Methods for Anomaly Detection and Classification in Network Traffic

Tingshan Huang

***Abstract*—Network traffic anomaly detection is important for maintaining daily network operations and has been researched for a long time. Many anomaly detection techniques have been developed over the years, some are signature-based, while the others are statistic-based; some are for specific applications while some are generic. In this article, we discuss three typical statistic-based generic techniques for anomaly detection, each with significant improvement compared with earlier work. These three techniques are: anomaly analysis using wavelets, anomaly analysis using *principal component analysis* (PCA) and anomaly detection technique based on *compressive sensing*. For each anomaly detection technique, we identify their assumptions that are used to differentiate normal and abnormal traffic, and present the mechanism of each technique. Furthermore, we discuss the advantages and limitations of each technique based on their performance in anomaly detection, and their further application in anomaly identification and classification. After evaluations of these three techniques, we point out some directions in the field of anomaly detection for further improvement.**

I. Introduction

Traffic anomalies are patterns in traffic data that do not conform to a well defined notion of normal behavior. It can be induced in traffic for a variety of reasons, such as malicious attack, system breakdown, or measurement faults. Most traffic anomalies can be a great threat to network management, examples are measurement anomaly, denial-of-service (DoS) attacks, flash crowds, port scanning and spreading of worms. It is critical for network operators to apply anomaly diagnosis to detect these anomalies quickly and accurately, and to identify the anomalies automatically.

Anomaly diagnosis consists of three parts: anomaly detection, identification and classification. The latter two are also known as root cause analysis. Anomaly detection systems monitor traffic data and send alarm messages whenever an abnormal change of any kind is observed. Root cause analysis tries to classify the anomaly based on features of the anomalous traffic and then quantify the amount of anomalous traffic.

Intuitively, anomaly detection seems no more difficult than performing a comparison: use existing statistical-analysis techniques to compare the measurements with a statistical model of normal behavior, and generate an output if there exist any statistical outliers. However, there are many pitfalls in this intuition. First, it is difficult to get statistical definitions of normal behaviors. Traffic varies drastically all the time, and different network systems show different traffic patterns. Besides, for some systems the normal behavior keeps evolving all the time, thus it always takes a long time to find a normal pattern. Even if we have a model for the normal pattern, we still have difficulty in finding the boundary between normal and abnormal behavior. Furthermore, limitations on available measurements and difficulties in parameter tuning will degrade the performance of our anomaly detection system.

Considering these problems, researchers have developed a lot of techniques for anomaly detection. There are two basic categories: signature-based techniques and statistic-based techniques. Signature-based techniques detect traffic anomalies by looking for a pattern that matches signatures of known anomalies [26], [27], [28], [30]. For example, Moore *et al*. used property of address uniformity for several popular DoS toolkits to diagnose DoS in [26]. Many software systems and toolkits such as Bro [27] and Snort [28] have been developed based on signature-based techniques. However, techniques in this category have the limitation that they could only detect the known anomalies and thus they would compromise network security when some unknown anomalies come into the system since these unknown anomalies would go undetected.

On the other hand, the statistic-based techniques do not require any prior knowledge about the anomalies. Techniques in this category use past traffic histories to derive a model of normal behavior, and look for significant changes in short-term behavior. Therefore, these techniques are also referred to as change detection, and can be effective for both known anomalies and unknown anomalies.

Many methods have been proposed for volume anomaly detection using statistical techniques. Methods in [1], [12], [29], [31] operate on timeseries traffic over a single link, and assume traffic from different links are independent. In this paper, we first discuss the one proposed in [1]: a wavelet-based approach which uses wavelets to distinguish predictable and anomalous traffic volume changes. The wavelet-based approach was among the first to develop volume-based anomaly detection schemes, which treat anomalies as deviations in the overall traffic volume. This anomaly detection technique applies wavelet transform on measurement data and utilizes a deviation score based on the local variance and global variance in the link/flow volume in different time-frequency scales.

Later works try to explore spatial correlations among multiple links in the network, such as [2], [6], [8], [12], [21], [25], [32], [33]. In this paper, we discuss two of them, the PCA-based approach proposed by Lakhina *et al.* in [2] and [6] which explore the deviation in the network-wide traffic volume and feature distributions caused by anomalies. The proposed anomaly detection scheme applies PCA on anomaly-free traffic data and separates the space of network traffic into two subspaces: the normal subspace and the anomalous subspace. The normal subspace captures high variance of normal traffic data and thus models the normal behavior of

a network, whereas projections of measurement data onto the anomalous subspace are used to signal, identify and classify anomalies.

While these methods have been shown to detect anomalies accurately, they only consider spatial correlations while temporal correlations are ignored. One latest work using the spatial-temporal *compressive sensing* (CS) framework is developed by Zhang *et al.* in [3]. This technique uses an algorithm called *sparsity regularized matrix factorization* (SRMF) which focuses on large-scale traffic matrix (TM) data and builds a spatial-temporal model that involves both the spatial and temporal property of the underlying TM. This algorithm method claims that normal component of a TM can be captured by a low-rank matrix. Therefore it tries to find out a low-rank matrix that best estimates the original TM and uses the differences between this low-rank matrix and the original matrix to signal anomalies. We will discuss this method as the third one.

The organization of this paper is as follows: In Section II, we introduce the fundamentals of wavelet analysis and then the wavelet-based method. In Section III, we first show how PCA works, and then present two methods using PCA. One of them uses PCA to diagnose volume anomalies while the other one tries to detect and classify anomalies according to how they change traffic feature distributions. In Section IV, we introduce the algorithm of SRMF and demonstrate how it can be used for anomaly detection. In Section V, we evaluate these three methods in detail, compare their advantages and limitations, and in Section VI we present some other recent work in the field of anomaly detection. We conclude the report in Section VII.

II. Anomaly Detection using Wavelet Transform

Barford *et al.* developed this wavelet-based approach to detect anomalies [1]. This method is based on the observation that most anomalies bring abrupt changes in traffic volume, and it tries to apply wavelet analysis on measured traffic data over a single link, with aims of rapid and correct detection as well as automatic identification of anomalies that would change traffic volume.

In this section, we will first present some fundamentals of wavelet analysis, and then describe the wavelet-based anomaly detection approach developed by Barford *et al.*

A. Fundamentals about Wavelet Analysis

Wavelet analysis is a type of *multiresolution analysis* (MRA) technique that can reveal time-frequency characteristics by applying wavelet transform on the signal. Wavelet transform is performed by convolving the input data with orthonormal series generated by a wavelet, and the products are called wavelet series. A MRA consists of a collection of nested subspace $\{\mathbf{V}_j\}_{j\in\mathcal{Z}}$ that satisfy the following set of properties [7]:

1) $\bigcap_{j\in\mathcal{Z}} \mathbf{V}_j = \{0\}$, $\bigcup_{j\in\mathcal{Z}} \mathbf{V}_j$ is dense in $\mathbf{L}^2(\mathcal{R})$, by saying dense we mean any point in $\mathbf{L}^2(\mathcal{R})$ belongs to $\bigcup_{j\in\mathcal{Z}} \mathbf{V}_j$ or is a limit point in $\bigcup_{j\in\mathcal{Z}} \mathbf{V}_j$.
2) $\mathbf{V}_j \subset \mathbf{V}_{j-1}$ for $j = 1, 2, \cdots$.
3) $x(t) \in \mathbf{V}_j \Longleftrightarrow x(2^j t) \in \mathbf{V}_0$.
4) there exists a *scaling function* $\phi_0(t) \in \mathbf{V}_0$ such that space $\mathbf{V}_j = \text{span}\{\phi_{j,k}(t), k \in \mathcal{Z}\}$, where $\phi_{j,k}(t)$ are scaled and shifted version of $\phi_0(t)$, $\phi_{j,k}(t) = 2^{-j/2}\phi_0(2^{-j}t - k)$.

Since $x(t) \in \mathbf{V}_j \Longleftrightarrow x(2t) \in \mathbf{V}_{j-1}$, projection of signal x onto subspace $\mathbf{V}_j$ represent higher resolution than its projection onto subspace $\mathbf{V}_{j-1}$. The projection of signal x onto approximation subspace $\mathbf{V}_j$ gives us an approximation of x:

$$A_{x,j}(t) = \sum_{k\in\mathcal{Z}} a_x(j,k)\phi_{j,k}(t), \tag{1}$$

where $\{a_x(j,k), k \in \mathcal{Z}\}$ are called *approximation coefficients* of x at level j, and are obtained by convolving the signal with shifted scaling functions at resolution level j:

$$a_x(j,k) = \langle x, \phi_{j,k}\rangle. \tag{2}$$

The MRA theory shows that there exists a function ψ_0 named *mother wavelet* that satisfies zero mean $\int \psi_0(t)dt = 0$ and unit square norm $\int |\psi_0(t)| = 0$. The shifted versions of ψ_0 at jth scaled level $\psi_{j,k}(t) = 2^{-j/2}\psi_0(2^{-j}t - k)$ form the basis for *wavelet subspace* $\mathbf{W}_j = \text{span}\ \{\psi_{j,k}(t), k \in \mathcal{Z}\}$. Wavelet subspaces also have the property of $\mathbf{W}_j \subset \mathbf{W}_{j-1}$ since $x(t) \in \mathbf{W}_j \Longleftrightarrow x(2t) \in \mathbf{W}_{j-1}$. Besides, $\mathbf{W}_j$ is the orthogonal complement of $\mathbf{V}_j$ inside $\mathbf{V}_{j-1}$. In other words, if the projections of signal $x(t)$ onto $\mathbf{W}_j$ are

$$D_{x,j}(t) = \sum_{k\in\mathcal{Z}} d_x(j,k)\psi_{j,k}(t), \tag{3}$$

where $\{d_x(j,k), k \in \mathcal{Z}\}$ are called *detailed coefficients* of signal x at level j and is obtained by convolving the signal with shifted wavelet functions at resolution level j:

$$d_x(j,k) = \langle x, \psi_{j,k}\rangle, \tag{4}$$

then $D_{x,j}(t) = A_{x,j-1}(t) - A_{x,j}(t)$ gives the approximation error at the jth level.

Therefore, signal $x(t)$ can be written as a collection of *details* $d_x(j,k)$ at J levels and *approximations* $a_x(J,k)$ at the Jth resolution level:

$$\begin{aligned} x(t) &= A_{x,0}(t) \\ &= A_{x,1}(t) + D_{x,1}(t) \\ &= A_{x,2}(t) + D_{x,2}(t) + D_{x,1}(t) \\ &= \cdots \\ &= A_{x,J}(t) + \sum_{j=1}^{J} D_{x,j}(t) \\ &= \sum_{k\in\mathcal{Z}} a_x(J,k)\phi_{J,k}(t) + \sum_{j=1}^{J}\sum_{k\in\mathcal{Z}} d_x(j,k)\psi_{j,k}(t). \end{aligned} \tag{5}$$

Performing wavelet analysis on signal $\mathbf{x}$ composes two steps:

1) *Decomposition*: This step produces the approximation coefficients and detailed coefficients at different resolution levels. An example for wavelet transform would be using Haar wavelet, given by:

$$\psi(t) = \begin{cases} 1, & 0 \le t < 1/2 \\ -1, & 1/2 \le t < 1 \\ 0, & \text{otherwise,} \end{cases}$$

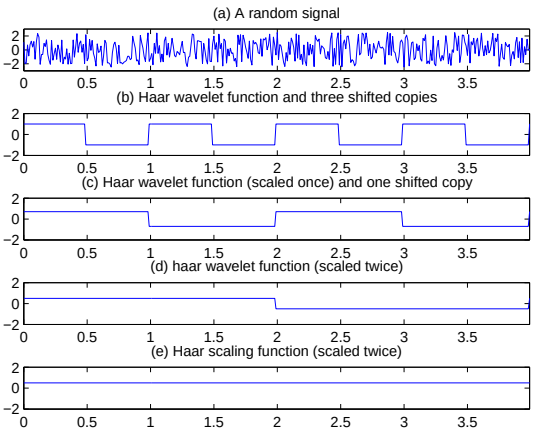

Fig. 1. Example of wavelet transform using the haar wavelet

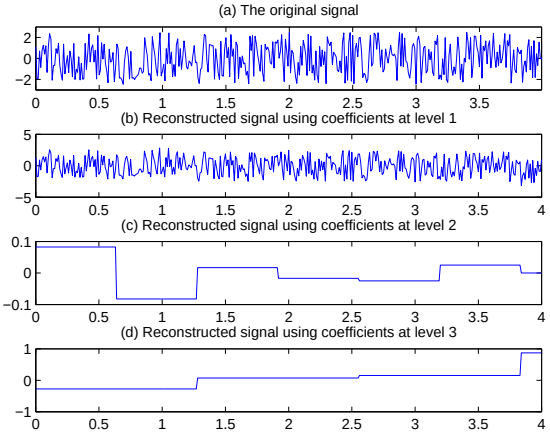

Fig. 2. Signal reconstruction using the haar wavelet

and scaling function, given by:

$$\phi(t) = \begin{cases} 1, & 0 \le t < 1 \\ 0, & \text{otherwise.} \end{cases}$$

As shown in Fig. 1, the wavelet transform is performed by convolving the input signal $\mathbf{x}$, with shifted wavelet functions which are dilated at different levels. At the first level, the original Haar wavelet function is convolved with the input signal, then the wavelet function shifts to the right by distance of one unit and get convolved again, as shown in Fig. 1-(b). Repeat the shifting till the end of the signal and we can obtain the detailed coefficients at the first level, $d_x(1,k), k \in \mathcal{Z}$. Next, the wavelet function get dilated by $\sqrt{2}$, and we'll get the detailed coefficients at the second level by repeating the same operation as in the first level, except that wavelet functions are shifted by two units, as shown in Fig. 1-(c). In this way we can obtain the detailed coefficients at the second level, $d_x(2,k), k \in \mathcal{Z}$. Repeat the dilation and shifting till some specified level J or until the length of dilated wavelet is larger than that of the signal, as shown in Fig. 1-(d), and we can get a full set of detailed coefficients for this signal, $d_x(j,k), j = 1, \cdots, J, k \in \mathcal{Z}$. As for the approximation coefficients $a_x(J,k), k \in \mathcal{Z}$, we simply need to construct the dilated scaling function at level J and convolve it with the signal, as shown in Fig. 1-(e).

As we can see from the above example, coefficients are generated by convolving the input signal with scaled and shifted wavelets. Therefore, coefficients will be large if the convolved part of signal is similar to the corresponding version of wavelet. Also, since wavelet functions at higher levels are more dilated, coefficients at higher levels capture lower frequency properties of signal. In other words, coefficients at higher levels capture the variation of signal $\mathbf{x}$ at lower frequencies, while coefficients at lower level are more likely to capture higher frequency components of $\mathbf{x}$ such as abrupt changes and influence of noise in the signal. Furthermore, because of the shifting, wavelet analysis exposes frequency characteristic of the input signal at different time. This is why wavelet analysis is able to isolate characteristics of signal via a combined time-frequency representation.

In some wavelet systems, for example in the *multi-wavelet system*, we use multiple wavelets in the decomposition process instead of using only one wavelet. The processes for decompositions and reconstructions stay the same, except that the wavelet coefficients at stage j in such systems become $d_{x,i}(j,k), i = 1, \cdots, r, k \in \mathcal{Z}$, where r is the number of wavelets used in the system.

2) *Reconstruction*: This step reconstructs the original signal using Eq. 5, where $d_x(j,k), j = 1, \cdots, J, k \in \mathcal{Z}$ and $a_x(J,k), k \in \mathcal{Z}$ are the wavelet coefficients and approximation coefficients obtained from the decomposition step.

To show signal in different bandwidths, wavelet-based algorithms usually try to reconstruct a new signal using the derived wavelet coefficients of the original signal. This can be accomplished by altering the coefficients corresponding to other bandwidths to zero. For example, to get rid of noise, we can threshold the coefficients at high-frequency levels and suppress those coefficients exceeding the threshold to zero. For another example, using coefficients at each level presents us the the signal in different bandwidths. The reconstructed signal for Fig. 1 is shown in Fig. 2. The design of such algorithms needs careful selection of wavelet(s) suitable for the nature of input signal and required performance.

A good balance between time localization and frequency localization characteristics is given as a criteria for selecting the proper wavelet transform. *Time localization* is measured by the length of filters used in the transform: the shorter the filter, the easier it is for us to observe short-lived changes. *Frequency localization* is measured by the number of *vanishing moments* or equivalently *approximation order*. If some wavelet has m vanishing moments, then its filters are able to calculate the mth order difference. For this reason, larger number of vanishing moments leads to higher accuracy. Therefore,

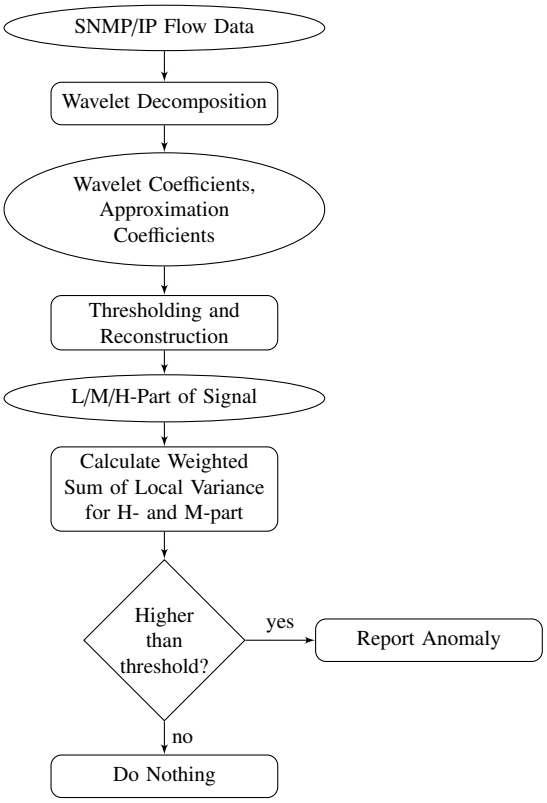

Fig. 3. Wavelet-based anomaly detection process

longer filters have higher number of vanishing moments, thus higher accuracy, and lower false alarm rates.

Another issue for the system design is *artifact freeness*. Artifact is the features in reconstructed signals resulting from the filters used rather than the input signal. We need wavelet filters that do not create undesired artifacts.

B. Diagnosing Volume Anomalies

Data used in [1] consists of link loads from Simple Network Management Protocol (SNMP) and IP flow measurements that are collected over a six month period at the border router of a large university.

The SNMP data was gathered at a five minute sampling interval, and shows the traffic transmitted by the router in bytes and packet counts. Collected data from IP flow monitors show the byte and packet counts for each flow at five minute intervals. Average IP packet size is also included in the data set, since they provide more information for exposing anomalies that use typical packet sizes.

In [1], Barford *et al.* developed an automated method for detecting and identifying irregularities in the data. They used a multiwavelet system. In their chosen system, the approximation order is 4, and there is one low-pass filter L and three high-pass filter H_1, H_2, H_3, with vanishing moments of 2, 3 and 4 respectively. The filters used for decomposition have 7 non-zero coefficients and the filters for reconstruction have 5 non-zero coefficients.

The process of this wavelet-based method is shown in Fig. 3. First, traffic data $\mathbf{x}$ is decomposed for several iterations and wavelet coefficient series at each stage are obtained.

Second, signal $\mathbf{x}$ is reconstructed in three bandwidths using wavelet coefficient series:

1) L-part, low frequency part of the signal. The L-part is obtained by applying the reconstruction process to all the wavelet coefficients at level 9 and higher, and thus can reliably show a high degree of regularity and consistency in the traffic data. It is supposed to capture normal traffic patterns and anomalies of very long duration, in this case several days and longer.
2) M-part, mid frequency part of the signal. The M-part is obtained by applying the reconstruction process to wavelet coefficients at levels 6, 7, 8. The M-part is supposed to capture daily variation in the data, and the amount of its traffic is 3% of the original data.
3) H-part, high frequency part of the signal. The H-part is obtained by first thresholding wavelet coefficients at the first five levels and then applying the reconstruction process to them. Since noise is usually short-lived, method of thresholding is applied on the low-level coefficients to remove noise. The method of thresholding works by going through all the coefficients and comparing them to a given threshold. If it find one coefficient smaller than the threshold, the coefficient is set to zero.

Third, a special term named *deviation score* is calculated based on variations observed in the H and M parts. The H-part and M-part are first normalized to have variance one, then local variability of the H-part and M-part are combined to form the V-part of the signal using a weighted sum. Here, V stands for Variable, and local variability is computed using data within a moving window of a size suitable for anomalies we want to detect.

Fourth, V-part of the signal is used to signal anomaly. As the way they are generated, the L-part shows normal traffic pattern. If any anomaly exists and introduces abrupt changes, we should observe a high variance in the H-part and M-part. Therefore, anomalies can be detected from the V-part of the signal. Using a specified threshold, the authors think there exists an anomaly where the value of V-part exceeds the threshold.

Finally, by measuring the height and width of the peaks shown in the V-part of the signal, we are able to identify anomalies based on their duration and relative intensity.

C. Results

Using a metric of false negative, results show that the wavelet-based method is effective in detecting volume anomalies and also in identifying them into two categories: the short-lived events and the long-lived events.

First, it is found that long-term anomalies such as flash crowds usually get exposed by the mid-band and low-band filters. Especially, if we use extra information of average packet size, the detection result is more accurate. On the other hand, short-term anomalies such as DoS attacks and measurement anomalies are best exposed by mid-band and high-band filters.

Results also show that if there exists a series of short-term anomalies in the same time window, the deviation score can be used to indicate each one of them. However, if there exists a significant anomaly, anomalies of less intensity within the same time window can not be detected. This is because their amplitudes get suppressed by normalization in the first step of deviation score. Anomalies that have duration much longer than the preset window size can not get detected by the deviation score, but can be detected by visual inspection on L-part, M-part and H-part of the signal.

Finally, this method is effective in exposing anomalies even if we use measurements of aggregated traffic data where anomalies are well hidden. Also, the position where we get the measurements does not affect the detection results.

III. PCA-based network-wide traffic analysis

It was first proposed by Lakhina *et al.* to apply subspace method on network-wide traffic measurements for anomaly detection [2]. They claim that the high-dimensional network-wide traffic data can be represented in a low-dimensional subspace that captures most of the variance in the data. PCA is applied on traffic data collected from multiple links, and the high-dimensional space of network traffic measurements are separated into two disjoint subspaces: a low-dimensional normal subspace that captures the normal pattern, and an anomalous subspace that can be used to signal anomalies. Later, traffic feature distribution is explored to identify a wider range of anomalies [6].

In this section, we first present how PCA works for extracting the subspace of low dimensions, then discuss how to use PCA to detect and identify traffic anomalies using two different set of data: one is the set of data for traffic volume, and the other is the set of entropy values for feature histograms.

A. Fundamentals about PCA

PCA is a coordinate transform method that converts a set of observations of possibly related variables into a set of values of unrelated variables. These variables that are independent of each other are called *principal components*, or *principal axes*. When the observations have zero mean, the principal components have the property that the first principal component points in the direction of maximum variance in the data, and each succeeding component has the maximum variance after the variance corresponding to the preceding components have been extracted. Thus, the principal components are ordered by the amount of variance in data that they can capture. The number of principal components is usually smaller than the size of original variables.

Define a $t \times m$ data matrix $\mathbf{X}$

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_t]^T \tag{8}$$

where $\mathbf{x}_i$ is the vector of measurements at time i. Subtracting each element in $\mathbf{X}$ by its column mean, we get a new $t \times m$ matrix $\mathbf{B}$ with zero column mean. Next we apply PCA on $\mathbf{B}$ and get m principal components $\{\mathbf{v}_i\}_{i=1}^{m}$ of $\mathbf{B}$, where $\mathbf{v}_i$ are $m \times 1$ vectors.

The first principal component $\mathbf{v}_1$ points in the direction of maximum variance in $\mathbf{B}$ and is obtained by

$$\mathbf{v}_1 = arg \max_{\|\mathbf{v}\|=1} \| \mathbf{B}\mathbf{v} \|. \tag{9}$$

By subtracting the first $k-1$ principal components from $\mathbf{B}$, the kth principal component can be found by

$$\mathbf{v}_k = arg \max_{\|\mathbf{v}\|=1} \| (\mathbf{B} - \sum_{i=1}^{k-1} \mathbf{B}\mathbf{v}_i\mathbf{v}_i^T)\mathbf{v} \| \tag{10}$$

The normalized projection of data onto principal axis i is

$$\mathbf{u}_i = \frac{\mathbf{B}\mathbf{v}_i}{\| \mathbf{B}\mathbf{v}_i \|}, i = 1, \ldots, m, \tag{11}$$

where $\mathbf{u}_i$ are vectors of size t and are orthogonal to each other.

The value of $\mathbf{u}_i$ is the result of data mapped to the i-th principal component and normalized to unit length. Therefore $\mathbf{u}_i$ can reveal the temporal variation of the whole measurements along the i-th principal component $\mathbf{v}_i$. Since the first principal component captures the maximum variance of the original data, we expect $\mathbf{u}_1$ to have the most significant temporal patterns of all measurements, $\mathbf{u}_2$ to capture the second strongest temporal pattern and so on. In this way, PCA divides the space of link traffic data path into two subspaces according to their variation:

1) the *normal subspace* $\mathbf{S}$ spanned by the first r principal components that capture most of the normal variation in traffic.
2) the *anomalous subspace* $\tilde{\mathbf{S}}$ spanned by all the other anomalous principal components that show anomalous variation if anomaly exists.

A threshold-based separation method is developed in [2] to separate the projections into normal and anomalous sets. It works as follows: examine the projection on each principal component in order from $\mathbf{u}_1$ to $\mathbf{u}_m$ until we find a projection $\mathbf{u}_{r+1}$ that exceeds a given threshold, then principal components $\{\mathbf{u}_j, j = 1, \cdots, r\}$ are assigned to normal subspace, and principal components $\{\mathbf{u}_j, j = r+1, \cdots, m\}$ are assigned to anomalous subspace. The threshold can be set as, for instance, 3σ deviation from the mean.

B. Diagnosing Volume Anomalies Using Link Data

The PCA-based method in [2] tries to diagnose volume anomalies in a backbone network where nodes are connected by m links. *Origin-Destination* (OD) flow is defined as the traffic that enters the backbone at an origin *Point of Presence* (PoP) and exits at a destination PoP. Traffic data is collected network-wide, and byte/packet counts over each link is recorded. It is observed that underlying OD flows of the backbone network have the attribute of low intrinsic dimensionality [8], and their aggregated form of link load data heritages this attribute. A complete system of training, anomaly detection, identification and quantification is built based on these observations.

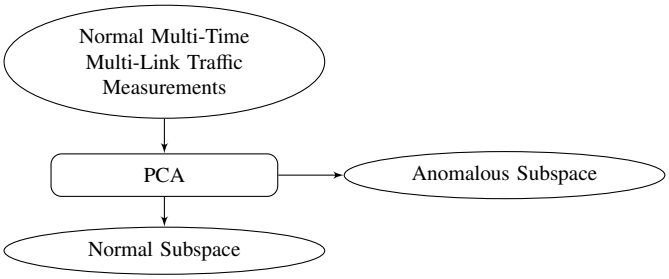

Fig. 4. The training process on normal volume data

1) Training: This process applies PCA on normal traffic data collected from multiple links to construct normal subspace and anomalous subspace.

Define a $t \times m$ link data matrix $\mathbf{Y}$

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_t]^T \tag{12}$$

where $\mathbf{y}_i$ is a vector of size m that records traffic counts on m links at time i.

As is shown in Fig. 4, before detection, we need a set of network traffic data matrix $\mathbf{Y}$ where no anomaly exists, and then subtract each element in $\mathbf{Y}$ by its column mean. If we denote the new $t \times m$ matrix as $\mathbf{B}$ and apply PCA on $\mathbf{B}$ as in the previous section, we can get principal components $\{\mathbf{v}_i\}_{i=1}^m$. By examining projection of normal data on each principal components, we can determine r, the number of principal components in $\mathbf{S}$. Then the linear operator that projects a traffic data vector onto normal subspace $\mathbf{S}$ is an $m \times m$ matrix $\mathbf{C} = \mathbf{P}\mathbf{P}^T$, where $\mathbf{P} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_r]$ is composed of the first r principal components, and the projection onto anomalous subspace $\tilde{\mathbf{S}}$ is also an $m \times m$ matrix $\tilde{\mathbf{C}} = \mathbf{I} - \mathbf{P}\mathbf{P}^T$.

Since $\| \mathbf{B}\mathbf{v}_i \|^2$ is proportional to the captured variance of data along direction of $\mathbf{v}_i$. the subspace method assigns normal traffic variations to the normal subspace $\mathbf{S}$, and volume anomalies can be detected from significant anomalous variations in anomalous subspace $\tilde{\mathbf{S}}$.

By examining $\| \mathbf{B}\mathbf{v}_i \|^2, 1 \leq i \leq m$, it was found in [2] that the majority of variance in link counts for a network of more than 40 links, i.e. $m > 40$, can be well captured by the first 3 or 4 principal components. This result shows that the effective dimensionality of link loads is low, a result that is consistent with the low dimensionality of the underlying OD flows. This is the basis for successful application of subspace methods such as PCA on network-wide link data.

2) Detection: This process examines projection of measurements onto the anomalous subspace and sends a message if an anomaly is found. This process is summarized in Fig. 5.

Given a link traffic vector $\mathbf{y}$ at any instant, first we need to subtract $\mathbf{y}$ by its mean and get a zero-mean vector $\mathbf{b}$. Next, we separate $\mathbf{b}$ into normal and anomalous components by:

$$\mathbf{b} = \hat{\mathbf{b}} + \tilde{\mathbf{b}} \tag{13}$$

where $\hat{\mathbf{b}} = \mathbf{C}\mathbf{b}$ is projection of $\mathbf{b}$ onto the normal subspace and is also referred to as the *modeled* part of $\mathbf{b}$, and $\tilde{\mathbf{b}} = \tilde{\mathbf{C}}\mathbf{b}$ is projection of $\mathbf{b}$ onto the normal subspace and is referred to as the *residual* part of $\mathbf{b}$.

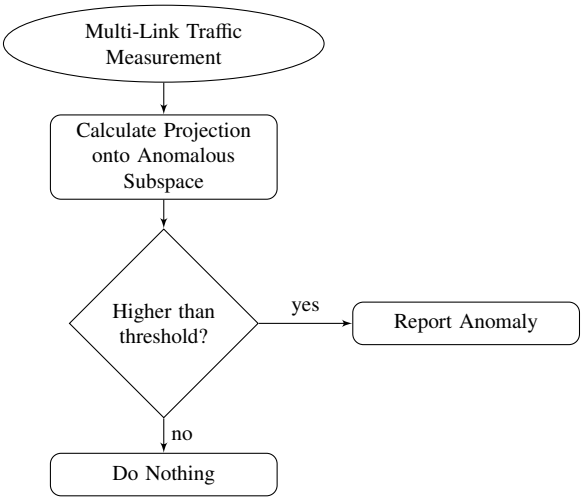

Fig. 5. The PCA-based volume anomaly detection process

As mentioned earlier, volume anomalies generally lead to changes in $\tilde{\mathbf{b}}$. Thus, we can use squared prediction error (SPE) to indicate abnormal changes in $\tilde{\mathbf{b}}$:

$$\text{SPE} \equiv \| \tilde{\mathbf{b}} \|^2 = \| \tilde{\mathbf{C}}\mathbf{b} \|^2, \tag{14}$$

and we consider network traffic to be normal if

$$\text{SPE} \leqslant \delta_\alpha^2, \tag{15}$$

where δ_α^2 is the threshold for SPE at $1 - \alpha$ confidence level. As a result of Q-statistical test for the residual vector $\tilde{\mathbf{b}}$, δ_α^2 is given in [5] as:

$$\delta_\alpha^2 = \phi_1 \left[\frac{c_\alpha \sqrt{2\phi_2 h_0^2}}{\phi_1} + 1 + \frac{\phi_2 h_0 (h_0 - 1)}{\phi_1^2} \right]^{\frac{1}{h_0}} \tag{16}$$

where $h_0 = 1 - \frac{2\phi_1\phi_3}{3\phi_2^2}$, $\phi_i = \sum_{j=r+1}^m \lambda_j^i$, for $i = 1, 2, 3$, $\lambda_j = \| \mathbf{B}\mathbf{v}_j \|^2$ is the variance captured by projecting the data along the j-th principal component, and c_α is the $1 - \alpha$ percentile in a standard normal distribution.

Assuming the sample vector $\mathbf{y}$ follows a multivariate Gaussian distribution, this setting of $1 - \alpha$ confidence limit for the Q-statistic ensures a false alarm rate of α. It is discussed in [5] that this result holds no matter what value r is chosen for the number of principal components in the normal subspace, and the result for Q-statistic does not change much even if the distribution of elements in $\mathbf{y}$ differs substantially from Gaussian.

3) Identification: This process finds out which type of anomaly is responsible for the anomalous volume change we observed in the detection process.

Suppose the network has m links and P OD flows, and assume the set of all potential anomalies is $\{\mathcal{F}_i, i = 1, \cdots, I\}$, with each anomaly $\mathcal{F}_i$ defined by a set of OD flows and an associated matrix Φ_i of size $m \times P$. $\Phi_i(m, n)$ describes the amount of changes in traffic volume over link m when anomaly $\mathcal{F}_i$ happens in OD flow n, and each column of Φ_i has unit

norm. Then, in case of anomaly $\mathcal{F}_k$, the link load can be presented by

$$\mathbf{y} = \mathbf{y}^* + \Phi_k \mathbf{f}_k, \tag{17}$$

where $\mathbf{y}_k^*$ denotes the portion of traffic data in normal conditions, and entries in vector $\mathbf{f}_k$ represent the the intensity of anomaly $\mathbf{f}_k$ in each OD flow. The best estimation of anomaly $\mathcal{F}_k$ should have minimum projection of $\mathbf{y}_i^*$ onto the abnormal subspace $\tilde{\mathbf{S}}$:

$$k = \arg\min_i \min_{\mathbf{f}_i} \| \tilde{\mathbf{C}}(\mathbf{y} - \Phi_i \mathbf{f}_i) \| . \tag{18}$$

4) Quantification: This process estimates the amount of anomalous traffic.

Define *routing matrix* $\mathbf{A}$ of size $m \times P$, where $\mathbf{A}_{ij}$ equals 1 if OD flow j passes over link i and 0 otherwise. Normalize $\mathbf{A}$ so that each column has unit sum: $\hat{\mathbf{A}}_i = \mathbf{A}_i / \sum_i (\mathbf{A}_i)$.

The quantity of anomaly $\mathcal{F}_k$ is estimated as $\hat{\mathbf{A}}_k^T (\mathbf{y}_k - \mathbf{y}_k^*)$, which is proportional to the estimated sum of the additional traffic due to anomaly $\mathcal{F}_k$.

C. Detection and Classification using Traffic Feature Distributions

It was first proposed by Lakhina *et al.* in [6] to use traffic features for anomaly detection. This method is based on the observation that most traffic anomalies result in a change in the distributions of network-wide traffic features, such as source/destination IP address and port numbers in header field of packets. Entropy is used to quantify feature distributions.

1) Traffic Features Affected by Various Anomalies: In Table I we list some common anomalies in network traffic and their impact on traffic features. As is shown, different anomalies have various impact on different features [25]. Based on this observation, Lakhina *et al.* claim in [6] that by examining distributions of traffic features we can detect anomalies and classify anomalies into more detailed categories.

2) The Multiway and Multivariate Data: To detect anomalies that could cause changes in the ensemble of OD flows and its traffic features, the data for detection needs to be multiway and multivariate: the data is multiway in that it spans multiple traffic features, and is multivariate in that it spans multiple OD flows. This feature-based method looks into four traffic features of each OD flow: source IP address (denoted *srcIP*), destination IP address (denoted *dstIP*), source port number (denoted *srcPort*), and destination port number (denoted *dstPort*).

Metric of *sample entropy* is used to quantify the concentration of a distribution. Suppose in S trials, feature i occurs n_i times, $i = 1, \cdots, N$, with $\sum_{i=1}^N n_i = S$. Thus the histogram is $X = \{p_i = \frac{n_i}{N}, i = 1, \cdots, N\}$ and the sample entropy for histogram X is given by

$$H(X) = -\sum_{i=1}^{N} p_i \log_2 p_i. \tag{19}$$

The value of $H(X)$ lies in $[0, \log_2 N]$, and the more concentrated is the histogram X, the larger is the value of $H(X)$. Especially, when X is a uniform distribution and thus least concentrated, $H(X) = 0$.

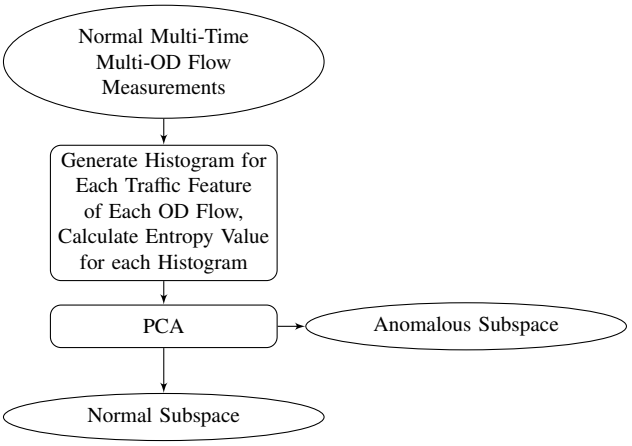

Fig. 6. The training process on normal entropy data

3) Training Process: This process constructs the normal subspace and anomalous subspace using a set of multiway and multivariate anomaly-free data. This process is summarized in Fig. 6.

For each OD flow in a five-minute window, the sample entropy values for four features are calculated. Suppose the data is collected from P OD flows for T time bins, we have four matrices: $\mathbf{H}$(srcIP), $\mathbf{H}$(dstIP), $\mathbf{H}$(srcPort), and $\mathbf{H}$(dstPort). Each matrix is of size $T \times P$, and $\mathbf{H}_{i,j}(*)$ records the entropy value of feature $*$ for OD flow j in the ith time bin. Next, we need to divide each element of each matrix by the energy of that matrix so that each matrix has unit energy.

Combine these four matrices into a $T \times 4P$ matrix $\mathbf{H}$:

$$\mathbf{H} = [\mathbf{H}(\text{srcIP}), \mathbf{H}(\text{dstIP}), \mathbf{H}(\text{srcPort}), \mathbf{H}(\text{dstPort})].$$

Next apply PCA on $\mathbf{H}$ to construct the normal subspace $\mathbf{S}$ and anomaly subspace $\tilde{\mathbf{S}}$. Then construct the linear operator $\mathbf{C}$ that projects a vector onto normal subspace, and the linear operator $\tilde{\mathbf{C}}$ that projects a vector onto anomalous subspace, as described in Section III-A.

4) Detection Process: Given a entropy vector $\mathbf{h}$ of size $4P \times 1$, which records the entropy value of the four features for P OD flows in the same time bin, we can separate it into two parts by

$$\mathbf{h} = \hat{\mathbf{h}} + \tilde{\mathbf{h}}, \tag{20}$$

where $\hat{\mathbf{h}} = \mathbf{C}\mathbf{h}$ is the projection of $\mathbf{h}$ onto normal subspace and $\tilde{\mathbf{h}} = \tilde{\mathbf{C}}\mathbf{h}$ is the projection of $\mathbf{h}$ onto abnormal subspace.

It is claimed in [6] that the point of time for anomalies can be detected by inspecting $\| \tilde{\mathbf{h}} \|^2$ and unusually large values of $\| \tilde{\mathbf{h}} \|^2$ shows anomalous conditions, following Section B. The detection process is summarized in Fig. 7.

5) Identification Process: This process identifies the underlying OD flows that are involved in the detected anomalies.

Construct a $4P \times 4$ matrix Φ_k such that $\Phi_k(k+(m-1)P, m) = 1$ for $m = 1, \cdots, 4$ and all zeros in other positions, then Φ_k specified the components of $\mathbf{h}$ belonging to OD flow k:

$$\mathbf{h} = \mathbf{h}^* + \Phi_k \mathbf{f}_k, \tag{21}$$

where $\mathbf{h}^*$ denotes the normal entropy vector and $\mathbf{f}_k$ is the amount of entropy changes belonging to OD flow k.

TABLE I
QUALITATIVE EFFECTS ON FEATURE DISTRIBUTIONS BY VARIOUS ANOMALIES [6]

Anomaly Label	Definition	Impact on Traffic Feature Distributions
Alpha Flows	Unusually large volume point to point flow	More concentrated in source address and destination address (possibly ports)
DoS	Denial of Service Attack from a single source or distributed sources	More concentrated in destination address, and more concentrated in source address if it's single-source attack
Flash Crowd	Unusual burst of traffic to single destination, from a "typical" distribution of sources	More concentrated in one destination address and destination port
Port Scan	Probes to many destination ports on a small set of destination addresses	More concentrated in a set of destination address, more dispersed in destination port
Network Scan	Probes to many destination addresses on a small set of destination ports	More dispersed in destination address, more concentrated in a set of destination port
Outage Events	Traffic shifts due to equipment failures or maintenance	Mainly source and destination address (No dominant feature changes)
Point to Multipoint	Traffic from single source to many destinations, e.g., content distribution	More concentrated in source address, more dispersed in destination address
Worms	Scanning by worms for vulnerable hosts (special case of Network Scan)	More dispersed destination address and port

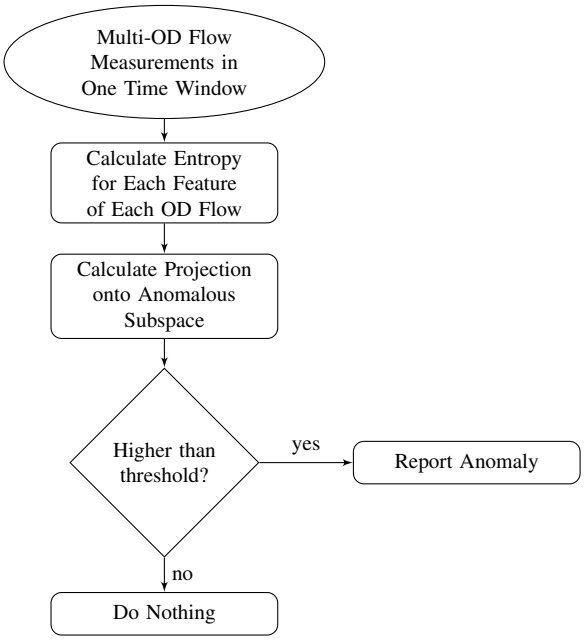

Fig. 7. The PCA-based feature anomaly detection process

The anomalous OD flow is then identified as the one which results in normal component $\mathbf{h}^* = \mathbf{h} - \Phi_k \mathbf{f}_k$ with minimum projection onto $\tilde{\mathbf{S}}$:

$$k^* = \arg\min_k \min_{\mathbf{f}_k} \| \mathbf{h} - \Phi_k \mathbf{f}_k \| \tag{22}$$

To identify more anomalous OD flows, repeat this step recursively on $\mathbf{h}^* = \mathbf{h} - \Phi_{k^*} \mathbf{f}_{k^*}$ until the resulting $\mathbf{h}^*$ falls below a threshold.

6) Unsupervised Classification: In the stage of classification, two clustering approaches are used: k-means algorithm and hierarchical agglomerative algorithm.

k-means algorithm is a representative of partitional algorithms which produce k partitions based on global properties of input data. The k-means algorithm starts with k random cluster centers. In each iteration each data point from the input is assigned to the cluster whose center is the nearest. At the end of each iteration the center-most point of the new cluster is set as the new cluster center. The algorithm ends when no further rearrangement is possible.

The hierarchical agglomerative algorithm is a representative of hierarchical algorithms which produce clusters based on local neighborhood structure. The hierarchical agglomerative algorithm starts with N clusters with each point in the input as the cluster center, where N is the size of the input. This algorithm goes on by clustering together the two points closest to each other. This algorithm ends when k clusters are left.

In these two algorithms, the metric for distances between two data points is the Euclidean distance between their corresponding residual part $\tilde{\mathbf{h}}$. Value of k is decided by maximizing *inter*-cluster variation while minimizing *intra*-cluster variation.

Results from [6] show that using these two simple clustering techniques, each set of labeled anomalies are distributed separately in entropy space because of their specific effects on traffic features. In this way, anomalies can be classified systematically using mining techniques.

D. Results

The PCA-based method separates the high-dimensional space spanned by network traffic into two subspaces, one of which is low-rank and captures the normal traffic pattern. In [2] a general method using PCA is proposed for anomaly diagnosis, and it is shown in [2] that this method is effective in diagnosing volume anomalies with high detection rate and low false alarm rate. Besides, this method is able to identify which anomaly out of a set of potential anomalies is responsible for the observed anomalous volume, and it is also effective in quantifying the amount of traffic involved in the anomalous OD flow.

In [6] entropy is used as the metric for traffic feature distributions, and PCA is applied on these entropy values. It is claimed that anomalies result in changes in distribution for traffic feature such as IP addresses and ports, and we can detect anomalies by observing changes in the entropy values. Results in [6] show that by treating anomalies as events that alter traffic feature distributions, we are able to detect new anomalies. Also, we can have a better understanding of the

structure of anomalies, which would be helpful for anomaly classification.

IV. Spatio-Temporal Detector using Compressive Sensing

In [3] Zhang *et al.* develop a technique called *Sparsity Regularized Matrix Factorization* (SRMF) which exploits the global low rank structure as well as local spatio-temporal structure of the network traffic for anomaly detection.

In this section, we first present the property of sparsity for traffic matrix and the algorithm of *Sparsity Regularized SVD* (SRSVD) for low-rank matrix estimation using compressive sensing theory. Then we present how to use the algorithm SRMF for anomaly detection.

A. Sparsity of Traffic Matrix

Suppose a network is composed of N routers, and each pair of routers is connected by a link. The *traffic matrix* (TM) for such network is defined as an $n \times m$ matrix $\mathbf{X}$ with $n = N^2 \ll m$, where m is the number of measurements and n is the number of links. The jth column $\mathbf{X}_j$ records traffic volume over n OD flows in the jth time window, and more specifically, the $\{i + (j-1) \times N\}$th element of $\mathbf{X}_j$ represents traffic volume of the OD flow entering the network at router i and exiting at router j.

First, we need to present a linear constraint on the TM. A general condition in practical usage for the link measurements $\mathbf{B}$ and TM measurements is given by:

$$\mathcal{A}(\mathbf{X}) = \mathbf{B}, \tag{23}$$

where $\mathcal{A}(\cdot)$ is a linear operator and matrix $\mathbf{B}$ contains the available measurements.

If measurements for TM are available, then $\mathcal{A}(\mathbf{X}) = \mathbf{X}$, $\mathbf{B} = \mathbf{X}$. However, in practice, sometimes TM can not be measured directly, and sometimes TM can not be obtained at some routers. For example, TM could be inferred from routing matrix $\mathbf{A}$ and link load measurements $\mathbf{Y} = \mathbf{AX}$, where $\mathbf{Y}$ is an $l \times m$ matrix with entry $\mathbf{Y}(i, j)$ denoting traffic over the ith link in the jth time bin, and $\mathbf{A}$ is an $l \times n$ matrix with entry $\mathbf{A}(i, j)$ denoting the proportion of OD flow j over link i. In this case, $\mathcal{A}(\mathbf{X}) = \mathbf{AX}, \mathbf{B} = \mathbf{Y}$. On the other hand, even if we are able to measure OD flow volume at routers, there could be missing values at some routers, and thus there could be missing rows in $\mathbf{X}$. In this case, $\mathcal{A}(\mathbf{X}) = \mathbf{B} = \mathbf{M}. \times \mathbf{X}$, and

$$\mathbf{M}(k, t) = \begin{cases} 1, & \text{if } \mathbf{X}(k, t) \text{ is available} \\ 0, & \text{otherwise} \end{cases} \tag{24a}$$

We are interested in how to infer anomalies in TM from observation $\mathbf{B}$ and linear operation $\mathcal{A}(\cdot)$. As noted in [8], the underlying OD flows of a backbone network have low effective dimensionality. Using this inherent structure of the TM, the authors claim that normal component of $\mathbf{X}$ can be captured by a low-rank matrix that best estimates the original TM. This is similar to PCA-based method which tries to find a subspace of low dimension to capture the normal traffic pattern. Next we will present the algorithm named *Sparsity Regularized SVD* (SRSVD) developed in [3] for the low-rank estimation of $\mathbf{X}$.

Recall that the Singular Value Decomposition (SVD) of matrix $\mathbf{X}$ is given by:

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T, \tag{25}$$

where $\mathbf{U}$ is an $n \times n$ matrix and $\mathbf{V}$ is an $m \times m$ matrix, both are unitary matrices. Σ is an $n \times m$ diagonal matrix, and its diagonal entries $\{\sigma_i : \sigma_i > \sigma_{i+1} \geq 0, i = 1, \cdots, m\}$ are singular values of $\mathbf{X}$. Furthermore, we can factorize $\mathbf{X}$ as:

$$\mathbf{X} = \mathbf{LR}^T \tag{26}$$

where $\mathbf{L} = \mathbf{U}\Sigma^{1/2}$ and $\mathbf{R} = \mathbf{V}\Sigma^{1/2}$.

Suppose the rank of $\mathbf{X}$ is r, then r should be much less than n and m, both $\mathbf{L}$ and $\mathbf{R}$ has low rank of r. Therefore, $\mathbf{L}$ can be reduced to be of size $n \times r$ and is composed of the first r columns of $\mathbf{U}\Sigma^{1/2}$, and $\mathbf{R}$ can be reduced to be of size $m \times r$ and is composed of first r columns of $\mathbf{V}\Sigma^{1/2}$.

Given measurements $\mathbf{B}$ and linear constraints on TM $\mathcal{A}(\cdot)$, our problem of estimating $\hat{\mathbf{X}} = \mathbf{LR}^T$ that satisfies Equation(23) is:

$$\begin{aligned} \text{minmize} \quad & \text{rank}(\mathbf{LR}^T) \\ \text{subject to} \quad & \mathcal{A}(\mathbf{LR}^T) = \mathbf{B}, \end{aligned} \tag{27}$$

where matrix $\mathbf{L}$ is of size $n \times r$ and matrix $\mathbf{R}$ is of size $m \times r$.

By compressive sensing theory, if the matrix for $\mathcal{A}(\cdot)$, $\mathbf{A}$, satisfies the s-restricted isometry property, we have

$$\text{minmize} \;\; \text{rank}(\mathbf{LR}^T) \iff \text{minimize} \;\; \| \mathbf{L} \|_F^2 + \| \mathbf{R} \|_F^2,$$

i.e., rank minimization of $\mathbf{L}$ and $\mathbf{R}$ can be performed by minimizing the nuclear norm of $\mathbf{L}$ and $\mathbf{R}$. Here $\mathbf{A}$ is said to have the s-restricted isometry property if for $s < n$, there exists a constant δ_s that for any s-sparse vector $\mathbf{x}$ of size l, we have $(1 - \delta_s)\| \mathbf{x} \|_F^2 \leq \| \mathbf{Ax} \|_F^2 \leq (1 + \delta_s)\| \mathbf{x} \|_F^2$. Also, to allow errors in the measurements, the linear constraints in Equation(23) can be satisfied by

$$\text{minimize} \; \| \mathcal{A}(\mathbf{LR}^T) - \mathbf{B} \|_F^2.$$

The SRSVD approach works as follows: given measurements $\mathbf{B}$ and the matrix for linear constraints on TM $\mathbf{A}$, the low rank component of $\mathbf{X}$ can be found by solving

$$\text{minimize} \quad \| \mathcal{A}(\mathbf{LR}^T) - \mathbf{B} \|_F^2 + \lambda(\| \mathbf{L} \|_F^2 + \| \mathbf{R} \|_F^2) \tag{28}$$

where matrix $\mathbf{L}$ and $\mathbf{R}$ are of size $n \times r$ and $m \times r$ respectively. The first term in Eq. (28) $\| \mathcal{A}(\mathbf{LR}^T) - \mathbf{B} \|_F^2$ is called *fitting error*, λ is a tunable regularization parameter to balance between the fitting error and low rank that can be achieved.

B. Anomaly Detection using SRMF

An algorithm named *Sparsity Regularized Matrix Factorization (SRMF)* is proposed in [3]. SRMF extends SRSVD by exploiting the spatio-temporal structure of traffic matrices. This algorithm tries to estimate a low-rank component of matrix TM $\mathbf{X}^* = \mathbf{LR}^T$ by inferring low-rank matrices $\mathbf{L}$ and $\mathbf{R}$ from:

$$\begin{aligned} \text{minimize} \quad & \| \mathcal{A}(\mathbf{LR}^T) - \mathbf{B} \|_F^2 + \lambda(\| \mathbf{L} \|_F^2 + \| \mathbf{R} \|_F^2) \\ & + \| \mathbf{S}(\mathbf{LR})^T \|_F^2 + \| (\mathbf{LR}^T)\mathbf{T}^T \|_F^2. \end{aligned}$$

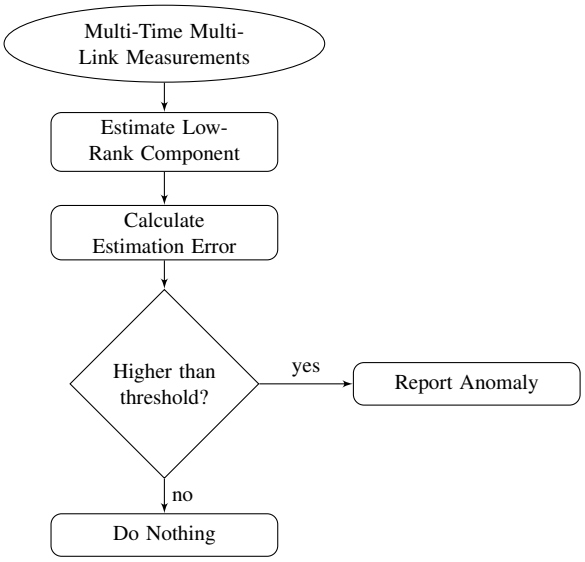

Fig. 8. The SRMF-based anomaly detection process

The added two terms involve a temporal constraint matrix $\mathbf{T}$ of size $(n-1)\times m$ and a spatial constraint matrix $\mathbf{S}$ of size $n\times m$, which shows the correlation of traffic measurements in time and over the links correspondingly. The selections of $\mathbf{S}$ and $\mathbf{T}$ are based on how well they express the spatio-temporal structure of the TM.

In [3], the temporal constraint matrix $\mathbf{T}$ is chosen as a Toeplitz matrix with 1 as the diagonal entries, -1 as the first upper diagonal entries and 0 for all other entries:

$$\begin{bmatrix} 1 & -1 & 0 & \cdots \\ 0 & 1 & -1 & \ddots \\ 0 & 0 & 1 & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}_{(n-1)\times m}, \tag{29}$$

Applying this temporal constraint matrix on a traffic matrix, each element in $\mathbf{X}\mathbf{T}^T$ reveals changes in two measurements consecutive in time. By minimizing $\| \mathbf{X}\mathbf{T}^T \| = \| (\mathbf{L}\mathbf{R}^T)\mathbf{T}^T \|_F^2$, we are minimizing the prediction error at each measurement step.

Since the spatial constant matrix $\mathbf{S}$ should show the spatial correlations between different OD flows, it needs to be built with regards to the measurements. This involves two steps:

1) Obtain an initial TM estimate $\tilde{\mathbf{X}}$. First, we need to use the *Baseline Approximation* to estimate baseline $\mathbf{X}_{base}$ of original TM $\mathbf{X}$. Once we have the baseline estimate, we set each entry of the initial TM estimate $\tilde{\mathbf{X}}(i,j)$ to be the measurement value when the measurement is available, otherwise it takes the value of $\mathbf{X}_{base}(i,j)$. Therefore, $\tilde{\mathbf{X}} = \mathbf{X}_{base}.*(1-\mathbf{M})+\mathbf{D}.*\mathbf{M}$, where $\mathbf{M}$ is defined in Eq. (24) and $\mathbf{D}$ is the direct measurement matrix.
   The baseline estimation $\mathbf{X}_{base} = \overline{\mathbf{X}} + \mathbf{x}_{row}\mathbf{1}^T + \mathbf{1}\mathbf{x}_{col}^T$ is a rank-2 approximation of $\mathbf{X}$, which can be used to estimate matrix mean $\overline{x}$, vector of row mean $\mathbf{x}_{row}$, and vector of column mean $\mathbf{x}_{col}$. Here $\overline{\mathbf{X}}$ is a matrix of size $n\times m$ with $\overline{\mathbf{X}}(i,j) = \overline{x}, 1 \le i \le n, 1 \le j \le m$, and $\overline{x} = \frac{1}{n\times m}\sum_{1\le i\le n, 1\le j\le m}\mathbf{X}(i,j)$ is the estimated mean of all elements in $\mathbf{X}$; $\mathbf{x}_{row}$ is a vector of size n with the i-th entry equal to the estimated mean of elements in the i-th row in $\mathbf{X}$, $\mathbf{x}_{row}(i) = \sum_j(\mathbf{X}(i,j)-\overline{x})/n$; $\mathbf{x}_{col}$ is a vector of size m with the j-th entry equal to the estimated mean of elements in the j-th column in $\mathbf{X}$, $\mathbf{x}_{col}(j) = \sum_i(\mathbf{X}(i,j)-\overline{x})/m$.
   $\overline{\mathbf{X}}$, $\mathbf{x}_{row}$, and $\mathbf{x}_{col}$ can be estimated by:

$$\begin{aligned} \text{minimize} \| \mathcal{A}(\overline{\mathbf{X}} + \mathbf{x}_{row}\mathbf{1}^T + \mathbf{1}\mathbf{x}_{col}^T) - \mathbf{B} \|_F^2 \\ + \lambda(\overline{x}^2 + \| \mathbf{x}_{row} \|_F^2 + \| \mathbf{x}_{col} \|_F^2), \end{aligned}$$

   where λ is a regularization parameter that balances between the fitting error of $\mathcal{A}(\mathbf{X}_{base}) - \mathbf{B}$ and the over-fitting error from the second term.
2) Choose spatial constraint $\mathbf{S}$ based on initial TM estimate $\tilde{\mathbf{X}}$. For the ith row of $\tilde{\mathbf{X}}$, perform linear regression to find the K most similar rows indexed by $\{j_1, \cdots, j_K\} \subseteq \{1, 2, \cdots, n\}\backslash\{i\}$ such that their combination $\sum_{k=1}^K w(k)\tilde{\mathbf{X}}(j_k, *)$ is the best estimate for row vector $\tilde{\mathbf{X}}(i, *)$. Then we set

$$\mathbf{S}(i,j) = \begin{cases} 1, & \text{for } j = i \quad (30a) \\ -w(k), & \text{for } j = j_k \quad (30b) \\ 0, & \text{otherwise} \quad (30c) \end{cases}$$

   In this way, the approximation errors are in form of $\mathbf{S}\tilde{\mathbf{X}}(i,j) = \tilde{\mathbf{X}}(i,j) - \sum_{k=1}^K w(k)\tilde{\mathbf{X}}(j_k, j)$ and should be small for the normal component of TM. Therefore, the matrix spatial $\mathbf{S}$ can work as spatial constraint for the algorithm.

For detection, SRMF does not require anomaly-free data sets for training, but instead can be applied directly to any traffic measurements $\mathbf{X}$ where anomalies could exist. The low rank approximation algorithm recovers low rank matrix $\mathbf{L}$ and low rank matrix $\mathbf{R}$, which are then used to estimate normal component of traffic matrix $\hat{\mathbf{X}} = \mathbf{L}\mathbf{R}^T$. Therefore, the differences between the traffic measurement $\mathbf{X}$ and estimated normal traffic $\hat{\mathbf{X}}$ can reveal the amount of anomalous traffic in the traffic measurements, and we can use $\| \mathbf{X} - \hat{\mathbf{X}} \|_F^2$ to signal anomalies by applying methods such as thresholding. The anomaly detection based on SRMF is outlined in Fig. 8.

C. Results

Using metrics of false alarm rate and detection rate, simulation results are given in [3] which shows the improved performance of SRMF compared to PCA and another simple method that uses *Differencing*.

For small anomalies, the Differencing method shows lower detection rate than PCA and SRMF. This is because both SRMF and PCA exploit the spatial property within the traffic matrix, thus even when the amount of anomalies is small and traffic volume does not change significantly, these anomalies can still be detected by SRMF and PCA. On the other side, Differencing only considers traffic changes in time. For this reason, when the amount of anomalies is small and traffic volume does not change significantly, these anomalies go undetected by Differencing.

For large anomalies, PCA shows the lowest detection rate. This is due to two limitations of PCA. First, PCA does not

consider the correlation of traffic data in time. Second, large anomalies could pollute the data which are used to construct the normal space in the training process, or they could spoof to reside in the normal space during detection process. We will talk about limitations of PCA in more detail in Section V-B. Because of its reliance on spatio-temporal model of traffic matrix, SRMF has the best performance among the three in detecting anomalies.

V. Discussion of the three methods

In this section, we discuss about the three methods that were introduced in the previous three sections. We compare their advantages and disadvantages with each other in terms of their applied techniques, assumptions, deployed properties and so on. A summary is given in Table II.

A. Remarks on Wavelet-based Detection

The wavelet-based method tries to detect traffic volume anomalies using statistical technique of wavelet analysis. Traffic measurements at one router are collected at both link and flow levels and then wavelets are used to decompose the measurements into low-band, mid-band, and high-band parts. Frequency characteristic of anomalous network traffic is studied. It is found that daily component of traffic can be shown in the L-part of the signal, while local variability of the M-part and the H-part can be used to reveal anomalies. Using the deviation score algorithm, this method successfully detects anomalies in aggregated signals and is also able to identify them in terms of duration.

This wavelet-based method shows how to use the statistical techniques to reveal the frequency characteristics of anomalous traffic over single link. However, this method needs further development because of the following concerns:

First and most importantly, this method only looks into the temporal property of traffic data collected at a single node, therefore it could only detect anomalies that result in anomalous traffic volume observed from a single node. In this way, this method would easily ignore anomalies that do not cause significant changes in traffic volume over a single link, for which a good example could be low-rate scanning. In [1], the only metric for its performance is false negative. Without evidence of its performance in terms of false positive, the fact that this method can not detect anomalies that result in low-volume changes is well hidden. Based on this observation, it is not convincing that this method can be effective in detecting all anomalies.

Secondly, in the identification stage, anomalies are roughly divided into long-lived and short-lived anomalies. Anomalies caused by DoS attacks and measurement failures, for instance, can not be distinguished because they have identical characteristics if we diagnose their resultant anomalous traffic using wavelets. We need better identification techniques to divide them into further detailed categories.

Thirdly, even though this detection method claims to be unsupervised and can be portable to different systems, it could perform poorly if they are configured poorly. For example, a good output for this algorithm requires a good selection of wavelets suitable for the nature of input signal and desired performance. The fact that it requires careful parameter tuning to get satisfying detection results makes it less feasible than traditional supervised detection techniques.

Finally, it is not practical to apply this method for online detection. In simulation, the authors of [1] allowed as much as 1.5 hours of discrepancy between the timestamps of anomalies and the timestamps at which the automated methods reported anomaly. For realtime applications, this latency is not tolerable.

B. Remarks on PCA-Based Anomaly Detection

1) Remarks on PCA-Based Volume Anomaly Detection: The PCA-based method is one of the first spatial techniques developed for anomaly detection. PCA is applied by first choosing r dimensions to form the normal subspace that captures most variation of normal traffic data, and then the projection of measurements onto the anomalous subspace is used to signal the existence of anomaly.

There are three significant aspects of this method carried out by Lakhina *et al.* First, it was one of the pioneering works in the area of anomaly detection at network layer. Instead of collecting IP flow traffic over a single link, they use link load data at multiple routers and exploit the spatial property of low-rank in the underlying data for detection. Therefore this method can detect anomalies that span multiple links over the network, a significant improvement compared to traditional methods which only look into traffic measurements observed from a single node.

Second, this method exploits spatial correlation among measurements over all links. Observing that normal network-wide traffic has low-rank, the authors introduced PCA to the field of anomaly detection. In this work, normal traffic pattern is represented by a normal subspace of low rank, and anomalies are revealed by projections onto the anomalous subspace. Simulation results proved its effectiveness in inference of volume anomalies in an Origin-Destination (OD) flow.

Finally, the authors built a complete system for anomaly detection, identification and quantification. Once an anomaly gets detected, a supervised identification process will be applied to diagnose which type of anomaly in a given set best describes the residual part. After this, a routing matrix is used to quantify the intensity of the anomaly.

While this method has been widely used and further developed by researchers afterwards, the PCA-based anomaly detection technique has several limitations. The first four challenges of using PCA are pointed out by Ringberg *et al.* in [18]. Firstly, the performance of applying PCA is very sensitive to how many principal components are chosen to form the normal subspace. Simulation results show that both false alarm and detection rate change much even if the number of components varies in a small range.

Secondly, the performance of PCA is sensitive to the extent to which measurements are aggregated. For comparison, PCA is applied to measurements aggregated at link level, at IP flow level and at OD flow level. It is shown in [18] that using measurements at OD flow level leads to the best tradeoff between

TABLE II
COMPARISON OF DIFFERENT ANOMALY DETECTION TECHNIQUES.

Category	Wavelet-Based	PCA-Based (Volume)	PCA-Based (Entropy)	SRMF-Based
Technique	Wavelet Analysis	Principal Component Analysis (PCA)	PCA	Sparsity Regularized Matrix Factorization (SRMF)
Fundamental Assumption	Normal traffic pattern are revealed in low frequency component	Normal network traffic pattern can be captured by a low-rank subspace	Anomalies are the root cause for changes in feature distributions	Normal component of traffic matrix can be represented by a low-rank matrix satisfying some spatial-temporal constraints
Properties that are Used	Temporal property (time-frequency characteristics) of traffic volume over single link	Partial spatial property (low-rank property) of multi-link traffic	Assumed low-rank property of entropy values for multi-OD flow traffic feature distributions	Complete spatial property and temporal property (low-rank property and spatial-temporal correlation)
Data	Link or flow traffic collected at a single node	Link traffic collected at multiple nodes	Histogram of OD flow traffic features distributions collected at multiple nodes	Link traffic collected at multiple nodes (allow missing data)
Need Training?	No	Yes	Yes	No
Detected Anomalies	Anomalies that introduce observable traffic changes over a single link	Anomalies that introduce observable traffic changes over multiple links	Anomalies that introduce changes in feature distributions for multi-link traffic	Anomalies that introduce observable traffic changes over multiple links or violate spatial-temporal correlations
Identified Content	Anomalous link	Responsible anomaly out of a set of potential anomalies	Anomalous OD flow	Not developed, possibly anomalous link
Classification	Categorized into short-lived and long-lived classes	Supervised classification: categorized into known anomaly classes	Unsupervised classification: categorized according to how they change the feature distributions	Not developed
Computational Complexity	$O(m)$ for m measurements over a single link	$O(m\times n^2)$ for m measurements over n links	$O(m\times n^2)$ for m measurements of n OD flows	$O((n+m)\times r^2)$ for m measurements over n links with r effective dimensions

detection rate and false alarm probability among the three levels of aggregation. However, measurements from OD flows are difficult to collect. Since the number of measurements m should be at least the number of OD flows $n = N^2$ in the network, the collection of measurements at OD flow level will be too large even for a moderate-sized network. This makes PCA-based method inappropriate for online detection.

Thirdly, a large anomaly may pollute the normal subspace produced by PCA. This is because when an anomaly is sufficiently large, it is possible that most of its variance resides in the normal subspace while its projection onto the anomalous subspace is comparatively small [18]. When this happens, this large anomaly gets undetected while a simple Differencing method can easily detect it. One solution to avoid such a "stealth poisoning attack" could be filtering out large anomalies before applying PCA, however, this will lead to missing value for some measurements and thus requires interpolation that cannot be solved by PCA. A good news is that we have SRMF for TM interpolation. We may choose to first filter measurements, then use SRMF for interpolation, and at last apply PCA, or we may simply apply SRMF on our measurements for anomaly detection. We will discuss this issue again in Section V-B2.

Finally, PCA cannot identify which flow contributes to the anomaly that gets detected. The effectiveness of anomaly identification in [2] relies on the assumption that we are aware of the routing matrix associated with each potential anomaly. However, in practice this assumption does not always hold, since we can never have a complete description of all potential anomalies. Besides, PCA does not provide a direct mapping between the subspace spanned by residual part and the original flow. Therefore, we are unable to figure out the original spatial location of the anomaly.

Later in [19], the reasons for all those limitations of PCA were discussed. It was found that to get an effective result, the data set on which PCA is applied should be values of linear random variables with sufficient statistics of their means and covariance, an example of which could be joint Gaussian variables. Measurements from the real network are complicated and may not necessarily follow such a distribution. Therefore, PCA is limited by its assumption that "the vector of link loads follows a multivariate Gaussian distribution". It was suggested in [19] to integrate temporal correlations to improve the performance of PCA.

Besides, PCA has high computational complexity because of its reliance on SVD. Consider an $n\times m$ traffic matrix recording m measurements of n links in a network. Since we always need measurements of size larger than the number of links in the network, we have $m > n$. Applying PCA on this matrix has computational complexity of $\min\{O(mn^2), O(nm^2)\} = O(mn^2)$. This high processing delay makes PCA not suitable for online detection in a large network.

2) Remarks on PCA-Based Feature Anomaly Detection: Based on the intuition that anomalies are the root cause of changes in traffic feature distribution, the anomaly diagnosis in [6] explores the properties of network-wide anomalies systematically in how they change traffic features, and shows strong capability in anomaly detection and classification.

This method goes beyond traditional anomaly detection techniques which look only at volume deviations, but instead looks into the altered distributions of traffic features, and thus can detect a broader range of anomalies. Entropy is used to capture distributional changes in traffic features, and

aids in exposing traffic behavior. Compared with previous methods that focus on volume anomalies, this method has three significant improvements.

First, this method is able to detect network-wide anomalies that span multiple flows and constitute low percentage of a single flow. These anomalies are difficult to detect by observing volume deviations only, since different anomalies may result in the same pattern of changes in traffic volume. On the other hand, unusual distributions can provide us valuable information about the structure of each anomaly [25]. It was shown in [6] that port scans, network scans or point-to-multipoint transfers which are buried in the aggregated traffic can be detected only when using entropy. This is because the multiway subspace method explores the properties of these anomalies over different links, rather than traffic volume over a single link. Anomalies of low-rate port scanning, for example, result in more dispersed distributions for destination port, and more concentrated distributions for destination IP address. Using this property, the entropy-based method is able to detect the existence of port anomalies, while the volume measurements of byte/packet counts over the links during the port scan attack will not show obvious changes. An empirical evaluation of entropy-based traffic anomaly detection is given in [20], where the detection power of using entropy-based analysis of multiple traffic distributions in conjunction with each other is illustrated with simulation results.

The second improvement of this method enables us to use the distributional structure of anomalies for automatic classification. By mining the entropy of traffic feature distributions, one can classify anomalies into distinct categories based on their various impact on network traffic. Traditional classifications, on the other hand, are based on volume changes only and thus could only classify anomalies into rough categories based on their features in time or frequencies.

The third improvement is that this method is less sensitive to packet sampling than the detection via volume [23]. We will discuss this in Section VI.

Although the entropy-based anomaly detection technique is able to detect a wider range of anomalies and shows low false alarm, this method has several drawbacks we should be concerned about.

First, this method does not provide a solid proof for low dimensionality in the entropy space. While it has been shown in [2] that the network traffic is low dimensional, there was no evidence in [6] showing that the space spanned by entropy for different traffic features also has low dimensionality. Since the low dimensionality in the entropy space is the basis for the correctness of applying subspace methods such as PCA, its validity may be limited.

Also, we should note that some anomalies do not bring an observable change to the entropy of traffic features even if they do change the traffic volume. The fact is, two completely different distributions can have the same entropy value, therefore sometimes the metric of entropy alone cannot identify significant differences between two distributions. For example, limited-scope host scanning attack will lead to an overall increase in the traffic to the victims, but will not trigger any notable change in entropy values of any feature. Therefore, although this feature-based method can detect a wider range of anomalies in the network-wide traffic compared with volume-based method, there are still some anomalies it fails to detect. Actually it was pointed out in [6] that "volume measures and entropy complement each other in detecting anomalies". A new approach is proposed in [21] as an extension to the entropy-based anomaly detection technique. In this approach, it is claimed that using entropy to quantify histograms will bury a lot of information revealed by a histogram. Therefore, this approach constructs detailed histogram models for different features, models histogram patterns and identifies deviations from the created models. Results in [21] show the effectiveness of this approach.

Finally, this method is not appropriate for online detection and classification. The action of looking into the header field of each packet and calculating the histogram for each time window results in high processing delay even if we use random packet sampling. Before detection, we always need to first generate the histogram for a time window and then calculate the entropy value, therefore the delay would be at least the window size, which is 5 minutes in [6]. This amount of delay makes it impractical to apply entropy-based technique for online detection. Also, online classification needs to be further developed as mentioned in [6].

C. Remarks on SRMF-Based Anomaly Detection

The algorithm of SRMF exploits both global structure and local structure of network-wide traffic, where the global structure of network-wide traffic refers to low-rank property of the underlying TM, and the local structure refers to the correlation of traffic in time over a single link and spatial correlation of the traffic over all links. Based on these properties, SRMF tries to find a low-rank matrix that best estimates the original TM and also has the spatial-temporal properties of the real TM. It is claimed in [4] that this low-rank estimation shows the normal component of the real traffic, and we can use squared error between estimated TM and real TM to signal anomalies.

First of all, compared with methods based on wavelet and PCA, the algorithm of SRMF has one significant improvement in that it not only exploits both the temporal correlation and spatial correlation but also the low-rank property of TM in the estimation process.

As discussed before, the wavelet-based method considers only the temporal correlation since it only observes how traffic volume over a single link changes in time. For this reason, the wavelet-based method does not work in case of anomalies such as point-to-point scans that involve small amount of traffic. For the same reason, the wavelet-based method cannot detect anomalies such as port scans and network scans that span several links and result in insignificant changes of traffic over a single link.

As for the PCA-based method, its detection result is based on the low-rank property of TM only, while correlation among measurements over time and spatial correlation over links are ignored in the detection process. Although it is claimed in [6] that spatial property of the underlying traffic is exploited in its detection process, it is worthwhile to point out that

concept of spatial correlation for PCA-based method and that for SRMF-based method are different. For PCA-based method, the spatial correlation is considered to be the root reason for low-rank property of the network-wide traffic, therefore the low-rank property is considered to be the spatial correlation. For SRMF-based method, however, the low-rank property is considered to be the global structure of the TM, while the spatial correlation defined for SRMF is truly the correlations among all the links in the network. A spatial constraint matrix is designed to represent this correlation in Eq. (30). From this clarification, we can see the superiority of SRMF-based method over the PCA-based method in that SRMF exploits extra property of spatial correlation over all links, as well as temporal correlation in measurements.

The second improvement is that SRMF does not require training process using anomaly-free data sets. As mentioned in Section III, the PCA-based method needs to first produce the normal traffic pattern using anomaly-free traffic data in the "training process", and then use traffic measurements with anomalies for detection. In practice, anomaly-free traffic data rarely exists. For this reason, anomalies in the training data can easily pollute the data in the training process and create a wrong normal traffic pattern, consequently degrading the detection performance. The SRMF algorithm, however, does not require a learning process and can be applied directly to any set of measurements. This is because this algorithm uses the global low-rank structure and the spatial-temporal correlation to estimate the normal traffic component. When a large anomaly dominates in certain dimension and spoofs to be part of normal traffic, it may succeed in PCA-based method [18], but it will fail in the SRMF-based method for its violation of spatial correlation and/or temporal correlation.

The third improvement is that the SRMF-based method works even when there are a significant number of missing values in the TM measurements. Recall that for linear constraints on TM, we deal with missing values by setting $\mathcal{A}(\mathbf{X}) = \mathbf{B} = \mathbf{M}. \times \mathbf{X}$, where $\mathbf{M}(k,t) = 1$ if measurement for OD flow k in time bin t is available, and setting $\mathbf{M}(k,t) = 0$ otherwise. As is mentioned in [4], one way for the PCA-based method to avoid large anomaly pollution is to first filter large anomalies before applying PCA. However, this will lead to missing values, and will require SRMF-based method for further detection.

The fourth improvement of SRMF is that it requires less computational time compared with PCA. Since both methods are based on SVD, the computational complexity is related to the size of matrices each method deals with. For the same m measurements for a network composed of n links and $m > n$, the computational complexity of PCA is $O(m \times n^2)$, while the computational complexity of SRMF is $O((n+m) \times r^2)$, where r is the effective dimensionality of the traffic matrix. As we mentioned in Section IV, because of the low dimensionality attribute we have $r \ll n$ and thus the computational complexity of SRMF is far less than that of PCA.

However, we should note that the SRMF-based method has several limitations. First, this method requires too much processing time and is not eligible for online detection. This method requires measurements in several time bins before it is able to tell whether there exists any anomaly during the corresponding period, therefore the processing delay for SRMF-based method is at least $m \times$ (length of a single time bin). Besides, this method can not tell in which time bin the detected anomalies actually happen. Because of this, the SRMF-based method does not work for anomaly classification. Furthermore, it is the volume anomalies that the SRMF-based method tries to detect. Therefore, it can not detect anomalies that compose small amounts of traffic.

Another defect of SRMF-based method is that it is usually hard to pick a general threshold that can perform well for all inputs. Since SRMF works with measurements input and the low-rank estimation, the threshold for anomaly detection will be input-dependent, and will vary for different systems at different times. In [3], the SRMF-based method are performed on simulated data, rather than real measurements, which enables the authors to choose a threshold that leads to good simulation result. If applied on the real data, SRMF is expected to perform less effectively.

Finally, for the temporal constraint matrix $\mathbf{T}$, the authors use a simple Toeplitz matrix to reveal changes in two measurements consecutive in time. Although this choice has already led to good results, we expect a more careful design of T for further improvement.

VI. Current Practice and Future Trends

A lot of recent work has focused on detection techniques aimed at special but common anomalies. For example, in [9] a method is proposed to detect peer-to-peer (P2P) voice over IP (VoIP) calls hidden in web traffic. For another example, authors of [10] work on IP level and try to find out the entry and exit point or the path of a Distributed DoS (DDoS) attack, while in [11] another approach is proposed to detect and filter DDoS attacks at the application level. Another technique is proposed in [15] which tries to detect anomalies in Internet traffic based on large deviation results for empirical measures. It was shown to be able to identify temporal and spatial anomalies in short time-scales. We expect many more new techniques specified for one kind of anomaly to come. However, for network monitors to practically detect and analyze anomalies, we will need a general technique which is able to signal both common and uncommon, known and unknown anomalies.

We also find tremendous work on distributed algorithms for anomaly detection. Observing that most detection techniques have processing delay that is linear to the size of measurements, these algorithms try to solve their scalability problems so that they can be used for online detection. In [12], a sketch-based change detection technique is presented which uses sketch for data structure and implements multiple time series forecast models for detection, however, this techniques still works offline. In [13], Kumar presents a distributed framework for cybersecurity using data mining, however, no details of simulation results are given in his work. A distributed algorithm is proposed in [14] where spatial anomaly detections are performed locally at each router and only adjacent neighbors exchange their detection results. In [16], another sketch-based algorithm is proposed which performs PCA for spatial

anomaly detection in a distributed manner. This algorithm is proved to require only logarithmic running time and space, and is more efficient than traditional PCA-based method. In [17], a system named "Monalytics" is proposed and implemented which combines continuous anomaly monitoring and analytic for large-scale networks in an online fashion. We expect more non-intrusive distributed algorithms with low computational complexity to be developed in the near future.

We also find some work investigating effect of sampling on anomaly detection in [22], [23], and [24]. In practice, sampling is usually used to reduce measurement overhead in terms of router CPU, memory and bandwidth. [23] applies four sampling methods (random packet/flow sampling, smart sampling, and sample-and-hold) for wavelet-based method and other methods, and shows that general sampling methods will degrade the performance of detection. Authors of [23] point out that entropy summarizations are more resilient to sampling than to volume metrics, but the performance is still not perfect. Although the impact of sampling on detection has been investigated for several years, we did not find any solution except in one recently published work [24], where the technique of security-aware packet sampling is developed to sample a larger fraction of malicious traffic. We expect many more sampling techniques to be developed for effective anomaly detection.

VII. Conclusion

In this article, we discussed three methods for anomaly analysis: wavelet-based method, PCA-based method and SRMF-based method. The wavelet-based method detects volume anomalies by considering only the temporal property of traffic collected at a single node. The PCA-based method works at the network level and exploits the low-rank property of traffic over multiple links in a network. Results for application of PCA on traffic volume and entropy metric are shown to complement each other for anomaly detection. The SRMF-based method also works at network-level and exploits the low-rank property, and it is superior to PCA in that it adds spatial-temporal property for TM estimation and can work when there are large anomalies or when there are missing values in the measurements. By comparing these three methods, we see the power of combining temporal properties and spatial properties for anomaly detection. In the near future, we expect further improvement of SRMF for anomaly classification. We also expect many more smart algorithms for sampling, as well as many more distributed algorithms for online anomaly analysis.

References

[1] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," *Proc. ACM SIGCOMM Internet Measurement Workshop*, Nov. 2002, pp. 71–82.

[2] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *ACM SIGCOMM*, Aug. 2004.

[3] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," *ACM SIGCOMM*, Aug. 2009.

[4] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for traffic anomaly detection," *Proc. of ACM SIGMETRICS*, 2007.

[5] J. E. Jackson and G. S. Mudholkar, "Control procedures for residuals associated with principal component analysis," *Technometrics*, 1979, pp. 341–349.

[6] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," *ACM SIGCOMM*, 2005, pp. 217–228.

[7] I. Daubechies, "Ten lectures on wavelets," *Society for Industrial and Applied Mathematics*, 1992.

[8] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," *ACM SIGMETRICS*, 2004.

[9] E. P. Freire, A. Ziviani, and R. M. Salles, "Detecting VoIP calls hidden in web traffic," *IEEE Transactions on Network and Service Management*, vol. 5, no. 4, pp. 204–214, Dec. 2008.

[10] V. L. L. Thing, M. Sloman, and N. Dulay, "Locating network domain entry and exit point/path for DDos attack traffic", *IEEE Transactions on Network and Service Management*, vol. 6, no. 3, pp. 163–174, Sept. 2009.

[11] Y. Xie, and S.-Z. Yu, "A large-scale hidden semi-markov model for anomaly detection on user browsing behaviors," *IEEE/ACM Transactions on Networking*, vol. 17, no. 1, pp. 54–65, Feb. 2009.

[12] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, "Sketch-based change detection: methods, evaluation, and applications," *Internet Measurement Conference*, 2003.

[13] V. Kumar, "Parallel and distributed computing for cybersecurity," *IEEE Distributed Systems Online*, Oct. 2005.

[14] P. Chhabra, C. Scott, E. D. Kolaczyk, and M. Crovella, "Distributed spatial anomaly detection," *IEEE INFOCOM*, pp. 1705–1713, 2008.

[15] I. C. Paschalidis, and G. Smaragdakis, "Spatio-temporal network anomaly detection by assessing deviations of empirical measures," *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, June, 2009.

[16] Y. Liu, L. Zhang, and Y. Guan, "Sketch-based streaming PCA algorithm for network-wide traffic anomaly detection," *International Conference on Distributed Computing Systems*, 2010.

[17] M. Kutare, G. Eisenhauer, C. Wang, K. Schwan, V. Talwar, and M. Wolf, "Monalytics: Online monitoring and analytics for managing large scale data centers," *ICAC*, June 2010.

[18] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for traffic anomaly detection," *ACM SIGMETRICS*, June, 2007.

[19] D. Brauckhoff, K. Salamatian, and M. May, "Applying PCA for traffic anomaly detection: problems and solutions," *IEEE INFOCOM*, 2009.

[20] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang, "An empirical evaluation of entropy-based traffic anomaly detection," *Internet Measurement Conference*, 2008.

[21] A. Kind, M. P. Stoecklin, and X. Dimitropoulos, "Histogram-based traffic anomaly detection," *IEEE Transactions on Network Service Management*, vol. 6, no. 2, pp. 110–121, June, 2009.

[22] D. Brauckhoff, B. Tellenbach, and A. Laknina, "Impact of packet sampling on anomaly detection metrics," *Internet Measurement Conference*, 2006.

[23] J. Mai, C.-N. Chuah, A. Sridharan, T. Ye, and H. Zang, "Is sampled data sufficient for anomaly detection?", *Internet Measurement Conference*, 2006.

[24] S. Ali, I. U. Haq, S. Rizvi, N. Rasheed, U. Sarfraz, S. A. Khayam, and F. Mirza, "On mitigating sampling-induced accuracy loss in traffic anomaly detection systems," *ACM SIGCOMM CCR*, vol. 40, no. 3, pp. 5-16, July, 2010.

[25] A. Lakhina, M. Crovella, and C. Diot, "Characterization of network-wide anomalies in traffic flows," *Internet Measurement Conference*, 2004.

[26] D. Moore, G. Voelker, and S. Savage, "Inferring internet denial of service activity," in *Proc. of the USENIX Security Symposium*, Washington D.C., August 2001.

[27] V. Paxson, "Bro: A system for detecting network intruders in real-yime," *Computer Networks*, pp. 2435–2463, 1999.

[28] M. Roesch, "Snort: Lightweight intrusion detection for networks," in *LISA: Proc. 13th USENIX Conf. System Administration*, Seattle, WA, Nov. 1999, pp. 229–238.

[29] J. Brutlag, "Aberrant behavior detection in timeseries for network monitoring," in *USENIX Fourteenth Systems Administration Conference (LISA)*, 2000.

[30] F. Feather, D. Siewiorek, and R. Maxion, "Fault detection in an ethernet network using anomaly signature matching," in *ACM SIGCOMM*, 1993.

[31] M. Roughan, T. Griffin, M. Mao, A. Greenberg, and B. Freeman, "Combining routing and traffic data for detection of IP forwarding anomalies (Poster)," in *ACM SIGMETRICS*, 2004.

[32] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan, "Network anomography," in *Proc. ACM SIGCOMM Internet Measurement Conf.*, Oct. 2005.

[33] M. Thottan and C. Ji, "Anomaly detection in IP networks," *IEEE Trans. Signal Processing*, vol. 51, no. 8, pp. 2191–2204, Aug. 2003.