs. Thus, an efficient method to detect traffic anomalies is to compare \( \|\hat{Y}_t\|^2 \) (where \( \|\cdot\|^2 \) is the \( L^2 \) norm) with a given threshold (\( \xi \)). In more detail, if \( \|\hat{Y}_t\|^2 \geq \xi \), the traffic is considered anomalous, and we mark the time-bin (\( t \)) as an anomalous time-bin.

Moreover, since when computing the sketch table, we have used \( d \) different hash functions \( h_j \), we have \( d \) data matrices, \( Y^j \) (one for each function). So, the previously described analysis (performed for each \( Y^j \)) returns \( d \) different responses. Thus, a voting analysis is performed. We evaluate the number of produced alarms and we decide if the time-bin is anomalous or not according to the following rule:

\[
\begin{align*}
\text{anomalous} & \quad \text{if number of alarms} > \frac{d}{2} - 1 \\
\text{normal} & \quad \text{otherwise}
\end{align*}
\]

E. Identification Phase

If an anomaly has been detected, the system performs a new phase called anomaly identification. To be noted that the PCA works on a single time-series, so in the detection phase we are able to identify the time-bin during which traffic is anomalous. But, at this point we do not know the specific network event that has caused the detection. In fact, it is worth noticing that an anomalous time-bin may contain multiple anomalous events.

In this phase we want to identify the specific flow responsible for the revealed anomaly. In more detail, at first, we search the specific traffic aggregation in which the anomaly has occurred. Since an anomaly is detected when \( \|\hat{Y}\|^2 \geq \xi \), the identification method consists of searching the particular aggregate that, if removed from the aforementioned statistics, would bring it under threshold. In this way we identify an element \( y_{\text{no}} \) of the vector \( Y \) that corresponds to a set \( A \) of candidate IP flows responsible for the detected anomaly. For this analysis we can use the information stored inside the \( \{Y^j\}_{j=1}^d \) data matrices that contain \( d \) analysis of the same traffic data using different hash functions. At this point, for each of these matrices we can detect the anomalous time-bin and the anomalous aggregate, so we can identify an element \( \{y_{\text{no}}\}_{j=1}^d \) of each matrix. Each of these elements correspond to an aggregate \( A_j \) of candidate anomalous IP flows. Given that, the responsible IP addresses can be found by simply evaluating the intersection of all these aggregates \( I = \bigcap_{j=1}^d A_j \).

IV. EXPERIMENTAL RESULTS

The proposed system has been tested using a publicly available data-set, composed of traffic traces collected in the Abilene/Internet2 Network [15], that is a hybrid optical and packet network used by the U.S. research and education community.

The used traces consist of the traffic related to nine distinct routers, collected in one week, and are organized into 2016 files, each one containing data about five minutes of traffic (Netflow data). To be noted that the last 11 bits of the IP addresses are anonymized for privacy reasons; nevertheless we have more than 220000 distinct IP addresses.

Since the data provided by the Internet2 project do not have a ground truth file, we are not capable of saying \textit{a priori} if any anomaly is present in the data. For this reason we have partially performed a manual verification of the data, analyzing the traces for which our system reveals the biggest anomalies. Moreover we have synthetically added some anomalies in the data, so as to be able to correctly interpret the offered results, at least partially.

In more detail we have added several anomalous traffic flows of different shapes (e.g., constant rate, increasing/decreasing rate). The anomalies we have added consist of 100 packets (155 anomalous flows in total), and could be associated to a DoS attack.

Before detailing the performance achieved by the system in terms of number of detected anomalies and detection rate, let us analyze the sensitivity of our method to two key parameters, i.e., the dimension of the normal subspace (number of dominant PCs, \( r \)) and the sketch size. It is important to say that the presented performance have been obtained varying the value of the threshold \( \xi \) in a range chosen based on the observation of the values of \( \|Y\|^2 \).

As previously said, for the selection of an appropriate number of PCs we have used the scree-plot method. In Figure 1 we report the scree-plot related to random aggregation for a sketch size \( w \) of 64 and bin-size of 5 minutes. From the plot we can easily notice that most of the data energy is captured by the first five PCs and that after the eighth PC, the contribution of the remaining PCs is less than \( 4\% \). Thus we have decided to perform our analysis with a number of PCs \( r \in [2, 7] \).

Figure 2 shows the detection rate (computed over the synthetically added anomalies) when varying the number of PCs. It is worth noticing that a very low value of \( r \) takes to correctly detect a good number of anomalies, also raising a big number of false alarms, due to the fact that also the “normal” components are considered in the anomalous subspace. Vice-versa, considering a high value of \( r \) takes to a bad detection rate, behavior due to the fact that considering a high number of PCs implies to insert in the normal subspace some anomalous components. Given this, it is evident that \( r \) is an important parameter and it has be chosen so as to obtain a good trade-off between detection rate and false alarm rate.

Concerning the sketch size, it is important to highlight that in our implementation we have used \( d = 8 \) distinct hash functions, which give output in the interval \([0, w - 1]\); so the resulting sketch tables will be \( \mathbb{N}_8 \times w \), where \( w \) is a parameter to set.

The choice of \( w \) is very important, since this parameter determines the number and the composition of the aggregates, significantly influencing the detection rate. For this reason, we
have studied the detection rate achieved by the system when varying the sketch size. Figure 3 shows the results of such analysis, when $r$ has been fixed equal to 5. Even tough the graph shows better performance for $w = 32$, in that case there are too few traffic aggregates taking to a huge number of false alarms. Thus, we have concluded that the best performance are achieved when $w = 64$.

In the following we show the performance achieved by our system, in terms of detection rate (computed over the synthetically added anomalies) and total number of detected anomalies.

Figures 4 and 5 respectively show the detection rate and the number of detected anomalies achieved when using entropy, while Figures 6 and 7 show the same quantities when using K-L divergence. By comparing these couples of graphs it easy to conclude that the performance are quite similar, even though when using K-L divergence there is a stronger dependance on the number of PCs for the detection rate, while the number of detected anomalies decreases faster when varying the value of the threshold.

To be able to perform a more significant comparison and to realize if, in fact, the use of entropy and K-L divergence takes to some improvements, we have evaluated some more parameters. Table I presents the increase in the detection rate offered by the use of the two techniques. In more detail for a given number of PCs, we have fixed the value of the detection rate to two different values and we have evaluated how many
anomalies are revealed with both entropy and K-L divergence and how many only by one of the two. It emerges that, as an example, if we consider to use 5 PCs with a detection rate of 70%, we can see that the 55% of the anomalies are detected by both the systems, while the 15% detected by using the entropy is different from the 15% detected by using K-L divergence, thus in this case the detection rate improves from the 70% to the 85%.

Finally we present the performance achieved in the identification phase. Figure 8 shows the percentage of anomalous flows correctly identified. As it appears clearly, the performance depend on the threshold value. In more detail, it is worth noticing that in correspondence of too low values the identification is not able to correctly work, since it is not possible to find any flow that, if removed, would bring $\|\tilde{g}\|^2$ under the threshold.

V. CONCLUSIONS

In this paper we have presented a novel anomaly detection method, based on the use of PCA. With respect to the “classical” approaches, our system achieves several improvements due to the use of K-L divergence. Moreover our system also goes beyond the state of the art, by introducing a method for identifying the traffic flows responsible for an anomaly detected at the aggregate level.

To assess the validity of the proposed solution, we have tested the system over traffic collected in the Internet2/Abilene network. The performance analysis has highlighted that, for a proper choice of the tuning parameter, the implemented system obtains very good results, detecting all the synthetic anomalies.

REFERENCES