

# ADAM: A Practical Approach for Detecting Network Anomalies Using PCA

Arshad Ali, Modood Ahmad Khan, S. Azam H. Bukhari and Waqar Mahmood<sup>1</sup>  
[drarshad@niit.edu.pk](mailto:drarshad@niit.edu.pk), [55Modood@niit.edu.pk](mailto:55Modood@niit.edu.pk), [55azam@niit.edu.pk](mailto:55azam@niit.edu.pk) and [drwaqar@niit.edu.pk](mailto:drwaqar@niit.edu.pk)  
NUST Institute of Information Technology, Rawalpindi, Pakistan

**Abstract:** Anomalies are abnormal behaviors that can arise in any network. This paper presents a practical statistical approach for anomaly detection, analyzing various network related parameters. Our approach ADAM (Anomaly Detection and Analysis Measures), initially uses two parameters i.e. average available bandwidth (ABW) and average round-trip time (AvgRTT). We analyze samples of these two parameters by collecting randomly changing values of these parameters periodically from various remote locations around the world. Our proposed algorithm performs rigorous statistical calculations on these samples. Passing collected data to PCA (Principal Component Analysis) algorithm, which performs separation of link traffic into disjoint normal and anomalous subspaces. We compared our anomaly detection approach with the existing commercial and open-source tools that requires behavior-based and signatures-based detection respectively.

## 1. INTRODUCTION

Anomalies are any abnormal, unusual significant behavior in the network. They can arise in any network irrespective of their geographical limits and boundaries. Network anomaly detection is the detection of abnormal traffic conditions on a monitored network. Detecting anomalies is a critical task for both network operators and end-users because it is really a cumbersome job to detect anomalous patterns of traffic from large amount of high dimensional and noisy-data. Some noteworthy rationales for sudden cropping up of this anomalous behavior in the network are router outages, configuration changes, flash crowd and abuse. In this paper we present here a practical approach for the detection of anomalies & filtering-out anomalous traffic which is based on statistically analyzing the data samples with the help of a data gathering tool called 'Abing'<sup>2</sup>. We also evaluated our approach for the detection of anomalies in contrast to the traditional commercial and open-source tools that requires pre-defined and pre-fed policies and signatures respectively; that are more prone to malicious behaviors on the network like warm attacks and DoS (Denial of Service) / DDoS (Distributed Denial of Service) attacks. To further elucidate our approach i.e. ADAM (Anomaly Detection and Analysis Measures), we collect and analyze random samples of average available bandwidth (ABW) and average round trip time (AvgRTT) using the 'Abing' utility on periodic basis. These data samples are further fed to our algorithm which first performs rigorous statistical calculations on the collected data set. These statistical calculations include the process

of normalizing the data, computing and formulating a covariance matrix for the collected data, finding appropriate eigenvalues & eigenvectors and then passing the normalized data to the PCA (Principal Component Analysis) algorithm. PCA is a coordinate transformation method that maps a given set of data points onto new axes. These axes are called the principal axes or principal components. When working with zero-mean data, each principal component has the property that it points in the direction of maximum variance remaining in the data, given the variance already accounted for in the preceding components [1]. By identifying this variance of the data from the average we can detect and perform separation of anomalous data on the link in an efficient and economical manner thus preventing ourselves from false alarms to a very large extent. In addition to the above approach we also have discussed the core reasons that are responsible for the occurrence of anomalies in the network, we also have described the categories of anomalies and various different approaches that are currently being deployed all over the world for the detection of anomalies.

## 2. TYPES OF ANOMALIES

To categories different types of anomalies that are very commonly seen in networks. In a more generalized form we categorize network anomalies as follow:

- Unintentional Anomalies
- Malicious Anomalies

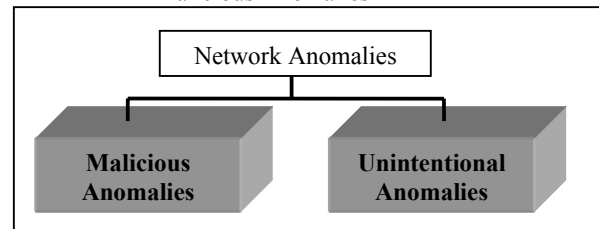


Figure 1: Basic types of network anomalies

Unintentional Anomalies are those that arise during peak hours when usually large volume of data is being transferred and exchanged, this kind of anomalous behavior can be commonly observed in universities and at organizational level when end-users usually are extensively availing facilities like file transfers, net browsing and video conferencing, such that the link capacity is being used at its maximum limit. One very common example of unintentional behavior of network traffic is flash crowd anomaly which usually takes place

<sup>1</sup> Authors are listed alphabetically <sup>2</sup> Abing is a tool using the packet pair dispersion technique to estimate the available Bandwidth/bit rate (unused capacity) for a path in the network

due to new software release, when more than the desired users want to access a single resource. In contrast to the unintentional anomalies networks are much more prone to malicious anomalies that can target and bog-down any network in the form of worms and DoS / DDoS attacks. These anomalies fall into the category of network abuse anomalies [2]. These kinds of anomalies may arise multiple times in a network and need to be tackled before hand. In this paper our approach is to target and detect anomalies that are related to the sudden bristly nature of traffic behavior on any network.

### 3. TEST-BED FOR ANALYSIS

For inspecting the viability and effectiveness of our algorithm we have operate on parameters like ABW and AvgRTT. The values for these parameters are taken by collecting data from links between our university NIIT and SLAC-Stanford, CERN and CALTECH respectively. These parameters are more or less the same that can be collected using SNMP (Simple Network Management Protocol) but all ISPs usually do not have SNMP enabled routers or even if they do have, detecting something abnormal and generating an alarm against it in real time environment is a demanding task which is not the case when we go for the option of SNMP. Some ISPs do employ the option of flow counts using commercial suites like CISCO NetFlow by deploying these applications on edge links, but again processing the collected data from the links is again a challenging task. Our approach is simple in a sense that by periodically monitoring and working with various network related measurements our application will not only be efficient in terms of processing, but it will be less CPU hungry as well. Also it does not have to sniff for each packet flowing across a particular link, instead it looks for the acute changes that arise in terms of bandwidth consumed and bandwidth that is available on the link. One more parameter that is part of this initial anomaly detection technique of ours is the average round trip time, which in case of some abnormal scenario occurring in the network will automatically cause late delivery of packets at the destination.

### 4. DATA

For data collection purposes we are using Abing utility. Its working is simple and its basic architecture is a combination of a client and a server, also known as reflectors. Similar kinds of server reflectors (abw\_rfl) are continuously running at various research vicinities. One such server is running at SLAC-Stanford since August 2002 and is first launched at iGrid2002 [3]. The set of parameters that we collect from Abing are shown below in table-1:

Table 1: List of Parameters returned by Abing

<b>Timestamp</b>	1099489009
<b>T/F</b>	A flag indicating the direction of traffic; where 'T' represents outgoing traffic and 'F' indicating it's vice versa
<b>IP-Address</b>	The host address where an Abing server reflector is running, may it be locally or remotely
<b>ABw</b>	Available Bandwidth
<b>Xtr</b>	Estimated Cross Traffic
<b>DBC</b>	Dominated Bottleneck Capacity
<b>ABW</b>	Average Available Bandwidth
<b>RTT</b>	Round Trip Time; minimum, average and maximum

When we run our implementation of algorithm it basically initiates as an Abing client which continuously sends request to the Abing server. These values are being collected periodically. We collect six such values after every 3 minutes and provide these readings to ADAM.

### 4. ADAM BASIC OPERATIONS

ADAM (*Anomaly Detection and Analysis Measures*) basically comprises of six steps. In other words we can say that these steps as a whole make up Principal Component Analysis technique for identifying and analyzing patches of abnormal patterns out of normal traffic. Certain matrix transformations are required in the ending phase but core steps still remains the same. These steps are further elucidated one by one in the following sub sections.

#### 4.1 Data Collection

As a first step we first start gathering results by running Abing continuously between our university and one of its research collaborator's sites (SLAC-Stanford, CALTECH and CERN). Once results starts coming in we parse through the results and extract our desired parameters i.e. ABW & AvgRTT. We collect six such random values returned by Abing in one go and then collect next six readings after 3 minutes and so on.

#### 4.2 Generating Zero-Mean Data

For PCA to work properly we have to first calculate the mean of the data collected, and subtract this mean from the original data. The first two columns of table-2 hold the original data that is being collected by parsing through the set of parameters that Abing returns. Whereas the last two columns are holding the mean subtracted data or zero-mean data.

Table 2: Sample data

ABW x (Mbps)	AvgRTT y (ms)	(x-x')	(y-y')
0.824	397.177	-0.74083	6.70733
5.1	397.177	3.53516	6.70733
0.984	398.753	-0.58083	-3.20666
0.141	398.753	-0.42383	-3.20666
0.740	384.632	-0.82483	-3.50066
1.6	384.632	0.03516	-3.50066

This zero-mean data is the average across both the dimensions. This data is further plotted in the form of a real time active graph which keeps itself updated. Below a snapshot of our application 'ADAM' is shown which executed continuously throughout the day, collecting its data samples from a remote location.

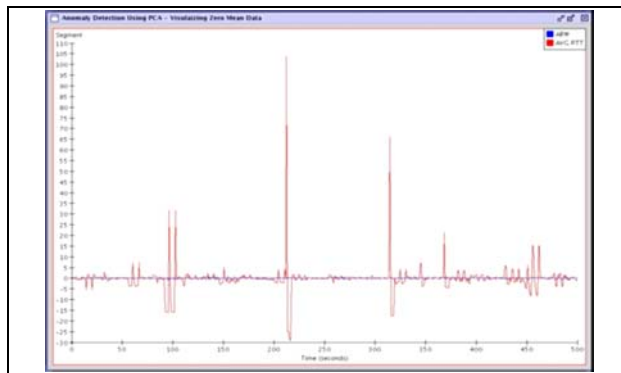


Figure 1: Histogram showing graph between zero-mean values of ABW & AvgRTT collected from CERN

From the above snapshot we can easily identify the spiky conditions and behaviors appearing in our link during the data collection phase. These points are plotted against the timeline and from there we can identify at which specific moment the underlying link is showing anomalous behavior. But still in conditions when factor of noise also overlaps or joins the normal traffic it is really hard and difficult to separate normal & anomalous patterns. Therefore from this point we carry out further operations to mitigate and minimize false positives or false alarms which might result into immediate halt of underlying network as a precautionary measure from the system administrators leaving the end-users in complete chaos.

#### 4.3 Formulating Covariance Matrix

After collecting six values of ABW & AvgRTT respectively we are now in a position to formulate a covariance matrix. Since we have two dimensions therefore our matrix will also have two dimensions. For efficient manipulation of data we store this matrix in the form of a double dimensional array. So every time a new set of values comes in we once again refresh this covariance array.

Table 3: Data stored as covariance matrix

<i>Covariance(ABW, ABW)</i>	<i>Covariance(ABW, AvgRTT)</i>
2.682082138	4.655863777
4.655863777	22.508566222
<i>Covariance(AvgRTT, ABW)</i>	<i>Covariance(AvgRTT, AvgRTT)</i>

#### 4.4 Getting Eigenvectors and Eigenvalues of the covariance matrix

Here we have to rely on a third party library called COLT 1.1.3 designed and developed by "CERN school of Computing" in JAVA with support for calculating eigenvector and eigenvalue. The library has its own predefined data structures therefore before doing any further calculation we have to explicitly transform our array holding the values of covariance matrix into this third party library's data structure. The library takes this matrix as an argument and returns an equivalent eigenvalues and eigenvector. Here is where the notion of reduced dimensionality comes into the scene. Basically to find out the principal component of the collected data set we have to find the eigenvector with the highest eigenvalue. Here a question arises i.e. what do eigenvectors mean anyway? Specially in this context what is their significance? To justify this if we take a closer look at the scatter plot of figure 2 below then we can easily identify how the data has quite a strong pattern in contrast to these eigenvectors. Here we have plot the eigenvectors that are derived from the covariance matrix.

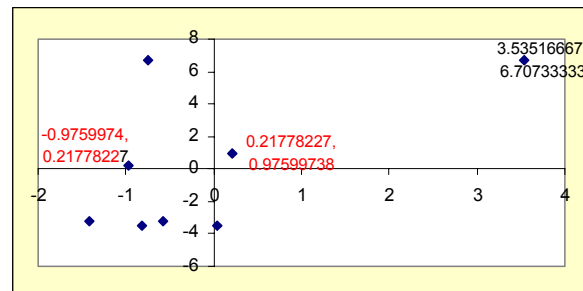


Figure 2: A plot of the normalized data (mean subtracted) with the eigenvectors of the covariance matrix overlaid

#### 4.5 Choosing the Desired Component

As a last final phase of PCA we take on the sorted list of eigenvectors that we hinted about in the previous section. Now we have components in order of significance. Here since we are dealing with eigenvalues of lesser significance therefore we omit eigenvalues of lesser connotations. Hence we now have final data set of lesser dimensions than the original. From here we move forward with the construction of another matrix called 'feature vector' which is responsible for holding the selected vectors column wise. As soon as we select these more significant components i.e. significant vectors or principal components in our case we multiply this matrix with the matrix of original data, but before multiplication we take transpose of both the matrices. This gives us another

matrix which in our case is called '*final data*' matrix with data items in columns and dimensions along rows. Basically what we have done here with our data is that we have changed it in terms of the patterns between them. These patterns are in the form of lines that most closely describe the relationship between the data sets. With the help of this line pattern we can pinpoint the exact location about the data points exactly where they lie i.e. (above/below) the trend lines. By using all the above discussed transformation we have basically represented and altered our dataset in terms of eigenvectors instead of usual axes. Thus by identifying the variations of dataset on the line patterns we can easily detect any intensifying threat that is about to transform into anomalous behavior in the network or in our case on the underlying link.

## 5. CONCLUSIONS

Here we demonstrate a statistical approach for detecting network anomalies. Our approach uses PCA (Principal Component Analysis) to pinpoint and identify any upcoming anomalous behavior on the underlying link. For this purpose we evaluated and tested our method on various links between our university and other institutions and organizations such as SLAC-Stanford, CERN and CALTECH. We have shown how our method works, in a step by step approach with additional graphs and images where necessary.

As a future direction we are concentrating on a full fledged diagnosis suite which not only help detecting abnormal behaviors arising in the network in the context of available

bandwidth and round trip time but also in the perspective of packet losses, link capacity and etc. Also such functionality must be incorporated into this suite so that apart from detecting anomalies it can also identify and diagnose these abnormal behaviors as well.

## ACKNOWLEDGMENT

We are thankful to R. Les. Cottrell, Connie Logg, Jiri Nivartili, William Jerrod of SLAC-Stanford, Ejaz Ahmed, Zaheer A. Khan, Fawad Nazir and Sajjad Haider of NUST Institute of Information Technology-Pakistan for their continuous support, guidance and appreciation.

## REFERENCES

- [1] Anukool Lakhina, Mark Crovella and Christophe Diot. "Diagnosing Network-Wide Traffic Anomalies," *In Proceedings of ACM SIGCOMM 2004, August 2004*
- [2] Paul Barford, David Plonka. "Characteristics of network traffic flow anomalies". Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement, San Francisco, pages 69 - 73, ISBN: 1-58113-435-5, 2001.
- [3] Jiri Navratli, R. Les. Cottrell. "ABwE: A Practical Approach to Available Bandwidth Estimation". Stanford Linear Accelerator Center (SLAC), 2575 Sand Hill Road, Menlo Park, California 94025