Sensitivity of PCA for Traffic Anomaly Detection

Haakon Ringberg
Department of Computer Science
Princeton University

Augustin Soule
Thomson Research

Jennifer Rexford
Department of Computer Science
Princeton University

Christophe Diot
Thomson Research

ABSTRACT

Detecting anomalous traffic is a crucial part of managing IP networks. In recent years, network-wide anomaly detection based on Principal Component Analysis (PCA) has emerged as a powerful method for detecting a wide variety of anomalies. We show that tuning PCA to operate effectively in practice is difficult and requires more robust techniques than have been presented thus far. We analyze a week of network-wide traffic measurements from two IP backbones (Abilene and Geant) across three different traffic aggregations (ingress routers, OD flows, and input links), and conduct a detailed inspection of the feature time series for each suspected anomaly. Our study identifies and evaluates four main challenges of using PCA to detect traffic anomalies: (i) the false positive rate is very sensitive to small differences in the number of principal components in the normal subspace, (ii) the effectiveness of PCA is sensitive to the level of aggregation of the traffic measurements, (iii) a large anomaly may inadvertently pollute the normal subspace, (iv) correctly identifying which flow triggered the anomaly detector is an inherently challenging problem.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations; C.4 [**Performance of Systems**]: Modeling Techniques

General Terms

Measurement, Performance, Reliability

Keywords

Network Traffic Analysis, Principal Component Analysis, Traffic Engineering

SIGMETRICS'07, June 12–16, 2007, San Diego, California, USA.

1. INTRODUCTION

Traffic anomalies, such as flash crowds, denial-of-service attacks, port scans, and the spreading of worms, can have detrimental effects on Internet services. Detecting and diagnosing these anomalies is critical to network operators, who must take corrective action to alleviate congestion, block attacks, and warn affected users. Sifting through an immense amount of measurement data to identify the anomalous traffic is an onerous task, best left to automated analysis. On the surface, anomaly detection seems straightforward: pick a statistical definition of an anomaly, feed the measurement data into a statistical-analysis technique, and classify the statistical outliers as anomalies. Unfortunately, anomaly detection is much more complicated than it seems. There are many ways to represent the traffic and pinpoint anomalies, each with its own set of assumptions, limitations, and tunable parameters that significantly affect the results. This paper focuses on that problem.

Principal Component Analysis [7] (PCA) is perhaps the best-known statistical-analysis technique for detecting network traffic anomalies. PCA is a dimensionality-reduction technique that returns a compact representation of a multi-dimensional dataset by reducing the data to a lower dimensional subspace. Recent papers in networking literature have applied PCA to the problem of traffic anomaly detection with promising initial results [14, 12, 10, 13]. Our research shows that a great deal of manual tuning is necessary to achieve such results, however, because PCA is very sensitive to its parameters and the proposed techniques for tuning them are inadequate. In this paper, we identify and evaluate four main challenges of using PCA for traffic anomaly detection:

The false-positive rate is very sensitive to the dimensionality of the normal subspace: PCA's effectiveness as a traffic anomaly detector is very sensitive to its two main tunable parameters—the dimensionality of the normal subspace (the top_k parameter) and the detection threshold. Previous research has required a great deal of manual tuning of these parameters. We show that the false-positive rate can vary by a factor of three or more within a small range of top_k values. The detection threshold, on the other hand, has a more predictable impact, and provides operators with an intuitive knob to strike a balance between the false-positive rate and the total number of detections.

The effectiveness of PCA is sensitive to the way the traffic measurements are aggregated: The large volume of IP flow traces must be aggregated before PCA is applied. We evaluate three different ways of aggregating the

data—by input link, by ingress router, and by OD pairs—and find that the choice has a significant impact on PCA's effectiveness. Choosing a representation that aggregates the data too much leads to highly smooth data that hides all but the most blatant anomalies, whereas aggregating too little often yields flows with wildly varying relative sizes and thus causes PCA to be overly sensitive to variations among the small flows. In addition, some representations of the traffic, such as OD flows, are only appropriate for very long traces, limiting the applicability of PCA.

Large anomalies can contaminate the normal subspace: Sufficiently large anomalies can inadvertently pollute PCA's normal subspace, thereby skewing the definition of normalcy and increasing the false-positive rate as a result. We support this contention with real measurements of a short-lived but drastic network event in the Geant network that goes largely undetected by PCA because of this phenomenon. This argues for preprocessing measurement data to detect, and filter, large anomalies before applying PCA.

Pinpointing the anomalous flows is inherently difficult: The problem of identifying which ingress router, for example, was responsible for a PCA detection is fundamentally hard. Unfortunately, there is no direct mapping between PCA's dimensionality-reduced subspace and the original spatial location of the anomaly. We show that the previously employed heuristic for associating a given PCA detection with specific location (e.g., an ingress router) relies on an assumption that does not hold in general, and has the side-effect of associating a large fraction of the detections with a very small set of locations.

In order to demonstrate these four points, we analyze one week of IP flow data for both the Geant and Abilene backbones, including a detailed examination of the feature time series for each detected anomaly to identify false positives. Comparing the results for Geant and Abilene allows us to conclude that the appropriate setting of PCA's tunable parameters varies from one network to another. Still, our comparison between Geant and Abilene illustrates that (i) the relative properties of traffic aggregations, in terms of false-positive rate and total number of detections, (ii) the difficulty of identifying which anomalous flow caused a PCA detection, and (iii) the contamination of PCA's normal subspace, appear to hold across networks.

The remainder of the paper is organized as follows. In Section 2 we present a brief overview of how to apply PCA to detect anomalous traffic. Next, Section 3 describes our software for aggregating the measurement data, applying PCA, and validating the resulting anomalies. Then, we evaluate the sensitivity of PCA to the top_k parameter and the detection threshold for the two networks and three traffic aggregations in Section 4. In Section 5 we introduce two limitations intrinsic to PCA, viz. contamination of the normal subspace, and identification of the flows responsible for triggering the detection of an anomaly. We present related research in Section 6 and conclude in Section 7.

2. PCA FOR TRAFFIC ANOMALIES

PCA is a dimensionality-reduction technique that has been applied to many kinds of data. In fact, PCA is the *optimal* such linear transform—that is, for any choice for the number of dimensions, PCA returns the subspace that retains the highest variance [7]. In this section, we describe how to use PCA to construct a model of "normal" traffic, and to detect and identify the statistical outliers.

2.1 Model Construction

Network-wide anomaly detection draws on measurement data from multiple locations and time periods. We define a *traffic matrix* as a timeseries of n measurement vectors $\vec{v_1}, \ldots, \vec{v_n}$, where each time-step i has m measurements (i.e., $|\vec{v_i}| = m$). We intentionally leave the precise meaning of the cell $v_{i,j}$ unspecified, since the choice may vary from one representation of the data to another. For example, $v_{i,j}$ could be the number of bytes or packets observed at time-step i at location j, or could be something more complex, such as the entropy of the distribution of source IP addresses in the traffic seen at location j during time interval i. Throughout the paper, we refer to the m columns of the traffic matrix as 'flows', so we can talk about the matrix without regard for how we choose to aggregate the data. When we need to refer to 'IP flows', we use that term explicitly.

When applied to a matrix, PCA returns a set of orthonormal vectors (called the principal components) such that for all $k \leq m$, the k-subspace defined by these vectors captures the highest variance in the original matrix. Adhering to previous terminology, we refer to the subspace defined by these first k principal components as the "normal subspace"; we refer to k itself as the top_k parameter. The basic underlying assumption of traffic anomaly detection is that the k-subspace corresponds to the primary regular trends (e.g., diurnal, weekly) of the traffic matrix. Previous work [14] has shown that traffic timeseries have low intrinsic dimensionality, which means that k can be a small number. In section 4.1 we investigate PCA's sensitivity to this parameter in the context of traffic anomaly detection.

Once the "normal" operation of the network has been gleaned from the traffic traces, one may assume that what is left is either statistical anomalies or mere noise. That is, when the normal subspace has been removed, one is left with a $(n-k)$-subspace that can then either be treated as wholly anomalous or be further split into a ℓ-subspace that is anomalous and a $(n-k-\ell)$-subspace of noise. These two or three subspaces are then the simplified model that is retained of the entire traffic trace. In constructing this model of the traffic, the m flows of the $m \times n$ traffic matrix can be thought of as random variables, of which there are n observations each. As such, it does not make sense to consider cases where $m > n$, i.e., one cannot draw statistically sound conclusions when one has fewer observations than one has variables[1]. We will therefore require that the number of time-steps n must be greater than or equal to the cardinality of the chosen traffic aggregation m, which we will demonstrate in section 3.3 to be a problem for certain traffic aggregations.

2.2 Detection and Identification

Classification of a new measurement vector $\vec{v}$, representing a given moment in time, occurs in relation to the model constructed in the previous step. $\vec{v}$ is projected onto the relevant subspaces in the model, which decomposes the vector into a linear combination of its normal and anomalous constituents. $\vec{v}$ is then classified as normal or anomalous depending on whether it is primarily expressed by the nor-

[1]Recent theoretical work [3] attempts to overcome this limitation in certain circumstances.

Network	Nodes	Sampling	Time Agg	Anon
Abilene	11	1%	5 min	11 bits
Geant	23	0.1%	15 min	0 bits

Table 1: Networks studied

mal or anomalous subspaces. The threshold that determines how statistically significant a given event (spike) must be for it to be flagged as anomalous is another parameter that can be tuned, and we investigate its impact in section 4.2.

Finally, if $\vec{v}$ is classified as anomalous, we must determine precisely which set of columns $C \subseteq [1, m]$ (i.e., flows) of the traffic matrix were primarily responsible for the detection. Knowing that an anomaly occurred at a particular time is typically not sufficient—knowing *where* it happened often matters as well. For example, the network operator may need to know which ingress routers were the entry point for the anomalous traffic. It is important to realize that this flow identification step is separate from PCA, and different heuristics may be employed here. It is a necessary step in the context of network traffic anomaly detection, however, and we will therefore evaluate the detector that is the combination of the above PCA technique and the heuristic employed by the line of work that followed Lakhina *et al.* [12, 10, 13, 15, 16]. We will further discuss this heuristic in section 5.2.

3. METHODOLOGY

We designed and implemented the architecture shown in figure 1 to evaluate PCA's effectiveness as a traffic anomaly detector. The diagram is organized into four sub-components, each of which will be individually explained in the following subsections. Specifically, section 3.1 will detail the data that were collected; section 3.2 explains how the data were preprocessed and aggregated according to either ingress routers, OD flows, or input links; section 3.3 deals with our interface with the PCA anomaly detector written by Lakhina *et al.* [12]; and section 3.4 describes how we labeled the detected anomalies as false positives or true positives.

3.1 Data Sources

For this work we studied both the Abilene [1] and Geant [6] networks. Their respective properties are summarized in table 1. Abilene is an 11-node research backbone that connects Internet2 universities and research labs across the continental United States. Abilene does not, however, provide transit services to the Internet at large; instead, its participants must maintain separate connections to the commodity Internet [2]. Geant is a 23-node network that connects national research and education networks representing 30 European countries; unlike Abilene, Geant does provide Internet connectivity to its participants.

Both the Abilene and Geant networks collect flow statistics using Juniper's J-Flow tool [8]. Abilene samples 1 out of every 100 packets for inclusion in the flow statistics whereas Geant samples packets at 1 out of 1000. In Abilene, packets are aggregated into five-minute time-bins compared to a fifteen-minute time window for Geant. Finally, Abilene anonymizes the last eleven bits of the IP address stored in the flow records to preclude a reader from identifying the source or destination host.

In order to aggregate the collected IP flows into OD-flows, we also need to parse the routing data from each network. Abilene deploys Zebra BGP monitors that record all BGP messages they receive. This means that for any ⟨ingress,prefix⟩ pair, it is sufficient to parse the BGP logs in order to identify the egress point of the IP flow. Geant has one Zebra BGP monitor embedded in an iBGP mesh that logs a single BGP record for all routers, which gives us a *set* of egress points for a given prefix. Subsequently we must parse the Geant IS-IS logs to produce a minimum-cost path from each ingress router to all egress routers, which, in conjunction with the set of egress routers for a given prefix, uniquely identifies the egress point for a given ⟨ingress,prefix⟩ pair.

For both networks, we studied a full week of data between November 21st and 27th of 2005, corresponding to 2016 datapoints for each flow in Abilene (e.g. 7 days × 24 hours × $\frac{60}{5}$ bins per hour) and 672 for Geant.

3.2 Timeseries Construction

In the following subsection we will elaborate on how the data was further preprocessed and transformed into entropy timeseries before being analyzed by the PCA anomaly detector itself.

3.2.1 Entropy Timeseries

Previous work [13] has demonstrated that entropy timeseries of the four main IP header features (source IP, destination IP, source port, and destination port) is a rich data type to analyze for traffic anomaly detection. That is, for every measurement vector $\vec{v_i}$ at time i there are four measurements $v_{i,j}, \ldots, v_{i,j+3}$ for every ingress router (for example). $v_{i,j}$ will be the entropy of the distribution of source IP addresses for this router, $v_{i,j+1}$ will be the entropy of the distribution of destination IP addresses for this router, etc.

$$H(X) = -\sum_{i=1}^{n} Pr[X = x_i] \log_2(Pr[X = x_i]) \qquad (1)$$

Entropy is studied because it provides a computationally efficient metric for estimating the dispersion or concentration in a distribution, and a wide variety of anomalies will impact the distribution of one of the discussed IP features. The entropy of a random variable X is defined in equation 1, where $Pr[X = x_i]$ is the probability of event $x_i \in X$ occurring. In our context, the events are observations of a given IP feature. For example, the probability of seeing port 80 is defined to be number of packets using port 80 divided by the total number of packets in the given time interval. A sudden flash crowd to a webserver, for example, will cause a specific destination IP (the webserver) and destination port (port 80) to become much more prevalent than in previous time-steps, which will cause a decrease in the destination IP and destination port entropy timeseries, respectively, and hence allow us to detect it. A more complete explanation of the benefits of using entropy for traffic anomaly detection can be found in [13].

3.2.2 Traffic Aggregation

In addition to studying two networks, we also studied several traffic aggregations. That is, IP flow traces must be further aggregated so that statistical methods such as PCA can detect correlations and periodic trends in the data. If the

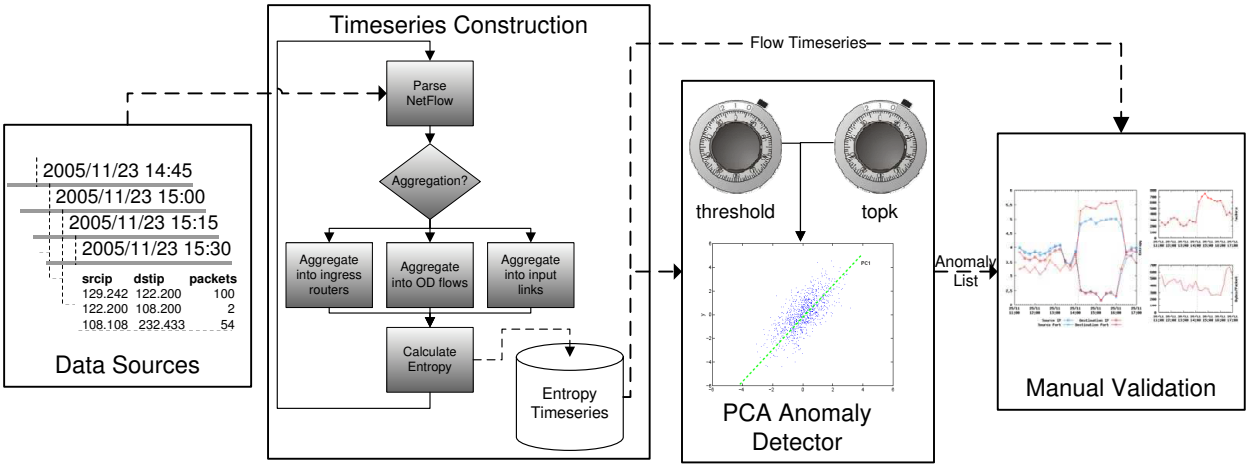


Figure 1: The Architecture

data type used for the traffic matrix was byte-counts instead of entropy values, then a cell item $v_{i,j}$ would uniquely map to the number of packets carried by router j, for example, at time i.

There are many ways to perform structural traffic aggregation, each with different statistical properties, e.g. different number of constituent flows, distribution in flow size, etc. Our research demonstrates that the choice of traffic aggregation can significantly impact the effectiveness of PCA as a traffic anomaly detector, and hence it is important to study several such formalisms.

It is often natural to perform this structural aggregation of IP flows according to where they enter and exit in the network. We analyzed three such aggregations, viz. ingress routers, OD flows, and input links. For ingress routers, the data is aggregated according to which router it entered the network, e.g. there are 11 such flows for Abilene because it has 11 routers and they all accept incoming traffic. Performing this aggregation is straightforward because there are separate IP flow logs for each ingress router. For the "input links" aggregation, IP flow records are aggregated by ⟨ingress router, input interface⟩ tuples, which is also computationally uncomplicated because IP flow records contain the necessary interface information. An OD, or origin-destination, flow uniquely identifies which ingress and egress router an IP flow traversed while inside the network. Identification of the egress point e for a given ⟨ingress router, prefix⟩ pair requires parsing of routing logs as explained in section 3.1.

3.3 PCA Anomaly Detector

The Matlab code that performs the PCA calculations was written by Lakhina *et al.* [14] and was graciously donated for our work. As detailed in section 2, it builds a model for normal traffic for the given traffic matrix and top_k parameter, and classifies a given time and flow as anomalous if the statistical outlier at that time exceeds the threshold parameter. We wrote wrapper code around this software in order to sweep a range of parameters, as is diagrammed by the dial knobs in figure 1, to evaluate PCA's sensitivity to these parameters.

Applying PCA to network-wide traffic measurements introduces several complications. First, because statistical tools such as PCA analyze *timeseries*, they classify individual *time bins* as anomalous, which are different from the underlying network events that may have caused the detection. In fact, a given anomalous time bin may contain multiple anomalous events of interest to a network operator and, vice versa, one anomalous event may span multiple time-bins. For simplicity, we use the term "anomaly" as shorthand for "anomalous time bin" in the remainder of this paper, consistent with previous work.

Second, PCA requires the length of the time series (i.e., n) to be greater than or equal to the number of measurements (i.e., m). In addition, the value of m depends not only on the number of locations (e.g., input links, ingress routers, or OD pairs), but also on the number of measurements included from each vantage point. Jointly analyzing entropy for the four IP traffic descriptors exploits PCA's ability to find correlations across dimensions, at the expense of requiring an even longer time series. For example, the Geant network has 23 routers, which produces 552 OD flows. This requires a minimum of $552 \times 4 = 2208$ time-steps, which is equal to $\frac{2208 \times 15}{60 \times 24} = 23$ days since Geant aggregates its flow records into 15-minute time bins. Analyzing such a large amount of data simultaneously can be impractical, which is why we do not include a Geant OD-flow dataset in our study (see table 3). Moderately sized networks may therefore be unable to run a PCA-based traffic anomaly detector on top of OD flows, which can be a very fruitful traffic aggregation [12].

In addition to the hard limit on how many time-steps must be analyzed concurrently, the increase in the number of variables processed by PCA also comes at a computational overhead. The algorithm most commonly used for performing PCA—singular value decomposition (SVD)—takes time $O(nm^2)$. Having ϱ measurements per vantage point not only increases m by a factor of ϱ but may (for the reasons explained in the previous paragraph) also increase n by the same factor, leading to an $O(\varrho^3)$ factor increase in the computational overhead associated with applying PCA[2].

[2]These issues could potentially be addressed by the technique proposed in [23] for conceptually combining routers according to topology, but we have not evaluated its effectiveness in this context.

Σ	meaning	value
δ	length of inspected window	3 hours
n	heavy hitters inspected	10
ω	min mean #packets	100
γ	max times #packets = 0	once an hour

Table 2: Heuristics used in the manual validation step

3.4 Manual Validation

To provide qualitative statements about the effectiveness of PCA, we need some measure of ground truth. The paucity of labeled data is a challenge facing research on network anomaly detection, and our study is no different. For our work, we manually validated the anomalies detected by PCA across the two networks, three traffic aggregations, and range of tunable parameters we explored. We manually inspected the entropy time series for a significant fraction of the suspected anomalies and classified them as real anomalies or false positives. We declared an anomaly to be a false positive if its entropy timeseries plots appeared merely to fluctuate in a random fashion. Unfortunately, statistical tools such as PCA do not always flag the precise moment in time that a trained operator might consider the beginning of an anomaly. Often, for example, the end of an anomaly is equally statistically significant as the beginning. As such, we investigate a time window of length δ around each detected anomaly. We found that inspecting a three-hour time window around any anomaly at time t was more than adequate (i.e., we inspect $t \pm 1.5$ hours).

In our initial exploration of the data, we discovered that many suspected anomalies involved flows that carry relatively little traffic. An input link, ingress router, or OD pair with a small amount of traffic can experience significant variations in load in response to a modest change in traffic conditions. These large variations in load often caused large variations in other metrics, such as the entropy features in the traffic matrix. In fact, we found that anomalies were often triggered by the addition of a single IP flow that is a sustained file-transfer (also known as an "alpha flow"). We decided to flag outliers that occurred on such relatively small flows as false positives because we deemed them uninteresting to network operators.

To illustrate the skew in traffic volumes, we computed the average number of packets for each input link, ingress router, and OD pair over all of the time-steps of our measurement data. Figure 2 plots the average number of packets in a flow for each traffic aggregation, as a function of the flow ID. The graph shows that the number of packets per flow can vary widely, especially for certain representations of the traffic. For the input-link aggregation, in particular, big flows carry up to eight orders of magnitude more traffic than small flows. To filter away such very small flows, we classified flows that had less than an average of ω sampled packets within the inspected time-window of length δ, or carried 0 packets γ times or more during this same period of time, as "small". ω was set to 100 in order to roughly correspond to the gap seen at the tail of the Abilene and Geant input link flow-size plots seen in in figure 2, and γ was set so that flows must carry more than 0 packets in a time-window at least once an hour (on average).

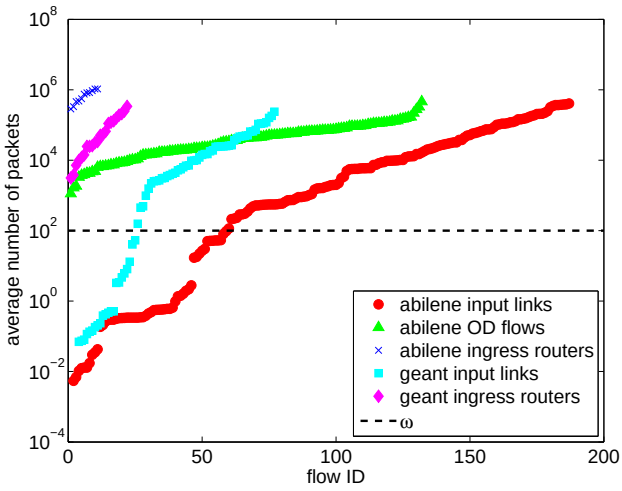


Figure 2: Average Packets Per Flow

Network	Aggregation	Flows	Total	FP
Abilene	ingress routers	11	955	557
Abilene	OD flows	132	749	382
Abilene	input links	187	937	398
Geant	ingress routers	23	421	110
Geant	input links	77	491	172

Table 3: Classified anomalies

Finally, to give us further confidence in our labeling of anomalies as true or false positives, we manually inspected the heavy-hitters for each of the four IP header features we studied. That is, we inspected the top-n source IP heavy-hitters, destination IP heavy-hitters, etc. We chose to set n to 10 because the vast majority of heavy-hitter traffic anomalies appear to involve only a handful of IP header features.

This validation step produced the datasets summarized in table 3, where the "Total" column refers to the total number of detections validated and the "FP" column refers to the total number of false positives. Both the number of total detections and false positives are across all top_k values and detection thresholds for the given dataset. The listed total number of detections are all that were detected by PCA in the studied week, except for the Abilene OD flows and input links aggregations, where a 22% and 23% random sample was chosen, respectively. The false-positive rate and total number of detections are the metrics we use to evaluate PCA's effectiveness across the studied parameter space.

The false-*negative* rate is a key metric that is conspicuously absent from both table 3 and the preceding discussion. The lack of reliable estimates of the false-negative rate is a long-standing problem in traffic anomaly detection that is complicated by the magnitude of the datasets studied. It is difficult to know what anomalies might go undetected when the size of the datasets studied approach the terabyte range. We cannot comment on PCA's false-negative rate in our study because the set of possible anomalies that we were considering was defined in terms of the suspected anomalies PCA detected. This is clearly a limitation of our study, as well as previous work on traffic anomaly detection.

4. TUNABLE PARAMETERS

The following section will evaluate PCA's sensitivity to its two key parameters, viz. the number of dimensions that constitute its normal subspace in section 4.1 and the detection threshold in section 4.2.

4.1 Size of Normal Subspace

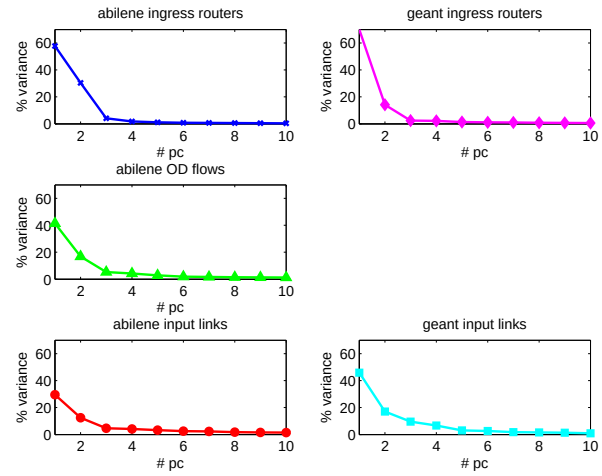
The number of principal components included in the normal subspace—the top_k parameter—is the most important parameter to be tuned when using PCA as a traffic anomaly detector. Past literature has made four important claims in this context: (i) traffic traces have low intrinsic dimensionality, which means that top_k can be small, (ii) these first few principal components capture the vast majority of the variance in the data, (iii) the same principal components are also highly periodic and thus capture the diurnal trends sought to be included in the normal subspace, and (iv) identifying the separation between normal and anomalous principal components can be done by retaining the first k principal components such that the projection of the traffic data does not contain a 3σ deviation from the mean [14, 12]. The following sub-sections show, in order, that the second claim does not hold across all networks and traffic aggregations, that the effectiveness of PCA is very sensitive to the top_k parameter, and that the previously proposed techniques for determining top_k are inadequate.

4.1.1 Decoupling Size from Captured Variance

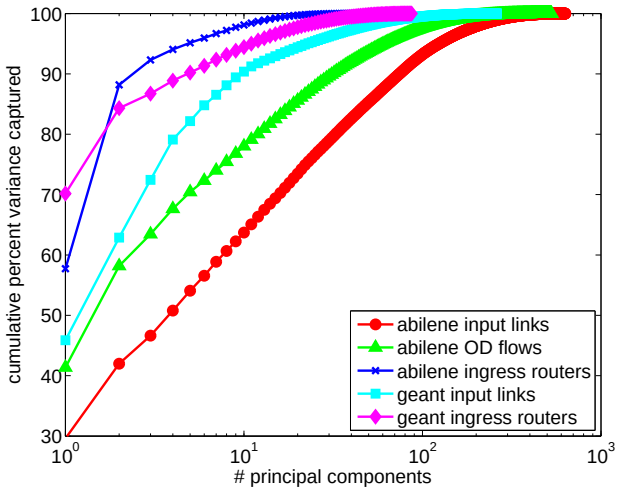
Each of our datasets support the previous finding that traffic traces have low intrinsic dimensionality, as can be seen from figure 3(a). The figure contains scree plots for each of the datasets used in our study. A scree plot is a plot of percent variance captured by a given principal component. We can conclude that traffic traces have low intrinsic dimensionality because the corresponding scree plots have very early knees relative to the original dimensionality of the datasets (seen in table 3). This is an important observation because it means that only a small fraction of all principal components need to be included in the normal subspace to capture the periodic trends that these first few principal components have been shown to exhibit [14].

However, our results do not support the earlier contention that the first few principal components necessarily capture the vast majority of variability in the traffic matrix, which is demonstrated by figure 3(b), which is the CDF of 3(a) in log-scale. While the plots in figure 3(a) have knees somewhere in the range $[2, 6]$, it is much more difficult to argue that setting top_k to a value in this range would correspond to a vast fraction of variance in figure 3(b). For example, if an Abilene network operator wanted to capture 90% of the variance for the input-link aggregation, he would need a top_k value that was an order of magnitude larger than previously reported in the literature (at least 95). If, on the other hand, he set top_k to match up with the knee seen in figure 3(a), he would capture less than half of the total variance.

The purpose of this section is not merely to debunk an earlier coupling of low intrinsic dimensionality and percent variance captured, but also to highlight that this distinction is an important one. One should not determine the top_k variable based on percent variance captured (i.e., plot 3(b)) because different networks have different natural levels of variability, and the normal subspace should capture *periodicity* as opposed to a certain fraction of variance. For example, a research backbone for universities such as Abilene will likely have a more variable matrix than a tier-1 network because Abilene is (i) smaller, (ii) is used as an experimental network, and (iii) only a very limited set of source hosts gain access to the network. The same heterogeneity is exhibited across different traffic aggregations also, where a more highly aggregated traffic aggregation such as ingress routers will have more stable statistical properties than input links, which may have lots of small flows that are highly variable. It is therefore important to highlight that the top_k parameter should not be determined based on cumulative percent variance captured.



(a) scree plots

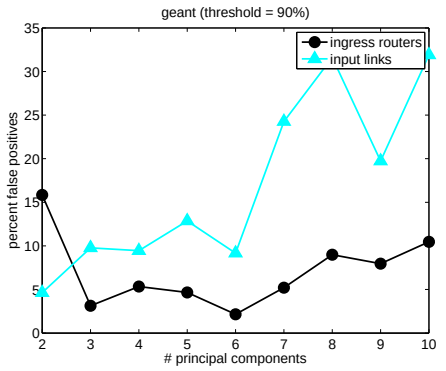


(b) CDF of scree plots

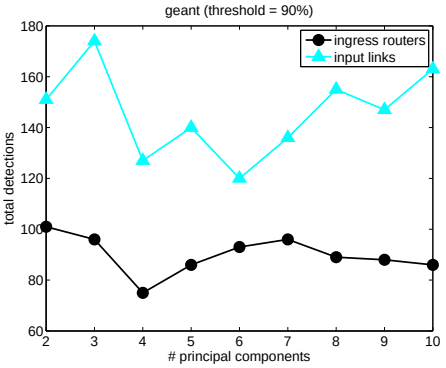
Figure 3: Intrinsic Dimensionality

4.1.2 Sensitivity Analysis

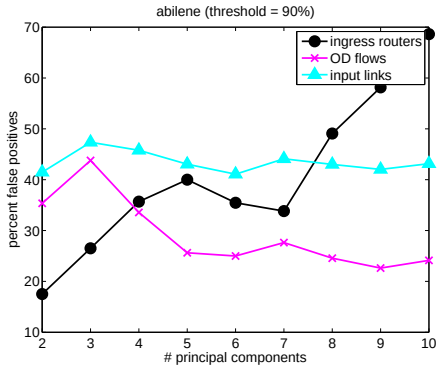
PCA is very sensitive to the top_k parameter: We noted previously that the scree plots for our datasets each appeared to have knees in the range $[2, 6]$. While that range might appear small, our results indicate that PCA is very sensitive to the number of principal components even within such a limited range. As can be seen from figure 4(a), within the $[2, 6]$ range, the false-positive rate for Geant ingress



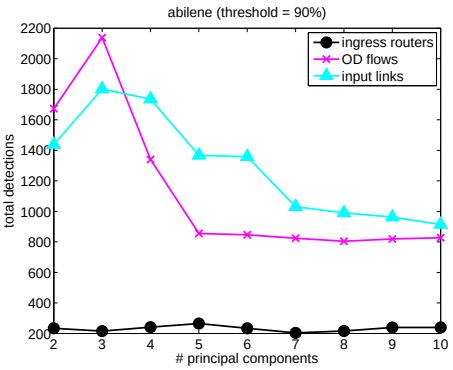
(a) Geant false-positive rate



(b) Geant total detections



(c) Abilene false-positive rate



(d) Abilene total detections

Figure 4: Impact of top_k on false-positive rate and total detections

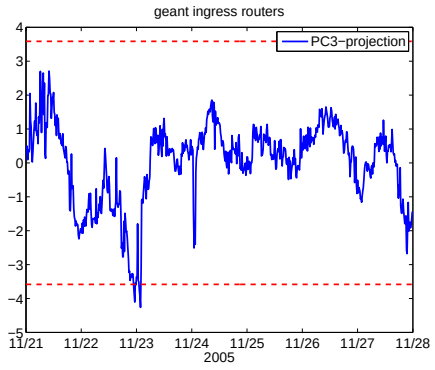
routers varies between 3.1% and 15.8%. If one ventures beyond this range, the performance degradation can be even more rapid. In the same figure we can see that the false-positive rate when going from 6 to 8 principal components for Geant input links increases from 9.2% to 31.6%. It is therefore extremely important that the top_k parameter be carefully tuned. For the remainder of this paper, we will define the 'appropriate' top_k value as the one that we consider achieves the best trade-off between the false-positive rate and the total number of detections.

The appropriate top_k value varies across networks and traffic aggregations: Figure 4 also shows that the appropriate number of principal components to incorporate into the normal subspace varies across networks and traffic aggregations. For example, the appropriate choice of principal components is probably 2 for Abilene ingress routers, 3 for Geant ingress routers, and 5 for Abilene OD-flows. It is interesting to note that the relative order of these three datasets in terms of top_k value is identical to their relative ordering in terms of original dimensionality (see: total number of flows in table 3). We hypothesize that this phenomenon will hold in general, and further research might provide rule-of-thumb guidelines that map $\langle$original dimensionality, scree knee$\rangle$ tuples to a top_k value. Guidelines are not sound methodology, however, and PCA's sensitivity to the top_k parameter necessitates a robust methodology.

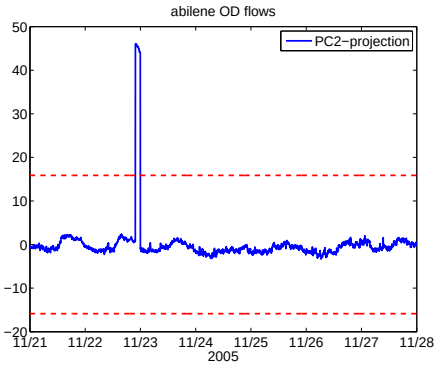
Comparison of traffic aggregations: Finally, figure 4 shows that the choice of traffic aggregation has a strong impact on PCA's performance. Choosing the right traffic aggregation is tricky: too much aggregation will lead to smooth and predictable flow curves whereas too little aggregation yields a very heavy-tailed flow-size distribution (see figure 2) and hence some highly variable small flows whose spikes are not of interest to network operators. In particular, for both Abilene in figure 4(d) and Geant in figure 4(b), it is clear that the ingress router aggregation consistently detects fewer anomalies than OD flows and input links. The reason for this is that at the level of ingress routers, the data is so aggregated and the flows are so large that most anomalies are effectively drowned. This also means that the anomalies that *are* flagged by PCA when using this aggregation-level tend to be large and obvious. Hence, at its appropriate top_k value (e.g. 3 for Geant and 2 for Abilene), the ingress routers aggregation has the lowest false-positive rate of the three traffic aggregations studied for both networks.

On the other end of our aggregation spectrum, input links' false-positive rate suffers as a result of a large fraction of small flows. Abilene input links is particularly bad in figure 4(c) in that its false-positive rate never goes below 40%. It holds across both networks that, at their respective appropriate top_k values, the input links aggregation has the highest false-positive rate of the three formalisms. We believe this can be largely contributed to an excess of small flows whose natural variance cause alarms to be raised by the PCA traffic anomaly detector.

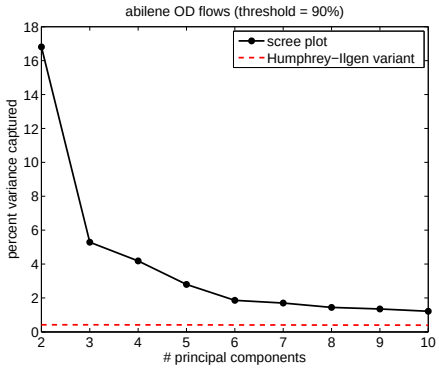
For the Abilene network (figures 4(c) and 4(d)), it seems clear that OD-flows is the traffic aggregation that achieves the best overall trade-off between total detections and false-positive rate. Our findings support earlier papers [14] that have demonstrated that OD-flows is a fruitful traffic aggregation for detecting network anomalies. For this same reason, it is doubly frustrating that we are prevented from try-



(a) Geant ingress routers



(b) Abilene OD flows



(c) Abilene OD flows

Figure 5: Determining the top_k parameter

ing the OD-flow aggregation for the Geant network due to the reasons explained in section 3.3.

4.1.3 Evaluating Top-K Selection Techniques

We've demonstrated that (i) PCA is very sensitive to the top_k parameter, and (ii) that its appropriate value varies from one setting to the next. For PCA's effective operation as a traffic anomaly detector, it is therefore essential that there are automated techniques for determining the proper setting of the top_k parameter. Unfortunately, current research does not provide any reliable such techniques. Two techniques that have been used include (i) determining top_k by visually inspecting the scree plot — a method referred to as Cattell's Scree Test [5] in the statistics literature, and (ii) retaining the first k principal components that do not contain a 3σ deviation from the mean when the traffic matrix is projected upon them.

We've evaluated the effectiveness of the 3σ heuristic in figures 5(a) and 5(b). Each figure shows the result of projecting the respective traffic matrices onto the first principal component that results in a 3σ deviation from the mean (the $\pm 3\sigma$ deviation is represented by the upper and lower dashed horizontal lines). That is, there is a 3σ deviation for these principal components because the solid lines exceed the boundary of the dashed lines. Specifically, figure 5(a) shows such a deviation for the third principal component, which means that the 3σ heuristic suggests retaining two principal components in the Geant ingress routers normal subspace. Our results in figure 4(a) indicate that this would lead to a false positive rate that is three times as high as ideal. Likewise, figure 5(b) shows that the same heuristic suggests keeping only a single principal component for the Abilene OD flow normal subspace. While we are not including the result here, the 3σ heuristic also suggests keeping *zero* principal components for the Abilene ingress routers normal subspace, which is not possible. It is therefore clear that this heuristic is not robust. On the other hand, however, one can legitimately question whether principal components with such large spikes can capture normalcy; we will address this question in section 5.1.

In figure 5(c) we have evaluated the effectiveness of Cattell's Scree Test. The knee of the scree plot appears to be at $k = 3$ but we determined previously that a top_k value of $k = 5$ seems to achieve the best results. In general, Cattell's Scree Test is within one or two principal components, but there appears to be no predictable pattern to the deviation.

Humphrey-Ilgen parallel analysis [17] is an automated statistical technique for determining the number of principal components to keep. The method determines the number of principal components to retain by the intersection point of two curves representing the cumulative eigenvalues of the traffic matrix and an equivalently-sized random matrix. The intuition behind this method is to only include principal components that contribute more variance than a random vector would (i.e., those before the intersection point). For our purposes, a more effective metric is to compare where the respective scree plots intersect. Figure 5(c) plots the scree plots for a traffic matrix from our study in addition to an equivalently-sized random matrix. It should not be surprising that the scree plot for the random matrix (i.e., "Humphrey-Ilgen variant") is nearly horizontal, given that every principal component of a random matrix is expected to capture the same variance. The scree plot for the traffic matrix appears to have a knee at $k = 2$ but Humphrey-Ilgen retains far more principal components than this (i.e., the two curves do not intersect anywhere in plotted interval).

We are therefore left with no reliable technique for tuning the top_k parameter[3]. Cattell's Scree Test performs the best in that it is often within one or two principal components of the operating point that minimizes the false-positive rate, but we've demonstrated that PCA's false-positive rate can be very sensitive even within such a small range. While our results indicate that there does appear to be small ranges of top_k values that perform better than others, there are

[3]We also evaluated Kaiser's Criterion [9], which is another automated technique for determining top_k, but omit the result because it performed even poorer than Humphrey-Ilgen

fundamental problems with even the *concept* of the top_k parameter that limit the potential success of any such scheme for determining which principal components to include in the normal subspace. We will discuss this intrinsic limitation in section 5.1.

4.2 The Detection Threshold

The threshold parameter specifies how statistically significant a given outlier must be for a PCA-based traffic anomaly detector to report it. Therefore the total number of detections will always decrease monotonically as a function of the threshold. The false-positive rate, while generally decreasing as a function of the threshold, need not always decrease. The reason for this is that one may cease to detect true positives (that are less statistically significant) before ceasing to detect false positives, as can be observed in figure 6(c).

Figure 6 provides further support for the conclusion that the relative properties of traffic aggregation formalisms appear to hold across networks. That is, for a given top_k value but across all thresholds and both networks, the input links aggregation generally detects more potential anomalies than OD-flows, which detects more potential anomalies than ingress routers (see figures 6(b) and 6(d)). We believe this to be the case because the input links aggregation tends to produce less multiplexed data than OD-flows, which in turn produces less multiplexed than the ingress router aggregation. For both Abilene in figure 6(c) and Geant in figure 6(a), one can conclude that the input links aggregation has a higher false-positive rate than each of the others. Finally, figures 6(c) and 6(d) reinforce the perception that OD flows is probably the aggregation that achieves the best balance between false-positive rate and total number of detections.

Our results indicate that the threshold provides operators with an intuitive knob to trade off the false-positive rate and total number of detections.

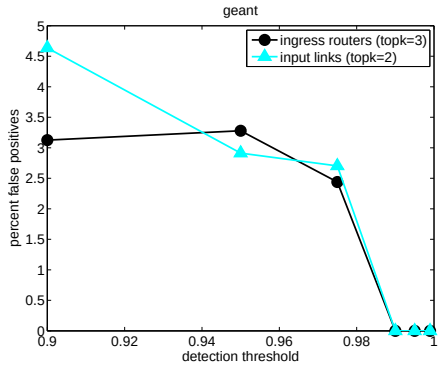
5. INTRINSIC LIMITATIONS OF PCA

In this section, we highlight two key limitations of PCA that limit its effectiveness as a traffic anomaly detector. Section 5.1 illustrates how a sufficiently large anomaly may inadvertently pollute PCA's definition of normal traffic, and section 5.2 examines the difficulty of identifying the set of flows responsible for a statistical anomaly.

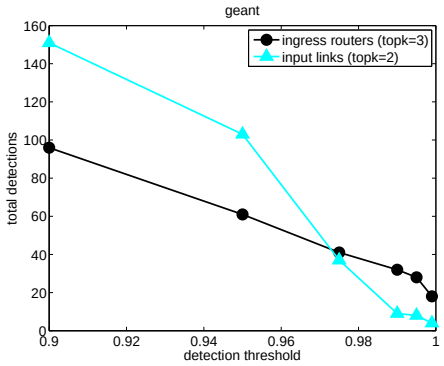
5.1 Contamination of the Normal Subspace

Using PCA to detect traffic anomalies relies on the assumption that the top few principal components represent the normal traffic, and the anomalies lie in the remaining components. However, in some cases, a sufficiently large anomaly may introduce so much variance in the traffic matrix that it is included in one of the first few principal components, thereby contaminating PCA's definition of normality. Our analysis in the previous section intentionally avoided time periods with dramatic network events to avoid unduly degrading PCA's false-positive rates[4]. Therefore, to demonstrate the effects of polluting the normal subspace, we analyze a separate (unlabeled) trace for the Geant network between November 12-20, 2005. Since we have not classified all of the detected anomalies during this period, we cannot

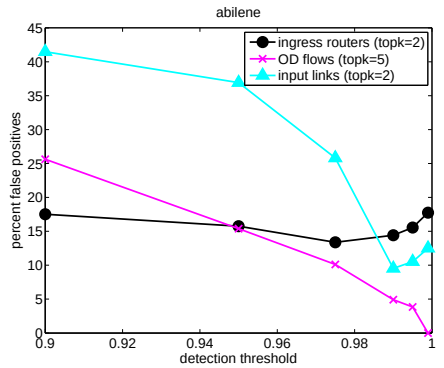
[4]However, figure 5(b) shows that even moderately sized events can contaminate the *very first* principal component



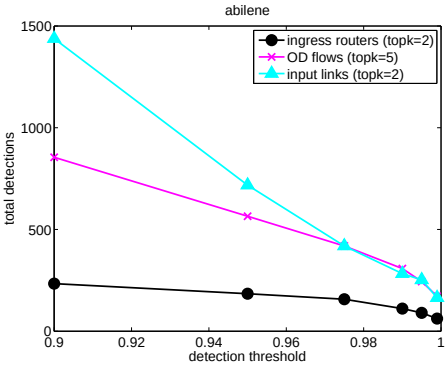
(a) Geant false-positive rate



(b) Geant total detections

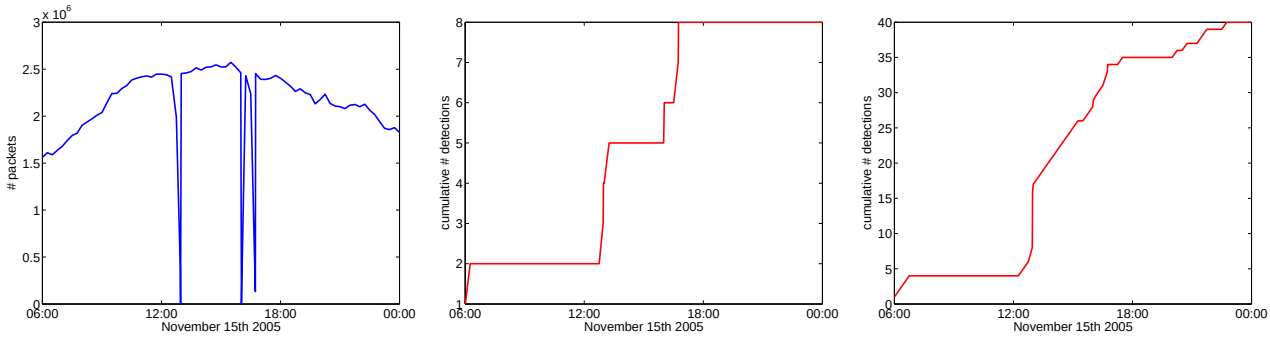


(c) Abilene false-positive rate



(d) Abilene total detections

Figure 6: Impact of detection threshold on false-positive rate and total detections



(a) Geant network-wide traffic (b) Geant ingress routers (c) Geant input links

Figure 7: A large outage is included in PCA's model of normalcy and hence goes largely undetected by PCA

produce false-positive rates for this trace, though we expect the erroneous definition of "normal" traffic would increase the false-positive rate.

Figure 7(a) plots the aggregate traffic on the Geant network for eighteen hours on November 15, 2005. The plot shows several clear outages that caused a significant drop in the aggregate traffic; in fact, only two routers carried any traffic at all during these fifteen-minute windows. Figures 7(b) and 7(c) plot the cumulative detections by PCA over this 18-hour period for the ingress-router and input-link aggregations, respectively. While figure 7(b) has three spikes in the number of detections that coincide with spikes seen in 7(a), the spikes in 7(b) correspond to only a small fraction of the total number of ingress routers. Although Geant has 23 ingress routers, only eight detections are made during the entire 18-hour period. The input-link aggregation in figure 7(c) fares even worse in that the only visible spike is correlated with the first drastic network event seen in figure 7(a). In addition, the spike in figure 7(c) corresponds to less than 13% of all links, whereas the outage at that moment caused 75 out of 77 input links to carry 0 packets.

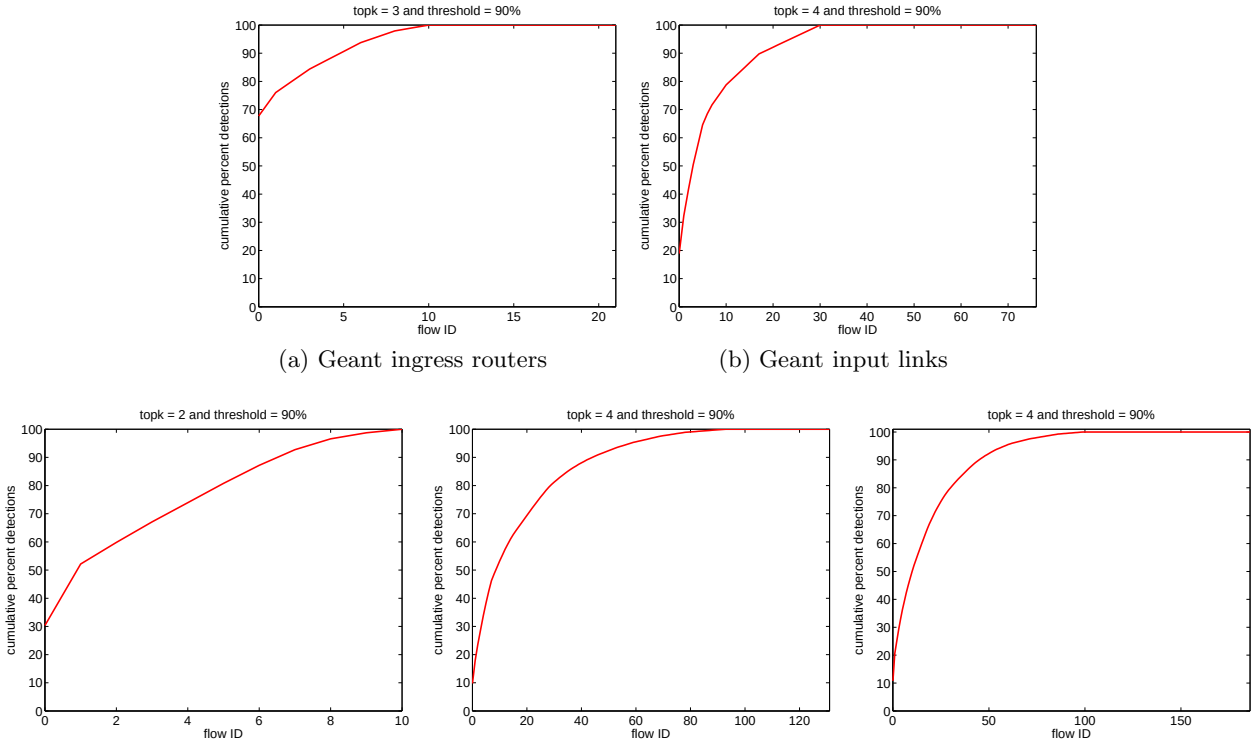
When a large network event contaminates the normal subspace, PCA may not detect the anomaly, and its inclusion in the normal subspace may yield false positives or false negatives for other traffic. Our results suggest that it is important to *preprocess* the data to identify and remove large anomalies before constructing the normal subspace. [11] presented a potential technique for identifying when such large outliers are included in the normal subspace, but no techniques have been evaluated for subsequently smoothing the normal subspace (to our knowledge). Even if smoothing techniques could be identified (e.g. exponential weighted moving average, EWMA), their applicability could potentially be limited to the large-scale anomalies seen in figure 7(a). Medium-sized anomalies would be more problematic, as they might easily evade a coarse-grained filtering scheme and still unwittingly pollute the normal subspace.

5.2 Identifying the Anomalous Flows

PCA detects anomalous *time bins*, not anomalous *flows*. That is, PCA reports an anomaly when a measurement vector $\vec{v_i}$ is expressed primarily by the anomalous subspace. However, PCA provides no direct mapping between these subspaces and the original flows, which makes it difficult to identify the flow(s) responsible for the anomaly. Previous studies have applied a heuristic [12, 10, 13] that associates an anomaly with the r flows with the largest contribution to $\vec{v_i}$, such that the r flows are big enough to account for the spike in the anomalous subspace. Unfortunately there is no *a priori* reason for why the r flows with the highest entropy value at time i must necessarily correspond to the flows that caused PCA to detect an anomaly. In fact, this heuristic can unduly trigger alarms in some flows much more frequently than others.

To illustrate this "heavy hitter" phenomenon, figure 8 plots the CDF of the percentage of the anomalies that are attributed to the various flows, where we ranked the flows in order of how many anomalies the heuristic associates with them. For example, in figure 8(a), a *single* ingress router is associated with 70% of all PCA alarms on the Geant network. Each of the other aggregations show similar types of skewed flow-identification distributions. For example, 29% of OD-flows in the Abilene network did not contribute to a single alarm during the studied week. Although we do not necessarily expect a uniform distribution, the skew in the five graphs is a natural consequence of a heuristic that ranks flows in order of their entropy values.

Mapping an anomalous time bin to one or more responsible flows is inherently challenging, since PCA operates on aggregated measurement data and remaps the data to another subspace. Moreover, the inaccuracies of the previously employed heuristic very likely increased the false-positive rates reported in our study. That is, the PCA technique itself may have identified a legitimate anomalous time-bin, but it was identified as a false-positive because the heuristic associated this anomaly with an incorrect flow. We therefore believe that creating more effective heuristics is a very important avenue for future work. For example, it may be better to identify the r flows that exhibit the greatest *variance* along the anomalous subspace around the time of detection. While still only a heuristic, with associated shortcomings, this approach may more closely capture the notion of a sudden anomalous event. The difficulty of identifying the anomalous flows is a fundamental problem of PCA, which begs the question of whether other anomaly-detection techniques (i.e., that operate on the raw data, rather than an aggregated and transformed variant of the data) are more appropriate for applications where network operators need to pinpoint the location(s) of an anomaly.



(a) Geant ingress routers (b) Geant input links

(c) Abilene ingress routers (d) Abilene OD flows (e) Abilene input links

Figure 8: The heavy-hitter flow phenomenon

6. RELATED WORK

Lakhina *et al.* popularized using PCA for traffic anomaly detection in [14, 12, 10, 13]. The work showed that traffic traces have low intrinsic dimensionality, that PCA can detect network-wide anomalies when analyzing the OD flows aggregation, and can detect a wide variety of types of anomalies when analyzing entropy timeseries of IP header features. PCA has also recently been combined with sketches [15] and distributed monitors [16] to provide more efficient traffic anomaly detection. This entire body of work used the same dataset, however, for which Lakhina's PCA code was highly optimized.

In [22], PCA was one of many algorithms evaluated in a general system that aimed to infer network-level anomalies from available data aggregates. PCA has also been used to correlate BGP updates with underlying network events such as link failures, resets, etc [21]. Other statistical methods that have been used for traffic anomaly detection include Kalman filters [19], wavelets [4], among others. Other inherent limitations of PCA have also been discussed in the statistics literature [18, 20].

7. CONCLUSION

Previous work has shown that PCA can detect real anomalies, but our work demonstrates that the challenges to using PCA as a traffic anomaly detector have been understated and current methods for tuning PCA are inadequate. Lakhina *et al.* were able to achieve such promising early results because of their great familiarity with both the technique and the data. Subsequent PCA work in this lineage used the same software, heuristics, and labeled data, which understandably yielded equally strong results by utilizing already highly optimized parameter-settings for the given circumstance.

Starting with new data sets and exploring a range of parameter settings, we show that selecting the appropriate value for top_k is surprisingly difficult; small changes in top_k in either direction can have a significant influence on the false-positive rate. In addition, existing techniques for selecting top_k are inadequate. In fact, we've shown that top_k is a flawed concept in and of itself because the first few principal components need not capture a vast majority of the variance in a traffic trace, nor are they necessarily periodic. The normal subspace may in fact become polluted by large anomalies, which degrades the effectiveness of PCA. We also demonstrated that identifying the flow that caused a PCA detection is a fundamentally hard problem. We showed that the previously employed heuristic could fail in many circumstances, and may have the inadvertent side-effect of associating the majority of detections with a small set of flows.

Our study suggests that using PCA for traffic anomaly detection is much more difficult than it appears. Before PCA can be used for automated, unsupervised detection of anomalous traffic, we need more effective techniques for determining the dimensionality of the normal subspace, preventing its contamination, and identifying flows responsible for a given PCA detection. In our ongoing work, we are also investigating other statistical techniques that may be able to detect and identify anomalous traffic in a more robust manner.

8. ACKNOWLEDGMENTS

The authors would like to thank Mark Crovella and Anukool Lakhina for invaluable advice and feedback on applying PCA to traffic anomaly detection. The first author would further like to thank advisor Kai Li for continued support, and David Gardner for providing housing in Paris where the work was performed.

9. REFERENCES

[1] Abilene Backbone Network. `abilene.internet2.edu/`.

[2] Abilene Participation Agreement. `abilene.internet2.edu/community/connectors/AbileneConnectionAgreement2006.pdf`.

[3] Bair, E., Hastie, T., Paul, D., and Tibshirani, R. Prediction by supervised principal components. *Journal of the American Statistical Association 101*, 473 (2006), 119–137.

[4] Barford, P., Kline, J., Plonka, D., and Ron, A. A signal analysis of network traffic anomalies. In *ACM Internet Measurement Workshop* (Marseille, France, 2002), pp. 71–82.

[5] Cattell, R. B. The scree test for the number of factors. *Multivariate Behavior Research 1* (1966), 245–276.

[6] Geant Network. `www.geant.net/`.

[7] Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Educ. Psy.* (1933), 417–441.

[8] Juniper J-Flow. `www.juniper.net/techpubs/software/erx/junose61/swconfig-routing-vol1/html/ip-jflow-stats-config2.html`.

[9] Kaiser, H. F. The varimax criterion for analytic rotation in factor analysis. *Psychometrika 23* (1958), 187–200.

[10] Lakhina, A., Crovella, M., and Diot, C. Characterization of network-wide anomalies in traffic flows. In *ACM Internet Measurement Conference* (Taormina, Sicily, Italy, 2004), pp. 201–206.

[11] Lakhina, A., Crovella, M., and Diot, C. Characterization of network-wide anomalies in traffic flows. Tech. Rep. BUCS-TR-2004-020, Boston University Department of Computer Science, May 2004.

[12] Lakhina, A., Crovella, M., and Diot, C. Diagnosing network-wide traffic anomalies. In *ACM SIGCOMM* (Portland, Oregon, USA, 2004), pp. 219–230.

[13] Lakhina, A., Crovella, M., and Diot, C. Mining anomalies using traffic feature distributions. In *ACM SIGCOMM* (Philadelphia, Pennsylvania, USA, 2005), pp. 217–228.

[14] Lakhina, A., Papagiannaki, K., Crovella, M., Diot, C., Kolaczyk, E. D., and Taft, N. Structural analysis of network traffic flows. In *ACM SIGMETRICS* (New York, NY, USA, 2004), pp. 61–72.

[15] Li, X., Bian, F., Crovella, M., Diot, C., Govindan, R., Iannaccone, G., and Lakhina, A. Detection and identification of network anomalies using sketch subspaces. In *ACM Internet Measurement Conference* (Rio de Janeiro, Brazil, October 2006).

[16] Li, X., Bian, F., Zang, H., Diot, C., Govindan, R., Hong, W., and Iannaccone, G. MIND: A distributed multi-dimensional indexing system for network diagnosis. In *IEEE INFOCOM* (Barcelona, Spain, April 2006).

[17] Montanelli, R.G. Jr., and Humphreys, L. G. Latent roots of random data correlation matrices with squared multiple correlations on the diagonal: A monte carlo study. *Psychometrika 41*, 3 (1976), 341–348.

[18] Schoelkopf, B., Smola, A., and Mueller, K. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation 10*, 5 (1998), 1299–1319.

[19] Soule, A., Salamatian, K., and Taft, N. Combining filtering and statistical methods for anomaly detection. In *ACM Internet Measurement Conference* (Berkeley, California, USA, October 2005).

[20] Tipping, M. E., and Bishop, C. M. Mixtures of probabilistic principal component analysers. *Neural Computation 11*, 2 (1999), 443–482.

[21] Xu, K., Chandrashekar, J., and Zhang, Z.-L. A first step toward understanding inter-domain routing dynamics. In *ACM MineNet Workshop* (Philadelphia, Pennsylvania, USA, 2005), pp. 207–212.

[22] Zhang, Y., Ge, Z., Greenberg, A., and Roughan, M. Network anomography. In *ACM Internet Measurement Conference* (Berkeley, California, USA, October 2005).

[23] Zhang, Y., Roughan, M., Duffield, N., and Greenberg, A. Fast accurate computation of large-scale IP traffic matrices from link loads. In *ACM SIGMETRICS* (San Diego, CA, USA, 2003), pp. 206–217.