Abstract

Internet has become an essential part of the daily life for billions of users worldwide, who are using a large variety of network services and applications everyday. However, there have been serious security problems and network failures that are hard to resolve, for example, Botnet attacks, polymorphic worm/virus spreading, DDoS, and flash crowds. To address many of these problems, we need to have a network-wide view of the traffic dynamics, and more importantly, be able to detect traffic anomaly in a timely manner. Existing network measurement and monitoring solutions often suffer scalability problems caused by overly large processing, space, or communication overhead. In this paper, we propose to develop sketch-based algorithms for network-wide anomaly detection that are able to detect both high-profile and coordinated low-profile traffic anomalies as an outlier in the regular traffic patterns. Our approach is based on the spatial analysis by using traffic measurements from multiple monitors. Spatial analysis methods have been proved to be effective in detecting network-wide traffic anomalies that are not detectable at a single monitor. To our knowledge, Principle Component Analysis (PCA) is the best-known spatial detection method for the coordinated low-profile traffic anomalies. However, existing PCA-based solutions have scalability problems in that they require $O(m^2 n)$ running time and $O(mn)$ space to analyze traffic measurements from $m$ aggregated traffic flows within a sliding window of the length $n$, which makes it often infeasible to be deployed for monitoring large-scale high-speed networks. We propose two novel sketch-based algorithms for PCA-based traffic anomaly detection in a distributed fashion. Our algorithm can archive $O(w \log n)$ running time and $O(w \log^2 n)$ space at local monitors, and $O(m^2 \log n)$ running time and $O(m \log n)$ space at Network Operation Center, where $w$ denotes the maximum number of traffic flows at a single local monitor. Additionally, our algorithm can protect the privacy of traffic measurements for Internet Service Providers.
Sketch-based Network-wide Traffic Anomaly Detection

1 INTRODUCTION

The Internet has become an essential part of the daily life for billions of users worldwide. People are using and relying on a large variety of services built on the top of the Internet, such as web browsing, online banking, shopping, entertainment, VoIP, Video on demand, auction, social networks, etc. However, everyday we are still reading news stories about major security breaches, new polymorphic worm/virus spreading, identity theft, Botnet activity, DDoS or phishing emails. To address many of these problems (e.g. DDoS, Botnet, worm/virus, etc.), we need to have a network-wide view of the traffic dynamics, and more importantly, be able to detect the traffic anomaly in a timely manner. Otherwise, the failure of doing so may cause catastrophic damages or unwanted results with impacts affecting online business, public safety, homeland security, personal privacy, the economy and the society at large.

Traffic anomalies can occur due to a variety of problems. Firstly, security threats like DDoS, worms, and Botnets, can generate extremely large-volume anomalous traffic. Secondly, unusual events can cause traffic anomalies, like equipment failures, vendor implementation errors, and software bugs. Thirdly, abnormal user behaviors can change the traffic patterns, for example, flash crowds, non-malicious large file transfers, etc. In the early days, traffic anomalies often involve unusual large-volume traffic, i.e. high-profile traffic, which are mainly caused by traditional DoS, worm, or flash crowds. In recent years, new threats like Botnets introduce low-profile but in a coordinated manner, which only generate a small amount of traffic but follow specific coordinated traffic patterns. Besides these, there are also some traffic anomalies that are low-profile and non-coordinated, e.g. Black mails and spam voice IP calls.

![A Distributed Framework for Network Measurement and Monitoring](image.png)

For the purpose of addressing problems like intrusion detection, fault detection and recovery, and QoS provisions, many ISPs have chosen to use a distributed architecture for the network monitoring as shown in Fig.1, which consists of local monitors and Network Operation Centers (NOC). In this framework, local monitors collect data from routers and other network devices, perform some processing at or close to the data sources, and then transfer their measurements to NOCs. NOCs are responsible for mining characteristics of interest from collected measurements, and identifying the problems and the roots thereof. Many of such measurements from these system are also the data source for the traffic anomaly detection. Monitoring and detecting network-wide traffic anomalies has been and is still challenging for the following reasons: Firstly, the Internet traffic exhibits huge fluctuations and long range dependence [1], which makes traffic anomalies often be hidden by large volumes of the normal traffic. Secondly, traffic anomalies show an extreme diversity and new varieties of traffic anomalies are emerging everyday. Thirdly, ISPs want to detect traffic anomalies when they are still at a low-profile volume in order to reduce the damage as much and early as possible. Last but not the least, there are many systems where
data, computing, and other resources are distributed and cannot be transported to a center for various reasons, e.g. low bandwidth, security, privacy, and load balancing issues.

The spatial analysis like Principal Component Analysis (PCA) [2] has been verified to be an effective method for the traffic anomaly detection. But it introduces several challenges for applying PCA online in practice [3]. This method only classifies specific time intervals as anomalies, but cannot identify responsible ones. Local monitors must send their data to the NOC periodically, which could cause the scalability problems. PCA requires a singular value decomposition (SVD) of a $n \times m$ matrix. The computation complexity of SVD is $O(nm^2)$ and the space requirement is $O(nm)$, which would become a bottleneck to perform PCA in high-spend networks. Because the bandwidth to a NOC may be limited, local monitors cannot send data at a high frequency. To address this problem, the NOC must set a long enough period for each monitor to update their measurements.

1.1 Our Contribution

In this paper, we propose sketch-based algorithms for the traffic anomaly detection based on network-wide traffic measurements, which can significantly improve the computation and storage overhead for the PCA-based methods [2], [4]–[7]. In our algorithms, each local monitor only maintains a series of sketches for each traffic flow rather than the raw measurements, which requires $O(w \log n)$ running time and $O(w \log^2 n)$ space, where $w$ denotes the maximum possible number of traffic flows at a local monitor. Traffic measurements are projected into random selected sketches that are sent to the NOC, and keep the original data in secret. It is very difficult to learn other information about network traffic from the sketches but traffic anomalies. The sketch computation and update can be done either at the local monitor or at the NOC with the consideration about the privacy protection and the communication cost. NOC can run PCA-based detection methods by using collected sketches with $O(m^2 \log n)$ running time and $O(m \log n)$ space. Our main contributions can be summarized as follows:

1) Our algorithm is efficient in both the running time and the space requirement.
2) Our algorithm is flexible for ISP to balance the computation and the storage in a distributed measurement and monitoring system.
3) Our algorithm can detect both high-profile and coordinated low-profile traffic anomalies as an outlier in the regular traffic patterns like the PCA-based methods.
4) Our algorithm provides an additional benefit, i.e. the privacy protection of traffic measurements.
5) Our algorithm can also keep the communication cost below a upper bound $l$ when the NOC only allows a long updating period due to the limited communication bandwidth.

1.2 Related work

Traffic anomaly detection has become an important issue for the network management in the Internet, which has obtained considerable research interests. For the high-profile traffic anomalies, researchers can apply some signal analysis methods on the traffic measurements from a single monitor to detect traffic anomalies [8]. To deal with the low-profile coordinated traffic anomalies, Lakhina et al. [2], [5] proposed a PCA-based detection method by utilizing traffic measurements from multiple links. Li et al. [7] aggregated traffic flows into sketch subspaces and detected traffic anomalies based on PCA. Due to the high communication cost in Lakhina’s method, Huang et al. [9] designed a local algorithm to filter data at the local monitor in order to avoid excessive use of the network-wide communication. A local monitor will send its data to the NOC only if the local error exceeds a user-specified tolerance. Furthermore, a general framework of detection methods [10] and a distributed multi-dimensional indexing system of traffic measurements [11] have been proposed for the traffic anomaly detection problem. Recently, several new methods are introduced to detect traffic anomalies based on various features in the network traffic [12]–[14]. Chhabra et al. [13] used the generalized quantile sets (GQSs) to identify a set of candidate anomalies at each local monitor. Then a local monitor communicates its detection results with other local monitors to finally detect traffic anomalies. Kline et al. [14] utilized Bayes Net to identify potential anomalous traffic from traffic volumes and correlations between ingress/egress packet and bit rates.
The rest of this paper is structured as following. We formalize the traffic anomaly detection problem in Sec.2. Next we present two detection algorithms in Sec.3 and Sec.4, respectively, where the second one is an advanced version of the first one that improves the algorithm at local monitors. The theoretical analysis of these two methods are also given in the same sections, respectively. We evaluate our algorithm by using the data from Abilene Observatory Data Collections [15] in Sec.6. We conclude our paper with future work in Sec.7.

2 Problem Definition

2.1 Background

The Internet is a global system of interconnected computer networks, which can provide data interchanging by using the standardized Internet Protocol Suite (TCP/IP). The computer networks are organized into several autonomous systems (AS), each of which is independently operated by an Internet Service Provider (ISP). The success of the Internet mainly owes to the end-to-end principle, which results in a simple network infrastructure. All the data transported by the Internet are divided into IP packets, and each packet is forwarded hop-by-hop by routers. There are a source address and a destination address in each packet's header, which are used by routers to determine the forwarding path from the source to the destination. Inter-domain routing protocol (BGP) is used to forward IP packets among different ASs. And there are more than one intra-domain routing protocol that can be used in a single AS. The communication between two computers is controlled by a transmission protocol like TCP, which creates an individual end-to-end traffic flow.

Due to the exponential increase in terms of the number of users and applications, it has become not feasible to maintain statistics for each individual end-to-end traffic flow if not impossible. Thus, ISPs often aggregate end-to-end traffic flows at different levels, such as origin autonomous systems, ingress links, applications, etc. For example, ISPs can use the origin-destination (OD) flow, defined as all packets that enter the network at one origin router and exits at another destination router.

Local monitors collect traffic measurements on aggregated traffic flows in real time. A traffic measurement, denoted by \( s \), can contain one or several traffic features, e.g.,

\[
\begin{align*}
s &= \{c_1(\text{IP}), c_2(\text{Port}), c_3(\text{Size})\},
\end{align*}
\]

where \( c_k(\cdot) \) denotes a function on IP addresses, TCP/UDP ports, packet size, or other traffic features. And \( c_k(\cdot) \) is computed over packets within a time interval, which can be the count, the entropy, or other quantities of the traffic features.

2.2 Traffic Anomaly Definition

![Fig. 2. DoS Traffic Flood ©CAIDA](image)

A high-profile traffic anomaly often means an unusual large volume of traffic from one or multiple sources to one or multiple destinations. As a simple example, we show the packet counts in a campus
link in Fig.2 from CAIDA [16]. There was a spike around 11:00 a.m. in both inbound and outbound traffic, which was caused by a flood of incoming TCP ACK packets directed to a campus computer. Flash crowds is another example of common traffic anomalies, which refer to large unexpected traffic spikes towards particular web-sites due to associated user behaviors. Usually, sudden events of great interest trigger flash crowds, for example the CNN broadcast on the terrorist attacks of September 11, 2001 [17]. The increase in the request rate is dramatic, but relatively short in duration. On September 11, CNN served over 132 million pages, and the traffic was almost doubling every 7 minutes.

Usually, a local monitor can detect such big changes in the traffic pattern, but it is difficult to identify potential traffic anomalies that arise from small fluctuations in thousands of traffic flows. For the low-profile coordinated traffic anomalies, some computers attempt to compromise vulnerable hosts, propagate marlacious software, or operate some botnets, which only involve small-volume traffic flows as shown in Fig. 3 including the traffic volumes in four OD flows on the Abilene network [15]. Such traffic anomalies like the Botnet [18] cannot be detected at a single local monitor and therefore require a network-wide traffic analysis.

![Graph](image)

**Fig. 3.** An example of coordinated traffic anomalies from the Abilene network

In a large-scale network, ISPs can utilize traffic measurements from multiple monitors to detect traffic anomalies that cannot be detected at a single monitor. An observation, denoted by $x = (s_1, \ldots, s_m)^T$, is defined as a vector of the measurements from multiple monitors, where $s_j$ denotes a measurement of the $j$-th flow and $m$ denotes the number of traffic flows. Although each $s_j$ may be a normal measurement individually, they can only be observed concurrently when there are some traffic anomalies occurring. In general, a traffic anomaly can be detected as an observation that doesn’t belong to an normal observation set $\Omega$, which contains all observations corresponding to the normal traffic.

The easiest way to check whether an observation $x$ belongs to $\Omega$ is to compare $x$ with each element in $\Omega$, which may require a large amount of time to process all possible observations. In order to detect traffic anomalies in nearly real time, we have to investigate the properties of normal traffic, and try to find some quantities that are different enough to distinguish normal traffic and anomalous traffic. The anomaly distance is defined as such a distance norm that the normal observations in $\Omega$ are close to each other with a high probability. We can pick a typical normal observation $x_0 \in \Omega$, and rank every observation based on the anomaly distance between itself and the typical observation $x_0$. Then the probability distribution
of the anomaly distance on normal observations can be computed, which will be used to detect traffic anomalies with a statistical confidence.

**Definition 1:** Given an observation set \( \Omega \) and an anomaly distance \( d_\Omega(\cdot, \cdot) \), an observation \( x \) is a traffic anomaly at \((1 - \varrho)\) confidence level if

\[
P(x' : d_\Omega(x', x_0) > d_\Omega(x, x_0)) \leq \varrho\tag{1}
\]

where \( P(\cdot) \) denotes the probability distribution of the anomaly distance, and \( \varrho \) is a real number between 0 and 1.

In this paper, we take the detection of traffic volume anomalies as an example to verify our method, which is a specific case study of the general problem. In our algorithm, we do not specify any aggregation method, and assume that the aggregation method is given by ISPs who can balance the computation overhead and their requirements. And we have no knowledge about the statistical properties of the traffic. As a consequence, we are given a previous measurement set \( \hat{\Omega} \), and estimate the probability distribution of the anomaly distance based on it. The traffic volume is defined as the total size of all IP packets in a traffic flow within a single time interval. Here \( x_{ij} \) denotes the traffic volume of the \( j \)-th flow at the \( i \)-th time interval. The measurement set \( \hat{\Omega} \) contains all recent observations within a sliding window of the length \( n \) from \( m \) traffic flows. And the probability density distribution of the anomaly distance is denoted by \( f_d(x) \). For simplicity, we use \( d_\hat{\Omega}(x) \) to denote the anomaly distance between \( x \) and \( x_0 \).

**Definition 2:** Given the measurement set \( \hat{\Omega} \), an observation \( x_i \) measured at the \( i \)-th time interval is a traffic anomaly at \( 1 - \varrho \) confidence level, if

\[
d_\hat{\Omega}(x_i) > \delta_\varrho\tag{2}
\]

where \( \delta_\varrho \) denotes the threshold of the anomaly distance assigned to the observation set \( \hat{\Omega} \), which can be obtained by solving the following equation,

\[
\int_{\delta_\varrho}^{\infty} f_d(x)dx = \varrho.
\]

### 2.3 Objective of this research

Our algorithm aims at detecting traffic anomalies with continuous traffic measurements updating without retrieving previous data in high-speed networks. To archive this goal, we must compute the anomaly distance for sampled observations based on the measurement set \( \hat{\Omega} \) with the consideration of computation and space constraints. Firstly, the computation should be very efficient at both the local monitors and the NOC. Secondly, the algorithm is implemented in a distributed system which requires careful consideration about the storage and communication overhead. Thirdly, it should support continuous update to adjust the detection method due to the evaluation of the traffic anomaly. Last but not the least, a privacy protection mechanism makes ISPs be able to share traffic measurements with each other, which can provide a network-wide detection of the traffic anomaly.

### 3 Simple Sketch Method (SSM)

![Fig. 4. System model of simple sketch method](image-url)
The system model of the simple sketch method is shown in Fig. 4, which utilizes Random Projection [19]–[21] to reduce the computation complexity. In this method, the measurement set $\Omega$ is converted into a compact representation, i.e. the sketches. The sketches require smaller storage space and therefore can be maintained in the memory. The computation of the threshold $\delta_\nu$ and the anomaly distance $d_\Omega(x_i)$ are based on the sketches rather than the original measurement set $\Omega$. There are five modules and the algorithm in each module is described in the following section. Then we give an example to show the detection procedure of the SSM. At last, we provide the theory for the SSM method to detect traffic anomalies and the analysis of its computation complexity and the error bound.

3.1 Algorithm

3.1.1 Volume Counter

ISP implements an aggregation method and reports a pair $(FlowID, Size)$ to the volume counter, where $Size$ denotes the packet size and $FlowID$ is the index of the traffic flow. The volume counter maintains a list of buckets for each flow. A bucket $U_{ij}$ stores the traffic volume $x_{ij}$ of the $j$-th flow at the $i$-th time interval. An array of buckets contains all traffic volumes within a sliding window of the length $n$ for each flow. When a pair $(FlowID, Size)$ with $FlowID = j$ comes at the $i$-th time interval, the corresponding bucket $U_{ij}$ will be increased by $Size$. When a time interval ends, we just delete the last bucket and create a new bucket for the new time interval. The local monitor reports

$$y_{ij} = x_{ij} - \bar{x}_{tj}$$  \hspace{1cm} (4)

at the time interval $t$, which possibly contains anomalous traffic to the NOC, where $\bar{x}_{tj}$ denotes the mean of traffic volumes within the sliding window at the current time interval $t$,

$$\bar{x}_{tj} = \frac{1}{n} \sum_{i=t-n+1}^{t} x_{ij}.$$  \hspace{1cm} (5)

3.1.2 Sketch Computation

The sketch is a compact representation of a sequence of traffic measurements in order to compute interested statistics efficiently [22], [23]. It is an extension of the Random Projection which projects a $n$-dimensional vector into a random-selected $l$-dimensional sketch with a set of random vectors. After the projection, the distance between the original vectors can be preserved with the benefit that the size $l$ of the sketches is much less than the dimension $n$ of the original vectors, i.e. $l = O(\log n)$. At each local monitor, we compute an $l$-dimensional sketch of the traffic volume for each flow. All monitors share the same random numbers for the sketch computation. At each time interval, we use some pseudo random generators to compute $l$ independent and identically-distributed (i.i.d.) random numbers, denoted by $r_{i1}, \ldots, r_{il}$, from a specific probability distribution $\mathcal{F}$. Here, $\mathcal{F}$ could be the standard normal distribution, or some simple distributions that are easy to generate random numbers [24]. The property of the probability distribution $\mathcal{F}$ is closely related to the detection accuracy. The sketch $z_{kj}$ is computed as,

$$z_{kj} = \frac{1}{\sqrt{l}} \sum_{i=t-n+1}^{t} r_{ik} (x_{ij} - \bar{x}_{tj}) = \frac{1}{\sqrt{l}} \sum_{i=t-n+1}^{t} r_{ik} y_{ij} $$  \hspace{1cm} (6)

for $i = t - n + 1, \ldots, t, j = 1, \ldots, m$, and $k = 1, \ldots, l$.

3.1.3 Principle Component Analysis

NOC gets all sketches $\{z_{kj} : k = 1, \ldots, l, j = 1, \ldots, m\}$ from local monitors, and organizes them into a $l \times m$ matrix $Z$. PCA is applied on $Z$ and treats each row as a point in $\mathbb{R}^m$ and each column as a variable. PCA performs a coordinate rotation that aligns the transformed axes with the directions that make the projections of the row vectors on each axis get as large variance as possible. Principal components are the unit vectors along these axes. The first principal component of $Z$, denoted by $a_1$, can be found as,

$$a_1 = \arg \max_{\|x\|=1} \|Zx\|$$  \hspace{1cm} (7)
where \( \text{arg max} \) stands for the vector \( x = (x_1, \ldots, x_m)^T \) that satisfies \( \|x\| = 1 \) and makes the function \( \|Qx\| \) get the maximum value. Here, \( \|x\| \) stands for the Euclidean norm

\[
\|x\| = \sqrt{x_1^2 + \cdots + x_m^2}.
\]  

(8)

With the first \( r - 1 \) principal components, i.e. \( a_1, \ldots, a_{r-1} \), the \( r \)-th principal component \( a_r \) can be found by subtracting the first \( r - 1 \) principal components from \( Z \),

\[
a_r = \text{arg max}_{\|x\| = 1} \| (Z - \sum_{j=1}^{r-1} Z_a_j a_j^T)x \|.
\]

(9)

The standard procedure to find principle components is the singular value decomposition (SVD). A pair of vectors \( \mathbf{a} \in \mathbb{R}^m \) and \( \mathbf{b} \in \mathbb{R}^l \) are singular vectors of \( Z \), if \( \mathbf{Za} = \lambda \mathbf{b} \) and \( \mathbf{b}^T \mathbf{Z} = \lambda \mathbf{a}^T \), where \( \lambda \) is a real number denoting the corresponding singular value. And the principle components, i.e. \( \mathbf{a}_1, \ldots, \mathbf{a}_m \), are one of the pairs of singular vectors. Usually, the corresponding singular values of each principle component are ordered, i.e. \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0 \). Then the matrix \( Z \) can be written as

\[
Z = \sum_j \lambda_j \mathbf{b}_j \mathbf{a}_j^T.
\]

(10)

### 3.1.4 Anomaly Distance Computation

Given the principle components, i.e. \( \mathbf{a}_1, \ldots, \mathbf{a}_m \), we choose the last few principle components to compute the anomaly distance of an observation \( \mathbf{y}_i = (y_{i1}, \ldots, y_{im})^T \),

\[
d_Z(\mathbf{y}_i) = \sqrt{\sum_{j=r+1}^{m} (\mathbf{a}_j^T \mathbf{y}_i)^2}
\]

(11)

where \( r \) is an integer, \( r < m \). Because the first \( r \) principle components can capture the main pattern in the traffic, a normal observation should reside in the subspace of \( \mathbf{a}_1, \ldots, \mathbf{a}_r \), and the last \( m - r \) principle components are assumed to contain only random fluctuations. Thus, the probability distribution of the anomaly distance can be approximated by a distribution of a sum of chi-square random variables, because \( \mathbf{a}_j^T \mathbf{y}_i \) follows the normal distribution.

Before computing anomaly distance, we need to determine the number of principal components which are corresponding to random fluctuations in the traffic volumes. The last \( m - r \) principle components are chosen to compute anomaly distance, where \( r \) usually refers as the size of the normal subspace. There are several techniques which can be applied to determine the size of the normal space, such as \( k\sigma \)-heuristic, Cattell’s Scree Test, and so forth. Here, we give a brief introduction about the \( 3\sigma \)-heuristic [2]. The projection of the matrix \( Z \) on the \( j \)-th principle component, i.e. \( \mathbf{Za}_j \), is examined one by one. When a projection is found that the value of an element in \( \mathbf{Za}_j \) exceeds \( 3\sigma_j \) from the mean, where \( \sigma_j \) is the standard deviation, this and all remaining principle components are selected.

### 3.1.5 Threshold Computation

The threshold computation is based on the fault detection in multivariate process control [25]. Because \( \mathbf{a}_1, \ldots, \mathbf{a}_m \) is an orthonormal set of vectors, we get

\[
\| \mathbf{y}_i \| = \sqrt{\sum_{j=1}^{m} (\mathbf{a}_j^T \mathbf{y}_i)^2}.
\]

(12)

Let \( \mathbf{Q} = [\mathbf{a}_1, \ldots, \mathbf{a}_r] \), and then

\[
d_Z(\mathbf{y}_i) = \sqrt{\sum_{j=r+1}^{m} (\mathbf{a}_j^T \mathbf{y}_i)^2} = \sqrt{\| \mathbf{y}_i^T \mathbf{y}_i - \sum_{j=1}^{r} (\mathbf{a}_j^T \mathbf{y}_i)^2 \| = \| (\mathbf{I} - \mathbf{QQ}^T) \mathbf{y}_i \|}.
\]

(13)
Therefore, the anomaly distance $d_{Z}(y_i)$ equals to the squared prediction error (SPE) [25]. We can compute a threshold $\delta_\varphi$ based on the $Q$-statistic developed by Jackson and Mudholkar,

\[
\delta_\varphi^2 = \varphi_1 \left[ \frac{c_\varphi \sqrt{2 \varphi_2 h^2}}{\varphi_1} + 1 + \frac{\varphi_2 h (h - 1)}{\varphi_1^2} \right]^{1/h}
\]

where

\[
c_\varphi = 1 - \varphi,
\]

\[
h = 1 - \frac{2 \varphi_1 \varphi_3}{3 \varphi_2^2},
\]

\[
\varphi_k = \frac{1}{(n - 1)^k} \sum_{j=r+1}^{m} \lambda_{j}^{2k} \quad (k = 1, 2, 3).
\]

NOC first computes the anomaly distance according to Eq.(11). At the last step, if $d_{Z}(y_i) > \delta_\varphi$, NOC identifies $y_i$ as a traffic anomaly.

3.2 Detection Example

![Fig. 5. An example of traffic anomalies by SSM](image)

Given an observation $y_i$, we can identify $y_i$ as a traffic anomaly with a probability at least $1 - \varphi$ if $d_{Z}(y_i) > \delta_\varphi$. Here, we use an example from the Abilene network [15] to show the detection procedure. There are 9 routers in the Abilene network, i.e. ATLA, CHIC, HOUS, KANS, LOSA, NEWY, SALT, SEAT, and WASH. We first get the traffic volume series $y$ of each OD flow within 10 days from the Volume Counter, e.g.

\[
y_{ATLA-CHIC} = (-4.07 \times 10^7, \ldots , 4.54 \times 10^7). \quad (18)
\]

Then we compute the sketch $z_{ATLA-CHIC}$ by using $n \times l$ random numbers $r_{ik}$ according to Eq.(6),

\[
z_{ATLA-CHIC} = (4.26 \times 10^7, \ldots , 8.73 \times 10^7). \quad (19)
\]

NOC collects $z_{O-D}$ for all OD flow and organizes them into a matrix $Z$. By applying PCA on $Z$, we get principle components and the eigenvalues. Given an observation

\[
y_{5857} = (2.24 \times 10^7, \ldots , -2.42 \times 10^8) \quad (20)
\]
NOC computes the anomaly distance $d_Z(y_{5857}) = 2.08 \times 10^8$ according to Eq. (11). The threshold $\delta_p = 1.79 \times 10^8$ is computed as Eq. (14). Therefore, $d_Z(y_i) > \delta_p$ and NOC identifies the $i = 5857$ time interval as a traffic anomaly.

### 3.3 Computation Complexity

SSM detects traffic anomalies based on the same principles as Lakhina’s method [2], which can detect low-profile coordinated traffic anomalies. If there is an increase on several flows at the same time, we can detect that the anomaly distance will exceed the threshold. In fact, SSM is an approximation algorithm for Lakhina’s method, which can improve the computation complexity.

**Theorem 1:** The computation complexity and the space requirement of SSM are both $O(wn \log n)$ at the local monitor, where $w$ denotes the maximum possible number of traffic flows. The computation complexity is $O(m^2 \log n)$ and the space requirement is $O(m \log n)$ at the NOC.

**Proof:** At a local monitor, we first compute the mean of traffic volumes and then subtract it from $x_{ij}$ in order to get $y_{ij}$, which takes $O(wn)$ running time. Then, $y_{ij}$ is multiplied by the random numbers $r_{ik}$, which need to compute $O(wn \log n)$ productions. We calculate the sum of the productions as the sketch, that requires $O(wn)$ running time. Therefore, the computation complexity is $O(wn \log n)$. The local monitor needs to save the traffic volumes and random numbers, which requires $O(wn)$ spaces.

At the NOC, the computation complexity of the SVD on a $l \times m$ matrix is $O(m^2 l)$, which means that the computation complexity is at most $O(m^2 \log n)$. In order to save the sketch matrix, NOC needs $O(ml) = O(m \log n)$ memory space. At each time step, NOC uses the observation $y$ and pre-computed principle components to detect traffic anomalies, which only requires $O(m^2)$ running time. In general, the SSM requires $O(m \log^2 n)$ running time and $O(m \log n)$ space at the NOC.

### 3.4 Error Bound Analysis

First of all, we give a brief introduction about PCA-based method, which was introduced by Lakhina to detect coordinated low-profile traffic anomalies. Next, we prove that SSM can bound the detection error at a use-specified accuracy level.

#### 3.4.1 PCA-based Anomaly Detection

In Lakhina’s method, all traffic volume $x_{ij}$ from $m$ traffic flows within the sliding window of the length $n$ are organized into a $n \times m$ matrix $X$, which is adjusted to a matrix $Y$ with zero column mean, i.e. $y_{ij} = x_{ij} - \bar{x}_{ij}$. PCA is applied on the matrix $Y$, and the SVD of $Y$ is denoted by

$$ Y = \sum_{j=1}^{m} \eta_j u_j v_j^T, \quad (21) $$

where $v_j$ is the principal component and $\eta_j$ is the corresponding singular value.

Lakhina found that the traffic volumes of multiple flows had a low intrinsic dimensionality, which means that the normal traffic can effectively reside in a $r$-dimensional subspace with $r \ll m$. An adjusted observation $y_i = (y_{i1}, \ldots, y_{im})^T$ can be decomposed into normal and abnormal subspaces,

$$ y_i = y_{i,normal} + y_{i,anomaly} \quad (22) $$

where

$$ y_{i,normal} = P P^T y_i, \quad y_{i,anomaly} = (I - P P^T) y_i, \quad (23) $$

with $P = [v_1, \ldots, v_r]$. The size of the normal subspace, denoted by $r$, is determined by the $3\sigma$-heuristic. The traffic observation is classified as a normal traffic if

$$ \|y_{i,anomaly}\| \leq Q_\delta, \quad (24) $$
where $Q_\varrho$ is defined as the same as Eq.(14),

$$Q_\varrho^2 = \varphi_1 \left[ \frac{c_\varrho \sqrt{2\phi_2 h_0^2}}{\varphi_1} + 1 + \frac{\phi_2 h_0 (h_0 - 1)}{\varphi_1^2} \right]^{1/h_0},$$

(25)

where

$$h_0 = 1 - \frac{2\phi_1 \varphi_3}{3\varphi_2^2},$$

(26)

$$\varphi_k = \sum_{j=r+1}^{m} \sigma_j^2 (k = 1, 2, 3),$$

(27)

and $\sigma_j$ is the standard deviation of the projection of the measurements on the $j$-th principal component, which can be estimated as

$$\sigma_j = \frac{1}{\sqrt{n-1}} \|Yv_j\| = \frac{\eta_j}{\sqrt{n-1}}.$$

(28)

### 3.4.2 Error Bound for SSM

In this section, we explain why SSM can compute principal components based on the sketch matrix $Z$ and further detect anomalies based on them like Lakhina’s method. All proofs assume that the standard normal distribution is used for the sketch computation. Let $R$ be the $n \times l$ random matrix, which consists of the random number $r_{ik}$ from the standard normal distribution. According to the sketch computation in Eq.(6), we have

$$z_j = \frac{1}{\sqrt{l}} R^T y_j \quad \text{and} \quad Z = \frac{1}{\sqrt{l}} R^T Y,$$

(29)

where $y_j$ and $z_j$ are a column in $Y$ and $Z$, respectively. The vector $z_j$ is also called the random projection of $y_j$, which has the following properties [20].

**Lemma 1:** Let $R$ be a $n \times l$ random matrix from the standard normal distribution and $z_j = \frac{1}{\sqrt{l}} R^T y_j$. We have

- $E(\|z_j\|^2) = \|y_j\|^2$;
- $P(\|z_j\|^2 - \|y_j\|^2 \geq \varepsilon \|y_j\|^2) < 2e^{-(\varepsilon^2 - \varepsilon^3)} l$ for $\forall \varepsilon > 0$.

**Proof:**

$$E(\|z_j\|^2) = E(\sum_{k=1}^{l} z_{kj}^2) = E\left( \sum_{k=1}^{l} \left( \frac{1}{\sqrt{l}} \sum_{i=t-n+1}^{t} r_{ik} y_{ij} \right)^2 \right) = \frac{1}{l} \sum_{k=1}^{l} \left( \sum_{i=t-n+1}^{t} y_{ij}^2 E(r_{ik}^2) + \sum_{i=t-n+1}^{t} \sum_{i'=i+1}^{t} 2y_{ij} y_{i'j} E(r_{ik}) E(r_{i'k}) \right) = \sum_{i=t-n+1}^{t} y_{ij}^2$$

Let $W_k = \frac{\sqrt{7}}{\|y_j\|} z_{kj} = \frac{1}{\|y_j\|} \sum_{i=t-n+1}^{t} r_{ik} y_{ij}$, which is a standard normal variable. We define

$$W = \frac{l}{\|y_j\|^2} \|z_j\|^2 = \sum_{k=1}^{l} W_k^2$$

(30)
It follows the chi-square distribution $\chi^2$. Therefore,
\[
P(\|z_j\|^2 \geq (1+\varepsilon)\|y_j\|^2) = P(W \leq (1+\varepsilon)k) = P(e^{\theta W} \geq e^{(1+\varepsilon)\theta}) \leq \frac{E(e^{\theta W})}{e^{(1+\varepsilon)\theta}}
\]  
using Markov’s inequality.
\[
P(\|z_j\|^2 \geq (1+\varepsilon)\|y_j\|^2) \leq \Pi_{k=1}^{l} \frac{E(e^{\theta W_k})}{e^{(1+\varepsilon)\theta}} = \left(\frac{E(e^{\theta W^2})}{e^{(1+\varepsilon)\theta}}\right)^l
\]  
Because $W_1$ follows the standard normal distribution,
\[
E(e^{\theta W^2}) = \frac{1}{\sqrt{1-2\theta}}.
\]  
The above equation holds for any $\theta < 1/2$. Thus we get,
\[
P(W \geq (1+\varepsilon)l) \leq \left(\frac{e^{-2(1+\varepsilon)\theta}}{1-2\theta}\right)^{k/2}
\]  
The optimal choice of $\theta$ is $\varepsilon/2(1+\varepsilon)$. So we get,
\[
P(W \geq (1+\varepsilon)l) \leq ((1+\varepsilon)e^{-\varepsilon})^{k/2} < e^{-\varepsilon^2-\varepsilon^3/4}
\]  
Similarly,
\[
P(W \leq (1-\varepsilon)l) \leq ((1+\varepsilon)e^{-\varepsilon})^{k/2} < e^{-\varepsilon^2-\varepsilon^3/2}
\]  
According to the above two equations, we get $P(\|z_j\|^2 - \|y_j\|^2 \geq \varepsilon\|y_j\|^2) < 2e^{-(\varepsilon^2-\varepsilon^3)/2}$ □

Besides the standard normal distribution, there are several probability distributions which have been proposed for the random projection. Alon introduced the tug-of-war algorithm [24], where the random matrix $R$ is generated from the probability distribution
\[
r_{ik} = \begin{cases} 
-1 & \text{with probability } 1/2 \\
+1 & \text{with probability } 1/2 
\end{cases}
\]  
Later, Achlioptas [19] gave a more efficient algorithm, i.e. the sparse random projection with
\[
r_{ik} = \sqrt{s} \begin{cases} 
-1 & \text{with probability } 1/2s \\
0 & \text{with probability } 1 - 1/s \\
+1 & \text{with probability } 1/2s 
\end{cases}
\]  
where $s$ is an integer. In the sparse random projection, only $1/8$ of the data need to be processed. Recently, very sparse random projection has been recommended by Li et. al. [21], which uses $R$ of entries in $\{-1, 0, 1\}$ with probability $\{\frac{1}{2\sqrt{n}}, 1 - \frac{1}{\sqrt{n}}, \frac{1}{2\sqrt{n}}\}$.

Lemma 2: Let $R$ be a $n \times l$ random matrix with entries in $\{-1, 0, 1\}$ with probabilities $\{1/2s, 1 - 1/s, 1/2s\}$ and $z_j = \frac{1}{\sqrt{l}}R^Ty_j$. For $\forall \varepsilon > 0$, we have

- $E(\|z_j\|^2) = \|y_j\|^2$;
- $P(\|z_j\|^2 - \|y_j\|^2 \geq \varepsilon\|y_j\|^2) < 2e^{-(\varepsilon^2/2-\varepsilon^3/3)s}$.

Proof: It is easy to verify that
\[
E(\|z_j\|^2) = \|y_j\|^2
\]  
The second part has not finished yet. □
In the following part, we can use either convolutional random projection or sparse random projection, both of which give the same result. We will not distinguish two kinds of random projections. For the SVD, i.e. $Z = \sum_j \lambda_j b_j a_j^T$ and $Y = \sum_j \eta_j u_j v_j^T$, the singular values are approximately preserved.

Lemma 3: If $l > C \frac{\log n}{\varepsilon^2}$ for a large enough constant $C$ and an arbitrary positive constant $\varepsilon$,

$$ (1 - \varepsilon) \sum_{j=1}^r \eta_j^2 \leq \sum_{j=1}^r \lambda_j^2 \leq (1 + \varepsilon) \sum_{j=1}^r \eta_j^2 $$

(41)

for $\forall r$ with the probability $1 - 2e^{-\frac{C}{4} \log n}$.

Proof: Because $\lambda_1^2, \ldots, \lambda_r^2$ are the first $r$ largest eigenvalues of the matrix $Z^T Z$ and $v_1, \ldots, v_m$ are an orthonormal set of vectors, we have

$$ \sum_{j=1}^r \lambda_j^2 \geq \sum_{j=1}^r v_j^T (Z^T Z) v_j $$

$$ = \sum_{j=1}^r \frac{1}{l} v_j^T Y^T R R^T Y v_j $$

$$ = \sum_{j=1}^r \frac{1}{l} \eta_j^2 u_j^T R R^T u_j $$

$$ = \sum_{j=1}^r \eta_j^2 \| \frac{1}{\sqrt{l}} R^T u_j \|^2. $$

(42)

According to the properties of Random Projection, we have

$$ \sum_{j=1}^r \lambda_j^2 \geq \sum_{j=1}^r \eta_j^2 (1 - \varepsilon) \| u_j \|^2 $$

$$ = (1 - \varepsilon) \sum_{j=1}^r \eta_j^2. $$

(43)

We also know that

$$ \lambda_j^2 = a_j^T Z^T Z a_j $$

$$ = \frac{1}{l} a_j^T Y^T R R^T Y a_j $$

$$ = \| \frac{1}{\sqrt{l}} R^T (Y a_j) \|^2 $$

$$ \leq (1 + \varepsilon) \| Y a_j \|^2. $$

(44)

Because $\eta_1^2, \ldots, \eta_r^2$ are the first $r$ largest eigenvalues of the matrix $Y^T Y$ and $a_1, \ldots, a_m$ are an orthonormal set of vectors, we have

$$ \sum_{j=1}^r \eta_j^2 \geq \sum_{j=1}^r a_j^T (Y^T Y) a_j $$

$$ = \sum_{j=1}^r \| Y a_j \|^2 $$

$$ \geq \sum_{j=1}^r \frac{1}{1 + \varepsilon} \lambda_j^2. $$

(45)

Therefore, we have

$$ \sum_{j=1}^2 \lambda_j^2 \leq (1 + \varepsilon) \sum_{j=1}^r \eta_j^2. $$

(46)
Next, we want to bound the error of the covariance matrix in order to get a good approximation of the anomaly distance. According to the fact that principal components consists of an orthonormal set of vectors and \( \| Y \|_F^2 = \sum_{j=1}^{m} \eta_j^2 \), we can easily get the following result.

**Lemma 4:** Let \( V = Y^T Y \) and \( A = Z^T Z \). If \( l > C \frac{\log n}{\varepsilon^2} \) for a large enough constant \( C \), then with the probability \( 1 - 2e^{-C^2 \log n} \), we have

\[
\| V - A \|_F \leq \sqrt{2\varepsilon} \| Y \|_F^2
\]

where \( \| X \|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}^2} \) is the Frobenius norm of a matrix \( X \in \mathbb{R}^{n \times m} \).

**Proof:** First, because \( a_1, \ldots, a_m \) are an orthonormal set of vectors,

\[
\| V - A \|_F^2 = \sum_{j=1}^{m} \| (V - A)a_j \|^2.
\]

For each \( j = 1, \ldots, r \), we have

\[
\| (V - A)a_j \|^2 = a_j^T V^T Va_j + a_j^T A^T Aa_j - a_j^T V^T Aa_j - a_j^T A^T Va_j = \| Va_j \|^2 + \lambda_j^2 - 2\lambda_j^2 a_j^T Va_j.
\]

Because

\[
\lambda_j^2 = a_j^T Aa_j = \frac{1}{l} a_j^T Y^T R R^T Y a_j = \| \frac{1}{\sqrt{l}} R^T (Y a_j) \|^2.
\]

According to the properties of Random Projection, we have

\[
(1 - \varepsilon)\| Ya_j \|^2 \leq \| \frac{1}{\sqrt{l}} R^T (Y a_j) \|^2 \leq (1 + \varepsilon)\| Ya_j \|^2
\]

with a high probability. According to \( \| Ya_j \|^2 = a_j^T Va_j \), we have

\[
a_j^T Va_j \geq \frac{1}{1 + \varepsilon} \lambda_j^2.
\]

Because \( a_1, \ldots, a_m \) are an orthonormal set of vectors, we have \( \sum_{j=1}^{m} \| Va_j \| = \sum_{j=1}^{m} \eta_j^4 \). Therefore,

\[
\| V - A \|_F^2 = \sum_{j=1}^{m} \left( \eta_j^4 + \lambda_j^2 - 2\lambda_j^2 a_j^T Va_j \right) \leq \sum_{j=1}^{m} \left( \eta_j^4 + \lambda_j^2 - \frac{2}{1 + \varepsilon} \lambda_j^4 \right).
\]

Second, \( v_1, \ldots, v_m \) are also an orthonormal set of vectors,

\[
\| V - A \|_F^2 = \sum_{j=1}^{m} \| (V - A)v_j \|^2.
\]

For each \( j = 1, \ldots, m \),

\[
\| (V - A)v_j \|^2 = v_j^T V^T V v_j + v_j^T A^T A v_j - v_j^T V^T A v_j - v_j^T A v_j = \eta_j^4 + \| Av_j \|^2 - 2\eta_j^2 v_j^T A v_j.
\]
We also have

\[ v_j^T A v = v_j^T Z^T Z v_j = \frac{1}{T} v_i^T Y^T R R^T v_j \]

\[ = \eta_j^2 \| \frac{1}{\sqrt{t}} R^T u_j \|^2. \]  

(56)

According to the properties of Random Projection, we have

\[ \| \frac{1}{\sqrt{t}} R^T u_j \|^2 \geq (1 - \varepsilon) \| u_j \|^2 = (1 - \varepsilon). \]  

(57)

Because \( v_1, \ldots, v_m \) are also an orthonormal set of vectors, we have \( \sum_{j=1}^m \| A v_j \| = \sum_{j=1}^m \lambda_j^4 \). Therefore, we have

\[ \| V - A \|_F^2 = \sum_{j=1}^m (\eta_j^4 + \lambda_j^4 - 2\eta_j^2 v_j^T A v_j) \]

\[ \leq \sum_{j=1}^m \left( \eta_j^4 + \lambda_j^4 - 2(1 - \varepsilon)\eta_j^4 \right). \]  

(58)

Finally, based on Eq. (53) and Eq. (58), we get

\[ \| V - A \|_F^2 \leq \sum_{j=1}^m \varepsilon \left( \eta_j^4 + \frac{1}{1 + \varepsilon} \lambda_j^4 \right) \leq \varepsilon \sum_{j=1}^m (\lambda_j^4 + \eta_j^4) \].  

(59)

Based on Lemma 3 and \( \| Y \|_F^2 = \sum_{j=1}^m \eta_j^2 \), we have the following result.

\[ \| V - A \|_F^2 \leq \varepsilon \sum_{j=1}^m (\lambda_j^4 + \eta_j^4) \]

\[ \leq \varepsilon \left( \left( \sum_{j=1}^m \lambda_j^4 \right)^2 + \left( \sum_{j=1}^m \eta_j^4 \right)^2 \right) \]

\[ \leq \varepsilon ((1 + \varepsilon)^2 + 1) \left( \sum_{j=1}^m \eta_j^2 \right)^2 \]

\[ = 2\varepsilon (1 + \varepsilon + \varepsilon^2/2) \| Y \|_F^4 \]

\[ \approx 2\varepsilon \| Y \|_F^4. \]  

(60)

Therefore,

\[ \| V - A \|_F \leq \sqrt{2\varepsilon} \| Y \|_F^2. \]

According to Eq. (47), we also get a perturbation error bound for the eigenvalues from the Mirsky’s theorem [26],

\[ \sqrt{\frac{1}{m} \sum_{j=1}^m (\lambda_j^4 - \eta_j^2)^2} \leq \| V - A \|_F \leq \sqrt{2\varepsilon} \| Y \|_F^2. \]

(61)

Based on Lemma 3 and Lemma 4, we know that \( \phi_k \) can be approximated by \( \varphi_k \) up to the multiplicative factor \( (1 \pm \varepsilon_k^{1/2}) \). Therefore, the threshold \( Q_\varepsilon \) can be approximated by \( \delta_\varepsilon \) up to the multiplicative factor \( (1 \pm \varepsilon^{1/2}) \).

In the following part, we want to prove that \( d_Y(y) \) can be approximated by \( d_Z(y) \). The column space of a matrix \( M \) is the subspace spanned by the columns, which is denoted by \( \mathcal{R}(M) = \{ Mx : x \in \mathcal{R}^m \} \). The set of all eigenvalues of the matrix \( M \) is denoted by \( \mathcal{L}(M) = \{ \lambda : Mx = \lambda x, \exists x \neq 0 \} \). And \( \Theta \) denotes
the canonical angle between two subspaces, \( \Theta(\mathcal{M}, \mathcal{N}) = \sin^{-1} \Sigma \), where \( \mathcal{M} \) and \( \mathcal{N} \) are \( r \)-dimensional subspaces of \( \mathbb{R}^m \). The columns of their orthogonal bases can be transformed by a unitary matrix to

\[
\begin{pmatrix}
I \\
0 \\
0
\end{pmatrix}
\text{ and }
\begin{pmatrix}
\Gamma \\
\Sigma \\
0
\end{pmatrix}
\text{ if } 2r \leq m,
\text{ or }
\begin{pmatrix}
I \\
0 \\
0
\end{pmatrix}
\text{ and }
\begin{pmatrix}
\Gamma \\
0 \\
1
\end{pmatrix}
\text{ if } 2r > m.
\]

(62)

The matrix permutation theorem can be written as following [27].

**Lemma 5:** Matrix Perturbation: Let \( M \) have the spectral resolution

\[
\begin{pmatrix}
U_1^T \\
U_2^T
\end{pmatrix}
M
\begin{pmatrix}
U_1 \\
U_2
\end{pmatrix} = \begin{pmatrix}
L_1 & 0 \\
0 & L_2
\end{pmatrix},
\]

(63)

where \( (U_1, U_2) \) is unitary with \( U_1 \in \mathbb{R}^{n \times r} \). Let \( B \in \mathbb{R}^{n \times r} \) have orthonormal columns, and for any symmetric \( H \) of order \( r \), let \( E = MB - BH \). If \( \nu = \min |\mathcal{L}(L_2) - \mathcal{L}(H)| > 0 \), then we have

\[
||\sin \Theta(\mathcal{R}(U_1), \mathcal{R}(B))||_F \leq \frac{\|E\|_F}{\nu}.
\]

(64)

We apply the matrix permutation theorem to the covariance matrices \( A \) and \( V \), and get an error bound for the anomaly distance.

**Theorem 2:** If \( l > C \frac{\log n}{r} \) for a large enough constant \( C \), then

\[
d_Z(y) - d_Y(y) \leq \frac{2\sqrt{\varepsilon}}{\eta_{r+1}^2 - \eta_r^2} \|Y\|_F \|y\|
\]

(65)

with the probability \( 1 - 2e^{-\frac{C}{r} \log n} \).

**Proof:** Let \( Q = [a_1, \cdots, a_r], Q_c = [a_{r+1}, \cdots, a_m], P = [v_1, \cdots, v_r], \) and \( P_c = [v_{r+1}, \cdots, v_m] \). We have the following spectral resolutions,

\[
\begin{pmatrix}
Q^T \\
Q_c^T
\end{pmatrix}
A
\begin{pmatrix}
Q Q_c
\end{pmatrix} = \begin{pmatrix}
A_1 & 0 \\
0 & A_2
\end{pmatrix},
\]

(66)

\[
\begin{pmatrix}
P^T \\
P_c^T
\end{pmatrix}
V
\begin{pmatrix}
P P_c
\end{pmatrix} = \begin{pmatrix}
M_1 & 0 \\
0 & M_2
\end{pmatrix},
\]

(67)

where \( A_1 = \text{diag}(\lambda_1^2, \cdots, \lambda_r^2), A_2 = \text{diag}(\lambda_{r+1}^2, \cdots, \lambda_m^2), M_1 = \text{diag}(\eta_1^2, \cdots, \eta_r^2), \) and \( M_2 = \text{diag}(\eta_{r+1}^2, \cdots, \eta_m^2) \). Here \( \text{diag}(\cdot) \) denotes a diagonal matrix. Let \( E = VQ - QA_1 \). Because \( QA_1 = AQ \), we have

\[
E = VQ - AQ.
\]

(68)

According to Lemma 5, we have

\[
||\sin \Theta(\mathcal{R}(P), \mathcal{R}(Q))||_F \leq \frac{\|E\|_F}{\nu} = \frac{\|V - A\|_F}{\nu}
\]

(69)

where \( \nu = |\eta_{r+1}^2 - \lambda_r^2| \approx |\eta_{r+1}^2 - \eta_r^2| \). The project matrices of \( \mathcal{R}(P) \) and \( \mathcal{R}(Q) \) are \( PP^T \) and \( QQ^T \), respectively. Then, according to Ref. [27], we have

\[
\|PP^T - QQ^T\|_F = \sqrt{2}||\sin \Theta(\mathcal{R}(P), \mathcal{R}(Q))||_F \leq \sqrt{2\|V - A\|_F}. \]

(70)
Then we get
\[
|d_Z(y) - d_Y(y)| = \|(I - QQ^T)y\| - \|(I - PP^T)y\| \\
\leq \|(I - QQ^T)y - (I - PP^T)y\| \\
= \|(PP^T - QQ^T)y\| \\
\leq \|PP^T - QQ^T\|_F \|y\| \\
\leq \sqrt{2} \|V - A\|_F \|y\| \\
\leq \frac{2\sqrt{\varepsilon}}{|n_{t+1} - n_t^2|} \|Y\|_F \|y\|. \quad (71)
\]

Therefore, the anomaly distance can be approximated up to the multiplicative factor \((1 \pm \varepsilon^{1/2})\).

4 Advanced Sketch Method (ASM)

Fig. 6. System model of advanced sketch method

If the length of the sliding window is so long that a local monitor can not hold all traffic measurements in the memory, we need to find an alternative method to compute the sketch online. In the ASM, we utilize a variance estimation algorithm to maintain an approximation of the sketch in order to reduce the computation complexity and the space requirement at a local monitor. The architecture of the advanced sketch method is shown in Fig.6. We use Variance Histograms (VH) to maintain an approximation of the sketch for each flow, which is a modification of a variance estimation algorithm [28]. In this ASM, the Volume Counter module maintains only a bucket for each traffic flow at the current time interval. The Sketch Computation module is replaced by Variance Histograms. All other modules are the same as the SSM. We first introduce the Variance Histograms and then give the sketch computation algorithm at each local monitor. At last, we use the same example as the SSM to show the detection procedure in the ASM.

4.1 Algorithm

4.1.1 Variance Histograms

A Variance Histogram (VH) contains a list of buckets for each traffic flow, which are maintained by the variance estimation algorithm in Fig. 7. The traffic volume \(x_{ij}\) at each time interval is treated as a data element for the variance computation in this section. Given a sequence of data elements \(\{x_{(t-n+1)j}, \ldots, x_{tj}\}\), the variance is defined as
\[
V_{tj} = \sum_{i=t-n+1}^{t} (x_{ij} - \bar{x}_{tj})^2 \quad (72)
\]
where \(\bar{x}_{tj} = \frac{1}{n} \sum_{i=t-n+1}^{t} x_{ij}\) is the mean of data elements. A bucket \(B_{pj}\) contains the following statistics information for a subsequence of traffic volumes \(x_{ij}\).
- \(\tau_{pj}\): time stamp;
- \(n_{pj}\): total number of data elements in the subsequence;
Step 1: Check the time stamp of the last bucket $B_{N_j}$

if $\tau_{N_j} \leq t - n$
    delete $B_{N_j}$;
endif

Step 2: Create a new bucket $B_{1j}$

$\tau_{1j} = t; n_{1j} = 1; \mu_{1j} = x_{1j}; V_{1j} = 0$;

for $k = 1, \ldots, l$
    $Z_{1kj} = x_{1j} r_{tk}; R_{1kj} = r_{tk}$;
endfor

Step 3: Traverse the bucket list to merge buckets

$p = 1; B_B = B_{1j}$;
while $B_{(p+2)j}$ exists.
    $B_A = B_{(p+1)j} \cup B_{(p+2)j}$;
    if $n_A + n_B > n/2$
        return
    endif
    if $n_A \leq \frac{\epsilon}{10} n_B$ and $V_{A \cup B} - V_B \leq \frac{\epsilon}{3} V_B$
        delete $B_{(p+2)j}; B_{(p+1)j} = B_A$;
    else
        $p = p + 1; B_B = B_B \cup B_{pj}$;
    endif
endwhile

Fig. 7. Procedures for updating VH

- $\mu_{pj}$: mean of data elements in the subsequence;
- $V_{pj}$: variance of data elements in the subsequence;
- $Z_{pkj}$: sum of $x_{ij} r_{ik}$ for all $x_{ij}$ in the subsequence;
- $R_{pkj}$: sum of the corresponding $r_{ik}$.

The algorithm starts with an empty list of buckets and updates the list of buckets with three steps as shown in Fig. 7. First, when a new data element $x_{tj}$ comes, the current time stamp is updated to $t$. We check the oldest bucket $B_{N_j}$ and delete it if it is expired, where $N$ denotes the number of buckets in the list. Second, the new element constitutes a new bucket $B_{1j}$ and each old bucket $B_{pj}$ becomes $B_{(p+1)j}$ for $p = 1, \ldots, N$. Last, we check whether there are qualified pairs of buckets that can be merged. Let $B_A = B_{(p+1)j} \cup B_{(p+2)j}$ and $B_B = \cup_{q=1}^{P-1} B_{qj}$. We merge two adjacent buckets $B_{(p+1)j}$ and $B_{(p+2)j}$ if and only if they satisfy the following merging rules.

- Rule 1: $V_{A \cup B} - V_B \leq \frac{\epsilon}{3} V_B$.
- Rule 2: $n_A \leq \frac{\epsilon}{10} n_B$.
- Rule 2: $n_A + n_B \leq n/2$.

When two adjacent buckets $B_{pj}$ and $B_{qj}$ merge into a new bucket $B_{(p,q)j}$, the merged bucket’s time stamp is set to be the time stamp of the older one, and the merged bucket’s statistics information can be calculated as follow,

$$n_{(p,q)j} = n_{pj} + n_{qj}$$  \hspace{1cm} (73)

$$\mu_{(p,q)j} = \frac{n_{pj} \mu_{pj} + n_{qj} \mu_{qj}}{n_{pj} + n_{qj}}$$  \hspace{1cm} (74)

$$V_{(p,q)j} = V_{pj} + V_{qj} + \frac{n_{pj} n_{qj}}{n_{pj} + n_{qj}} (\mu_{pj} - \mu_{qj})^2$$  \hspace{1cm} (75)

$$Z_{(p,q)kj} = Z_{pkj} + Z_{qkj}$$  \hspace{1cm} (76)

$$R_{(p,q)kj} = R_{pkj} + R_{qkj}$$  \hspace{1cm} (77)
Let $B_{all,j} = \bigcup_{p=1}^{N} B_{pj}$ denote the bucket by merging all buckets together, and $\hat{V} = V_{all,j}$ be the estimated variance. We get the following result [28].

**Lemma 6:** Variance Histogram maintains a $\varepsilon$-approximate variance,

$$(1 - \varepsilon)V \leq \hat{V} \leq V,$$  

with $O(\frac{1}{\varepsilon} \log n)$ space and $O(1)$ running time.

### 4.1.2 Sketch Computation Algorithm

At a local monitor, we implement a VH for each traffic flow and $n$ pseudo random number generators shared by all traffic flows among local monitors. The architecture for the sketch computation at a local monitor is shown in Fig. 8. The volume counter only uses a bucket to maintain the traffic volume at the current time interval $t$ for each traffic flow. When a time interval ends, the volume counter reports the traffic volume $x_{tj}$ to the Variance Histogram $VH_j$. The $VH_j$ updates its buckets as shown in Fig. 7. At each time interval, we can compute an approximation of the sketch as,

$$\hat{z}_{kj} = \frac{1}{\sqrt{l}}(Z_{all,kj} - n_{all,j} \mu_{all,j} R_{all,kj}).$$  

(79)

where $n_{all,j}$, $\mu_{all,j}$, $Z_{all,kj}$, and $R_{all,kj}$ are the elements in $B_{all,j} = \bigcup_{p=1}^{N} B_{pj}$.

### 4.2 Detection Example

NOC collects the sketches $z_{kj}$ from local monitors, and follows the same procedure in the SSM to identify traffic anomalies. Here, we use the same example in Section 3.2. The sketch of the measurement $y_{ATLA-CHIC}$ in Eq.(18) is

$$\hat{z}_{ATLA-CHIC} = \left(1.00 \times 10^8, \ldots, -2.04 \times 10^8\right)$$  

(80)

First, NOC organize $z_{Origin-Destination}$ into a matrix $\hat{Z}$. Second, NOC uses the same method as the SSM to compute principle components and eigenvalues. Last, the anomaly distance and the threshold are computed according to Eq.(11) and Eq.(14), respectively.

$$\hat{\delta}_b = 1.82 \times 10^8, \quad d_{\hat{Z}}(y_{5857}) = 2.11 \times 10^8.$$  

(81)

Therefore, we get $d_{\hat{Z}}(y_i) > \hat{\delta}_b$ and thus identify $i = 5857$ as a time interval containing traffic anomalies, which gets the same result as the SSM.
4.3 Computation Complexity

ASM follows the same procedure as SSM to detect traffic anomalies at the NOC, which has the same computation complexity and the space requirement as SSM. Because the variance estimation algorithm only needs $O(\frac{1}{\varepsilon} \log n)$ space and $O(1)$ running time, we have the following theorem.

**Theorem 3:** ASM requires $O(w \log n)$ running time and $(w \log^2 n)$ space at the local monitor.

*Proof:* According to the variance estimation algorithm [28], VH only needs $O(1)$ running time to update the variance buckets. But we also need to update the sketches $Z_{pkj}$ and the random numbers $R_{pkj}$. Therefore, ASM needs $O(l)$ running time to update the variance histograms. A bucket needs $O(l)$ space to storage the statistics information and we have at most $O(\log n)$ buckets for each traffic flow. In general, the local monitor need $O(w \log^2 n)$ space and $O(w \log n)$ running time, because $l = O(\log n)$. □

4.4 Error Bound Analysis

The approximated sketch $\hat{Z}_{kj}$ is a sketch of a subsequence of the traffic volumes within the sliding window as shown in Fig.9. We organize $\hat{Z}_{kj}$ into a $l \times m$ matrix $\hat{Z}$. Then we have the following result.

**Lemma 7:** Let $\hat{A} = \hat{Z}^T \hat{Z}$. If $l > C \frac{\log n}{\varepsilon}$ for a large enough constant $C$, then

$$\|\hat{A} - V\|_F \leq 2\sqrt{\varepsilon}\|Y\|^2_F$$

with the probability $1 - 2e^{-\frac{C}{\varepsilon} \log n}$.

*Proof:* The variance estimation algorithm maintain the variance $\hat{V}$ of a subsequence of the data elements in the sliding window of the size $n$, which have the property,

$$(1 - \varepsilon)V < \hat{V} < V$$

according to Ref. [28]. Let

$$\hat{y}_j = (\overbrace{0, \ldots, 0, y_{(t-\Delta_j+1)j}, \ldots, y_{lj}}^n)^T.$$  

Because $V = \sum_{i=t-n+1}^{t}(x_{ij} - \bar{x}_{tj})^2 = \|y_j\|^2$ and $\hat{V} = \sum_{i=t-\Delta_j+1}^{t}(x_{ij} - \bar{x}_{tj})^2 = \|\hat{y}_j\|^2$, then we have

$$\|\hat{y}_j - y_j\|^2 = |\hat{V} - V| < \varepsilon V = \varepsilon\|y_j\|^2.$$  

According to Eq.(79), we have

$$\hat{z}_{kj} = \frac{1}{\sqrt{l}} \left( \sum_{i=t-\Delta_j+1}^{t} (x_{ij} - \bar{x}_{tj})r_{ik} \right) = \frac{1}{\sqrt{l}} r_k\hat{y}_j.$$  

where $r_k = (r_{(t-n+1)k}, \ldots, r_{lk})$. Therefore,

$$\hat{z}_j - z_j = \frac{1}{\sqrt{l}} R(\hat{y}_j - y).$$

where $\hat{z}_j$ and $z_j$ are the $j$-th column in $\hat{Z}$ and $Z$, respectively. We apply the properties of the Random Projection on the vector $\hat{z}_j - z_j$,

$$\|\hat{z}_j - z\|^2 \leq (1 + \varepsilon)\|\hat{y}_j - y\|^2.$$  

Fig. 9. An illustration of sketch approximation
Based on Eq.(85) and Eq.(88), we have
\[\|z_j - \hat{z}_j\|^2 \leq \varepsilon \|y_j\|^2.\]  
(89)
\[\|\hat{Z} - Z\|^2_F < \varepsilon \|Y\|^2_F.\]  
(90)
Because \(A = Z^T Z\), we have
\[\|\hat{A} - A\|^2_F = \|\hat{Z}^T \hat{Z} - Z^T Z\|^2_F = \|\hat{Z}^T \hat{Z} - \hat{Z}^T Z + \hat{Z}^T Z - Z^T Z\|^2_F \leq \|\hat{Z}^T \hat{Z} - Z^T Z\|^2_F + \|\hat{Z}^T Z - Z^T Z\|^2_F \leq 2\varepsilon \|Y\|^4_F.\]  
(91)
According to Lemma 4, we have
\[\|\hat{A} - \hat{V}\|^2_F = \|\hat{A} - A + A - \hat{V}\|^2_F \leq \|\hat{A} - A\|^2_F + \|A - \hat{V}\|^2_F \leq 4\varepsilon \|Y\|^4_F.\]  
(92)
Using Lemma 7, we can bound the threshold \(\hat{\delta}_e\) in ASM. The anomaly distance \(d_2(y_i)\) can be bounded by similar method in Theorem 2.

**Theorem 4:** If \(l > C \frac{\log n}{\varepsilon^2}\) for a large enough constant \(C\), then
\[|d_2(y) - d_Y(y)| \leq \frac{2\sqrt{2\varepsilon}}{|\eta_{r+1}^2 - \eta_r^2|} \|Y\|^2_F \|y\|\]  
(93)
with the probability \(1 - 2e^{-\frac{C}{2\varepsilon}}\).

**Proof:** Let \(\hat{Z} = \sum_j \lambda_j \hat{b}_j \hat{a}_j^T\), and we have \(\hat{Q} = [\hat{a}_1, \ldots, \hat{a}_r]\), \(\hat{Q}_c = [\hat{a}_{r+1}, \ldots, \hat{a}_m]\). The spectral resolution of the matrix \(\hat{A}\) can be written as,
\[
\begin{pmatrix}
\hat{Q}^T \\
\hat{Q}_c^T
\end{pmatrix}
\hat{A}

\begin{pmatrix}
\hat{Q} \\
\hat{Q}_c
\end{pmatrix}
= \begin{pmatrix}
\hat{A}_1 & 0 \\
0 & \hat{A}_2
\end{pmatrix},
\]  
(94)
where \(\hat{A}_1 = \text{diag}(\hat{\lambda}_1^2, \ldots, \hat{\lambda}_r^2)\) and \(\hat{A}_2 = \text{diag}(\hat{\lambda}_{r+1}^2, \ldots, \hat{\lambda}_m^2)\).

Let \(\hat{E} = V \hat{Q} - Q \hat{A}_1\) where \(V = Y^T Y\) as before. Because \(\hat{Q} \hat{A}_1 = \hat{A} \hat{Q}\), we have
\[\hat{E} = V \hat{Q} - \hat{A} \hat{Q}.\]  
(95)
According to Lemma 5 (Matrix Permutation Theorem), we have
\[\|\sin \Theta(R(P), R(\hat{Q}))\|_F \leq \frac{\|\hat{E}\|_F}{\hat{\nu}} = \frac{\|V - \hat{A}\|_F}{\hat{\nu}}\]  
(96)
where \(\hat{\nu} = |\eta_{r+1}^2 - \hat{\lambda}^2| \approx |\eta_{r+1}^2 - \eta_r^2|\). The project matrices of \(R(P)\) and \(R(\hat{Q})\) are \(PP^T\) and \(\hat{Q} \hat{Q}^T\), respectively. Then, according to Ref. [27], we have
\[\|PP^T - \hat{Q} \hat{Q}^T\|_F = \sqrt{2}\|\sin \Theta(R(P), R(\hat{Q}))\|_F \leq \sqrt{2}\frac{\|V - \hat{A}\|_F}{\hat{\nu}}.\]  
(97)
Then we get
\[
|d_z(y) - d_Y(y)| = \| (I - \hat{Q}\hat{Q}^T)y - (I - PP^T)y \| \\
\leq \| (I - \hat{Q}\hat{Q}^T)y - (I - PP^T)y \| \\
= \| (PP^T - \hat{Q}\hat{Q}^T)y \| \\
\leq \| PP^T - \hat{Q}\hat{Q}^T \|_F \| y \| \\
\leq \sqrt{2}\| V - \hat{A} \|_F \| y \| \\
\leq \frac{2\sqrt{2}ε}{|η_{r+1}^2 - η_r^2|} \| Y \|_F \| y \|. 
\]

(98)

5 DISCUSSIONS

We can bound the error of the estimated threshold and the anomaly distance in terms of $\eta_j$ and $Y$. Therefore, our algorithms are an approximation of Lakhina’s method, and the accuracy of our algorithms depends on the properties of the covariance matrix $V$ of the traffic measurements. In fact, the PCA-based detection method can have a high false alarm rate when all the eigenvalues of $V$ are close to each other. Fortunately, Lakhina observed that the above situations were rare in the traffic volume measurements [6]. There are only a few eigenvalues which are much larger than 0, and the other are close to 0. In this situation, the values of the threshold and the anomaly distance only have a small fluctuation in both SSM and ASM. Therefore, our algorithms can detect the traffic anomalies with a low false negative rate.

Random projection method has been proposed for privacy preserving distributed data mining [29]. If NOC doesn’t know the sequence of the random number $r_{ik}$ for the sketch computation, there is no way to guess the traffic volume series. Although NOC may know $r_{ik}$, NOC doesn’t know the length due to the VH algorithm. Also, local monitors can introduce an additional permutation $\epsilon'$ into the random numbers, e.g., $r'_{ik} = r_{ik} + \epsilon'w_{ik}$ where $w_{ik}$ is an independent random number from the standard normal distribution. Then NOC needs to solve a set of linear equations,
\[
\hat{z}'_j = R\hat{y}_j 
\]
where $\hat{z}'_j = R'\hat{y}_j$ and $R' = R + \epsilon'W$ ($W$ is a matrix with entries $w_{ik}$). The local monitor can specify the parameter $\epsilon'$ in order to increase the estimation error of the traffic volume $y_{ij}$.

For the communication cost, supposing the NOC set the period of updating traffic measurements as $T$, Lakhina’s method needs to send a vector of measurements of the length $T$, and our algorithm sends a vector of the length $l$. Because $l$ is regardless of $T$, our algorithm can reduce the communication cost by $l/T$ if $T > l$. If the NOC has enough bandwidth, the sketch computation can also be done at the NOC side. Then our algorithm can use the same communication cost as Lakhina’s method.

6 EXPERIMENTAL EVALUATION

In the evaluation, we want to determine the size of the sketches which is enough to get a good approximation of Lakhina’s method, because a smaller size of the sketches means that our algorithms require less computation resource. Our algorithm is very useful when the time length of the traffic measurements is so long that the local monitors don’t have enough space to save them. Thus we only implement the ASM in the following evaluation.

Abilene Observatory Data Collections [15] are used as the data set to evaluate the performance of our algorithm. Abilene is the Internet2 backbone network, which spans the continental USA. We use the data collected by Juniper’s J-Flow tool between 06/09/2008 and 06/29/2008. We use both BGP and ISIS routing information to aggregate packets into OD flows.

When a packet arrives, we first update a temporary list which saves the total traffic volume of each OD flow in every five-minutes interval. In this way, we can construct the traffic volume time series. We first apply Lakhina’s method to detect anomalies, and then use these detected anomalies as the real
anomalies to evaluate the detection accuracy of our method. We compute both Type I errors and Type II errors with different size of the sketches.

\[
\text{Type I} = \frac{\text{number of false anomalies}}{\text{total number of true normal observations}}, \quad \text{Type II} = \frac{\text{number of false normal observations}}{\text{total number of true anomalies}}
\]

(100)

Fig. 10. Detection Errors in Abilene Data

We check the eigenvalues of the measurement matrix \( Y \), and choose the size of the normal subspace as \( r = 6 \) which is proper for our data. We check each observation just after the sliding window, and show Type I errors and Type II errors of the last week in Fig.10. We find that both type I errors and type II errors decrease quickly at the beginning and then reach a nearly optimal value. If the size \( l \) of the sketch is more than 100, there is no remarkable decrease in the mean of errors and only the variance of the errors becomes smaller.

Because the number of true anomalies is much less than the number of true normal observations, the optimal type II error is higher than the optimal type I error. Because there is always some randomness in the data and the length of the sketch should be less than the length of the sliding window, we cannot reduce type I and type II errors to zero in practice.

7 CONCLUSION AND FUTURE WORK

In this paper, we study the network-wide traffic anomaly detection problem. Our algorithm archives \( O(\log n) \) running time and \( O(l \log^2 n) \) space at local monitors. The NOC could run PCA-based detection method with \( O(m^2 \log n) \) running time and \( O(m \log n) \) space. Our algorithm also make the ISPs be able to implement the detection method by paying careful consideration about the privacy protection, the communication cost, and other resources over a distributed computing environment. In the future, we will study the detection of traffic anomalies by using various traffic features like the communication patterns.

ACKNOWLEDGMENT

This project has benefited from the use of measurement data collected on the Internet2 network as part of the Internet2 Observatory Project.

REFERENCES

23


