2010 International Conference on Distributed Computing Systems

Sketch-based Streaming PCA Algorithm for Network-wide Traffic Anomaly Detection

Yang Liu, Linfeng Zhang and Yong Guan
Department of Electrical and Computer Engineering
Iowa State University
Ames, IA 50011, USA
yangl@iastate.edu, zhangl@gmail.com, guan@iastate.edu

***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 anomalies in a timely manner. 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 linear running time and space to analyze the traffic measurements within a sliding window, which makes it often infeasible to be deployed for monitoring large-scale high-speed networks. We propose a sketch-based streaming PCA algorithm for the network-wide traffic anomaly detection in a distributed fashion. Our algorithm only requires logarithmic running time and space at both local monitors and Network Operation Centers (NOCs), and can detect both high-profile and coordinated low-profile traffic anomalies with bounded errors.**

***Keywords*-Traffic Anomaly; Principle Component Analysis; Data Streams;**

I. Introduction

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 traffic anomalies 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.

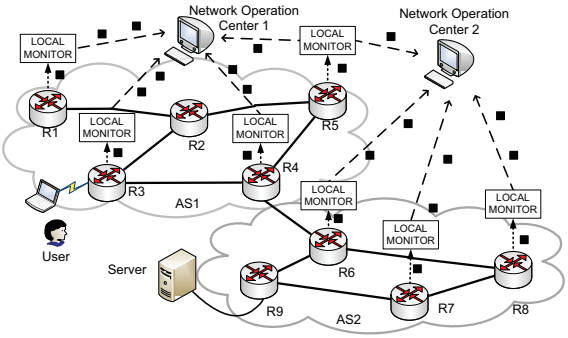

Figure 1. A Distributed Framework for Network Measurement and Monitoring

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.

For the purpose of addressing problems like intrusion detection, fault detection and recovery, and QoS provision, many ISPs have chosen to use a distributed architecture for the network monitoring as shown in Fig. 1. In this framework, local monitors collect data from routers and other network devices, perform some processing at or close to the data sources, and transfer their data to the NOCs. Then, NOCs are responsible for mining characteristics of interest from collected data, and identifying the problems and the roots thereof. Many of such measurements from

1063-6927/10 $26.00 © 2010 IEEE
DOI 10.1109/ICDCS.2010.45

these systems are also the data sources for the traffic anomaly detection. Monitoring and detecting network-wide traffic anomalies have been and are still challenging for the following reasons. Firstly, the Internet traffic exhibits huge fluctuations and long range dependence, which makes traffic anomalies often be hidden by large volumes of 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.

Due to the above challenges, the spatial analysis method like PCA [1] has been introduced for the traffic anomaly detection and verified to be effective for the coordinated low-profile traffic anomalies. But there are several challenges for applying PCA in practice. Currently, there is no well-known method to determine the parameters in the PCA-based detection methods [2]. Furthermore, large-volume traffic anomalies and the stealthy poisoning attacks can contaminate normal traffic patterns [3]. Last, 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 the high-speed network [1], [4], [5].

In this paper, we focus on the last challenge, i.e. the performance of the PCA-based detection method, and propose a novel sketch-based streaming algorithm, which can significant improve the computation and storage overhead. Because large-volume traffic anomalies can contaminate the normal traffic patterns, NOCs should use as many data as possible to train the traffic anomaly detector. By updating the traffic anomaly detector frequently, NOCs can detect the stealthy poisoning attacks with a high probability. Therefore, efficient algorithms for the PCA computation are very useful for the NOCs to detect network-wide traffic anomalies. Our main contributions are summarized as follows:

1) Our algorithm is efficient in both space and running time, which can achieve $O(w \log n)$ 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 the NOCs.
2) We also provide theoretical guarantees on the performance of our algorithm.
3) Our algorithm is flexible for ISPs to balance the computation and the storage in a distributed measurement and monitoring system.
4) 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.
5) Experimental results show that our algorithm can use very small sketches to detect traffic anomalies for all possible parameters. And the computation overhead is much less than the existing method.

The rest of this paper is structured as following. Related works are discussed in Sec.II. We formalize the traffic anomaly detection problem in Sec.III. Next we present our sketch-based algorithm in Sec.IV, which is a streaming PCA algorithm for the network-wide traffic anomaly detection. The theoretical analysis of our sketch-based algorithm is also given in Sec.V. We evaluate our sketch-based algorithm by using the data from Abilene Observatory Data Collections in the Internet2 project [6] in Sec.VI. We conclude our paper with future work in Sec.VII.

II. Related work

Traffic anomaly detections have become an important issue for the network management in the Internet, which have obtained considerable research interests [7], [8], [9], [10], [11], [12]. For the high-profile traffic anomalies, researchers can apply signal analysis methods to detect them [13]. To deal with the low-profile coordinated traffic anomalies, Lakhina et al. [1], [4] proposed PCA-based detection methods by utilizing traffic measurements from multiple links. The packets were aggregated into origin-destination (OD) flows and the traffic volume for each OD flow was updated to the NOC for every 5-minutes interval. Lakhina's method required $O(mn)$ space to maintain traffic volumes and $O(m^2 n)$ running time to compute PCA, where n was the number of intervals and m was the number of OD flows. Li et al. [7] aggregated flows into sketch subspaces and also detected traffic anomalies based on the PCA. Their method can help ISPs to identify the IP addresses related to the traffic anomalies but it involved a large number of aggregated flows which required at least as much computation as Lakhina's method. Due to the high communication cost in Lakhina's method, several methods have been proposed. Huang et al. [8] 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 would send its data to the NOC only if the local error exceeded a user-specified tolerance. Chhabra et al. [10] also proposed a method to reduce the communication cost, which used the generalized quantile sets (GQSs) to identify a set of candidate anomalies at each local monitor. Then a local monitor communicated its detection results with other local monitors to finally detect traffic anomalies. However, the computation overhead at the NOC cannot be reduced by using the above methods. Kline et al. [11] utilized Bayes Net to identify potential anomalous traffic from traffic volumes and correlations between ingress/egress packet and bit rates. Also, the temporal correlation was introduced to improve the accuracy of the PCA methods [12]. Such methods used more traffic information than Lakhina's method, and also required higher computation complexity.

Data streaming algorithms have been widely deployed for the traffic monitoring and analysis in real time [14]. Such algorithms usually use limited memory space to compute some functions with continuous traffic measurement updating and without retrieving previous measurements [15], [16]. PCA has also been applied for mining multiple data streams [17], [18], [19]. A more challenging problem is the sliding window model [20], where the streaming algorithm only focuses on the recent elements within a time window. According to our knowledge, there has been no work about the PCA computation in the sliding window model. Apparently, ISPs can apply the streaming PCA algorithm for the network-wide traffic anomaly detection in real time. This paper provides a novel streaming algorithm for the PCA computation over traffic streams in the sliding window model.

III. Problem Definition

A. Background

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. The communication between two computers is controlled by a transmission protocol like TCP, which creates an individual *end-to-end* 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* flow if not impossible. Thus, ISPs often aggregate *end-to-end* 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.

B. Principle Component Analysis

Lakihina et al. [1] applied PCA on the aggregated flows to build the statistical model for normal traffic, and further detected traffic anomalies. All traffic measurements from m aggregated flows within the sliding window of the length n are organized into a $n \times m$ matrix $\mathbf{X}$. Let x_{ij} denote the traffic measurement of the j-th flow at the i-th time interval, which can be the traffic volume, the entropy of IP addresses, the frequency of the byte values in the payload, and so forth. The matrix $\mathbf{X}$ is adjusted into a matrix $\mathbf{Y}$ with zero column mean, i.e. $y_{ij} = x_{ij} - \bar{x}_{tj}$ with $\bar{x}_{tj} = \sum_{i=t-n+1}^{t} x_{ij}/n$.

PCA is applied on $\mathbf{Y}$ and treats each row as a point in m-dimensional space 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 the matrix $\mathbf{Y}$, denoted by $\mathbf{v}_1$, can be found as,

$$\mathbf{v}_1 = \arg\max_{\|\mathbf{x}\|=1} \|\mathbf{Y}\mathbf{x}\| \tag{1}$$

where $arg\ max$ stands for the vector $\mathbf{x} = (x_1, \ldots, x_m)^T$ that satisfies $\|\mathbf{x}\| = 1$ and makes the function $\|\mathbf{Y}\mathbf{x}\|$ get the maximum value. Here, $\|\mathbf{x}\|$ stands for the Euclidean norm. With the first $r-1$ principal components, i.e. $\mathbf{v}_1, \ldots, \mathbf{v}_{r-1}$, the r-th principal component $\mathbf{v}_r$ can be found by subtracting the first $r-1$ principal components from $\mathbf{Y}$,

$$\mathbf{v}_r = \arg\max_{\|\mathbf{x}\|=1} \|(\mathbf{Y} - \sum_{j=1}^{r-1} \mathbf{Y}\mathbf{v}_j\mathbf{v}_j^T)\mathbf{x}\|. \tag{2}$$

A pair of vectors $\mathbf{v} \in \mathcal{R}^m$ and $\mathbf{u} \in \mathcal{R}^n$ are singular vectors of the matrix $\mathbf{Y}$, if $\mathbf{Y}\mathbf{v} = \eta\mathbf{u}$ and $\mathbf{u}^T\mathbf{Y} = \eta\mathbf{v}^T$, where η is the corresponding singular value. The principle components, i.e. $\mathbf{v}_1, \ldots, \mathbf{v}_m$, are one of the pairs of singular vectors. Usually, the corresponding singular values of each principle component are ordered, i.e. $\eta_1 \geq \eta_2 \geq \cdots \geq \eta_m \geq 0$. The matrix $\mathbf{Y}$ can be decomposed by the singular value decomposition (SVD),

$$\mathbf{Y} = \sum_{j=1}^{m} \eta_j \mathbf{u}_j \mathbf{v}_j^T. \tag{3}$$

C. Traffic Anomaly Detection

Because the first r principle components with $r \ll m$ can capture the main patterns of the normal traffic [1], a measurement vector $\mathbf{y}_{i*}$ should reside in the subspace of $\mathbf{v}_1, \ldots, \mathbf{v}_r$, and the last $m-r$ principle components are assumed to contain only random fluctuations. Therefore, $\mathbf{y}_{i*} = (y_{i1}, \ldots, y_{im})^T$ can be decomposed into normal and abnormal subspaces,

$$\mathbf{y}_{i*} = \mathbf{y}_{i,normal} + \mathbf{y}_{i,anomaly} \tag{4}$$

where $\mathbf{y}_{i,normal} = \mathbf{P}\mathbf{P}^T\mathbf{y}_{i*}$ and $\mathbf{y}_{i,anomaly} = (\mathbf{I} - \mathbf{P}\mathbf{P}^T)\mathbf{y}_{i*}$ with $\mathbf{P} = [\mathbf{v}_1, \ldots, \mathbf{v}_r]$. The distance of a measurement vector $\mathbf{y}_{i*}$ from the normal pattern can be computed as,

$$d_Y(\mathbf{y}_{i*}) = \|\mathbf{y}_{i,anomaly}\| = \sqrt{\sum_{j=r+1}^{m} (\mathbf{v}_j^T\mathbf{y}_{i*})^2}. \tag{5}$$

The distance $d_Y(\mathbf{y}_{i*})$ equals to the squared prediction error (SPE) [21]. The observed traffic is considered malicious if

$$d_Y(\mathbf{y}_{i*}) > Q_\varrho, \tag{6}$$

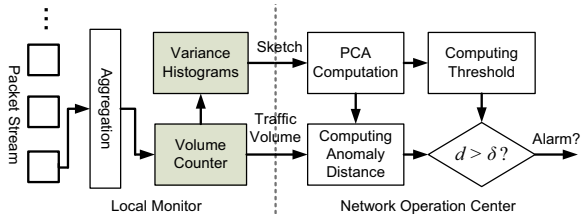

Figure 2. System model of the sketch-based algorithm

where Q_ϱ denotes the threshold that is computed by the *Q-statistic* developed by Jackson and Mudholkar [21].

$$Q_\varrho^2 = \phi_1 \left[\frac{c_\varrho \sqrt{2\phi_2 h_0^2}}{\phi_1} + 1 + \frac{\phi_2 h_0 (h_0 - 1)}{\phi_1^2} \right]^{1/h_0}, \quad (7)$$

where $c_\varrho = 1 - \varrho$,

$$h_0 = 1 - \frac{2\phi_1 \phi_3}{3\phi_2^2}, \quad \phi_k = \sum_{j=r+1}^{m} \sigma_j^{2k} (k = 1, 2, 3), \quad (8)$$

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

$$\sigma_j = \frac{1}{\sqrt{n-1}} \|\mathbf{Y}\mathbf{v}_j\| = \frac{\eta_j}{\sqrt{n-1}}. \quad (9)$$

D. Objective of this research

Our algorithm aims at computing PCA in a distributed system, and further detecting traffic anomalies. Firstly, the computation should be very efficient at both the local monitors and the NOCs. Secondly, the algorithm can be implemented in a distributed environment, which requires careful considerations about the storage and communication overhead. Last but not the least, it should support continuous updating to adjust the traffic anomaly detector due to the evolution of the traffic.

IV. Sketch-based Algorithm

The system model of our sketch-based algorithm is shown in Fig. 2, which utilizes the random projection [22] and the variance estimation [23] to reduce the computation complexity. There are five modules, each of which is described in the following subsection.

A. 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 aggregated flow. The volume counter maintains a bucket for each flow. A bucket U_j stores the traffic volume of the j-th flow at the current time interval. When a pair $(FlowID, Size)$ with $FlowID = j$ comes at the current time interval, the corresponding bucket U_j will be increased by $Size$. When a time interval ends, it just reports the traffic volume to the Variance Histogram (VH) and the NOC. Next, the value in the bucket is set to zero for the next interval.

Step1: *Check the time stamp of the last bucket* B_{Nj}
if($\tau_{Nj} \le t - n$) { delete B_{Nj}; }
Step2: *Create a new bucket* B_{1j}
$\tau_{1j} = t$; $n_{1j} = 1$; $\mu_{1j} = x_{tj}$; V_{1j}=0;
for($k = 1, \ldots, l$) { $Z_{1kj} = x_{tj} r_{tk}$; $R_{1kj} = r_{tk}$; }
Step3: *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; }
  if($n_A \le \frac{\varepsilon}{10} n_B$ and $V_{A \cup B} - V_B \le \frac{\varepsilon}{5} V_B$) {
    delete $B_{(p+2)j}$; $B_{(p+1)j} = B_A$;
  }
  else { $p = p + 1$; $B_B = B_B \cup B_{pj}$; }
}

Figure 3. Procedures for updating VH

B. Variance Histograms

A VH contains a list of buckets for each flow, which is maintained by the variance estimation algorithm in Fig. 3. The traffic volume x_{ij} at each time interval is treated as a data element for the variance computation. Given a sequence of data elements $\{x_{(t-n+1)j}, \ldots, x_{tj}\}$, the variance is defined as

$$V_{tj} = \sum_{i=t-n+1}^{t} (x_{ij} - \bar{x}_{tj})^2 \quad (10)$$

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}.

- τ_{pj}: time stamp;
- n_{pj}: total number of data elements in the subsequence;
- μ_{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. Firstly, when a new data element x_{tj} comes, the current time stamp is updated to t. We check the oldest bucket B_{Nj} and delete it if it is expired, where N denotes the number of buckets in the list. Secondly, 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 $B_B = \cup_{q=1}^{p} B_{qj}$. 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: $V_{A \cup B} - V_B \le \frac{\varepsilon}{5} V_B$.
- Rule 2: $n_A \le \frac{\varepsilon}{10} n_B$.
- Rule 2: $n_A + n_B \le n/2$.

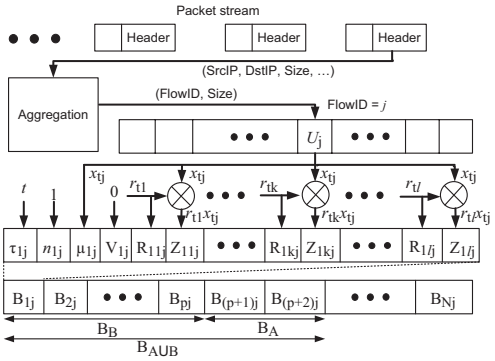

Figure 4. Sketch computation with VH

When two adjacent buckets B_{pj} and B_{qj} merge into a new bucket $B_{(p\cup q)j}$, 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 following,

$$n_{(p\cup q)j} = n_{pj} + n_{qj}, \tag{11}$$

$$\mu_{(p\cup q)j} = \frac{n_{pj}\mu_{pj} + n_{qj}\mu_{qj}}{n_{pj} + n_{qj}}, \tag{12}$$

$$V_{(p\cup q)j} = V_{pj} + V_{qj} + \frac{n_{pj}n_{qj}}{n_{pj} + n_{qj}}(\mu_{pj} - \mu_{qj})^2, \tag{13}$$

$$Z_{(p\cup q)kj} = Z_{pkj} + Z_{qkj}, \tag{14}$$

$$R_{(p\cup q)kj} = R_{pkj} + R_{qkj}. \tag{15}$$

Let $B_{all,j} = \cup_{p=1}^{N} B_{pj}$ denote the bucket by merging all buckets together, and $\hat{V} = V_{all,j}$ be the estimated variance. We get the following result [23].

Lemma 1: Variance Histogram maintains a ε-approximate variance,

$$(1-\varepsilon)V \le \hat{V} \le V, \tag{16}$$

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

At each local monitor, we implement a VH for each flow and n pseudo random number generators shared by all flows among local monitors. The architecture for the sketch computation at a local monitor is shown in Fig. 4. The volume counter only uses a bucket to maintain the traffic volume at the current time interval t for each flow. When a time interval ends, the volume counter reports the traffic volume x_{tj} to VH_j. Then VH_j updates its buckets as shown in Fig. 3. At each time interval, we can compute an approximation of the sketch as,

$$\hat{z}_{kj} = \frac{1}{\sqrt{l}}(Z_{all,kj} - n_{all,j}\mu_{all,j}R_{all,kj}), \tag{17}$$

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

C. PCA Computation

The PCA computation is done in a lazy mode. If there is no anomaly, the NOC will use previous PCA result and don't require the current sketches from the local monitors. Otherwise, the NOC gets all sketches $\{\hat{z}_{kj} : k = 1,\ldots,l,\ j = 1,\ldots,m\}$ and the means $\mu_{all,j}$ from local monitors, and organizes them into a $l \times m$ matrix $\hat{\mathbf{Z}}$. PCA is applied on the matrix $\hat{\mathbf{Z}}$ and its SVD is

$$\hat{\mathbf{Z}} = \sum_j \hat{\lambda}_j \hat{\mathbf{b}}_j \hat{\mathbf{a}}_j^T. \tag{18}$$

D. Anomaly Distance

Given the principle components, i.e. $\hat{\mathbf{a}}_1,\ldots,\hat{\mathbf{a}}_m$, we choose the last few principle components to compute the anomaly distance of the traffic vector $\mathbf{y}_{i*} = \mathbf{x}_i - (\mu_{all,1},\ldots,\mu_{all,m})^T$,

$$d_{\hat{Z}}(\mathbf{y}_{i*}) = \sqrt{\sum_{j=r+1}^{m} (\hat{\mathbf{a}}_j^T \mathbf{y}_{i*})^2} \tag{19}$$

where r is an integer less than m.

Before computing the anomaly distance, we need to determine the size of the normal subspace, denoted by r. There are several techniques which can be used 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 3σ-heuristic that was used in [1]. The projection of the matrix $\hat{\mathbf{Z}}$ on the j-th principle component, i.e. $\hat{\mathbf{Z}}\hat{\mathbf{a}}_j$, is examined one by one. When a projection is found that the value of an element in $\hat{\mathbf{Z}}\hat{\mathbf{a}}_j$ exceeds $3\sigma_j$ from the mean, where σ_j denotes the standard deviation, this and all remaining principle components are selected to compute the anomaly distance.

E. Threshold Computation

The threshold computation is based on the fault detection in multivariate process control [21]. Because $\hat{\mathbf{a}}_1,\ldots,\hat{\mathbf{a}}_m$ are m orthonormal vectors, we get

$$\|\mathbf{y}_{i*}\| = \sqrt{\sum_{j=1}^{m} (\hat{\mathbf{a}}_j^T \mathbf{y}_{i*})^2}. \tag{20}$$

Let $\hat{\mathbf{Q}} = [\,\hat{\mathbf{a}}_1,\ldots,\hat{\mathbf{a}}_r\,]$, and then we have

$$d_{\hat{Z}}(\mathbf{y}_{i*}) = \sqrt{\mathbf{y}_{i*}^T\mathbf{y}_{i*} - \sum_{j=1}^{r} (\hat{\mathbf{a}}_j^T \mathbf{y}_{i*})^2} = \|(\mathbf{I} - \hat{\mathbf{Q}}\hat{\mathbf{Q}}^T)\mathbf{y}_{i*}\|. \tag{21}$$

We can compute the threshold δ_ϱ based on the *Q-statistic*.

$$\delta_\varrho^2 = \varphi_1 \left[\frac{c_\varrho \sqrt{2\varphi_2 h_1^2}}{\varphi_1} + 1 + \frac{\varphi_2 h_1 (h_1 - 1)}{\varphi_1^2} \right]^{1/h_1} \tag{22}$$

where $c_\varrho = 1 - \varrho$,

$$h_1 = 1 - \frac{2\varphi_1\varphi_3}{3\varphi_2^2}, \quad \varphi_k = \frac{1}{(n-1)^k} \sum_{j=r+1}^{m} \hat{\lambda}_j^{2k} \quad (k = 1, 2, 3). \tag{23}$$

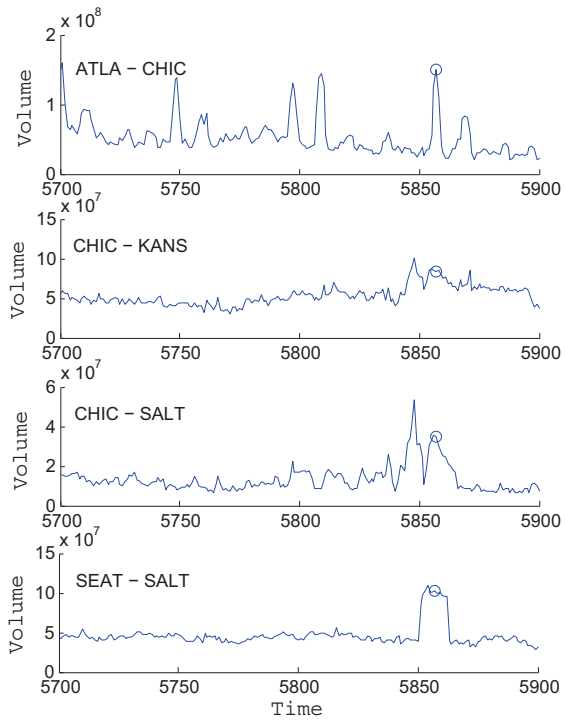

Figure 5. An example of coordinated traffic anomalies from the Internet2 network [6]

The NOC first computes the anomaly distance according to (21). If $d_{\hat{Z}}(\mathbf{y}_{i*}) < \delta_\varrho$, there won't be any traffic anomalies and the NOC will do nothing. If $d_{\hat{Z}}(\mathbf{y}_{i*}) > \delta_\varrho$, there could be two reasons. One is that the sketches at the NOC are out of date. The NOC will first get current sketches from local monitors, and recompute the anomaly distance and the threshold. The other is that there is an anomaly in the traffic. After the re-computation based on the current sketches, if the NOC still gets $d_{\hat{Z}}(\mathbf{y}_{i*}) > \delta_\varrho$, it will identify $\mathbf{y}_{i*}$ as a traffic anomaly. Otherwise, the NOC won't report any anomalies and will only update the PCA.

V. Theoretical Analysis

Our sketch-based algorithm detects traffic anomalies based on the same principles as Lakhina's method [1], which aims at detecting 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, as shown in Fig. 5. In fact, our algorithm is an approximation algorithm for Lakhina's method, which can improve the computation complexity. We first analyze the computation complexity and then prove the error bounds in our sketch-based algorithm.

A. Performance Analysis

Because the variance estimation algorithm only needs $O(\frac{1}{\varepsilon}\log n)$ space and $O(1)$ running time [23], we have the following theorem.

Theorem 1: The sketch-based method requires $O(w\log n)$ running time and $(w\log^2 n)$ space at the local monitor. The computation complexity is $O(m^2\log n)$ and the space requirement is $O(m\log n)$ at the NOC.

Proof: According to the algorithm in [23], we need only $O(1)$ running time to update the variance buckets. But we also need to update the sketches Z_{pkj} and the random numbers R_{pkj}. Therefore, the sketch-based method 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 flow. In general, the local monitor needs $O(w\log^2 n)$ space and $O(w\log n)$ running time for $l = O(\log n)$.

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 matrix $\hat{\mathbf{Z}}$, NOC needs $O(ml) = O(m\log n)$ memory space. At each time step, NOC uses the traffic vector $\mathbf{y}_{i*}$ and pre-computed principle components to detect traffic anomalies, which only requires $O(m^2)$ running time. In general, the sketch-based method requires $O(m^2\log n)$ running time and $O(m\log n)$ space at the NOC. ■

If the local monitors only have limited computation resources or bandwidth, we can maintain the VH and compute the sketches at the NOC side. In this way, the local monitors only need to implement the Volume Counter which only requires $O(1)$ running time to process each packet. The NOC needs $O(m\log n)$ running time and $O(m\log^2 n)$ space in this case.

B. Error Bounds

In this subsection, we explain why our method can compute principal components based on the sketches and further detect traffic anomalies like Lakhina's method. Due to the page limitation, we only provide the basic ideas to prove our results. Before the proof of our algorithm, we first provide some properties of the random projection of the traffic matrix $\mathbf{Y}$.

Let $\mathbf{R}$ be an $n\times l$ random matrix, which consists of the random number r_{ik} from the standard normal distribution. We define the random projection of $\mathbf{Y}$ as a matrix $\mathbf{Z}$,

$$\mathbf{z}_j = \frac{1}{\sqrt{l}}\mathbf{R}^T\mathbf{y}_j \quad \text{and} \quad \mathbf{Z} = \frac{1}{\sqrt{l}}\mathbf{R}^T\mathbf{Y}, \tag{24}$$

where $\mathbf{y}_j$ and $\mathbf{z}_j$ are a column in $\mathbf{Y}$ and $\mathbf{Z}$, respectively. The vector $\mathbf{z}_j$ is also called the random projection of $\mathbf{y}_j$, which has the following properties [22].

Lemma 2: Let $\mathbf{R}$ be an $n\times l$ random matrix from the standard normal distribution and $\mathbf{z}_j = \frac{1}{\sqrt{l}}\mathbf{R}^T\mathbf{y}_j$. We have

- $E(\|\mathbf{z}_j\|^2) = \|\mathbf{y}_j\|^2$
- $P(|\|\mathbf{z}_j\|^2 - \|\mathbf{y}_j\|^2| \geq \varepsilon\|\mathbf{y}_j\|^2) < 2e^{-(\varepsilon^2-\varepsilon^3)\frac{l}{4}}$

where ε is an arbitrary positive constant.

Besides the standard normal distribution, there are several probability distributions which have been proposed for the

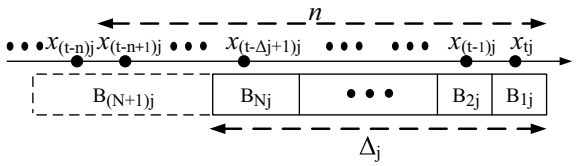

Figure 6. An illustration of sketch approximation

random projection. Alon [24] introduced the tug-of-war algorithm, where the random matrix $\mathbf{R}$ is generated from the probability distribution,

$$r_{ik} = \begin{cases} -1 & \text{with probability } 1/2 \\ +1 & \text{with probability } 1/2 \end{cases}. \quad (25)$$

Later, Achlioptas [25] gave a more efficient algorithm, i.e. the sparse random projection, in which $r_{ik} = -1$, 0, or 1 with a probability $\frac{1}{2s}$, $1-\frac{1}{s}$ and $\frac{1}{2s}$, respectively, where s is an integer. In the sparse random projection, only $1/s$ of the data need to be processed. Recently, the very sparse random projection has been recommended by Li [26], which uses $\mathbf{R}$ of entries in $\{-1, 0, 1\}$ with probability $\{\frac{1}{2\sqrt{n}}, 1-\frac{1}{\sqrt{n}}, \frac{1}{2\sqrt{n}}\}$. For the sparse random projection, we have the following properties,

Lemma 3: Let $\mathbf{R}$ be an $n \times l$ random matrix with entries in $\{-1, 0, 1\}$ with probabilities $\{1/2s, 1-1/s, 1/2s\}$ and $\mathbf{z}_j = \frac{1}{\sqrt{l}}\mathbf{R}^T\mathbf{y}_j$. For $\forall \varepsilon > 0$, we have

- *$E(\|\mathbf{z}_j\|^2) = \|\mathbf{y}_j\|^2$;*
- *$P(|\|\mathbf{z}_j\|^2 - \|\mathbf{y}_j\|^2| \geq \varepsilon\|\mathbf{y}_j\|^2) < 2e^{-(\varepsilon^2/2-\varepsilon^3/3)\frac{l}{2}}$.*

In the following part, we can use either the standard normal distribution or the sparse random projection, both of which give the same result.

The sketch $\hat{z}_{kj}$ is a sketch of a subsequence of the traffic volumes within the sliding window as shown in Fig. 6. We organize $\hat{z}_{kj}$ into a $l \times m$ matrix $\hat{\mathbf{Z}}$. Based on the properties of the random projection, we can prove that our sketch $\hat{\mathbf{z}}_j$ has similar properties to the random projection.

Lemma 4: Let r_{ik} for $k = 1, \ldots, l$, be generated from the probability distribution of the random projection, and $\hat{z}_{kj}$ be the sketch maintained by our algorithm. If $l > C\frac{\log n}{\varepsilon^2}$, we have

$$|\|\hat{\mathbf{z}}_j\|^2 - \|\mathbf{y}_j\|^2| \leq 2\varepsilon\|\mathbf{y}_j\|^2 \quad (26)$$

with a probability at least $1 - 2e^{-\frac{C}{4}\log n}$.

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

$$(1-\varepsilon)V < \hat{V} < V \quad (27)$$

according to [23]. Let $\hat{\mathbf{y}}_j$ denotes the subsequence maintained in the VH_j, and we have

$$\|\hat{\mathbf{y}}_j - \mathbf{y}_j\|^2 = |\hat{V} - V| < \varepsilon V = \varepsilon\|\mathbf{y}_j\|^2. \quad (28)$$

According to (17), we have

$$\hat{\mathbf{z}}_j - \mathbf{z}_j = \frac{1}{\sqrt{l}}\mathbf{R}(\hat{\mathbf{y}}_j - \mathbf{y}) \quad (29)$$

where $\hat{\mathbf{z}}_j$ and $\mathbf{z}_j$ are the j-th column in $\hat{\mathbf{Z}}$ and $\mathbf{Z}$, respectively. We apply the properties of the random projection on the vector $\hat{\mathbf{z}}_j - \mathbf{z}_j$, and get

$$|\|\hat{\mathbf{z}}_j\|^2 - \|\mathbf{y}_j\|^2| \leq 2\varepsilon\|\mathbf{y}\|_j^2 \quad (30)$$

with a probability $1 - 2e^{-\frac{C}{4}\log n}$. ■

Based on the above lemma, we can get

$$\|\hat{\mathbf{Z}} - \mathbf{Z}\|_F^2 \leq \varepsilon\|\mathbf{Y}\|_F^2, \quad (31)$$

where $\|\mathbf{X}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n x_{ij}^2}$ is the Frobenius norm of a matrix $\mathbf{X} \in \mathcal{R}^{n\times m}$. Therefore, we have

$$\begin{aligned} \|\hat{\mathbf{Z}}^T\hat{\mathbf{Z}} - \mathbf{Z}^T\mathbf{Z}\|_F &\leq \|\hat{\mathbf{Z}}^T\hat{\mathbf{Z}} - \hat{\mathbf{Z}}^T\mathbf{Z}\|_F + \|\hat{\mathbf{Z}}^T\mathbf{Z} - \mathbf{Z}^T\mathbf{Z}\|_F \\ &\leq 2\varepsilon\|\mathbf{Y}^T\mathbf{Y}\|_F \end{aligned} \quad (32)$$

Let $\hat{\mathbf{Z}} = \sum_j \hat{\lambda}_j\hat{\mathbf{b}}_j\hat{\mathbf{a}}_j^T$ and $\mathbf{Y} = \sum_j \eta_j\mathbf{u}_j\mathbf{v}_j^T$, we first prove that the singular values are approximately preserved.

Lemma 5: If $l > C\frac{\log n}{\varepsilon^2}$ for a large enough constant C and an arbitrary positive constant ε,

$$(1-3\varepsilon)\sum_{j=1}^r \eta_j^2 \leq \sum_{j=1}^r \hat{\lambda}_j^2 \leq (1+3\varepsilon)\sum_{j=1}^r \eta_j^2 \quad (33)$$

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

Proof: Because $\hat{\lambda}_1^2, \ldots, \hat{\lambda}_r^2$ are the first r largest eigenvalues of the matrix $\hat{\mathbf{Z}}^T\hat{\mathbf{Z}}$ and $\mathbf{v}_1, \ldots, \mathbf{v}_m$ are an orthonormal set of vectors, we have

$$\begin{aligned} \sum_{j=1}^r \hat{\lambda}_j^2 &\geq \sum_{j=1}^r \mathbf{v}_j^T(\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})\mathbf{v}_j \geq \sum_{j=1}^r \mathbf{v}_j^T(\mathbf{Z}^T\mathbf{Z} - 2\varepsilon\mathbf{Y}^T\mathbf{Y})\mathbf{v}_j \\ &= \sum_{j=1}^r \eta_j^2\|\frac{1}{\sqrt{l}}\mathbf{R}^T\mathbf{u}_j\|^2 - 2\varepsilon\sum_{j=1}^r \eta_j^2. \end{aligned} \quad (34)$$

According to the properties of the random projection,

$$\sum_{j=1}^r \hat{\lambda}_j^2 \geq (1-3\varepsilon)\sum_{j=1}^r \eta_j^2. \quad (35)$$

We also know that $\hat{\lambda}_j^2 \leq (1+3\varepsilon)\|\mathbf{Y}\hat{\mathbf{a}}_j\|^2$, and can further get

$$\sum_{j=1}^2 \hat{\lambda}_j^2 \leq (1+3\varepsilon)\sum_{j=1}^r \eta_j^2. \quad (36)$$

■

Using the above lemmas, we can 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 $\|\mathbf{Y}\|_F^2 = \sum_{j=1}^m \eta_j^2$, we can prove the following lemma.

Lemma 6: Let $\mathbf{V} = \mathbf{Y}^T\mathbf{Y}$ *and* $\hat{\mathbf{A}} = \hat{\mathbf{Z}}^T\hat{\mathbf{Z}}$. *If* $l > C\frac{\log n}{\varepsilon^2}$ *for a large enough constant* C, *then with the probability* $1 - 2e^{-\frac{C}{4}\log n}$, *we have*

$$\|\mathbf{V} - \hat{\mathbf{A}}\|_F \leq \sqrt{6\varepsilon}\|\mathbf{Y}\|_F^2. \tag{37}$$

Proof: Firstly, because $\hat{\mathbf{a}}_1, \ldots, \hat{\mathbf{a}}_m$ are an orthonormal set of vectors,

$$\|\mathbf{V} - \hat{\mathbf{A}}\|_F^2 = \sum_{j=1}^{m} \|(\mathbf{V} - \hat{\mathbf{A}})\hat{\mathbf{a}}_j\|^2. \tag{38}$$

For each $j = 1, \ldots, r$, we have

$$\|(\mathbf{V} - \hat{\mathbf{A}})\hat{\mathbf{a}}_j\|^2 = \|\mathbf{V}\hat{\mathbf{a}}_j\|^2 + \hat{\lambda}_j^4 - 2\hat{\lambda}_j^2\hat{\mathbf{a}}_j^T\mathbf{V}\hat{\mathbf{a}}_j. \tag{39}$$

Therefore,

$$\|\mathbf{V} - \hat{\mathbf{A}}\|_F^2 \leq \sum_{j=1}^{m}\left(\eta_j^4 + \hat{\lambda}_j^4 - \frac{2}{1+3\varepsilon}\hat{\lambda}_j^4\right). \tag{40}$$

Similarly, we have

$$\begin{aligned}\|\mathbf{V} - \hat{\mathbf{A}}\|_F^2 &= \sum_{j=1}^{m}\left(\eta_j^4 + \hat{\lambda}_j^4 - 2\eta_j^2\mathbf{v}_j^T\hat{\mathbf{A}}\mathbf{v}_j\right) \\ &\leq \sum_{j=1}^{m}\left(\eta_j^4 + \hat{\lambda}_j^4 - 2(1-3\varepsilon)\eta_j^4\right). \end{aligned}\tag{41}$$

Finally, based on (40) and (41), we get

$$\|\mathbf{V}-\hat{\mathbf{A}}\|_F^2 \leq \sum_{j=1}^{m} 3\varepsilon\left(\eta_j^4 + \frac{1}{1+3\varepsilon}\hat{\lambda}_j^4\right) \leq 3\varepsilon\sum_{j=1}^{m}\left(\hat{\lambda}_j^4 + \eta_j^4\right). \tag{42}$$

Based on Lemma 5 and $\|\mathbf{Y}\|_F^2 = \sum_{j=1}^{m}\eta_j^2$, we get the following result

$$\|\mathbf{V} - \hat{\mathbf{A}}\|_F^2 \leq 6\varepsilon\|\mathbf{Y}\|_F^4. \tag{43}$$

■

Next, we want to prove that $d_Y(\mathbf{y})$ can be approximated by $d_{\hat{Z}}(\mathbf{y})$. For this purpose, we first cite a theorem from the matrix perturbation theory. The column space of a matrix $\mathbf{M}$ is the subspace spanned by the columns, which is denoted by $\mathcal{R}(\mathbf{M}) = \{\mathbf{Mx} : \mathbf{x} \in \mathcal{R}^m\}$. The set of all eigenvalues of the matrix $\mathbf{M}$ is denoted by $\mathcal{L}(\mathbf{M}) = \{\lambda : \mathbf{Mx} = \lambda\mathbf{x}, \exists\mathbf{x} \neq \mathbf{0}\}$. And Θ denotes the canonical angle between two subspaces, $\Theta(\mathcal{M}, \mathcal{N}) = \sin^{-1}\mathbf{\Sigma}$, where $\mathcal{M}$ and $\mathcal{N}$ are r-dimensional subspaces of $\mathcal{R}^m$. The columns of their orthogonal bases can be transformed by a unitary matrix to

$$\begin{pmatrix}\mathbf{I}\\ \mathbf{0}\\ \mathbf{0}\end{pmatrix} \text{ and } \begin{pmatrix}\mathbf{\Gamma}\\ \mathbf{\Sigma}\\ \mathbf{0}\end{pmatrix} \tag{44}$$

if $2r \leq m$, or

$$\begin{pmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\\ \mathbf{0} & \mathbf{0}\end{pmatrix} \text{ and } \begin{pmatrix}\mathbf{\Gamma} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\\ \mathbf{\Sigma} & \mathbf{0}\end{pmatrix} \tag{45}$$

if $2r > m$. Then we have the following property [27].

Lemma 7: Matrix Perturbation: Let $\mathbf{M}$ *have the spectral resolution*

$$\begin{pmatrix}\mathbf{U}_1^T\\ \mathbf{U}_2^T\end{pmatrix}\mathbf{M}\,(\mathbf{U}_1\,\mathbf{U}_2) = \begin{pmatrix}\mathbf{L}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{L}_2\end{pmatrix} \tag{46}$$

where $(\mathbf{U}_1, \mathbf{U}_2)$ *is unitary with* $\mathbf{U}_1 \in \mathcal{R}^{n\times r}$. *Let* $\mathbf{B} \in \mathcal{R}^{n\times r}$ *have orthonormal columns, and for any symmetric* $\mathbf{H}$ *of order* r, *let* $\mathbf{E} = \mathbf{MB} - \mathbf{BH}$. *If* $\nu = \min|\mathcal{L}(\mathbf{L}_2) - \mathcal{L}(\mathbf{H})| > 0$, *then we have*

$$\|\sin\Theta[\mathcal{R}(\mathbf{U}_1), \mathcal{R}(\mathbf{B})]\|_F \leq \frac{\|\mathbf{E}\|_F}{\nu}. \tag{47}$$

We apply the above lemma to the covariance matrices $\hat{\mathbf{A}}$ and $\mathbf{V}$, and then we can get an error bound for the anomaly distance.

Theorem 2: If $l > C\frac{\log n}{\varepsilon^2}$ *for a large enough constant* C, *then*

$$|d_{\hat{Z}}(\mathbf{y}) - d_Y(\mathbf{y})| \leq \frac{2\sqrt{3\varepsilon}}{|\eta_{r+1}^2 - \eta_r^2|}\|\mathbf{Y}\|_F^2\|\mathbf{y}\| \tag{48}$$

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

Proof: Let $\hat{\mathbf{Q}} = [\,\hat{\mathbf{a}}_1, \cdots, \hat{\mathbf{a}}_r\,]$, $\hat{\mathbf{Q}}_c = [\,\hat{\mathbf{a}}_{r+1}, \cdots, \hat{\mathbf{a}}_m\,]$, $\mathbf{P} = [\,\mathbf{v}_1, \cdots, \mathbf{v}_r\,]$, and $\mathbf{P}_c = [\,\mathbf{v}_{r+1}, \cdots, \mathbf{v}_m\,]$. We have the following spectral resolutions,

$$\begin{pmatrix}\hat{\mathbf{Q}}^T\\ \hat{\mathbf{Q}}_c^T\end{pmatrix}\hat{\mathbf{A}}\left(\hat{\mathbf{Q}}\,\hat{\mathbf{Q}}_c\right) = \begin{pmatrix}\mathbf{\Lambda}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{\Lambda}_2\end{pmatrix}, \tag{49}$$

$$\begin{pmatrix}\mathbf{P}^T\\ \mathbf{P}_c^T\end{pmatrix}\mathbf{V}\,(\mathbf{P}\,\mathbf{P}_c) = \begin{pmatrix}\mathbf{M}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{M}_2\end{pmatrix}, \tag{50}$$

where $\mathbf{\Lambda}_1 = diag(\hat{\lambda}_1^2, \ldots, \hat{\lambda}_r^2)$, $\mathbf{\Lambda}_2 = diag(\hat{\lambda}_{r+1}^2, \ldots, \hat{\lambda}_m^2)$, $\mathbf{M}_1 = diag(\eta_1^2, \ldots, \eta_r^2)$, and $\mathbf{M}_2 = diag(\eta_{r+1}^2, \ldots, \eta_m^2)$. Here $diag(\cdot)$ denotes a diagonal matrix.

Let $\mathbf{E} = \mathbf{V}\hat{\mathbf{Q}} - \hat{\mathbf{Q}}\mathbf{\Lambda}_1$. Because $\hat{\mathbf{Q}}\mathbf{\Lambda}_1 = \hat{\mathbf{A}}\hat{\mathbf{Q}}$, we have $\mathbf{E} = \mathbf{V}\hat{\mathbf{Q}} - \hat{\mathbf{A}}\hat{\mathbf{Q}}$. According to Lemma 7, we have

$$\|\sin\Theta[\mathcal{R}(\mathbf{P}), \mathcal{R}(\hat{\mathbf{Q}})]\|_F \leq \frac{\|\mathbf{E}\|_F}{\nu} = \frac{\|\mathbf{V} - \hat{\mathbf{A}}\|_F}{\nu} \tag{51}$$

where $\nu = |\eta_{r+1}^2 - \lambda_r^2| \approx |\eta_{r+1}^2 - \eta_r^2|$.

The project matrices of $\mathcal{R}(\mathbf{P})$ and $\mathcal{R}(\hat{\mathbf{Q}})$ are $\mathbf{PP}^T$ and $\hat{\mathbf{Q}}\hat{\mathbf{Q}}^T$, respectively. Then, according to [27], we have

$$\begin{aligned}\|\mathbf{PP}^T - \hat{\mathbf{Q}}\hat{\mathbf{Q}}^T\|_F &= \sqrt{2}\|\sin\Theta[\mathcal{R}(\mathbf{P}), \mathcal{R}(\hat{\mathbf{Q}})]\|_F \\ &\leq \sqrt{2}\frac{\|\mathbf{V} - \hat{\mathbf{A}}\|_F}{\nu}. \end{aligned}\tag{52}$$

Then we get

$$\begin{aligned}|d_{\hat{Z}}(\mathbf{y}) - d_Y(\mathbf{y})| &\leq \|(\mathbf{I} - \hat{\mathbf{Q}}\hat{\mathbf{Q}}^T)\mathbf{y} - (\mathbf{I} - \mathbf{PP}^T)\mathbf{y}\| \\ &\leq \frac{2\sqrt{3\varepsilon}}{|\eta_{r+1}^2 - \eta_r^2|}\|\mathbf{Y}\|_F^2\|\mathbf{y}\|. \end{aligned}\tag{53}$$

■

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

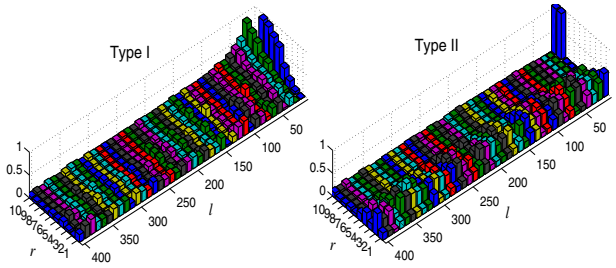

Figure 7. Type I and Type II errors with 5-minutes interval

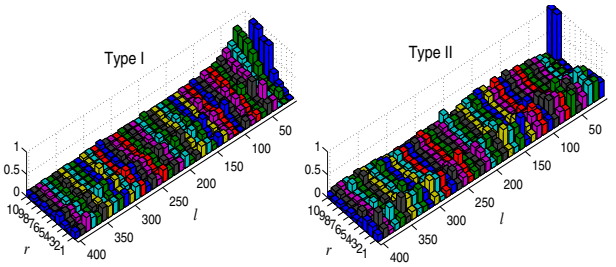

Figure 8. Type I and Type II errors with 1-minute interval

VI. Experimental Evaluation

In this section, we evaluate our methods on the data from Abilene Observatory Data Collections [6]. The data collection is running on nine routers after Feb 2008, i.e. ATLA, CHIC, HOUS, KANS, LOSA. NEWY, SALT, SEAT, and WASH. We use an one-month data collection between June 9th, 2008 and July 9th, 2008. The packets are aggregated into origin-destination (OD) flows based on both BGP and ISIS routing information. Because the traffic anomalies are unknown and hard to be determined only based on the Netflow traces, we first apply Lakhina's method to detect anomalies by using a fix size r for the normal subspace. And these detected anomalies are used as the 'real' anomalies to evaluate the detection accuracy of our algorithm. The length of the sliding window is two weeks. Currently, there is no good method to choose r, and thus we try possible values for the size r from 1 to 10. We set $\varepsilon = 0.01$ in the VH algorithm and $\varrho = 0.01$ for the threshold computation in the *Q-statistics*. We use our sketch-based method as an approximation of the Lakhina's method and compute both Type I errors and Type II errors with different size l of the sketches, i.e. $l = 10, 20, \ldots$

$$\text{Type I} = \frac{\text{number of false anomalies}}{\text{total number of true normal observations}},$$

$$\text{Type II} = \frac{\text{number of false normal observations}}{\text{total number of true anomalies}}.$$

We show Type I and Type II errors with five-minutes interval and one-minute interval in Fig.7 and Fig.8, respectively. When the size r is too small, we get large errors due to that the normal traffic cannot be retained in the normal subspace.

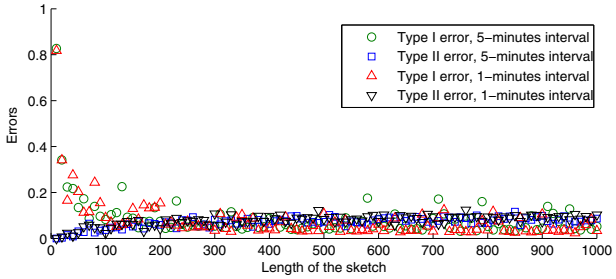

Figure 9. Type I and Type II errors with $r = 6$

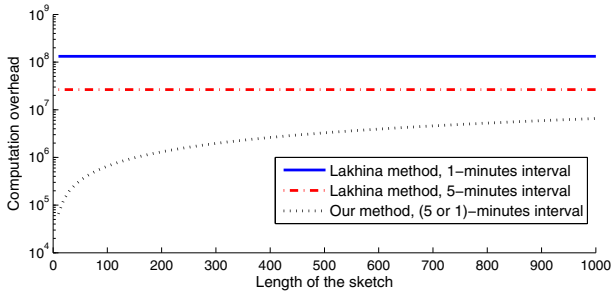

Figure 10. The computation overhead at the NOC in the logarithmic scale

If $r \geq 5$, 90% energy is retained in the normal subspace, i.e. $\|\mathbf{y}_{i,normal}\|^2/\|\mathbf{y}_i\|^2 \simeq 0.9$. In this case, both Type I and Type II errors decrease quickly at the beginning and then reach a nearly optimal value.

We check the eigenvalues of the measurement matrix $\mathbf{Y}$, and choose the size of the normal subspace as $r = 6$ which is proper for our data. We show the Type I and Type II errors with $r = 6$ and $l = 10, 20, \ldots, 1000$ in Fig.9. If the size l of the sketch is more than 200, there is no remarkable decrease in both errors. The computation overhead at the NOC is determined by the PCA computation, which requires m^2n and m^2l for both Lakhina's method and our method respectively. We show the computation overhead at the NOC in Fig. 10. Our method requires much less computation than Lakhina's method.

If we want to further reduce the errors, we must set smaller values for ε and ϱ. We can bound the error of the estimated threshold and the distance in terms of η_j and $\mathbf{Y}$. Therefore, the accuracy of our algorithm depends on the properties of the covariance matrix $\mathbf{V}$ of the traffic measurements. If all the eigenvalues of $\mathbf{V}$ are close to each other, we first have $\nu = \eta_{r+1}^2 - \eta_r^2$ that can be very small. Therefore, we cannot bound the error of the distance. When η_j is close to each other, the value of the threshold δ_ϱ can also be far from Q_ϱ. In fact, the PCA-based detection method also has a high false alarm rate because the traffic measurements cannot reside in a low dimensional subspace.

VII. Conclusion and Future Work

In this paper, we study the network-wide traffic anomaly detection problem. Our algorithm achieves $O(w \log n)$ running time and $O(w \log^2 n)$ 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 makes the ISPs be able to implement the detection method by paying careful consideration about the space requirement, the communication cost, and other resources over a distributed computing environment. In the future, we will apply our sketch-based method on various statistical anomaly detection methods, e.g. Markov models, Bayesian networks, etc.

Acknowledgment

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

This work was partially supported by NSF under grants No. CNS-0644238, CNS-0626822, and CNS-0831470. We appreciate anonymous reviewers for their valuable suggestions and comments.

References

[1] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 219–230, 2004.

[2] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of pca for traffic anomaly detection," *SIGMETRICS '07*, pp. 109–120, 2007.

[3] B. I. P. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S. Lau, S. Rao, N. Taft, and J. D. Tygar, "Stealthy poisoning attacks on pca-based anomaly detectors," *SIGMETRICS '09 (poster)*, 2009.

[4] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," *SIGCOMM '05*, pp. 217–228, 2005.

[5] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," *SIGMETRICS '04/Performance '04*, pp. 61–72, 2004.

[6] "Abilene observatory data collections," *www.internet2.edu/observatory/*.

[7] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina, "Detection and identification of network anomalies using sketch subspaces," *IMC '06*, pp. 147–152, 2006.

[8] L. Huang, X. L. Nguyen, M. Garofalakis, J. Hellerstein, M. Jordan, A. Joseph, and N. Taft, "Communication-efficient online detection of network-wide anomalies," *INFOCOM '07*, pp. 134–142, 2007.

[9] Y. Huang, N. Feamster, A. Lakhina, and J. J. Xu, "Diagnosing network disruptions with network-wide analysis," *SIGMETRICS '07*, pp. 61–72, 2007.

[10] P. Chhabra, C. Scott, E. Kolaczyk, and M. Crovella, "Distributed spatial anomaly detection," *INFOCOM '08*, pp. 1705–1713, 2008.

[11] J. Kline, S. Nam, P. Barford, D. Plonka, and A. Ron, "Traffic anomaly detection at fine time scales with bayes nets," *ICIMP '08*, pp. 37–46, 2008.

[12] D. Brauckhoff, K. Salamatian, and M. May, "Applying pca for traffic anomaly detection: Problems and solutions," in *INFOCOM 2009, IEEE*, 2009, pp. 2866–2870.

[13] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," *IMW '02*, pp. 71–82, 2002.

[14] C. C. Aggarwal, *Data Streams: Models and Algorithms (Advances in Database Systems)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[15] K. Kumar, J. Xu, J. Wang, O. Spatschek, and L. Li, "Space-code bloom filter for efficient per-flow traffic measurement," *INFOCOM '04*, pp. 1762–1773, 2004.

[16] Q. G. Zhao, A. Kumar, J. Wang, and J. J. Xu, "Data streaming algorithms for accurate and efficient measurement of traffic and flow matrices," *SIGMETRICS '05*, pp. 350–361, 2005.

[17] S. Guha, D. Gunopulos, and N. Koudas, "Correlating synchronous and asynchronous data streams," *KDD '03*, pp. 529–534, 2003.

[18] S. Papadimitriou, J. Sun, and C. Faloutsos, "Streaming pattern discovery in multiple time-series," *VLDB '05*, pp. 697–708, 2005.

[19] S. Papadimitriou and P. Yu, "Optimal multi-scale patterns in time series streams," *SIGMOD '06*, pp. 647–658, 2006.

[20] M. Datar, A. Gionis, P. Indyk, and R. Motwani, "Maintaining stream statistics over sliding windows: (extended abstract)," *SODA '02*, pp. 635–644, 2002.

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

[22] S. S. Vempala, *The Random Projection Method*. Rhode Island: American Mathematical Society, 2004.

[23] L. Zhang and Y. Guan, "Variance estimation over sliding windows," *PODS '07*, pp. 225–232, 2007.

[24] N. Alon, P. B. Gibbons, Y. Matias, and M. Szegedy, "Tracking join and self-join sizes in limited storage," *PODS '99*, pp. 10–20, 1999.

[25] D. Achlioptas, "Database-friendly random projections: Johnson-lindenstrauss with binary coins," *Journal of computer and System Sciences*, vol. 66, no. 4, pp. 671–687, 2003.

[26] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," *KDD' 06*, pp. 287–296, 2006.

[27] G. Stewart and J. guang Sun, *Matrix perturbation theory*. Boston: Academic Press, 1990.