New Methods for Network Traffic Anomaly Detection

Tahereh Babaie

School of Information Technologies

University of Sydney

A thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

2014

Abstract

Outlier detection is a well developed topic in data mining and has made remarkable inroads into many application domains. In this thesis we examine the efficacy of applying outlier detection techniques to understand the behaviour of anomalies in communication network traffic. We have identified several shortcomings. Our most finding is that known techniques either focus on characterizing the spatial or temporal behaviour of traffic but rarely both. For example DoS attacks are anomalies which violate temporal patterns while port scans violate the spatial equilibrium of network traffic. To address this observed weakness we have designed a new method for outlier detection based spectral decomposition of the Hankel matrix. The Hankel matrix is spatio-temporal correlation matrix and has been used in many other domains including climate data analysis and econometrics. To the best of our knowledge it has not been used for analysis of network traffic before. Using our approach we can seamlessly integrate the discovery of both spatial and temporal anomalies. Comparison with other state of the art methods in the networks community confirms that our approach can discover both DoS and port scan attacks. The spectral decomposition of the Hankel matrix is closely tied to the problem of inference in Linear Dynamical Systems (LDS). We introduce a new problem, the *Online Selective Anomaly Detection* (OSAD) problem, to model the situation where the objective is to report new anomalies in the system and suppress know faults. For example, in the network setting an operator may be interested in triggering an alarm for malicious attacks but not on faults caused by equipment failure. In order to solve OSAD we combine techniques from machine learning and control theory in a unique fashion. Machine Learning ideas are used to learn the parameters of an underlying data generating system. Control theory techniques are used to model the feedback and modify the residual generated by the data generating state model. Experiments on synthetic and real data sets confirm that the OSAD problem captures a general scenario and tightly integrates machine learning and control theory to solve a practical problem.

To My Parents...

Acknowledgements

The most rewarding aspect of the long but fulfilling journey that my thesis took me on is not the satisfaction of a finished project (as much as research can ever truly be finished), but rather being able to look back and remember all who have assisted and encouraged me during the successes and failures along the way.

I wish to thank my supervisor, Professor Sanjay Chawla, who provided encouraging and constructive supports throughout my degree.

I would also like to thank my degree collaborator the National ICT of Australia, NICTA.

Publication

1- T. Babaie, S. Chawla, R. Abeysuriya. "*Sleep Analytics and Online Selective Anomaly Detection*". Published in The Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD14, 2014, New York, NY, USA. pp. 362-371. isbn:978-1-4503-2956-9

2- T. Babaie, S. Chawla, S. Ardon, Y. Yu. "*A Unified Approach to Network Anomaly Detection*". Will be Published in The IEEE International Conference on Big Data 2014, IEEE BigData 2014.

3- K. Nguyen, T. Babaie, and S. Chawla. "*Network Anomaly Detection Using A Commute Distance Based Approach*". Published in The workshop of 10th The IEEE International Conference on Data Mining, ICDM2010. 2010, pp. 943 950. isbn: 978-1-4244-9244-2.

4- T. Babaie, S. Chawla. S.Ardon. "*Network Traffic Decomposition for Anomaly Detection*". Published in CoRR, abs/1403.0157, 2014. arXiv:1403.0157 [cs.LG], url: http://arxiv.org/abs/1403.0157.

Contents

List of Figures

List of Tables

1

Introduction

THE aim of this thesis is to pose and address the problem of outlier detection in the context of complex and big multivariate time series by combining learning and control theory. In this chapter we describe the main contributions of our thesis and provide an overview of our methodology and results.

Outlier mining is the identification of unexpected, rare, and suspicious objects which do not conform to an expected pattern or other items in data volumes [4]. Examples of outliers could be fraudulent activities in financial transaction records, Internet intrusions, medical and health problems, measurement errors in data derived from sensors or community outliers in information networks. Determining outliers is highly dependent on the context of the study and there is no rigid definition of what constitutes an anomaly.

In particular in the context of network anomaly detection (either malware attacks or failures), the outliers are often not rare items, but sudden eruptions in network activity. This pattern does not follow the common statistical definition of an outlier as an exceptional objective, and many outlier detection methods will fail on such data, unless it has been aggregated applicably.

To model the statistical properties of big data, it is often sensible to assume each observation to be correlated to the value of an underlying latent variable with less dimension, called state, that is evolving over the course of the sequence. In the other word, to separate the normal from the abnormal, a key idea is to operate in the latent as opposed to the observational space. The latent variable(s) captures the intrinsic (albeit unknown) state of the data and gives rise to the observational data. True anomalies will cause changes in the intrinsic state of the data which will then be reflected in observations. For example, normal network traffic in Internet can be considered as a superposition of several periodic trends (half-day, daily, weekly, etc.). These trends are not overtly visible and are obfuscated by the normal variation in traffic. Random noise will only cause a change in measurement and will not have an effect on latent variables. Thus by directly operating with the latent variable, will lead to higher recall and precision and ultimately better detection capability.

The question will be just raised here is how information about the changes in the latent space can be retained using only a fractional measurement or observation? Or, is this possible to build the intrinsic structure of a dynamic sequence by observing a partial of its behaviour? A key theorem from Taken in 1981 [5], called *Embedding Theorem*, replied to this fundamental problem in control theory, by proving that a sufficiently long set of observations from a dynamic object has enough information for recovering its unknown latent variables. In the other word, we do not have to measure all the latent variables of the system. The theorem specifically illustrates that the latent variables can be retained using *method of delays* which builds a *Hankel matrix* from observations.

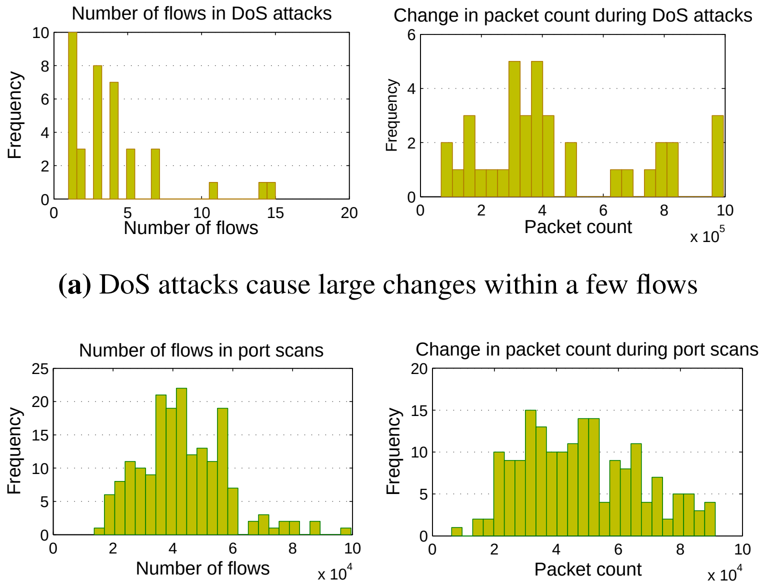
We show that how Hankel matrices make us able to recover the structural change in internal state of an observation in both model-based and model-free schemes.

One application we mainly focus is *Network Anomaly Detection*. Malicious Internet attacks are increasingly growing in both volume and sophistication. Experts have estimated that cybercrime now costs businesses hundreds of billions a year with a Web-based attack was blocked every 0.35 seconds in 2012[1].

Current state of the art techniques are either designed or able to detect a certain class of network anomalies at the cost of others. For example, wavelets analysis is quite accurate for detecting denial of service attacks (DoS) but is less accurate for identifying port scans [6, 7]. On the other hand, the recently introduced ASTUTE technique has exactly the opposite performance and displays high accuracy for detecting port scans but not DoS attacks [8, 9].

DoS and port scan attacks are emblematic of two types of deviations in network traffic. Fig. 4.12 shows how the number of flows and their packet counts change during real DoS attacks and port scans in a real network trace (the data will be introduced

[1] Source: http://www.symantec.com/threatreport/.



(a) DoS attacks cause large changes within a few flows

(b) A port scan attack causes small changes across many flows

Figure 1.1: Characterization of DoS attacks and port scans by the number of flows and the change in packet volume, for a traffic trace observed on a link in Abilene network, April 2007.

in section 3.5). DoS attacks are characterized by large changes in a (relatively) small number of flows as the attacking hosts send a large number of small packets, typically TCP SYN segments, to deplete system resources in the attacked host. Thus DoS like anomalies cause high temporal variation in the flows packet volume and can be detected using techniques based on time series analysis [6, 10, 11, 12, 13].

A port scan attack is typically accomplished by sending small packets as connections requests to a large number of different ports on a single destination IP address. At the flow level, they are therefore characterized as small increases in a large number of flows. Thus time series approaches often fail to detect port scans. This has prompted the introduction of new techniques, like the recently introduced ASTUTE method, to detect for spatial correlation across flows in order to find port scan attacks [8, 9].

In order to simultaneously capture both attacks we need to capture deviations from both the inherent spatial and temporal correlation in network traffic. In this project we present Multivariate Singular Spectrum Analysis (M-SSA), as a technique which can unify the detection of network anomalies. M-SSA is the successor of Singular Spectrum Analysis (SSA) as a robust version of the Taken's idea to reconstruct latent space from a time series [14, 15]. M-SSA requires the construction of a spatio-temporal covariance matrix which is then factorized using Singular Value Decomposition (SVD).

We show that (M-SSA) can significantly be applied to Internet traffic flows in order to provides a model for anomaly detection.

In the case where the state evolving through time by a linear function and the noise terms are assumed to be Gaussian, the resulting model is called a *Linear Dynamical System (LDS)*. The term dynamic model accounts for the behaviour of an object over time, in contrast a static (or steady-state) model calculates the objects behaviour in equilibrium, and thus is time-invariant. LDSs are an important tool for modelling time series in engineering, controls and economics as well as the physical and social sciences.

If data is generated by a LDS, then the SVD decomposition of the Hankel matrix can be used to estimate the LDS parameters. Several algorithms have been proposed including those based on gradient descent, Expectation Maximization, subspace identification and spectral approaches [16, 17, 18, 19].

The standard approach to detect outliers using an LDS is to use the inferred **A** and **C** matrices to compute the latent and observed error variables as:

$$\begin{aligned} \varepsilon(t) &:= x(t) - \hat{x}(t) \\ e(t) &:= y(t) - \hat{y}(t) \end{aligned}$$

where $\hat{x}$ and $\hat{y}$ are estimated using LDS. Then given a threshold parameter δ, an anomaly is reported whenever, $e(t) > \delta$.

There exists situations in which the objective is not to report all anomalies but suppress some known user-defined patterns or even known anomalous pattern. As an instance, port scanning represent a sizable portion of Internet anomalies which some times administrator wants to ignore and instead focus on more significant illegal activities like DoS attacks. Another example is Sleep EEG data in which tow significant anomalies are Sleep Spindle (SS) and K-Complexes (KC). Around 100 sleep spindles will occur during the course of a night. The number of K-Complexes is much fewer. For some experiments scientists are interested in identifying both sleep spindles and K-Complexes but only want to be notified with an alert when a non-spindle anomaly occurs (for example K-Complexes).

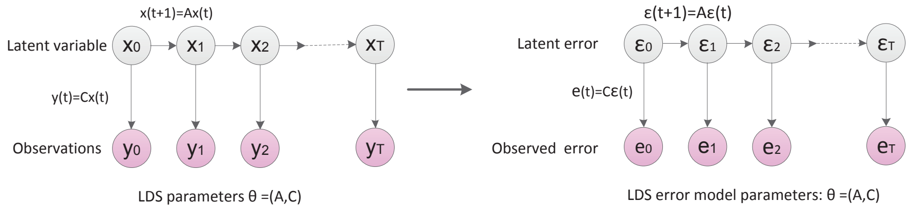


Figure 1.2: A linear dynamic system is a model which defines a linear relationship between the latent (or hidden) state of the model and observed outputs. The LDS parameters **A** and **C** need to be estimated from data. The LDS can also be used to model the relationship between the latent and the observed residuals (right figure).

In this research, We introduce the Online Selective Anomaly Detection (OSAD) problem which captures a particular scenario in sleep research.

The solution of the OSAD problem combines techniques form both data mining and control theory. Data Mining is used to model and infer the normal EEG pattern per subject. Experiments have shown that model parameters do not transfer accurately across to other subjects. In our case we will use a Linear Dynamical System (LDS) to model the EEG time series. Then based on frequency analysis, we infer the sleep spindle (SS) pattern and integrate the pattern as a disturbance into the LDS. The control theory part is used to *design* a new residual which suppresses SS signals but faithfully represents other errors generated by the LDS model. Thus by selectively suppressing SS pattern, the objectives of the OSAD problem are achieved.

for example, consider Figure 4.1. The top frame shows a typical EEG time series with both the SS and KC highlighted. The middle frame shows a typical residual time series based on an LDS model. The bottom frame shows a new residual designed to solve the OSAD problem. Notice that the error due to the presence of SS is suppressed but the residual due to the appearance of KC remains unaffected.

1.1 Contribution

We make the following four contributions:

- We have carried out an exhaustive survey of the traffic anomaly detection problem by creating a taxonomy based which includes: type of anomaly, time and

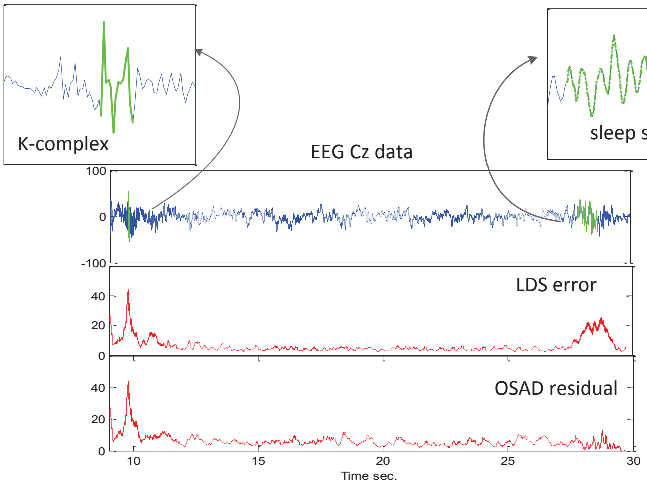


Figure 1.3: Sleep spindles (SS) along with K-Complexes (KC) are defining characteristics of stage 2 sleep. Both SS and KC will show up as residuals in an LDS system. The OSAD problem will lead to a *new* residual time-series where SS will be automatically suppressed but KC will remain unaffected. Due to relatively high frequency of SS, there are certain situations where sleep scientists only want to be alerted when a non-SS anomaly occurs

network granularity, detection method.

- A key observation that we have made is that network anomalies can be distinguished on the basis of within (temporal) and between (spatial) correlation. For example, DoS attacks are distinguished by violating the existing within correlation in a time series, while port scans are violate spatial correlation. Based on this observation we have designed a unified approach for network anomaly detection based on Hankel (trajectory) matrix decomposition. To the best of our knowlege, this is the first approach which can accurately detect both DoS attacks and port scans.

- We introduce a new computational problem, the Online Selective Anomaly Detection (OSAD), to model a specific scenario emerging while analysing time series data obtained from sleep experiments. The OSAD problem was introduced to design a residual system, where all anomalies (known and unknown) are detected but the system only triggers an alarm when non-SS anomalies appear.

- In order to solve OSAD, we combine techniques from data mining and control theory. In particular we will use a Linear Dynamic System (LDS) to model the underlying data generating process and use control theory techniques to design an appropriate residual system. In particular a function of the residual will be used to manipulate the changes in the error. The design objective will be to map the anomalies generated by the P pattern into the null space of the new residual. We claim that is one of the rare occasions where control theory techniques have been integrated with a data mining solution.

1. INTRODUCTION

2

Background on Network Anomaly Detection

CYBERATTACKS are now widely reported in the media and their frequency is growing. The aim of network intrusion detection techniques is to identify the digital signatures of known and predefined attacks in network traffic. However, cyber-attacks are constantly evolving and traditional intrusion detection systems are unable to detect what are called zero-day attacks. A new class of detection techniques which are based on the statistical analysis of network traffic have emerged for identifying zero-day attacks. These systems are often called network anomaly detection systems (NADS). The aim of this chapter is to survey known techniques in NADS and to suggest directions for future research. We present an exhaustive survey for the problem of traffic anomaly detection containing all the information the problem-solver needs for understanding and addressing the problem. Then we provide a taxonomy of current solutions in order to identify their contributions and point out their impacts as well as their drawbacks.

2.1 Preliminaries

Experts have estimated that cybercrime now costs businesses hundreds of billions a year. In 2012 a Web-based attack was blocked every 0.35 seconds[1]. Malicious Internet

[1] Source: http://www.symantec.com/threatreport/.

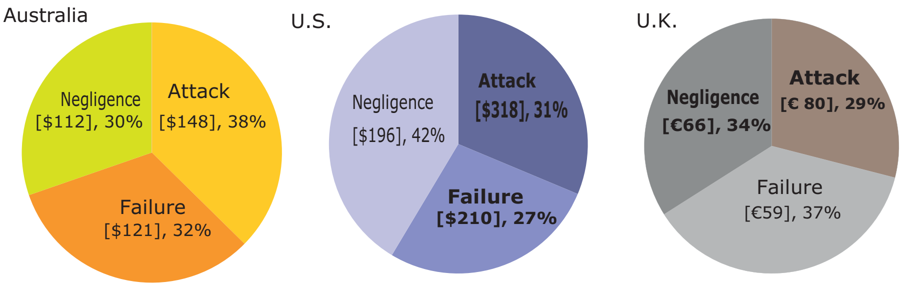


Figure 2.1: The 2010 costs per compromised record of data breaches by primary causes, along with frequency of the causes. The highest costs were due to malicious attacks.[2]

attacks are increasingly growing in both volume and sophistication. For example, the *Internet Security Threat Report* published by Symantec[1] in early 2012 noted web-based attacks increased by 36%, compared to the previous year, with over 4500 new attacks each day. The report also stated that 403 million new variants of malware were created in 2011, a 41% increase over 2010. Some high profile attacks which have made it to world headlines include the *Stuxnet* computer worm attack in 2010 and the denial of service attacks on credit card companies by supporters of Wikileaks.

Besides public and government concern about the security of the Internet infrastructure, considerable costs are incurred by organizations and companies to repair the damage after a cyber attack. A recent study in the US, UK and Australia estimated the cost per data record compromised by data breaches caused by malicious attacks, negligence and system failures. The cost due to malicious attacks were highest in all three countries (see Figure 1). For example, in the United States, the cost per affected data record caused by a malicious attack was 318USD compared to 210USD due to negligence and 196USD due to system failure. Similarly the *Annual US Cost of Data Breach Study*[2] notes that the average of total per-incident costs in 2010 was nearly 7.2 million, an increase of 7% from 2009, while the most expensive data breach event cost one organization 35.3 million to resolve. To fix network problems quickly and thus limit losses, we must be able to detect abnormal events in an acceptable time. Most commercially available security products use a signature-based model

[1] Symantec Corp., Internet Security Threat Report, http://www.symantec.com/

[2] A benchmark study of 51 U.S. companies, 38 UK companies, and 19 Australia companies related to breaches of sensitive information conducted by Ponemon Institute, LLC , http://www.symantec.com/

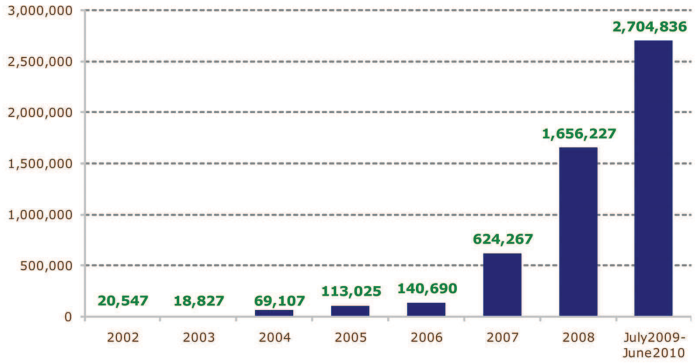


Figure 2.2: The number of new signature codes created by Symantec as malicious events. An explosive growth of new attack patterns is noticed in the Symantec reports from 2002 to 2010.[1]

[20, 21, 22, 23, 24, 25, 26] to prevent against malicious attacks. Systems which use signatures for detecting network anomalies are often called Intrusion Detection Systems (IDS). A signature is a distinctive pattern associated with a known attack(s). Once an attack is identified a signature is created and then registered with the system. For example, a common signature is to check if both the SYN and FIN flags in a TCP packet are simultaneously set. These are mutually exclusive flags as they determine the beginning and end of a tcp transmission sequence. How a system will react to these packets will depend upon the underlying operating system in place. Thus this attack can be used to determine the operating system in use.

There are two major limitations of signature-based systems. The first is that signature-based systems are vulnerable to new and previously unknown attacks. These are referred to as zero-days attacks. The second, is the fact that the number of attacks in growing at a rapid pace. According to a Symantec report, more than 286 million new threats were detected just in 2010, which is a huge increase compared to previous years. Fig.2.2 shows the number of new signatures created by Symantec each year from 2002 to 2010. There has been a dramatic rise in the number of new attack patterns discovered and documented during recent years.

Due to the above noted limitations of signature-based attacks the research focus has shifted to a statistical approach for detecting network anomalies. The key idea behind a statistical-based approach is to create a statistical profile of "normal traffic" and report deviations away from the normal behaviour as anomalies. The aim of this survey is to elaborate on this idea and survey the various techniques which have been used for both creating normal profiles but also detection systems which report deviations from normal behavior.

Detecting anomalous behavior through monitoring network resources is the main purpose of anomaly identification systems. Anomaly detection in the context of computer networks is finding unusual and large changes of interest in network traffic. Anomalies can be caused by many reasons, ranging from intentional attacks, e.g distributed denial of service (DDoS), to unusual network traffic, e.g flash crowds. Anomaly identification can be implemented on a traditional intrusion detection system (IDS) or a network anomaly detection system (NADS). Traditional IDS are based on finding attacks corresponding to predefined pattern data sets, known as signatures.

In response to the need for more effective identification, NADS have been introduced not only to detect zero-day attacks without any pre-identified signature, but to profile normal behavior of the network and address suspected incidents. Anomaly detection is an emerging research topic, although various commercial intrusion detection tools have been developed. Despite significant progress in the field of security, considerable research gaps remain. Addressing this issue involves developing an effective design approach for finding abnormal patterns in network behavior. Such a problem can be well addressed in data mining framework. However, there are reasons that make network anomaly detection a hard target for data mining approaches. First, network anomaly detection has not been clearly defined or mathematically clarified. Computer networks are huge in size and varied in data. Despite research progress in explaining traffic behavior in computer networks, the relation between network topologies and data transfer is still largely an open question. Second, a lack of agreement about how anomalies are defined makes it difficult to solve the problem of finding them. Most of the available data-sets lack confirmed labels of actual attacks experienced in a real network. Anomaly investigators manually identify and categorize anomalies using ground truth. Third, there is no substantial model for describing the behavior of computer networks in the context of data crossing over them. As a result, statistical non-parametric approaches are thus far the best data mining techniques for determining abnormalities in network data. However, few parametric approaches have been introduced to explain network behavior as a priori, in which any deviation from typical model would be considered as anomalous.

Developing an effective design approach to ensure efficient practical performance is therefore a high priority if we are to devote the next generation of network anomaly detection systems.

2.2 The NAD Problem Statement

Network providers are concerned about any change in traffic that might impact their Service-Level Agreements (SLAs) with their customers, including faulty or misconfigured routers, unexpected traffic such as flash crowds, and malware threats posed to their networks. Network anomaly detection dates back to the 80s when James Anderson introduced the notion of intrusion detection in a seminal paper [27]. This was the first prominent discussion of the concept of detecting misuses and determining user behaviors, and led to developments of auditing subsystems in every operating system. This work provided the foundation for future intrusion detection systems. In 1984, Dorothy Denning from SRI International helped to launch the intrusion detection expert system (IDES) on the original internet, ARPANET. Traditional intrusion detection solutions that have grown out of these efforts are almost all signature-based methods. A signature-based intrusion detection system uses a set of pre-configured and predetermined attack patterns, known as signatures, to catch a specific, malicious incident in network traffic. This is usually referred to as *misuse detection* in network. The set of the signatures must be frequently brought up to date to recognise new emerging threats to reach a high level of security performance [21].

In 1987, Denning published her important paper – *An Intrusion Detection Model* – in which she introduced the concept of network anomaly detection systems as an alarm scheme for abnormal system behavior [28]. Putting together an activity profile of normal activities over time and finding the deviation from these typical behaviors, she established a NAD approach, in contrast with the traditional IDS approach. This concept provided computer/network security field with the foundation for developing commercial IDS. An anomaly-based IDS sets up a routine activity baseline based on normal network traffic assessments. Thus, the behavior of network traffic activity can be monitored to enable action when network behavior varies from the typical activity profile.

In summary, while an IDS detects a known misuse signature in network traffic, NAD tries to identify a new or previously unknown abnormal behavior.
The research community has proposed a number of technical solutions to look for unusual changes in traffic behaviour and subsequently determines the causes of these changes. Traffic packets flowing through an Internet point consist of actual data, payload, routed by the headers which contain identification information such as the source and destination IP address of the traffic.

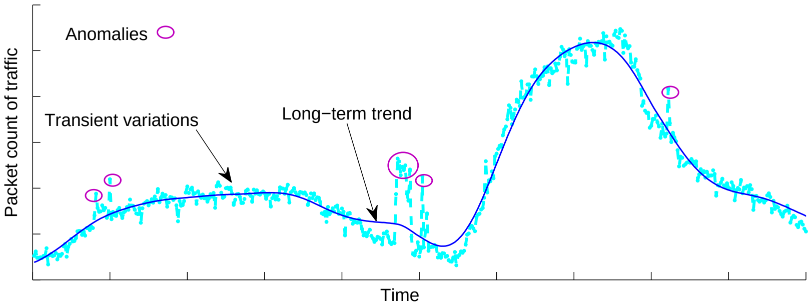


Figure 2.3: Traffic data is very dynamic, showing a long term trend pattern, transient oscillations with high frequency and significant changes in these oscillations are typically associated with anomalies.

Usually, sampled traffic from each node is processed for a period of time and a predefined sampling rate. Also, in order to avoid synchronization issues, usually traffic flow data is aggregated into time bins which can be some defined minutes. Anomaly detection procedures typically consist of two steps: (1) building a model that represents the time series, and (2) using the model to flag an *anomaly* whenever the observed traffic deviates from it. We employ an example taken from the Abilene network[1] traffic to demonstrate the concept of an anomaly detection in networks. The dotted line in Fig.2.3 shows a time series of the number of packets counted every five minutes on a link connecting users of Internet2 to a backbone router in New York. The long term trend is decoupled from the transient oscillation by using a Fourier analysis; shown with the solid line. Next, the detector method provides the tolerance for deviation

[1] http://www.internet2.edu/

from the baseline model, and consequently a time point is flagged anomalous if the observed value violates the tolerance.

2.2.1 Traffic Metrics

Early anomaly detection techniques investigated so-called volume metrics, i.e., the total number of packets, bytes, or connections observed on a single network link [6, 10, 11, 12, 13, 23, 29, 30, 31]. Volume metrics are immediately related to link's utilization so that high utilization can indicate attacks or flash crowds, while unusually low utilization can indicate link failures and routing changes. Since the early detectors were introduced to be installed in access links (e.g., in academic campuses [6] and enterprises [10]) where traffic is less aggregated than in backbones, volume investigation could easily expose the unusual events of these networks. Although, many anomalies could be covered by gigabytes of background traffic when an anomaly detector is deployed in the internet's core, but there are many anomalies that are difficult to be detected by volume analysis. This type of anomalies could be caught by analysing non-volume metrics such as number of flows observed in a link, or a network. Port scans are a prototype example of this sort of anomalies. Flow is a significant traffic metric widely used for anomalous traffic behaviour.

A TCP/IP flow is uniquely identified by as a unidirectional sequence of packets all sharing all of the following 5 values of header parameters, called 5-tuple, within a certain time period:

- Source IP address
- Destination IP address
- Source port number
- Destination port number
- Layer 4 protocol (TCP/UDP/ICMP)

With the increase in availability of flow-level traces (e.g., Cisco Net Flow1 and Juniper J-Flow2), Lakhina et al. [32] proposed using the entropy of header features of the flows, e.g. IP addresses and ports, as an effective metric for anomaly detection.

This was on the ground that entropy is a measure of dispersion and concentration in a given distribution. For example, the entropy of source IP addresses decreases during a DoS attack because the distribution of packets per address is concentrated in attacker IP address.

Some succeeding works have explored the use of other information-theoretic metrics in anomaly detection: Gu et al. [33] proposed the Kullback-Leibler (KL) divergence, to compare the distribution of packet classes inside a time bin to a baseline distribution obtained through a Maximum-Entropy optimization problem.

Later, Nychis et al. [34] showed that the entropies of flow size and degree distributions can flag low-volume anomalies in their dataset that go unnoticed in the entropies of features like addresses and ports.

2.2.2 Traffic Aggregation

Network monitoring systems needs to use data reduction practices to handle overload situations and generate practical traffic time series. This usually consists of packet/ traffic sampling, flow aggregation or a combination of them. Cisco's NetFlow [1], perhaps the most deployed solution in todays routers, uses packet sampling schemes to handle the large volumes of data exported and to reduce the load on the router. The sampling rate is defined at configuration time, and network administrators set it to a conservative value e.g, 1/100 or 1/1000 packets.

Various solutions for sampling techniques are available including Adaptive NetFlow [35], which is able to tune the sampling rate to the memory consumption , Flow Slices [36] ,which uses a combination of packet sampling and a variant of thresholds adapted to runtime conditions.

Another solution is using an aggregation technique, instead of sampling, to handle memory and CPU limitations [37]. [38] extended the Cisco's NetFlow into a report of 12 traffic summaries which are the answers for a number of predefined questions.

Note that using any traffic aggregations would generally lead to less accurate analysis as aggregations can contain too little, or too much, traffic, presenting a mixture of both legitimate and anomalous flows.

[1] Cisco NetFlow http://www.cisco.com/web/go/netflow/

2.2.3 Network Known-Anomalies Characterisation

Network attacks have evolved during recent years and became progressively more complicated. Malicious attacks are classified in distinct categories including a wide variety of viruses, worms and vicious programs. In the other side, there are some legitimate events that lead to abnormal behaviors in network. Here some of common network abnormalities and threats are described in order to a provide a general view of network anomalies.

DoS/DDoS: The goal of a denial of service attack is to make a target system's resource unavailable to prevent legitimate users from gaining access to the service provided. Typically attacker floods the target server until it becomes overloaded and cannot route legitimate traffic because of capacity deficient. The main feature of *DoS* attacks is the emergence of a spike in traffic data towards a dominant destination IP [39, 40].

Port scan: In a port scan attack, intruder scans TCP or UDP vulnerable ports to find services they can break into. Any spike in traffic data from a dominant source IP is assumed to be a suspected *port scan* attack [41].

Flash crowd: Flash-crowds occur when there is an unusually large demand for a resource, and are the non-malicious version of distributed denial of service (DDoS). The distinctive feature of a *flash crowd* is again a spike in traffic data to a dominant destination IP [39].

Worm: Computer worms are self-replicating program codes that are executed independently and spread across a network. Abusing security flaws in a target computer, a worm send copies of itself to other computers on the network without user involvement.

Ingress shift: An ingress shift anomaly happens when a customer shifts traffic from one ingress point to another. As an instance, when a client changes addresses of services or modifies routing policies in a service there will be an ingress shift. The main attribute of an *ingress shift* is that traffic in one group of OD flows, which include the existing ingress point; decrease while there will be a spike in another group of OD flows which involved a new ingress point [41].

Point to multipoint: It can be defined as distribution of content from a single source, e.g. one server, to many destinations, e.g. users. During a *point-to-multipoint* event there will be a spike in traffic from a dominant source to the same port of numerous destinations [41].

Outage events: Outage anomalies are equipment failures or maintenance events that cause decrease (or even to zero) in traffic exchanged between an origin and destination pair. When an *outage* happens, there is a dramatic decrease in traffic from a dominant source to a dominant destination. *Outage* anomalies are equipment failures or maintenance events that cause a decrease (even to zero) in traffic exchanged between an origin and destination pair.

Network anomalies can be classified based on their traffic properties, regardless of whether they represent malicious attacks, legal but abnormal incidents, or technical failures [42, 43].

Table 2.1: Characterizing network anomalies based on their features, number of bytes (#B), packets (#P) , flows (#F) and the entropy of flows features.

Anomalies	$volume$		$non-volume$	
	$\#B$	$\#P$	$\#F$	**entropies**
DoS/DDoS		⇑	⇑	⇕
port scan			⇑	⇕
flash crowd		⇑	⇑	⇕
worm			⇑	⇕
ingress shift	⇑	⇑	⇑	⇕
point to multi-point	⇑	⇑		⇕
outage	⇑			

The arrows ⇑ show spike in the feature.
The arrows ⇕ show change in the feature.

Traffic volume measurement in each time bin can be based on packet counting or byte counting. Packet-count traffic is the total number of packets counted in an OD flow or link measurement, while byte-count traffic is the total number of bytes. Some volume anomalies involve byte-count traffic change and some make changes in packet-count traffic; some anomalies can affect both. Network anomalies and threats

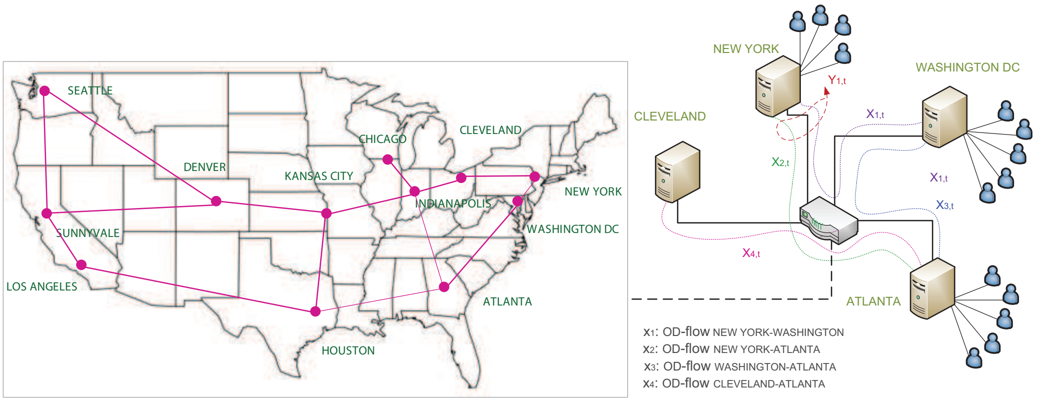


Figure 2.4: The Abilene network includes 11 regional aggregation points (giga-PoPs).

can be characterised based on their feature properties, as Table 2.1 shows a summary of anomalies feature characterisation.

2.2.4 Network-Wide Anomaly Detection

Diagnosing traffic anomalies spanning multiple links in a network is called network-wide anomaly detection. Significant traffic demand in a whole network is known as origin-destination flows (OD flows); described as a volume of traffic flows between all pair of PoPs in a specified network [44, 45, 46]. The links, where each OD flow passes through the network between source and destination, is determined in a routing matrix and consequently the superposition of those OD flows results in the traffic observed on each links.

In this work we use the Abilene network[1], used widely for network anomaly detection [8, 12, 32, 47, 48, 49].

This network includes 11 regional network aggregation points (giga-PoPs) with an OC-192c (10 Gbps) backbone connecting many universities, research labs and affiliate member institutions. The geographical topology of this network is shown in Fig.2.4. Consider a subset of the network consisting of four nodes: Cleveland, New York, Washington DC, and Atlanta, as shown in Fig.2.4. This network is being observed at time

[1] Internet2 - http://www.internet2.edu/

t. Suppose there are Four OD flows between NewYork and Washington DC (denoted by $x_{1,t}$), NewYork and Atlanta (denoted by $x_{2,t}$), Washinton DC and Atlanta (denoted by $x_{3,t}$) and Atlanta and Cleveland (denoted by $x_{4,t}$). Therefore, the traffic observed on the NewYork node, denoted by $y_{1,t}$ and called NewYork link, is the following superposition of passing OD flows:

$$y_{1,t} = x_{1,t} + x_{2,t}$$

And if it is done respectively for all links, the equations will be:

$$\begin{pmatrix} y_{1,t} \\ y_{2,t} \\ y_{3,t} \\ y_{4,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{1,t} \\ x_{2,t} \\ x_{3,t} \\ x_{4,t} \end{pmatrix}$$

Or in vector form:

$$\mathbf{y_t} = A_t \mathbf{x}_t$$

A_t is called the *Routing* matrix and that describes the routes of all OD flows. In a general network of M links and N OD flows, Routing matrix $A = (a_{mn})_{(M \times N)}$ is defined as:

$$a_{mn} = \begin{cases} 1 & \textit{if OD flow n pass through link m} \\ 0 & \textit{otherwise} \end{cases}$$

Assuming A_t is constant, we expand the traffic equation over time interval of $t = [1, ..., T]$, where vector $\mathbf{y}_t$ of size M is replaced by matrix $Y_{(M \times T)}$; which shows the traffic over links during time interval $[1, ..., T]$; and vector $\mathbf{x}_t$ of size N is replaced by matrix $X_{(N \times T)}$; which is the traffic volume over N routes during same time interval. So the equation in matrix form will be:

$$Y_{(M \times T)} = A_{(M \times N)} X_{(N \times T)}$$

Column vectors of Y and X represent the traffic volume of all M links or all N OD flows at different times, while row vectors in them display time series of traffic volume in links and OD flows, respectively. This traffic equation describes the relation between two multivariate time series in networks, OD flows and link matrices, which are connected to each other via Routing matrix.

Every sudden change in an OD flow traffic X is formally considered to be a *volume* anomaly, which often spans over several links in a network [41, 50, 51]. Such changes can be due to a range of anomalies surrounding changes in volume metrics in traffic. These anomalies are known as volume anomalies. For instance, when a DoS attack is launched, a large number of packets is sent from one host to a target server, which means that number of packets between this OD pair should dramatically increase during the attack time.

Early detectors relied on *volume metrics*, i.e., the total number of packets, bytes, or connections observed per time slot on a link (for example NewYork link in Abilene network). This was due to the availability of volume metrics through SNMP (*Simple Network Management Protocol* is an Internet-standard protocol for managing devices on IP networks). Then OD flow matrix X would be estimated through link measurements Y by solving an inverse problem. This process, called *Network Tomography* and its connection to anomaly detection will be discussed in next section. In a major work, Lakhina et al. [47] proposed analysing the spatial correlation across link measurements Y from multiple links in a network, to find the so-called network-wide anomalies. Accurately estimating OD flows (which typically are only estimated from link counts) can be very complex. Soule et al. [12, 52] discussed that traffic matrix X is indeed better than links and routers Y, because they lead to fewer false alarms. Later in a followed work, Lakhina et al. [32] aggregated traffic according to origin-destination (OD) flows and applied spatial correlation analysis directly to the traffic matrix X in a network, leading to a wider range of anomalies.

2.2.5 Network Anomography

Network anomography was first introduced by Zhang et. al, [13], to refer to the problem of finding network anomalies in the context of network tomography schemes. The term network tomography was coined by Vardi in 1996 [53], as the problem of estimating OD flow matrix X through link measurements Y. Network tomography has opened up an area of network study that involves solving an inverse under-determined linear equation system. A number of studies have attempted to solve the network inference problem, a major problem in traffic engineering. Network anomography framework tries to infer network anomalies (changes in OD flows) from non-direct measurements (link load measurements) [13]. To describe the scheme, again consider the network shown in Fig.2.5 and the corresponding traffic equation:

$$\mathrm{Y}_{(\mathrm{M}\times\mathrm{T})} = \mathrm{A}_{(\mathrm{M}\times\mathrm{N})}\mathrm{X}_{(\mathrm{N}\times\mathrm{T})}$$

Assuming only link data measurements (matrix Y) are available, there are two solution strategies for finding traffic anomalies in the above equation: early inverse and late inverse. The early inverse, which normally senses more instinctively, comprises two steps:

1. ***Network tomography:*** *finding OD flow matrix by solving the inverse equation of $X = A^{-1}Y$, which is an inference problem.*

2. ***Anomaly detection:*** *finding anomalies in inferred OD flow matrix, which is a detection problem.*

Despite this simple and straightforward concept, the early inverse strategy deals with an ill-posed inverse problem whose solutions cannot be always available or accurate, so any imprecision will affect the results at next step. Late inverse strategy has been proposed [13] by moving the inverse problem to a later step and substituting anomaly detection in truthful link-load measurements for unavailable OD flow matrix masses:

1. ***Anomaly detection:*** *finding link anomalies in link-load measurements, which is a detection problem.*

2. ***Anomaly Inference:*** *inferring OD flow anomalies from link anomalies, which is an inverse problem.*

Thus, network anomography involves solving an inverse problem. Two solutions have been discussed for the linear inverse problem: classical Pseudoinverse and recently proposed maximum sparsity. Pseudoinverse is the common solution for finding inverse matrix in general, while maximum sparsity has been proved to show better results [13].

Pseudoinverse solution: With common assumption that A has full-column rank, its Pseudoinverse, denoted by A^+, gives a unique solution for inferred anomaly vector$\mathbf{x}$, denoted by $\tilde{\mathbf{x}}$, based on least square error. Since the matrix A normally has fewer rows (number of links) than columns (number of OD pairs), so it is an under-determined case. We only need to search for a vector $\tilde{\mathbf{x}}$ with minimum Euclidean norm l^2 – in other words, to minimize the difference between $A^+\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$:

$$\parallel A^+\tilde{\mathbf{x}} - \tilde{\mathbf{y}} \parallel_2$$

Minimise $\parallel \tilde{\mathbf{x}} \parallel_2$ subject to $\parallel A^+\tilde{\mathbf{x}} - \tilde{\mathbf{y}} \parallel_2$ is minimal
Euclidean norm is defined by:

$$\parallel \tilde{\mathbf{x}} \parallel_2 = \sqrt{\sum_i \tilde{\mathbf{x}}_i^2}$$

Pseudoinverse is the classical solution to inverse problems, but the results in most applications are not useful because unknown coefficients seldom have zero effects [13].

Maximum sparsity solution: More recent studies have focused on enforcing the sparsity constraint when solving for the under-determined system of linear equations. Since there are typically just a few large values of anomalies at each point of time, the data is sparse. Consequently, we can maximize the sparsity of $\tilde{\mathbf{x}}$ by minimizing its l^0, which means maximizing the number of zero coefficients. Minimize $\parallel \tilde{\mathbf{x}} \parallel_0$ subject to $\tilde{\mathbf{y}} = \mathrm{A}^+\tilde{\mathbf{x}}$ where:

$$\parallel \tilde{\mathbf{x}} \parallel_0 = \sum_i \tilde{\mathbf{x}}_i^0$$

This minimisation is computationally intractable and NP-hard because of the non-convexity of l^0. In practical terms, there are two strategies to deal with minimising the l^0 norm: either using heuristics such as greedy algorithms as good examples, or using a convex function to approximately minimise the l^0 norm. Based on a recent work on under-determined systems [54, 55], minimising the l^1 norm is equivalent to minimising the l^0 norm in sparse solutions. In fact, l^0 is convexified by replacing with l^1, defined as:

$$\parallel \tilde{\mathbf{x}} \parallel_1 = \sum_i \mid \tilde{\mathbf{x}}_i \mid$$

Minimising the l^1 norm is done as a linear program by available efficient methods.

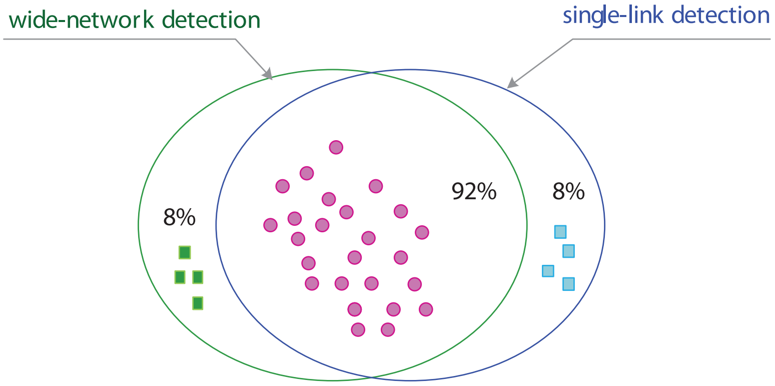


Figure 2.5: Running Kalman filter on Abilene data shows the overlap between anomaly found by each approach is large while both approaches find some anomalies that the other misses.

2.2.6 Single-link Vs. Network-wide Detection

Separating normal and anomalous network-wide traffic conditions we are able to find anomalies spanning multiple links in a network. There are many advantages related with network-wide traffic analysis but they are also followed by some disadvantages. The first drawback that has to be faced while using this framework is the need for ISP (*Internet Service Providers*) support. Some other drawbacks related with network-wide methods are that it is computationally expensive, it needs centralised algorithm and provides only single time scale analysis. There has been no quantitative evaluation of the advantage that network-wide frameworks have over single-link methods. However, in a sole work, Silveira et al. [56] run Kalman Filter as one of the network-wide anomaly detection techniques, using the data from all links in Inetnet2 at once, and also individually for each link. They report two important observations: first the intersection of anomalies found by both approaches is about 92%, and second both approaches have same complement anomalies around 8%. In the other word, both of them miss some anomalies which are only found by the other.

2.2.7 A Unified Statement

Network anomaly detection schemes aim at defining network traffic as

$$\text{Traffic} \approx \mathcal{F}(\text{ [normal component] , [abnormal component] }) + \text{Noise}$$

Assume that the variability of Internet traffic is characterised by a distribution which generates the observational data and is corrupted by external anomalous events and internal noise in the system as

$$y(t) = \mathcal{F}(\alpha_1, ..., \alpha_m,) + \varepsilon(t)$$

where $\mathcal{F}$ corresponds to the unknown distribution and $\varepsilon(t)$ captures the noise in the system. By monitoring the traffic flowing through an Internet point we obtain a time series of scalar measurements:

$$\mathbf{y} = (y(1), y(2), y(3), \ldots\ldots y(t), \ldots, y(n))$$

Here, each element of the time series could represent a certain time varying network traffic characteristic, for example, the volume of traffic or the entropy of flows in a certain time granularity. The "*Residual function*" defined by

$$R(\mathbf{y}; \alpha_0; \alpha_1; ...; \alpha_M) = \mathbf{y} - \mathcal{F}(\alpha_1, ..., \alpha_m,)$$

captures the difference between observations and the expected variations. Since the residual function $R(x; \alpha_m)$ is identical to $\varepsilon(t)$ which captures the noise in the system for the normal traffic behaviour, the first challenge then is to choose the proper fit $\mathcal{F}$ for normal behaviour and the second is how to investigate the residual space to flag an anomaly. The different NAD solutions differ mainly in their strategies for facing these challenges. Robust and reliable solutions to the above abstract problem require very accurate traffic models that have the ability to capture the statistical characteristics of the actual traffic on the network. However, the complexity of network traffic variability due to long-range dependence, self-similarity and, more recently, multifractality leads many NAD methods into failure.

2.3 The Solutions

2.3.1 Classification of NAD Solutions

Many approaches have been cited in the literature to address the network anomaly detection problem. We identify the contributions from proposed works by dividing them into sub-problems within traffic anomaly detection: (1) the network layer where traffic data are observed; and (2) the traffic metrics that can expose anomalies; and (3) the granularity of observation and (4) the time scale for observation and finally (5) the statistical techniques used to flag outliers in these traffic metrics. These sub-problems are identified in a taxonomy shown in Fig.2.6. Note that we separate this work from those analyse control data e.g., routing messages; also from those that analyse traffic from specific applications e.g., e-mail. The purpose of this work is to survey those techniques that have been analysing the general traffic data regardless of the application.

Network stack - Early NAD solutions have treated anomalies as non-conformities in the overall traffic volume measured in the Data-Link layer of the network stack. The volume indices can be either byte counts or packet counts. In wide-network granularity, the OD flow anomalies must be inferred by solving an under-determined inverse problem in the network traffic equation (discussed in section 2.2).
Next generation solutions developed with the increase in availability of flow-level traces by tools like Cisco NetFlow[1] and Juniper J-Flow[2]). Measurements in network layer include flow data on top of traffic volume count.

Granularity - There tow classes of approach within the current anomaly detection techniques; one exploits measurements from multiple vantage points called network-wide detection and the other one focuses on measurements from a single link. A wide-network anomaly detection approach can expos a wider range of suspicious events; although it would lead to a greater computational cost.

Traffic metric - NADS initially used traffic volume resource and gradually improved by introducing metrics of higher level of information, e.g. number of flows. Volume metrics including either packet counts or byte counts are available through protocols such as SNMP at Data-Link Layer. SNMP, a monitoring and management protocol, can measure the data at the link level by counting the number of packets or bytes entering the node, which results in matrix Y in the network traffic equation.

[1] Cisco NetFlow http://www.cisco.com/web/go/netflow/

[2] Juniper JFlow http://www.juniper.net/products/junos/

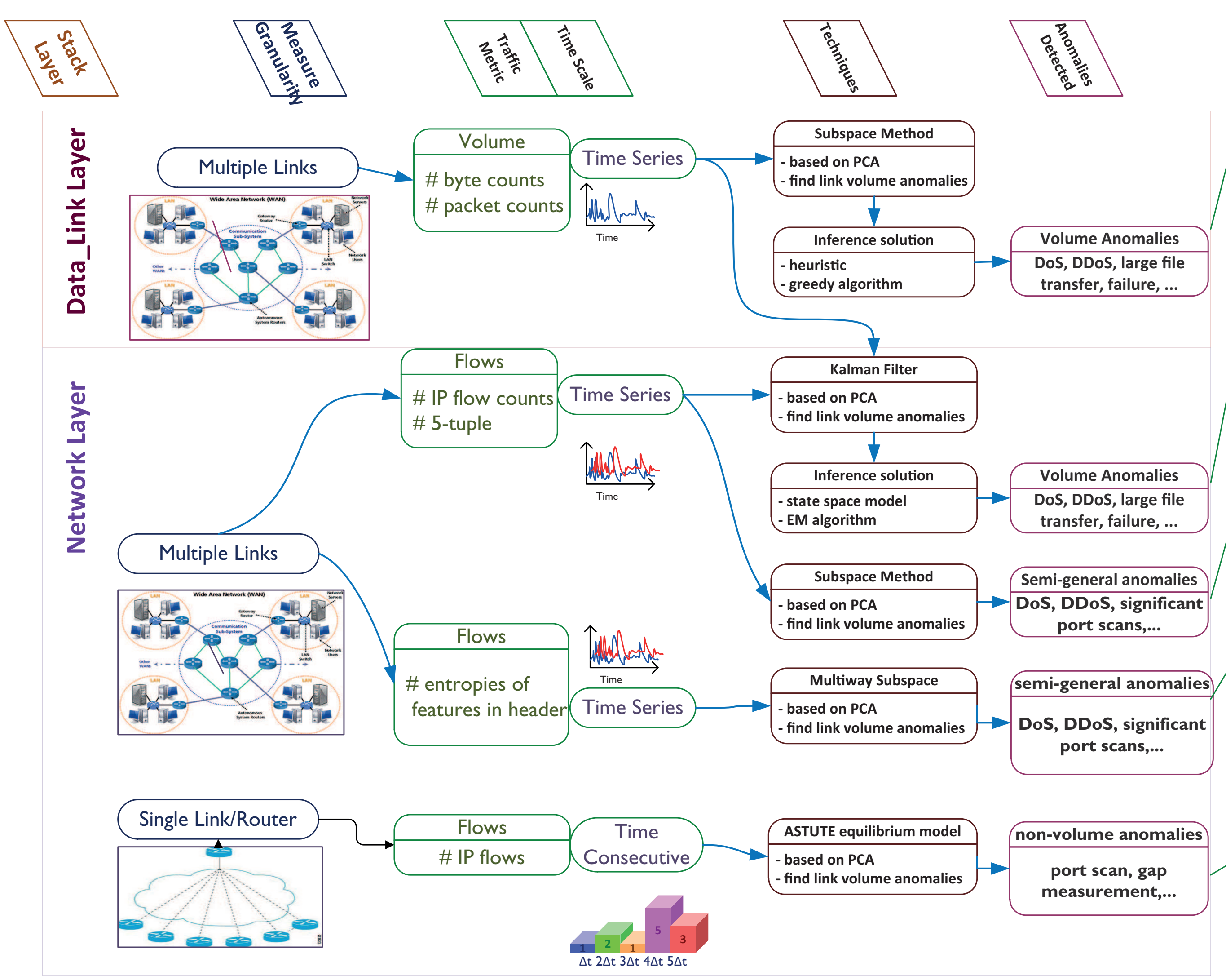
The number of traffic flows has been commonly used as an effective resource metric for identifying a wide range of volume and non-volume anomalies as large classes of anomalies do not cause noticeable change in traffic volume e.g., a slight port scan. Newer traffic analysing tools, e.g. Cisco's NetFlow, which was developed by Cisco Systems [57] and soon after became an industry standard for IP flow traffic measurement, measures the data at the network layer.

Another innovative metric for traffic measured in IP networks is the entropy of feature distributions. This metric calculates the normalised entropy of information in packet header. It has been shown that the patterns of many network anomalies have significant effects on this introduced measure [34].

Time scale - Most of the proposed techniques are based on time series analysis, in which a daily, weekly or monthly window of time bins is specified for constructing univariate/multivariate traffic matrices. Alternatively, consecutive time approach focuses only on every two consecutive time bins for analysis and decision making. Therefore, consecutive time analysis performs a local search of outliers in data, compared to the time series analysis that mines outliers within a whole set of data as global outliers.

Potential anomalies - In general, based on granularity, time window in data presentation, and statistical technique, each methods is capable of finding different types of anomalies in a network.

According to this classification, the first generation of NAD solutions is subspace method using link-level data, and involves detecting link anomalies and then inferring flow anomalies from them by solving an under-determined inverse problem. The subspace method using link-level data successfully spots volume anomalies. This subspace method is directly applied to the network-level time series in the next generation of the techniques and consequently, the inference problem would not be involved. The subspace method using network-level data could identify a wider range of volume and non-volume anomalies. The subspace method is the basis of many methodologies introduced to find anomalies, but Ringberg et al. have shown [58] that it is incapable of finding normal-space anomalies, and is very sensitive to parameter tuning.



1st generation was able to approximate anomalies in OD flow traffic from detected anomalies in link data measurement.

2nd generation was able to approximate OD flow anomalies in a wide network, from detected anomalies in link data.

3rd generation was able to detect anomalies by applying the same method of 1st generation on flow level data.

4th generation was able to detect a wider range of anomalies using a new metric.

5th generation has the capability of finding a range of anomalies which is difficult to be found by the others, however it is not able to find anomalies can be detected easily.

Figure 2.6: Our proposed taxonomy provides a classification framework for network anomaly diagnosis methodologies.

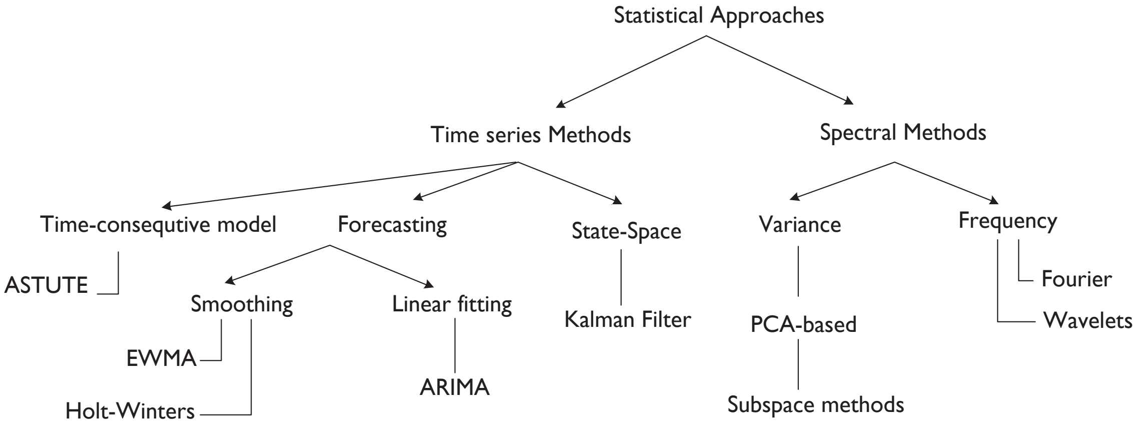


Figure 2.7: Classification of network anomaly detection methodologies based on the applied approaches.

Soule et al. [12] proposed the Kalman filter as a forecasting model for detecting and inferring anomalies in multiple links of a network, using both link measurements and IP flows counts. The third generation appeared in another work by Lakhina et al. [32], in which they applied the subspace method once again on the entropies metric and showed it yields a wider range of traffic anomalies. ASTUTE is the latest effort which offers using a new statistical model to expose anomalies that are more difficult to find by other approaches [59, 60, 61]. Since ASTUTE is incapable of finding anomalies associated with large number of flows, Silveira et al. suggested using a hybrid system consisting of ASTUTE and one of the common methods. However, implementing a hybrid system is likely to be complicated in practice.
We also classify the proposed methods based on characteristics derived from their technical approach, shown as Fig.2.7.

2.3.2 Timeseries Forecasting

Early traffic anomaly detectors have been developed on conventional time series forecasting techniques. They use historical observed traffic to build a predictive model; and consequently a prediction error between the observed and forecast values is computed. If this error violates a given detection threshold the method flags an alarm. Two types of time series forecasting approaches used for building traffic forecast prototype: (1) smoothing models and (2) Box-Jenkins models. Brutlag [10] and Krishnamurthy et al. [11, 39] proposed smoothing models: Moving Average, its weighted successors S-shaped Moving Average and Exponentially Weighted Moving Average (EWMA) and a Holt-Winters model. Consider the traffic $\mathbf{Y}$ of $m \times T$ measured as volume counts/ flow counts or entropies. then: Moving average model:

$$\hat{y}_t = \frac{\sum_{i=1}^{i=Q} \hat{y}_{t-i}}{Q}$$

Exponentially Weighted Moving Average (EWMA):

$$\hat{y}_t = \alpha y_{t-1} + (1-\alpha)\hat{y}_{t-1}$$

and Holt-Winters model which accounts for linear and seasonal trends:

$$\hat{y}_t = a_{t-1} + b_{t-1} + c_{t-m}$$

The three components correspond to a baseline, a linear trend and a seasonal trend respectively. The formulas to update these coefficients are

$$a_t = \alpha(y - t - c_{t-m}) + (1-\alpha)(a_{t-1} + b_{t-1})$$

$$b_t = \beta(a_t - a_{t-1}) + (1-\beta)b_{t-1}$$

$$c_t = \gamma(y_t - a_t) + (1-\gamma)c_{t-m}$$

2.3.3 Frequency Analysis

Frequency techniques, in general, transform a time series into a new frequency space, decomposing it into low, medium, and high frequency components. A NAD method using this approach assumes that normal traffic patterns lie in the low frequency components, and that changes in the medium and high frequency components are due to anomalies. So NAD frequency methods filter out high and medium frequencies components as they are supposed to capture the fast changes in traffic.

2.3.3.1 Fourier Analysis

Zhang et al. [13] applied a Fourier transform to decomposes the traffic time series into a linear mixture of Sine and Cosine principal components to investigate its frequency variations. Let $y[t]$ be the traffic measured as volume counts, flow counts or entropies.

- Transform $y[t]$ into the frequency domain using a Discrete Fourier Transform (DFT) defined by

$$f[t] = \frac{1}{T}\sum_{k=0}^{T-1} y[k]e^{-j\frac{2\pi}{T}kt} \quad for \quad 0 \leq k \leq T-1$$

- Set low frequency components to 0

$$f[k] = 0 \quad for \quad k \in [1, f_c] \cup [T - f_c, T]$$

  where f_c is called *cut-off frequency*. For example, for a 5-minute aggregated traffic data a cut-off frequency of one cycle per hour corresponds to $f_c = \frac{5}{60}T$.

- Use the Inverse Discrete Fourier Transform (IDFT) to reconstruct the traffic in time domain by

$$\tilde{y}[t] = \sum_{k=0}^{T-1} f[k]e^{j\frac{2\pi}{T}kt} \quad ,for \quad 0 \leq k \leq T-1$$

This is the residual space resulting from the Fourier analysis.

$$R(t, f_c) = \tilde{y}[t]$$

The most popular algorithm for calculating both DFT and IDFT is Fast Fourier Transformation (FFT).

2.3.3.2 Wavelet Analysis

A more sophisticated spectral method used for network anomaly detection is Wavelet analysis, as developed by Barford et al. [6] and discussed by [13, 62]. Since classical Fourier analysis is not able to tackle fast and intermittent variation in data, the new wavelet-based spectral approach is based on embedding a manifold in time domain to study time-frequency variations of the data. In theory, a Wavelet transformation decomposes a time series into localised and scaled component by discriminating between fast and slow oscillations. Hence, it projects the data onto a set of non-orthogonal functions. A NAD algorithm based on Wavelets shares the same principle as the FFT based approach: filter out mid and high frequency components as residual space. The algorithm can be summarised as following.

- Apply a multi-level one-dimensional Wavelet decomposition on each row of traffic matrix to get the Wavelet transformed of the traffic series

  $$f[t,\tau] = \frac{1}{\sqrt{c}} \sum_{k=0}^{T-1} y[k]\psi[(\frac{k}{c} - \tau)T]$$

  where c is scaling factor, τ is time factor and ψ represents the Wavelet function. The Wavelet function used in the implementations is Daubechies [63] Wavelet of order 6.

- Set the coefficients at frequencies higher than a cut-off level ω_c. In [6, 8, 13] ω_c of 3 used meaning only coefficient at frequency level 1,2 and 3 are kept.

- Reconstruct the high frequency space of the traffic by applying the Wavelet reconstruction procedure

  $$\tilde{y}[t] =$$

This is the residual space resulting from the Fourier analysis.

$$R(t,\omega_c) = \tilde{y}[t]$$

Later, Lu et al. in [7] attempt to extract a wider range of anomalies using wavelet approximation and prediction by ARX (AutoRegressive with eXogenous) model.

2.3.4 Subspace Method Using Link Data

Lakhina et al. [51] introduced the subspace method using principle component analysis to detect anomalies through link-data measurements, and combined it with a greedy algorithm to infer wide-network anomalies.

Principal component analysis: Known as one of the most common appearances of non-parametric methods in the dimensionality reduction problem [64, 65, 66, 67], principal components analysis (PCA) projects a multivariate space into a new subspace with the smaller number of uncorrelated variables while the rebuilt data has as little as possible change in variance. The new set of axes are called the principle component of main data. The first principal component represents the variable that captures maximal variance of data, and the next subsequent component corresponds to the variable capturing remaining maximal variance and is orthogonal to the first. The next ones have smaller variances and point to the directions of remaining orthogonal principal components.

Mathematical calculation includes eigenvalue decomposition of the covariance matrix of the data set, after removing the mean of the data for each attribute. By applying PCA to a matrix $\mathbf{Y}$ of size $T \times m$, a set of m principle components $\{\mathbf{u}_1, ..., \mathbf{u}_j, ..., \mathbf{u}_m\}$ is computed. Supposing $(\| \mathbf{Yu} \|_\mathbf{2})^\mathbf{2}$ is the variance captured by each principal component, then the first principal component $\mathbf{u_1}$ is given by:

$$\mathbf{u}_1 = \underset{\|\mathbf{u}\|_2=1}{\arg\max} \| \mathbf{Yu} \|_2$$

While the j^{th} principal component $\mathbf{u}_j$ is:

$$\mathbf{u}_j = \underset{\|\mathbf{u}\|_2=1}{\arg\max} \| (\mathbf{Y} - \sum_{\mathbf{i=1}}^{\mathbf{j-1}} \mathbf{Yu_j u_j}') \mathbf{u} \|_\mathbf{2}$$

Subspace method based on PCA : There are a number of statistical tests for anomalies using PCA. Dunia and Qin [68, 69] introduced a subspace approach based on the decomposition of a main space of data into normal and anomalous subspaces, using the projection of the data on the first few principle components and on the last few, respectively. In [51] and [50], Lakhina et al. applied this approach to the network anomaly detection problem using link measured volume traffic. In general, the last few principal components are likely to contain information that does not conform to the normal data [65]. Since the first few principal components capture most of variance in the dataset, they are strongly related to one or more of the original variables.

Consider the traffic $\mathbf{Y}$ of $m \times T$ measured as volume counts from m links of a wide-network over T times and apply PCA. Once ”Principle Components” have been determined by singular value decomposition of the covariance matrix, they are ordered by higher captured variance. Since PCs with higher variance characteristics are associated with common periodic and deterministic trends, they tend to capture the most significant normal behaviour. So, the first, let say k, eigenvectors form a subspace called normal subspace $\bar{S}$. PCs with low variance qualities are related to atypical and abnormal activities, so the remaining $m-k$ eigenvectors form an anomalous or residual subspace $\tilde{S}$. Followed by projecting the link traffic data $\mathbf{Y}$ onto these two assemblies, normal and abnormal subspace of data appear:

$$\mathbf{Y} = \bar{\mathbf{Y}} + \tilde{\mathbf{Y}}$$

Where $\bar{Y}$ is a projection of $\mathbf{Y}$ onto $\bar{S}$ and $\tilde{\mathbf{Y}}$ is its projection onto $\tilde{S}$. Eventually if the norm of a vector $\tilde{\mathbf{y}}_t$, $t = \{1, ..., T\}$, is large, then it is associated with an anomaly. Thus, in last step, an alert threshold is applied across anomalous subspace to quantify the significance of its vectors and detect suspected anomalies. When an anomaly occurs, the normal residual component will change significantly, while it is reasonable to assume that the normal components do not change. One of the proposed thresholds is squared prediction error (SPE) , defined as:

$$R = (\parallel \tilde{\mathbf{y}} \parallel_2)^2 = (\parallel \mathbf{y} - \bar{\mathbf{y}} \parallel_\mathbf{2})^\mathbf{2}$$

By applying a Q-statistics threshold for the squared prediction error at the $1-\beta$ confidence level, the time of an anomaly is detected [51, 70]. The subspace method's detailed procedure has been presented in algorithm 2.

Anomography: Now, one needs to infer anomalies in OD flows from detected anomalies in link data. Lakhina et al. [51] applied a greedy algorithm to identify anomalies. Let denote $\mathrm{A}_{(:,\mathrm{i})}$ as the i^{th} column of the matrix A; define $\alpha_\mathrm{i} = \mathrm{A}_{(:,\mathrm{i})} / \parallel \mathrm{A}_{(:,\mathrm{i})} \parallel_2$ and consider a set of hypothesized anomalies $\{\chi_\mathrm{i}\}_{\mathrm{i}=1}^{\mathrm{I}}$ such that every anomaly vector χ is a $\mathrm{M} \times 1$ vector. Each non-zero element in χ_i represents an OD flow that participates in the anomaly. For each anomalous time bin, the state vector can be decomposed to:

$$\mathbf{y} = \dot{\mathbf{y}} + \alpha_\mathbf{i} \chi_\mathbf{i}$$

As a maximum sparsity approach, the best estimate can be computed as:

$$\hat{\chi} = \arg\min_{\chi_i} \| \tilde{\mathbf{y}} - \hat{\alpha}\chi_i \|_2$$

This gives $\hat{\chi} = (\tilde{\alpha}'\tilde{\alpha})^{-\mathbf{1}}\tilde{\alpha}'\tilde{\mathbf{y}}$ by applying $\tilde{\mathbf{y}} = (\mathbf{I} - \mathbf{UU}')\mathbf{y}$ and $\tilde{\alpha} = (I - UU')\alpha$. Let $\tilde{U} = (I - UU')$, now the anomaly $\tilde{\chi}$ can be identified in the following two-steps:

1. compute $\dot{\mathbf{y}}$ by:
$$\dot{\mathbf{y}} = \mathbf{y} - \alpha\chi_i = (I - \alpha(\tilde{\alpha}'\tilde{\alpha})^{-1}\tilde{\alpha}'\tilde{U})\mathbf{y}$$
2. anomaly $\tilde{\chi}_i$ is given as $i = \arg\min_i \| \tilde{U}\dot{\mathbf{y}}_\mathbf{i} \|_\mathbf{2}$

2.3.5 Subspace Method Using Flow Data

Traffic flows are the source of rich information in networks. New flow monitoring tools such as Cisco's NetFlow understand the origin, the traffic destination, the time of day and the application utilization. Recording this data, we can construct rich information resources for traffic flows, which until now has not been easily available [71]. Again, consider the network with specified N OD flows (in terms of byte counts, packet counts and IP flow counts) over a specified time interval $[1 \quad T]$. In this network, OD flow traffic matrix is constructed as $\mathrm{X}_{(\mathrm{N}\times\mathrm{T})}$; where each row $\mathrm{X}(\mathrm{i},:)$ corresponds to the time series of i^{th} OD flow, $(1 \leq \mathrm{i} \leq \mathrm{N})$, and each column $\mathrm{X}(:,\mathrm{t})$ represents the observed OD flows at time t, $(1 \leq \mathrm{t} \leq \mathrm{T})$. The OD flow matrix can be constructed in terms of byte counts, packet counts, or IP flow counts. Applying PCA subspace method directly to this data will detect network anomalies, for which the inference problem is not involved.

2.3.6 Kalman Filter Method

Soule et al. [12] proposed using a Kalman filter as a forecasting model for detecting anomalies in multiple links of a network. The Kalman filter has its root in dynamic systems theory where the changes over time that occur in a set of variables is described by employing some equations. The main advantage of this model over the subspace method is that it is able to exploit both temporal and spatial correlations in data while the subspace method only exploits spatial correlation.

The idea is that since the OD flows are not directly measured while link measurements are observed, the whole network can be considered as a linear dynamic system with some hidden variables (OD flows) and observable dependent variables (link measurements). Assuming that OD flows are hidden states of a dynamic system and the link data are measurable output of the system, one can build a linear dynamic state space model. Therefore, this dynamic model incorporates both OD flow data and link data, and captures both temporal and spatial correlation in traffic flows. Such a model can be used for estimating the OD flows from link measurements and then finding anomalies through filtering the estimated states (estimated OD flows).

Modelling a linear dynamic system includes two steps: relating the observations to the states and capturing the dynamic behaviour of the states as the evolution of states in time. The first step is easily done by using traffic equation $Y_t = A_t Xt + V_t$, where the term V_t represents the stochastic measurement errors associated with the data collection step; all of these parameters are defined for a general discrete time t. The second step includes modelling the variability in states behaviour. Because of the diverse range of variability, such as daily periodic trends and small magnitude random fluctuations, the traffic in the network presents highly variable behaviour. Modelling this dynamic behaviour involves complexity but provides rich information about a system. For this purpose, a linear predictive model is constructed to relate the states between time $t+1$ and t as $X_{t+1} = B_t X_t + W_t$, where B_t is called transition matrix, which captures temporal and spatial correlations in the system, and W_t, which is a noise process representing both the randomness in the fluctuation of a flow and the imperfection of the prediction model. The completed model based on both described equations is given by:

$$\begin{cases} \mathrm{X_{t+1} = B_t X_t + W_t} \\ \mathrm{Y_t = A_t Xt + V_t} \end{cases}$$

Assuming that both the state-noise W_t and the measurement-noise V_t are uncorrelated, and are zero-mean Gaussian white-noise processes and with covariance matrices Q_t and R_t, and given a set of observations $\{Y_1, \ldots, Y_{t+1}\}$, the aim of the problem is to establish an estimation filter to generate an optimal estimate of the states in time $t+1$, denoted by $\hat{X}_{t+1}$, while the optimality function is defined as *Minimum Variance Error Estimator*:

$$E[\|X_{t+1} - \hat{X}_{t+1}\|^2]$$
$$= E[(X_{t+1} - \hat{X}_{t+1})'(X_{t+1} - \hat{X}_{t+1})]$$

The Kalman filter deals with this problem by using a two-step updating strategy, (1) prediction (or time update) and (2) correction (or measurement update) that iterate for each time t. The time update projects the current state estimate ahead in time. The measurement update adjusts the projected estimate by an actual measurement at that time. Denote $\hat{X}_{t|i}$ as the estimate of the state at time t given observations up to i. Because of the noise term W_t with covariance Q_t, this estimation will have an associated variability, which is the covariance of the error denoted by $P_{t|i}$. Based on this description, the the filter updates the estimation of the states in an iteration loop as follows:

Prediction: Given $\hat{X}_{t\|t}$ and $P_{t\|i}$, the state is predicted and the variance of this estimation is computed:

$$\begin{cases} \hat{X}_{t+1|t} = B_t\hat{X}_{t|t} \\ P_{t+1|t} = B_tP_{t|t}B_t' + Q_t \end{cases}$$

Correction: Given the result from prediction step, the error between predicted and observed outputs is:

$$\varepsilon_{t+1} = Y_{t+1} - A_t\hat{X}_{t|t}$$

This is called *innovation* or *measurement residual*. And also the residual covariance is:

$$S_{t+1} = A_tP_{t+1|t}A_t' + R_{t+1}$$

And finally the states are updated as:

$$\hat{X}_{t+1|t+1} = \hat{X}_{t+1|t} + P_{t+1|t}A'S_{t+1}^{-1}\tilde{Y}_{t+1}$$

$$P_{t+1|t+1} = (I - P_{t+1|t}A'S_{t+1}^{-1}A)P_{t+1|t}$$

The term that multiplied to $\tilde{Y}_{t+1}$ in correction equation is known as *Optimal Kalman Gain*:

$$K_{t+1} = P_{t+1|t}A'S_{t+1}^{-1}$$

By substituting the Kalman gain in the above equations, they are simplified to:

$$\hat{X}_{t+1|t+1} = \hat{X}_{t+1|t} + K_{t+1}\varepsilon_{t+1}$$

$$P_{t+1|t+1} = (I - K_{t+1}A)P_{t+1|t}$$

By populating an optimal estimation of traffic matrix $\hat{X}_{t+1|t+1}$, The Kalman filter provides us with a rich information that can be used for anomaly detection. Supposing that the Kalman filter estimates traffic matrix well, so the correction factors added to an a priori estimate $\hat{X}_{t+1|t}$ to adjust an a posteriori estimate $\hat{X}_{t+1|t+1}$ should be negligible. Thus the appearance of a large correction of the model can be considered an anomaly event, and the error that is generated by the predictor should be examined for anomalies. The first error is called "innovation" denoted by ε, which has been already defined:

$$\varepsilon_{t+1} = Y_{t+1} - A_t\hat{X}_{t+1|t}$$

This is considered to be the white Gaussian noise, with covariance matrix $E[\varepsilon_{t+1}\varepsilon'_{t+1}] = S_{t+1}$. Since anomalies in the OD flows are sought, so the error between the estimated state $\hat{X}_{t+1|t+1}$ and predicted $\hat{X}_{t+1|t}$ is defined as:

$$R_{t+1} = \hat{X}_{t+1|t+1} - \hat{X}_{t+1|t} = K_{t+1}\varepsilon_{t+1}$$

This is also called the *innovation process*, which is a zero-mean Gaussian process with variance:

$$\begin{aligned} \Delta_{t+1} &= E[R_{t+1}R'_{t+1}] \\ &= K_{t+1}(AP_{t+1|t}A' + R_{t+1})K'_{t+1} \end{aligned}$$

Any non-zero residual can be interpreted as the occurrence of an anomaly; therefore examining these residuals achieves anomaly detection. The detailed procedure of using Kalman filter for anomaly detection has been presented in algorithm 3.

Model calibration: Using the Kalman filter model requires the matrices A, B, Q, R. These matrices represent the dynamic behaviour of a network system. Finding such a dynamic model for a network is not straightforward; however, any estimation of these dynamic variables can be very useful for a diverse application of network issues such as network monitoring, load prediction and anomaly detection. Assuming that the matrices A and R are known, matrices B and Q can be estimated using an expectation-maximisation algorithm [12, 52]. There is a need for the historic data of states of the system (OD flows) to be used in the algorithm to estimate B and Q. So, OD flow data is used to calibrate the dynamic equations and then the Kalman filter is used for anomaly prediction. In [12] it has been showed that the Kalman filter needs to be re-calibrated every few days (using OD flow data), while it filters the unpredictable process of network dynamics using only link measurements.

2.3.7 Multiway Subspace Method Using Feature Distributions

Traffic volume plays an important role in the characterisation of volume anomalies, but other features of traffic data must be investigated for non-volume anomaly detection. Some anomalies, such as port scans, do not change volume metrics during attack time, but rather change the distribution of packet attributes. When a port scan occurs, regardless of volume changes it causes the distribution of traffic towards many destination addresses while checking only a few destination ports. Thus, there will be a concentration of destination addresses of the victim along with a dispersion of destination ports. both feature distributions changed over the time that scan is running. On the other hand, volume measurements do not present much information about difference in anomalies, apart from volume, while unusual distributional properties in traffic features reveal rich information about the structure of different anomalies, which can be used for valuable automatic classification.

However high-dimensionality curses in the distributions of traffic features prevent the direct analysis of all distributions at the same time. It is necessary to summarise effectively the different aspects of feature distributions in a way that is suitable for both detection and classification. The ”entropy” is assumed to be a descriptive metric capable of finding distributional changes in time series.

Lakhina et al. [32], proposed the entropy of feature distributions in traffic data as a new metric for anomaly detection. They employed an extended version of the subspace method to expose a wider range of anomalies.

Entropy of feature distributions: As mentioned above, we look for unusual changes in a distribution while we are also interested in the degree of dispersal or concentration of the changes. The term ”entropy” has been suggested as a metric that satisfies both. Intuitively, the entropy measures the diversity of the data through a predefined time window. Suppose F is a random variable that can take a range of values $\{f_1 ... f_N\}$, while $p(f_i)$ denotes the probability that F takes the value f_i:

$$p(f_i) = Pr[F = f_i]$$

the entropy of the random variable F will be:

$$H(F) = -\sum_{i=1}^{N} (p(f_i) \times \log_2 p(f_i))$$

The minimum value of the entropy is zero if all the observed data points are the same; this is maximum concentration. Maximum value of entropy is $\log_2 N$, when each data point appears exactly the same number of times; this is maximum diversity. Divided by $\log_2 N$, the entropy is normalised to be in the $[0,1]$ interval.

To reveal this information from traffic flows, the entropy can be calculated for any traffic features. Traffic features emerge as the field in the header of a packet. The four most important components of traffic features are source address, destination address, source port and destination port. Suppose the source address is the feature F whose distribution we want to construct. Suppose there are N distinct PoPs with N IP addresses from f_1 to f_N, so for the time of observation:

$$p(f_i) = \frac{\text{the number of observed } f_i \text{ in the sample}}{\text{the number of total observations in the sample}}$$

So the distribution for the source address will be:

$$p(f_i) = \frac{\text{the number of packets with source address } f_i}{\text{the total number of packets in the sample}}$$

By constructing this equation for source port, destination IP address and destination port, respectively, we achieve four time series of feature distributions.

In addition, since the number of distinct values in the sampled set of packets N, affects directly the sampled entropy, so unusual traffic volume may also produce unusual entropy values.

Multiway subspace method: Anomalies typically change the distributions of multiple features of traffic. For example, when a DoS Attack happens, there will be a significant change in destination and source address distributions. In fact, entropies of both features decrease in the time interval that attack occurs. The first requirement is to isolate the change spanning to the multiple feature distributions. In addition, anomalies in a wide network topology, have significant impact on multiple OD flows; it means the second requirement is to exploit the changes spanning to multiple OD flows. Based on both requirements, Lakhina et al. in [32] introduced a multiway subspace method that can look for correlations between ensembles of OD flows along with all features of each OD flow. Putting together four obtained entropy series, defined for important features of traffic, they framed a multiway multivariate data matrix H, size $T \times 4N$, which contains four sub-matrices, all with size $T \times N$ as H_{srcIP}, h_{desIP}, H_{srcPn}, H_{desPn}:

$$H = \begin{pmatrix} H_{srcIP} & H_{desIP} & H_{srcPn} & H_{desPn} \end{pmatrix}_{T \times 4N}$$

The employed features include source and destination IP addresses along with port source and port destination. For example, H_{srcIP} is defined as multivariate entropy of source IP address series of length T time bins for each OD flow as follows:

$$H_{srcIP}(t,n) = \text{the entropy value at time } t \text{ for OD flow } n$$

This process is run for other features as well to construct the above discussed multiway multivariate data matrix H.

By applying the standard subspace method discussed in section 2.3.4, the normal and abnormal subspaces, denoted by $\tilde{H}$ and $\bar{H}$, will appear. Each OD flow feature-space can be expressed as:

$$H = \bar{H} + \tilde{H}$$

Setting an alert threshold corresponding to a desired false alarm rate, unusually large values of residuals $\| \tilde{H} \|^2$ are considered as anomalies:

$$R = (||H - \bar{H}||_2)^2 = (||\tilde{H}||_2)^2$$

2.3.8 ASTUTE: An Equilibrium Model

Recently a new model based on equilibrium analysis has been proposed under the name ASTUTE (A Short-Timescale Uncorrelated Traffic Equilibrium) [8, 9, 49]. The working assumptions of ASTUTE are that normal traffic consists mainly of uncorrelated flows and a signal for correlation in flows is indicative of an anomaly.

ASTUTE is based on a statistical test for inferring strong correlations among active flows on a single link. Supposedly, ASTUTE is established to find anomalies triggered by strongly correlated flow changes, i.e., events where several flows change their volume all together. They showed that many types of events (e.g., scanning and DDoS attacks) cause strongly correlated flows.

The new approach is not building up a normal traffic model using historical data and is therefore immune against data poisoning, but rather is based on two assumptions for empirical properties of the flows observed on a single link: timescale and correlation. If observed flows satisfy two specified assumptions regarding these properties, the traffic flows show equilibrium. These two assumptions, which form the core of the ASTUTE statistical test, are:

Assumption 1: Different flows that arrive at a link are statistically independent in terms of three properties: arrival time bins, number of time bins where the flow is active, and the vector flow volume for each active time bin.

Assumption 2: The distributions of these three vector properties are time-stationary.

Consider two consecutive time bins, t and $\mathrm{t}+1$. Suppose there are N active flows $\mathcal{F} = \{\mathrm{f}_1, \mathrm{f}_2, \ldots, \mathrm{f}_\mathrm{N}\}$ in time t or $t+1$. For each flow in $\mathcal{F}$, the volume change of f_i from t to $\mathrm{t}+1$ is denoted by $\delta_{\mathrm{f,t}}$ and $\Delta_t = \{\delta_{f_{1,t}}, \delta_{f_{2,t}}, \ldots, \delta_{f_{N,t}}\}$ represents the set of changes for all the active flows. The consequence of the ASTUTE model is determined in the following theorem, which is the basis of the proposed approach.

Theorem 1 *If assumptions 1 and 2 hold, the variables of* Δ_t *are i.i.d. random variables*[1].

[1] Independent and identically distributed random variable with zero means

Result 1 *A set of active flows $\mathcal{F}$ satisfies the ASTUTE model if the computed confidence interval for $\hat{\delta}_{\mathrm{t}}$ (average volume changes across flows) includes zero. Otherwise, there is an ASTUTE anomaly at the time bin* t.

If the changes of the flows in the above model Δ_{t} are considered as a sample population with sample mean of $\hat{\delta}_{\mathrm{t}}$ and the sample standard deviation of $\hat{\sigma}_{\mathrm{t}}$, then for large N, $\hat{\delta}_{\mathrm{t}}$ will have a confidence interval of $(1-\beta)$:

$$\mathrm{CI}_{\hat{\delta}_{\mathrm{t}}} = [\hat{\delta}_{\mathrm{t}} - \mathrm{T}_{\beta}\hat{\sigma}_{\mathrm{t}}/\sqrt{\mathrm{N}}, \hat{\delta}_{\mathrm{t}} + \mathrm{T}_{\beta}\hat{\sigma}_{\mathrm{t}}/\sqrt{\mathrm{N}}]$$

The quantity T' is clearly the smallest value of T_{β} that leads to an interval containing zero:

$$\mathrm{T}' = \frac{\hat{\delta}}{\hat{\sigma}}\sqrt{\mathrm{N}}$$

T' is called the ”ASTUTE Assessment Value” (AAV) of a given time bin t. It has been shown that for a large number of flows (at least 100) in a time bin, the AAV distribution is close to the Gaussian distribution [8, 9].

Since the T_{β} is the $(\frac{1-\beta}{2})^{th}$ percentile of the standard Gaussian distribution and AAV for large number of flows has same distribution too, so the ASTUTE is violated if and only if:

$$\|\mathrm{T}'\|_2 > \|\mathrm{T}_{\beta}\|_2$$

Now suppose that the ASTUTE is violated because the confidence interval does not contain the zero. There are three reasons for this violation:

- First, the confidence interval does not contain the zero for a fraction β of time bins. So it is expected for this fraction that ASTUTE model to be violated by normal traffic. This is called false positive rate and for decreasing it we should increase the T_{β} in the above inequality.

- Second, based on the ASTUTE theorem, some sets of flows do not hold the second assumption. The author in [8, 9] has established an experiment to pinpoint the time scales where stationarity assumption is held. It has been shown in [8, 9] that the flow properties are stationary for time bins between 1 and 5 minutes. Therefore, if flows are measured in 5 minutes time bins, they would hold the second assumption.

- Third, eventually the trigger must be the violation of the flow independence assumption. So an anomaly occurred because of observing strongly correlated flows.

ASTUTE uses a statistical test based on an arguable flow independence assumption. Assuming that flows are independent is reasonable for backbone links as the author discussed in [8]. But if the nature of the Internet traffic become strongly correlated then the ASTUTE approach is not applicable anymore [8].

In addition to the flow independence assumption, there are limitations arising from the other assumptions employed in ASTUTE. The first limitation concerns the active capacity of the link. This method is only valid on the unsaturated link because for a fully saturated link, the AAV (based on the average changes) is equal to zero and no anomaly will be triggered [8].

The next limitation arises from the determined nature of the model. ASTUTE seeks strong correlations due to concurrent changes among several flows. Based on this, ASTUTE cannot trigger an anomaly for a single high-volume flow because the ratio of $\frac{\hat{\delta}_t}{\hat{\sigma}_t}$ in AAV does not reach a large amount when both nominator and denominator increase at the same rate. Thus, large volume anomalies caused by one or a few flows cannot be detected as a result of ASTUTE. Therefore, the main drawback of ASTUTE as a detector is that it cannot detect anomalies if they are aggregated into a few large flows. But finding these anomalies is simple for most other methods. Therefore the question is how strong the correlation should be to be triggered by ASTUTE? The answer will depend on the threshold T_β, as Silveiria and et al. [8] showed that for a large enough number of changes (greater than 100) at least T_β^2 anomalous flows should be involved to be detected by ASTUTE.

The main advantage of ASTUTE as a detector is that it can detect a different range of anomalies which are difficult to be found by other methods. In the computational aspect ASTUTE has low complexity (looking only at that time bin and the previous one) and in sensitivity aspect it is quite robust as it has only one threshold to be tuned. Although all the results are under certain statistical assumptions. The ASTUTE's procedure has been presented in algorithm 4.

2.4 Evaluation Scheme

2.4.1 Anomaly Decision

The residual space (or residual assessment value, e.g. AAV in the ASTUTE technique) resulting from the solutions must be investigated for anomalies. The general property of the proposed traffic anomaly detection solutions ensures that the residual is an uncorrelated random process with a specific variance (and mean). This variance is an essential part of any anomaly filtering algorithm and is used to determine if an observation is normal or not. The Kalman filter and ASTUTE model as well as PCA based methods are defined in the context of a Gaussian hypothesis. This means the residual resulting from the normal behaviour is an uncorrelated white Gaussian process with a known variance. This property is an inherent part of Kalman filtering, due to the whitening property of the filter, and ASTUTE, due to the main fundamental theorem of the model. The subspace method using PCA is also defined in the context of a Gaussian hypothesis. Note that Kalman filter and PCA based methods can be used in non-Gaussian situations but they will not be optimal. Let's assume an anomaly $a[t]$ happens at time t. The residual space at time t can be decomposed as:

$$R[t] = R_{Gussion}[t] + R_{anomaly}[t].$$

In the other word, an anomaly $a[t]$ can be detected only if for some value of t, the R[t] deviates from a decision threshold.

Under the Gaussian assumption for the normal residual process, two basic assessments of accuracy ,(1) the false positive probability and (2) true detection probability, can be computed as the result of threshold application.

Detection probability is sensitive to true positives and disregards false alarms, and false positive probability is sensitive to false alarms while ignores false negatives. To estimate the comprehensive combinations of false and true positive rates that a solution is able to provide, the ”Receiver Operating Characteristic” (ROC) is used. A ROC curve can be derived by plotting the points (false positive rate, detection rate) for varying values of the decision threshold. ROC measures the ability of the detection to discriminate between two alternative outcomes, thus measuring resolution. Selecting every threshold will produce a specific contingency table that generates one point of the ROC curve. ROC curves describe the full trade-off between false positives and false negatives over all possible threshold settings, in particular, and over operating conditions, in general.

The area under the ROC curves, AUCs, is frequently used as a score for performance accuracy because the more rapidly a ROC curve climbs towards the upper left corner, the better the performance of its generating result. Therefore, comparing the AUCs corresponding to different experiments enables the evaluation of their performance. Therefore, ROC curve, are well studied to addressing the challenging aspect of trade-offs between two important evaluation metrics. Moreover, note that AUS in different ROCs corresponding to different approaches can be used for a comparative evaluation.

2.4.1.1 Decision Variable (D_β)

Decision threshold value (i.e., when to raise an alarm for any anomaly investigating R space) is critically functioned by false positive rates. Decision threshold must have associations with different residual spaces (are to be compared) in order to permit a direct and fair comparison between them. This is the major challenge in the evaluation where different techniques are to be compared.

Kalman assessment value, innovation process (see section 2.3.6), follows a Gaussian white noise process if the traffic changes are normally distributed. Thus, given a target false positive rate β, the corresponding Kalman threshold is the percentile $1-\beta/2$ of the standard normal distribution. This is analogous to the AAV in ASTUTE (see section 2.3.8).

For the decision threshold value, the PCA-based methods in [32, 47, 48] use the variables proposed by Jackson et al. [70] and Jensen et al. [72]. The threshold D_β is defined as

$$\begin{aligned} D_\beta &= Q(\lambda_{k+1} : \lambda_{\ell \times m}, \beta) \\ &= \phi_1 [\tfrac{(1-\beta)\sqrt(2\phi_2 h^2)}{\phi_1} + 1 + \tfrac{\phi_2 h(h-1)}{\phi_1^2}]^{1/h} \end{aligned}$$

denotes the threshold for the $1-\beta$ confidence level, corresponds to a false alarm rate of β, and

$$h = 1 - \frac{2\phi_1\phi_3}{3\phi_2^2}, \quad \phi_i = \sum_{j=k+1}^{\ell m} \lambda_i \quad for \quad i = 1,2,3.$$

Based on Jensen et al. [72] the Q in the above equation follows a Gaussian distribution, and this convergence is robust even when the original data deviates from a Gaussian distribution.

Therefore a given false positive rate, one can calculate the threshold has for all PCA-based techniques, ASTUTE and Kalman filter.

Unlike the above discussed techniques, there is no well-known relationship between the frequency-based techniques threshold value and the target false positive rate.

Zhang et al. [13] addressed the problem of comparing techniques with different threshold scales. Specifically, for evaluation one pick D_β so that non-scalable techniques (e.g. Fourier, Wavelet) can catch as many anomalies as the techniques with adjusted threshold.

2.4.2 Bayesian Detection Rate

So far we have described the basic measures of accuracy for evaluating effectiveness of anomaly detection techniques. In general, the *effectiveness* of a detector can be described as ”identifying intrusive events while keeping the false alarm rate at a tolerant point”. Axelsson et. al [73] showed that due to the *base-rate fallacy* phenomena, the key factor that limits the performance of detection systems is the false alarm rate. The base-rate fallacy is described in terms of the Bayesian detection rate that is actually the probability of being a real anomaly event for a positive alarm. Let us describe this by explaining the proper formulations. Suppose $\mathrm{A_R}$ and $\neg\mathrm{A_R}$ denote real anomalous and non-anomalous events, and $\mathrm{A_D}$ and $\neg\mathrm{A_D}$ denote the detected as anomaly and not detected as anomaly by system, which determines whether the alarm is set off. Based on these assumptions:

$$\mathrm{PoD} = \mathrm{P(A_D|A_R)}$$

$$\mathrm{PoFD} = \mathrm{P(A_D|\neg A_R)}$$

No suppose that an alarm rises; how likely is it that the detected event is a real attack? This probability, which quantifies our ultimate interest in detection systems, is called *Bayesian detection rate* and is defined as:

$$\text{Bayesian detection rate} = \mathrm{P(A_R|A_D)}$$

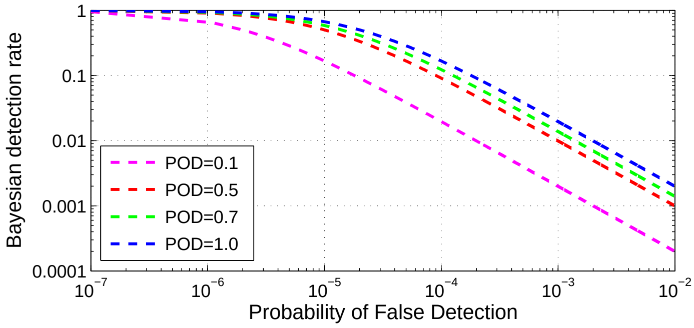


Figure 2.8: Bayesian detection rate versus False alarm rate

The reverse aspect of this measure is that when there is no alarm, how much should we worry? In other words, how likely is it that the normal tagged events would be a missed attack i.e. how much is $\mathrm{P}(\neg \mathrm{A_R}|\neg \mathrm{A_D})$? Therefore we need to maximize both probabilities for a proficient detection system. $P(A_R|A_D)$ can be calculated using Bay's theorem:

$$\mathrm{P}(\mathrm{A_R}|\mathrm{A_D}) = \frac{\mathrm{P}(\mathrm{A_R})\mathrm{P}(\mathrm{A_D}|\mathrm{A_R})}{\mathrm{P}(\mathrm{A_R})\mathrm{P}(\mathrm{A_D}|\mathrm{A_R}) + \mathrm{P}(\neg\mathrm{A_R})\mathrm{P}(\mathrm{A_D}|\neg\mathrm{A_R})}$$

Here, to clarify the main factors in effectiveness of detection systems, we make some assumptions based on a hypothesized network, including 10 audit records per anomaly, 2 anomalies per day, and $1,000,000$ audit records per day. Therefore:

$$\mathrm{P}(\mathrm{A_R}) = (\frac{1.10^6}{2.10})^{-1} = 2.10^{-5}$$

$$\mathrm{P}(\neg\mathrm{A_R}) = 1 - \mathrm{P}(\mathrm{A_R}) = 0.99998$$

and

$$\mathrm{P}(\mathrm{A_R}|\mathrm{A_D}) = \frac{2.10^{-5}.\mathrm{P}(\mathrm{A_D}|\mathrm{A_R})}{2.10^{-5}.\mathrm{P}(\mathrm{A_D}|\mathrm{A_R}) + 0.99998.\mathrm{P}(\mathrm{A_D}|\neg\mathrm{A_R})}$$

As the equation shows, the dominant factor is the factor of false positive detection rate in the denominator. Fig.2.8 depicts the values of Bayesian detection rate versus false positive rates for different detection rates. To achieve the desired maximum for $\mathrm{P}(\mathrm{A_R}|\mathrm{A_D}) = 100\%$, which is unattainable in practice, we would need to limit false alarm rate on the order of 10^{-5} to have only 66% of Bayesian detection rate! Thus, with this very low false alarm rate along with perfect detection rate, only about two-thirds of alarms are true abnormalities. Considering a realistic detection rate of 70% and the same false alarm rate, the Bayesian detection rate would be about 58%, meaning that

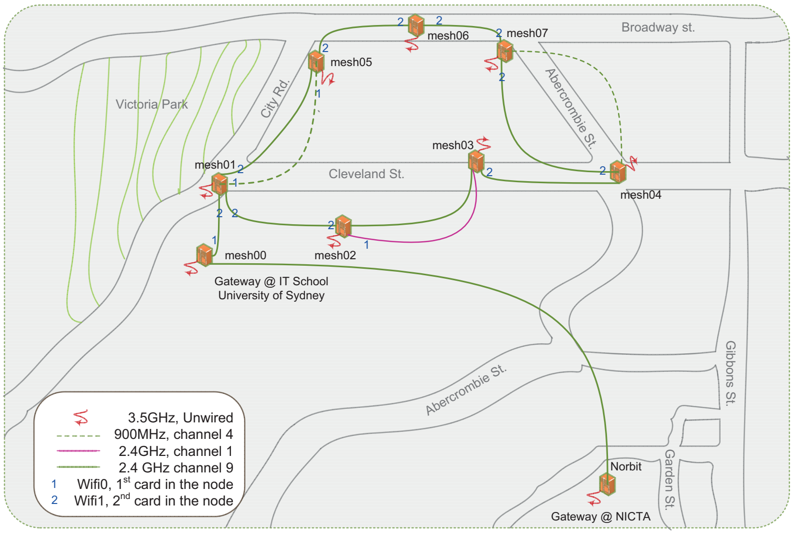


Figure 2.9: Layout of NICTA testbed, wireless types and channels. Note that all the links are wireless.

only half of alarms are true positives! The main cause of this circumstance, which is called base-rate fallacy, is because the problem is looking for a few rare points of anomalous events in an overwhelming number of normal points. As networks become huge and facilities faster, audit data sets become larger, but it is unlikely that attack activities will increase at the same rate.

2.5 Experiments and Discussion

Data: We use traffic data measured in an outdoor test-bed network developed as the smart transport and roads communications (STaRComm) project at National ICT Australia in Sydney[1]. This network has been used to produce a real traffic data set polluted with some common representative attacks. The network structure consists of seven nodes that have been connected through wireless channels and one gateway mesh node inside the School of IT at the University of Sydney[2]. The layout of the testbed in Sydney with all wireless links attributes, is shown in Fig.2.9. More detail of design, structure and measurement can be find at [74, 75].

[1] National ICT Australia www.nicta.com.au

[2] IT School, The University of Sydney www.it.sydney.edu.au

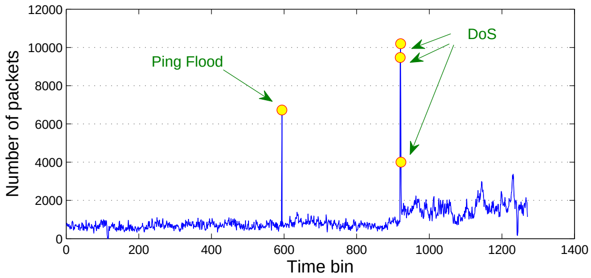


Figure 2.10: The time series of the number of transmitted packets during an observation window; DoS and Ping flood can be distinguished through flow count.

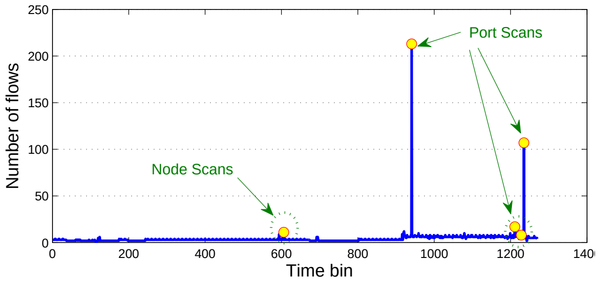


Figure 2.11: The time series of number of OD-flows involved in each time bin; port Scan can be distinguished through flow count.

Fig.2.10 shows the time series of the number of transmitted packets during an observation window – the sum of all packets in each time bin; Fig.2.11 represents the time series of number of OD-flows involved in each time bin (the number of flows in each time bin with no-zero packet). As these plots show, anomalies associated with DoS and Ping Flood can be distinguished via packet count analysis because of significant changes in number of packets, while the anomalies associated with port scan can be distinguished through flow count analysis because of significant changes in the number of flows. The anomalies associated with node scans cannot be distinguished by either analysis, because the testbed is only a small network of seven nodes, and therefore node scans do not produce a significant number of packets or flows.

To show how different proposed techniques detect attacks in traffic data we apply them to the constructed time series of NICTA testbed. We use PCA, ASTUTE and Kalman filter to find anomalies from the same data in order to compare their detection performance. The setting used in PCA includes top k = 1, and confidence interval for $1-\beta = 95\%$. We used same setting of confidence interval for ASTUTE and Kalman

Table 2.2: Number of anomalies detected by different introduced techniques

Anomalies	#Total	#found by PCA	#found by ASTUTE	#found by Kakman filter	#not-found
DoS attacks	3	3	0	3	0
Ping flood	1	1	0	1	0
port-scan attacks	4	1	4	0	1
node-scan	1	0	1	0	1
all attacks	9	4	5	5	1

filter. Table 2.2 shows the number of anomalies per type found by each method.

Result: We observe that the PCA can only find DoS attacks (including ping flood) as long as $k = 1$ is used for separating normal space from abnormal one. Note that this result is highly dependent on this setting so that if we chose $k = 4$ then non of the volume anomalies are detected. This will be discussed as the main issue of PCA in the section 2.5.1. Furthermore, ASTUTE is more effective at finding port scans as these attacks usually involve a large number of small flows comparing with DoS attacks which contain a few large flows. The Kalman filter is also capable of finding volume anomalies such as DoS and Ping flood while it cannot distinguish port scans as they contain small flows. We shall summarize that ASTUTE is accurate at finding non - volume anomalies; PCA is moderately effective at detecting volume attacks but the result is highly dependent on the tuning parameters; and the Kalman Filter is capable of finding various high volume attacks but it needs to be recalibrated frequently which involves a high computational cost.

2.5.1 PCA: Efficient Technique with Limitations

Principal component analysis is thus far the best statistical technique to detect network traffic anomalies. This approach, however, has empirical limitations.The inherent limitation of PCA in the statistical literature has been already discussed by [76] and [12]. The sensitivity of PCA's effectiveness has been addressed by Ringberg et al [58] as well. They showed experimentally that the PCA method is not only sensitive to the parameter that must be tuned , but that good results depend on the aggregation of the data that has been used. Another challenging point is related to the transformation stage. Although PCA's transformation helps to detect correlations, it makes it difficult to identify the original location of anomalies. Therefore, the main problems with the

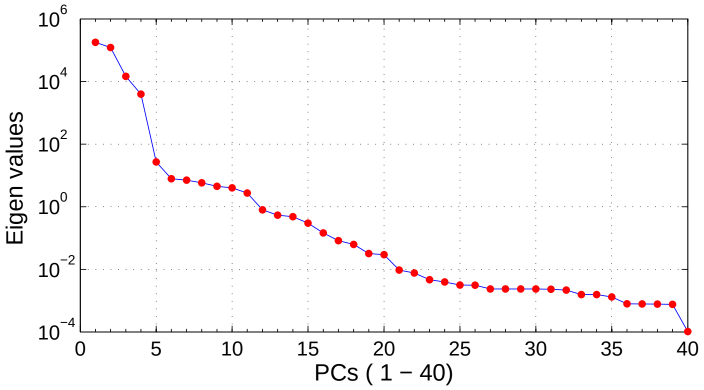


Figure 2.12: The first 40 eigenvalues calculated for NICTA dataset.

PCA-subspace methodology can be explained in terms of two processes: defining normalcy and remapping the coordinates.

The PCA-subspace method can be determined in three steps: modelling, detection and identification. The modeling phase includes separating normal and abnormal space by selecting the number of top principal components. In fact, the number of top PCs will determine the dimensionality of normal subspace. Different number of top PCs generates different normal and anomalous subspaces which is the main challenge of the methodology. Different methods have been proposed for tuning the number of top PCs, such as the 3σ deviation heuristic [50, 51], but results have shown that the methodology is still highly sensitive to the number of PCs included. The second issue in the modelling stage concerns contaminated normal subspace. If a very large anomaly appears in data, it will capture a large fraction of variance and consequently is included among top PCs. Therefore, large anomalies not only pollute the definition of normalcy but invalidate the intuitive assumption that the top PCs are semi-periodic by causing a spike in the first few PCs. The PCA-subspace method cannot detect sufficiently large anomalies in normal subspace because of its phenomenon.

The detection step includes analysing residuals to raise an alarm for the events on top of a defined threshold. So this step provides us with another tunable parameter, the detection threshold, which has a great impact on false positive rate. This threshold should be tuned so that the best trade-off between the false positive rate and the total detection rate is managed. Finally, we need to identify anomalies original location in the last step. The PCA-subspace method finds spikes on data projected on anomalous

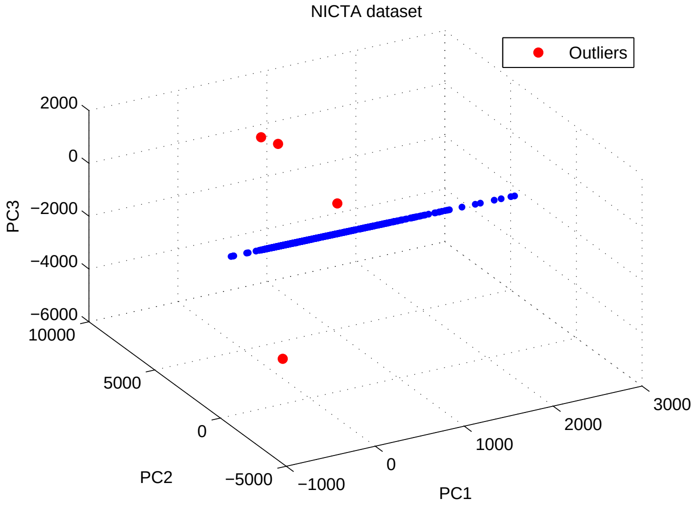


Figure 2.13: The NICTA dataset and the anomalies on the first three principal components. The variance was 0.99 for the three principal components.

subspace, whereas we need to know where it happened. In other words, we must identify which host, for example, is responsible for the detected anomaly to generate more actions for solving the problem or blocking the attackers. So far, all PCA-subspace methods have employed heuristics to associate a PCA-detected anomaly associated with a specific location.

To clarify PCA's limitations, we applied PCA to NICTA's traffic data, including a diverse range of volume and non-volume anomalies. We used a packet count traffic matrix as test data set. In this data set (except for the four volume anomalies), the remaining time bins are in the direction of the first eigenvector; thus, it is not difficult to detect these anomalies using the PCA. Fig.2.12 and Fig.2.13 show the first 50 eigenvalues and the PCA plot of the data set, respectively. The PCA result depends largely on the number of eigenvectors for normal subspace. In this data set, by choosing the number of eigenvectors $k = 1$ we successfully detected all four volume anomalies. If $k = 2$, only one volume anomaly was found, and if $k = 3$, all anomalies found were incorrect. Port Scan anomalies and node scans were not detected at all, which is predictable, because these types of anomaly do not have impact on packet count data.

2.5.2 ASTUTE: Anomalies and Correlation

Giving an example, in this section we describe how ASTUTE model spots anomalies. Suppose there are 2 active flows $\mathcal{F} = \{f_1, f_2\}$ in an arbitrary three consecutive time bins, t to $t+2$.

$$\begin{aligned}\mathcal{F}(t+0) &: f_1^{(0)} \quad f_2^{(0)}\\ \mathcal{F}(t+1) &: f_1^{(1)} \quad f_2^{(1)}\\ \mathcal{F}(t+2) &: f_1^{(2)} \quad f_2^{(2)}\end{aligned}$$

Suppose matrix $\mathbf{X}$ as volume time series of flows as:

$$\mathbf{X} = \begin{bmatrix} x_{f_1^{(0)}} & x_{f_2^{(0)}} \\ x_{f_1^{(1)}} & x_{f_2^{(1)}} \\ x_{f_1^{(2)}} & x_{f_2^{(2)}} \end{bmatrix}$$

where $x_{f_i^t}$ denotes the volume of f_i at time t.

ASTUTE basically assumes that these flows f_1 and f_2 are uncorrelated, otherwise an anomaly happens. So the default situation is a zero correlation and the correlation is sought in a temporal variability.

Such important assumption elicits the resulting conclusion: the changes of the active flows volume are standard Gaussian i.i.d. variables. Based on this result, ASTUTE's threshold for having zero mean changes in the flows volume is given by:

$$AAV = \frac{\hat{\delta}}{\hat{\sigma}}\sqrt{N}$$

ASTUTE model is looking for the time bins in which the active flows are correlated. In the other word, any point which violates the AAV threshold presents the existence of correlated flows in the related time bin. Note that ASTUTE determines the correlated flows by looking at two time consecutive bins. Let's define:

$$X_i^t = \begin{bmatrix} x_{f_i,t} \\ x_{f_i,t+1} \end{bmatrix}$$

then:

$$\Delta_1 = \{\delta_{f_{1,0}}, \delta_{f_{2,0}}\}$$

where $\delta_{f_{1,0}} = x_{f_1^{(1)}} - x_{f_1^{(0)}}$ and $\delta_{f_{2,0}} = x_{f_2^{(1)}} - x_{f_2^{(0)}}$. AAV is calculated by main statistical properties of the change vector $\Delta_1 = \{\delta_{f_{1,0}}, \delta_{f_{2,0}}\}$ with mean of δ_1 and standard deviation of σ_1:

$$\delta_1 = \frac{\delta_{f_{1,0}} + \delta_{f_{2,0}}}{2}$$

and:

$$\sigma_1 = \frac{1}{\sqrt{2}} \sqrt{(\delta_{f_{1,0}} - \delta_1)^2 + (\delta_{f_{2,0}} - \delta_1)^2}$$

Substituting δ_1 and σ_1 in $AAV_1 = \frac{\delta_1}{\sigma_1}\sqrt{2}$ we will achieve:

$$AAV_1 = \frac{\delta_{f_{1,0}} + \delta_{f_{2,0}}}{|\delta_{f_{1,0}} - \delta_{f_{2,0}}|}$$

We consider all possible scenarios for changes in the volume of flows from $t = 0$ to $t = 1$ and discuss how *AAV* threshold spots an abnormal flow or flows. We depict *AAV* variation over the change in the flows volume in Fig.2.14 by assuming a green point as initial point $(x_{f_1,0}, x_{f_2,0})$. We observe how the change over time can cause an *AAV* violation alarm.

Scenario 1: If the volume of f_1 and f_2 change but the changes stay the same, *AAV* reaches infinity. This reveals in fact the two flows are absolutely correlated. Setting $AAV = 7$, equivalent to a false positive rate of $p = 2 \times 10^{-5}$ [8], shows we make the model to tolerate $\frac{\delta_{f_{1,0}}}{\delta_{f_{2,0}}} = \frac{p+1}{p-1}$ and still consider f_1 and f_2 correlated. This has been shown in Fig.2.14 when the initial green point goes to any red point in hatched area.

Scenario 2: If the volume of f_1 and f_2 change in opposite way (one increases while the other decreases) but the absolute changes stay the same, *AAV* reaches zero. This shows that the two flows are absolutely negative correlated. Despite ASTUTE model is to find correlated flows, but it is unable to find this type of correlated flows.

Scenario 3: If the volume of one of the flows changes but the other one stays the same, *AAV* reaches one. This two flows are absolutely uncorrelated. Basically ASTUTE consider flows in these areas uncorrelated.

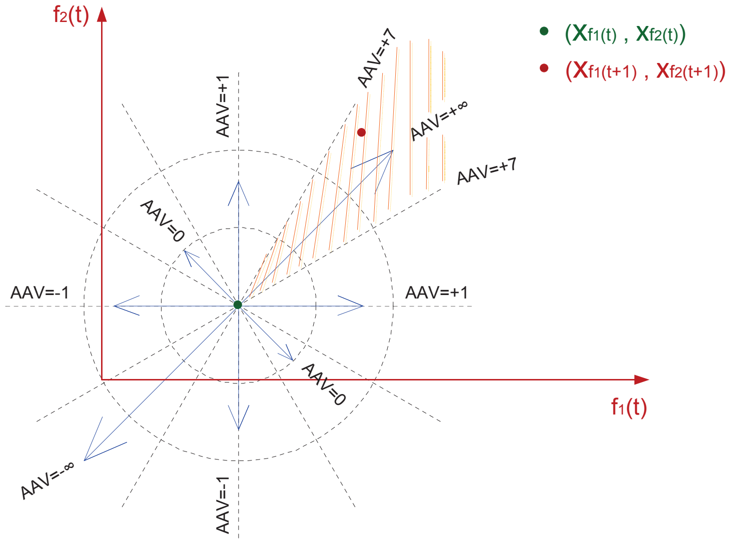


Figure 2.14: Schematic illustration of embedding a linear manifold into a time sires of M variables using a window of length L. By sliding this M × L window in the horizontal (x,τ)-plane, we look for spatio-temporal patterns, E-EOFs.

The above discussion can be generalised to N flows using the mean and standard deviation calculated for N changes and some algebraic calculation:

$$AAV_t = \frac{\sum_{i=1}^{N} \delta_{f_{i,t}}}{\sqrt{\sum_{i,j=1:N}^{i \neq j} (\delta_{f_{i,t}} - \delta_{f_{j,t}})^2}} \sqrt{N-1}$$

Theorem 2 *AAV is relatively large when port scans happen i.e.*

$$AAV_{tp} \gg 1$$

Proof: Port scans are a large number of flows change their volumes but the changes are small. The numerator of the ratio AAV_t is highly greater than the denominator because changes are small and their difference is much smaller.

$$(\delta_{f_{an,t}} - \delta_{f_{j,t}})^2 \quad \approx \quad \varepsilon$$

Then:

$$\begin{aligned} AAV_t &= \frac{\sum_{i=1}^{N} \delta_{f_{i,t}}}{\sqrt{\sum_{i,j=1:N}^{i \neq j} (\delta_{f_{i,t}} - \delta_{f_{j,t}})^2}} \sqrt{N-1} \\ &\approx \frac{\sum_{i=1}^{N} \delta_{f_{i,t}}}{\sqrt{\varepsilon}} \gg 1 \end{aligned}$$

Theorem 3 *AAV is approximately equal to 1 when DoSs happen i.e.*

$$AAV_{td} \approx 1$$

Proof: DoS attacks are a few number of flows change their volume dramatically. Suppose only one flow f_{an} is responsible for a DoS attack with a large number of packet change:

$$\delta_{f_{an,t}} \gg \delta_{f_{j,t}} \quad \forall j \in [2:N]$$

Based on this, the difference between this flow's change and the other flows $(\delta_{f_{an,t}} - \delta_{f_{j,t}})$ is is dominated by the $\delta_{f_{an,t}}$:

$$(\delta_{f_{an,t}} - \delta_{f_{j,t}})^2 \quad \approx \quad \delta_{f_{i,t}}^2$$

In the other side the $\delta_{f_{an,t}}$ appears in $N-1$ expression of differences:

$$\begin{aligned} AAV_t &= \frac{\sum_{i=1}^{N} \delta_{f_{i,t}}}{\sqrt{\sum_{i,j=1:N}^{i \neq j} (\delta_{f_{i,t}} - \delta_{f_{j,t}})^2}} \sqrt{N-1} \\ &\approx \frac{\delta_{f_{an,t}}}{\sqrt{(N-1)\delta_{f_{i,t}}^2}} = \pm 1 \end{aligned}$$

2.6 Summary

Traditional intrusion detection systems are based on finding attacks corresponding to predefined patterns known as signature. In contrast, network anomaly detection systems have been introduced to detect zero-day attacks without pre-identified signatures, to profile normal behavior automatically, and to address suspected incidents.

In this chapter we defined NAD problem in the context of data mining as a mathematical problem. We showed that the complexity of the problem is due to various requirement of a detection problem in comparison with today's huge networks. Some of the most influential solutions have been described. A taxonomy of the techniques based on the their approaches to solve NAD is proposed. Most methods have been based on the non-parametric PCA, which shows considerable drawbacks in practice. Furthermore, almost all the techniques show the capacity to find only some specific types of anomalies.

In summary, mining anomalies in network traffic has been researched and some approaches have been proposed, however, finding general anomalies in today's huge and complex networks remains a big challenge.

3

Spatio/Temporal Decomposition for Anomaly Detection

In this chapter we focus on the detection of network anomalies, such as Denial of Service (DoS) attacks and port scans, in a unified manner. While there has been extensive research on network anomaly detection, current state of the art methods are only able to detect one class of anomalies at the cost of others. Some anomalies (e.g. DoS attacks) are marked by a temporal variation while others (e.g. port scans) exhibit correlation across multiple traffic flows. Our method can identify both of these phenomenon using an approach based on the spectral decomposition of a Hankel matrix. We show this can detect deviations from correlations between traffic flows, as well as temporal variations within a flow, present in observed network traffic data. Detailed experiments on synthetic and real network traces show a significant improvement in detection capability over competing approaches. In the process we also address the issue of robustness of anomaly detection systems in a principled fashion.

3.1 Introduction

In its most abstract form, network traffic can be described by a time series $y(t)$, where y represents the observed state of the traffic. For example, $y(t)$ could simply be the total number of packets or could be a vector, where each component represents an active flow. A flow is an aggregation of packets by attributes such as source and destination

IP address.

In order to detect anomalies in network traffic we must first model the generative process, which gives rise to the observable time series $< y(t) >$. Assume that the latent variables $x(t)$. The relationship between $y(t)$ and $x(t)$ can be abstractly represented by a model as $y(t) = M(x(t))$. We can learn the model and obtain an estimation as $\hat{y} = M(\hat{x}(t))$. Then an anomaly occurs of time t if $y(t) - \hat{y}(t)$ is greater than a predefined threshold. In order to design the generative model we have to capture different forms of correlation between variables of the system which we describe here.

3.1.1 Between and Within Flow Correlation

An important aspect that needs to be captured in any model of network traffic is the presence of *between* and *within* correlation in packet flows. For example, consider Fig.3.1(a), which shows the the time series of two flows, $f_1(t)$ and $f_2(t)$. The point labeled D is an example where the correlation *within* flow $f_1(t)$ flows has deviated from the expected norm. Similarly, the point labeled P is where the correlation *between* the two flows f_1 and f_2 has deviated in a localized time window. The anomaly D is an example of a Denial of Service (DoS) attack while an anomaly P is an example of port scan. Discovering events like P and D is the focus of this paper.

3.1.2 The Trajectory/Hankel Matrix

A key tool that we will use to detect correlation deviation in network traffic, is the trajectory (or Hankel) matrix that will be constructed from the observed time series (see [5, 14, 15, 17, 18, 19, 77]). For example, given two flows $\{f_1(i), f_2(i)\}_{i=1}^{T}$, the Hankel matrix ($H$) of window length $L < T$ of the two flows is given by

$$\left[\begin{array}{ccc|ccc} f_1(1) & \dots & f_1(L) & f_2(1) & \dots & f_2(L) \\ f_1(2) & \dots & f_1(L+1) & f_2(2) & \dots & f_2(L+1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_1(T-L+1) & \dots & f_1(T) & f_2(T-L+1) & \dots & f_2(T) \end{array}\right]$$

Now the key insight of the paper, is that the SVD of correlation (or covariance) matrix

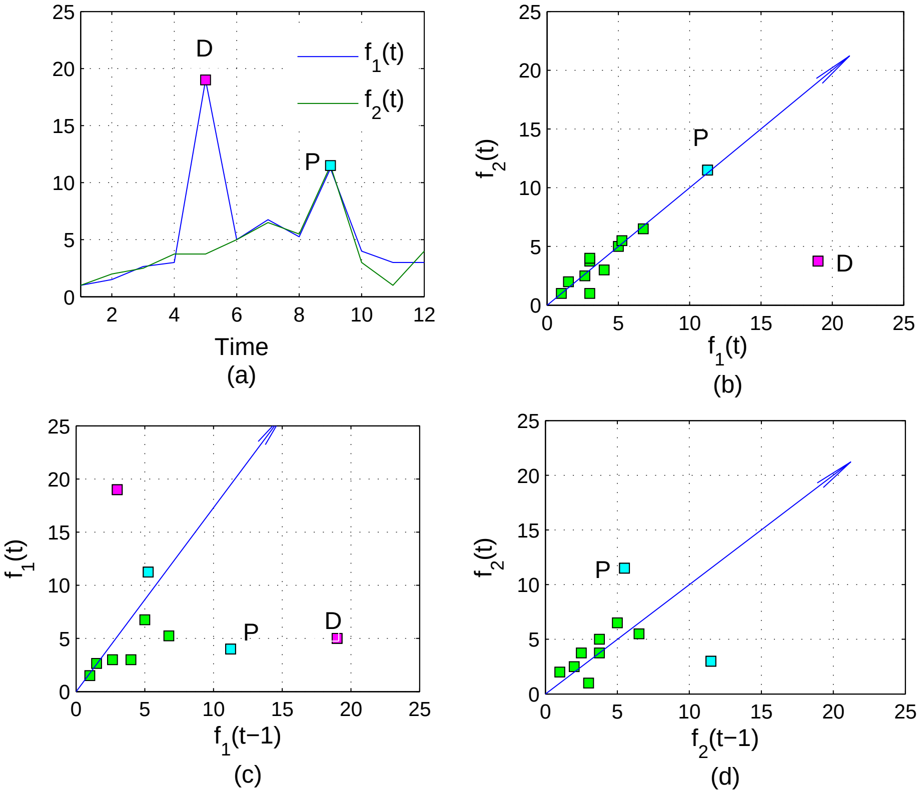


Figure 3.1: (a) An example of two flows f_1 and f_2 experiencing two different anomalies DoS attack (D) and port scan (P). (b) SVD finds D anomaly and misses P one as it is in its normal space. (c) and (d): mapping the f_1 and f_2 vector into a 2-dimensional space and applying SVD both P and D anomalies are detectable.

of the Hankel matrix (H), will capture both *between* and *within correlation* in network flows. Thus a low rank decomposition of H will characterize the manifold structure M between the flows as well as help identify the anomalies which deviate from the inferred manifold structure. For example, Fig. 3.1(b), shows the relationship between the flows f_1 and f_2 and also the direction of the most dominant eigenvector of the standard correlation matrix (without the time lag). This decomposition is unable to capture the port scan (P) anomaly because, P is not a simple violation of the between flow correlation but the existing correlation is violated only in a localized time window. In Fig. 3.1(c), it is clear that a time window lag ($L = 1$), captures the spatial correlation in a small time window and thus the P anomaly is away from the main eigenvector. In Fig. 3.1(d), there is no correlation violation within flow f_2 and thus the P anomaly is in the direction of the main eigenvector.

The remainder of this paper is structured as follows. Section 3.2 describes the important role of Hankel matrices in capturing the underlying dynamic of a system. Section 3.3 explains the algorithm behind the singular spectrum analysis. The characterization of network anomalies is abstractly presented in section 3.4. Section 3.5 presents a validation of the different analysis algorithm based on SSA on a real traffic data and analyses their capability for anomaly detection. A brief background is presented in section 3.6 and we discuss some conclusion remarks in section 3.7.

3.2 Hankel Matrix and Generative Model

We now justify the decomposition of the Hankel matrix based on a generative model of the data. In particular we show that if data is generated by a Linear Dynamical System (LDS), then the SVD decomposition of the Hankel matrix can be used to estimate the LDS parameters. Assume a LDS given by:

$$\begin{array}{ll} x(t+1) & = Ax(t) + w(t) \\ y(t) & = Cx(t) + v(t) \end{array}$$

where

- $x(t) \in \mathbb{R}^n$ is the system state vector,
- A defines the system's dynamics,
- w is the vector that captures the system error, e.g. a random vector from $\mathcal{N}(0,Q)$,
- $y(t) \in \mathbb{R}^m$ is the observation vector,
- C is the measurement function,
- v is the vector that represents the measurement error, e.g. a random vector from $\mathcal{N}(0,R)$,

Fig.4.2 presents a graphical model of LDS.

Problem 1 *Assume that some data is generated from a LDS governed by the equation above. Given a sequence of observations $\{y_i\}_{i=1}^n$, estimate A, C, Q and R.*

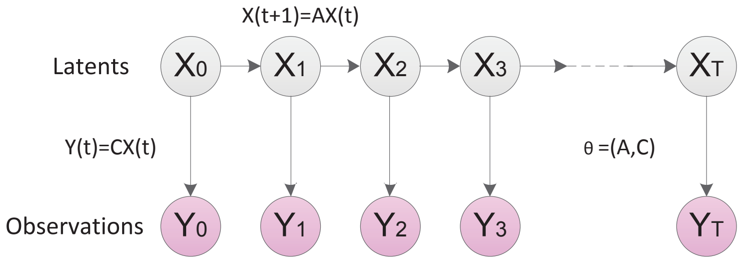


Figure 3.2: The graphical model of a linear dynamic system (LDS)

To solve the above problem, we need to define the Hankel matrix of the observations as

$$H(t)=\begin{pmatrix} y(t) & y(t+1) & y(t+2) & \dots & y(n-\ell+1) \\ y(t+1) & y(t+2) & & \ddots & \vdots \\ \vdots & & & & \\ y(t+\ell) & \dots & & & y(n) \end{pmatrix}$$

where y(t) is $m \times n$ observation at time t, and H is a $\ell \times n'$ where $n' = n - \ell + 1$. Equivalently, H is a Hankel matrix if and only if there exists a sequence $s_1, s_2, s_3, \dots$ such that $H_{i,j} = s_{i+j-1}$ (see [78]). Therefore, every Hankel matrix uniquely determines a time series and every time series can be transferred into a Hankel matrix, i.e.

$$H(t-i) \Leftrightarrow y^i(t)$$

where $y^i(t) = \{y(i), y(i+1), \dots, y(t), \dots\}$.
By replacing the entries of the Hankel matrix with their equivalent from the LDS:

$$H(1)=\begin{pmatrix} CAx(0) & CAx(1) & CAx(2) & \dots & CAx(n-\ell) \\ CAx(1) & CAx(2) & \ddots & & \vdots \\ \vdots & & & & \\ CAx(\ell-1) & \dots & & & CAx(n-1) \end{pmatrix} = \begin{pmatrix} CAx(0) & CA^2x(0) & CA^3x(0) & \dots & CA^{n-\ell+1}x(0) \\ CA^2x(0) & CA^3x(0) & \ddots & & \vdots \\ \vdots & & & & \\ CA^{\ell}x(0) & \dots & & & CA^n x(0) \end{pmatrix}$$

$$= \begin{pmatrix} CA & CA^2 & CA^3 & \dots & CA^{\ell} \end{pmatrix}^T \cdot \begin{pmatrix} x(0) & Ax(0) & A^2x(0) & \dots & A^{n-\ell-2}x(0) \end{pmatrix}$$

Define:

$$\begin{array}{l} P \doteq \begin{pmatrix} CA & CA^2 & CA^3 & ... & CA^{\ell} \end{pmatrix}^T \\ Q \doteq \begin{pmatrix} x(0) & Ax(0) & A^2x(0) & \dots & A^{n-\ell-2}x(0) \end{pmatrix} \end{array}$$

then:

$$H(1) = \quad PQ$$

The shifted Hankel matrices can be described by:

$$H(i) = \quad PA^{i-1}Q$$

To obtain the matrices A and C, perform SVD of $H(1)$:

$$H(1) \quad = U\Sigma^2 V^T$$

where Σ^2 is a diagonal $\ell \times \ell$ matrix containing the singular values and the ℓ columns of U are the singular vectors. Selecting the top-k $(1 < k < \ell)$ singular values from the matrix Σ^2,denoted by Σ_k, and k associated singular vectors, denoted by U_k, we define reduced rank matrices:

$$P_k \doteq \quad U_k\Sigma_k$$

$$Q_k \doteq \quad \Sigma_k V^T$$

Using the reduced rank matrices P_k and Q_k, the shifted Hankel matrix $\hat{H}(2)$ is estimated as:

$$\begin{aligned} \hat{H}(2) \quad &= P_k A_k Q_k \\ &= U_k \Sigma A_k \Sigma V^T \end{aligned}$$

The matrix A_k can be approximated as:

$$A_k \quad = (U_k\Sigma_k)^{+}\hat{H}(2)(\Sigma_k V^T)^{+}$$

where $^{+}$ marks the pseudoinverse of the matrices. Given A_k we can estimate C_k as:

$$C_k \quad = P_1^{-1}A_k$$

where P_1 is the first m rows of the matrix P.

Therefore, using SVD of a Hankel matrix we can estimate a generative model for any given time series. Now, given A_k and C_k (assuming $x(0)$ is known), we can estimate the error space as:

$$\begin{aligned} \delta_k(t) = & \; y(t) - \hat{y}(t) \\ = & \; y(t) - C_k\hat{x}(t) \\ = & \; y(t) - C_kA_k\hat{x}(t-1) \end{aligned}$$

and an outlier is reported whenever $|\delta_k|$ exceeds a predefined threshold.

In practice, we are able to use the decomposition of the Hankel matrix to identify outliers without any assumption for underlying dynamic, e.g. linearity. Perform the SDV of the Hankel matrix and chose $(1 < k \leq \ell)$ to divide the spectral decomposition into two disjoint spaces:

$$\begin{aligned} H(1) & = U\Sigma^2V^T \\ & = \sum_{i=1}^{k} \lambda_i^{1/2} U_i V_i' + \sum_{i=k+1}^{\ell} \lambda_i^{1/2} U_i V_i' \end{aligned}$$

Define $\hat{H}(1) \doteq \sum_{i=1}^{k} \lambda_i^{1/2} U_i V_i'$ and $\Delta_k \doteq \sum_{i=k+1}^{\ell} \lambda_i^{1/2} U_i V_i'$, then:

$$\Delta_k \quad = H(1) - \hat{H}(1)$$

Δ_k is the error between the original Hankel matrix and the estimated one $\hat{H}(1)$. Since every Hankel matrix is associated with a time series, if $\hat{H}(1)$ would be Hankel then the error space Δ_k is a Hankel and uniquely defines a time series, e.g. δ_k. This Time series represents the residual time series to be investigated for anomalies. The tool for achieving this residual is Hankelization operator, which transforms an arbitrary $\ell \times n'$ matrix to the form of a Hankel matrix. A Detailed procedure of Hankeliztion is given in Appendix B.

We showed that by the use of spectral decomposition of a Hankel matrix constructed based on a lag window of ℓ we can estimate the residual space of time series with respect to the high variations. A neat step-by-step algorithm for this approach is explained next.

3.3 Multivariate Singular Spectrum Analysis Algorithm

The application of SVD to Hankel matrix is known as SSA or M-SSA. The key advantage of M-SSA is its ability to succinctly capture both between (spatial) and within (temporal) correlation in the underlying network traffic flows. Here we give a step-by-step introduction to SSA, as a method of discovering anomalies.

1. Assume the network flow volume through a router at a pre-specified level of granularity (e.g.five minutes) is given by the time series.

   $$y_1, y_2, \ldots, y_m, w_{m+1}, w_{m+2}, \ldots, w_n, y_{n+1}, y_{n+2}, \ldots$$

   We have used both y and w to indicate that the nature of traffic has changed for $n-m+1$ time steps after y_m. In practice we of course don't know where and when the traffic changes and is precisely what we want to infer.

2. Choose an integer $\ell < m$, known as the embedding dimension and form the *Hankel matrix* for the x part of the time series.

   $$\mathbf{Y} = \begin{pmatrix} y_1 & y_2 & \cdots & y_\ell \\ y_2 & y_3 & \cdots & y_{\ell+1} \\ \cdots & \cdots & \cdots & \cdots \\ y_{m-\ell+1} & y_{m-\ell+2} & \cdots & y_m \end{pmatrix}$$

   Where each $\mathbf{Y_i} = (y_i, y_{i+1}, \ldots, y_{i+\ell})'$, is of dimension ℓ. In SSA, the assumption is that $\mathbf{Y}$ captures the main dynamics of the network flow. We now apply the Singular Value Decomposition (SVD) of $\mathbf{Y}$ as follows.

3. For the $\ell \times \ell$ covariance matrix of Y give by

   $$C = Y \times Y'$$

4. Compute the eigen-decomposition of $C = [U, D]$ where U is matrix where each column is a eigenvector and D is the diagonal matrix of eigenvalues. The relationship between C, U and D is given as

   $$CU(:,i) = D(i,i)U(:,i) \text{ for each } i$$

5. Form an k-dimensional subspace M of R^{ℓ} where $k \leq \ell$, by using the top-k eigenvectors of U, i.e., $\mathbf{M} = U_s U_s'$. The space $\mathbf{M}$ is where the ”normal” traffic lives and our objective is to look for changes in the flow which cannot be explained by $\mathbf{M}$. This is achieved by projecting a sliding window of ℓ dimensional vectors on M and raising an alarm whenever the deviation between a vector and its projection on M becomes large.

6. For example, consider a ℓ-dim vector which contains parts of the changed traffic $y_i's$.
$$\mathbf{z} = (y_{m-1}, y_m, w_1, \ldots, w_{\ell-m-1})'.$$
Then, the deviation between $\mathbf{z}$ and its projection on M is given by $\mathbf{e} = ||\mathbf{z} - \mathbf{Mz}||$. Assuming that the w_i's were generated by anomalous traffic, then the deviation $\mathbf{e}$ will be large relative to deviations caused by normal traffic.

7. To reconstruct the refined time series we proceed in a manner inverse to the step 2. On the other hand, if the objective is to reconstruct the original time series then we have to apply a Hankelization (inverse) operator. The network anomaly detection process remains unaffected by the inverse operation. More details can be found in [79, 80, 81, 82].

Before we go into further details about SSA we illustrate the key steps using a simple example.

Example 1 *Assume that a sample time series is given as*

$$\mathbf{y(t)} = \begin{cases} sin(.2t) + \varepsilon(t) & if \quad 1 \leq t \leq 175 \\ sin(.3t) + \varepsilon(t) & if \quad 176 \leq t \leq 375 \\ sin(.2t) + \varepsilon(t) & if \quad 376 \leq t \leq 560 \end{cases}$$

Here $\varepsilon(t)$ is gaussian $\mathcal{N}(0,1)$ noise. Notice that there is a change in the time series between $t = 176$ and $t = 376$. Fig. 4.1(a and b) show the example time series without the noise and the time series with added noise. Fig. 4.1(c) shows the deviation of the signal for different values of ℓ and k. It is clear that the deviation becomes larger near

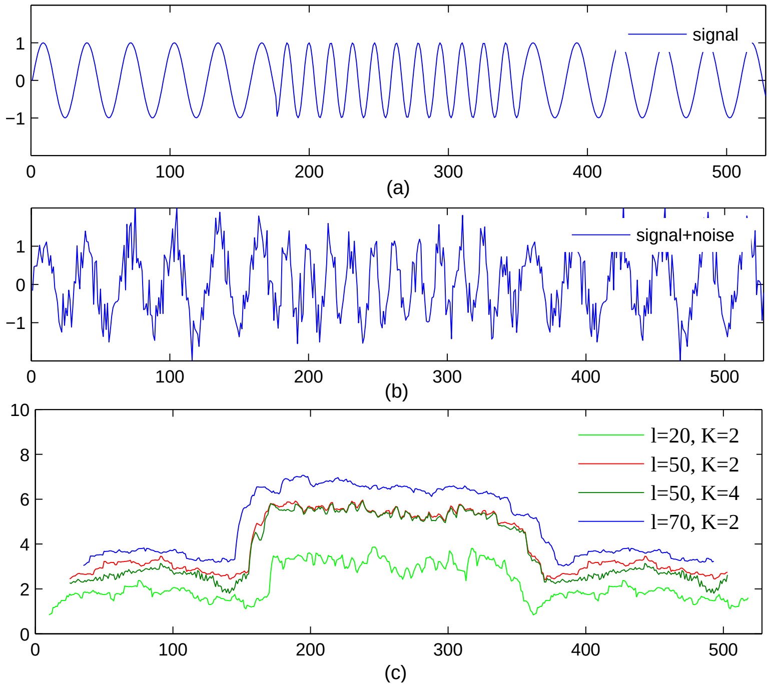


Figure 3.3: An example of using SSA to detect changes in a time series for various combination of parameter values ℓ and k. The time series changes in the middle which is reflected in the deviation in the bottom figure.

time step 176 *and then returns to its normal value after the change signal disappears around time step* 376.

3.3.1 Choice of Parameters in SSA

The key idea in SSA is the use of a trajectory matrix $\mathbf{Y}$ which then factorized using SVD. The formal relationship between Y and the underlying dynamics of the time series has been extensively researched in both the statistics and physics community. The key take away from the theoretical literature is that for an appropriate choice of ℓ, the trajectory matrix will capture the appropriate dynamics of the underlying system (see [5, 14, 15, 82]). The choice of ℓ along with k (the dimensionality of the projected subspace) and the threshold $(\mathbf{e})$ are three important parameters that need to calibrated and set. These parameters are like “knobs” which a network administrator can use to adapt to specific network characteristics.

Example 1 above already provides some indication of how the choices of ℓ and k have on time series monitoring. For example, for $\ell = 20$, the deviation $\mathbf{e}$ is less than for other values of ℓ. This may surprising at first but notice the initial part of the time series has an intrinsic dimensionality of 1 (as it is composed of one sin term). Thus a smaller value of ℓ is better at capturing the dynamics of the time series than a larger value $\ell = 50, 70$. Now consider, the two cases where $L = 50$ but $k = 2$ or $k = 4$. Notice that the projected error (in the middle) is almost identical but at the tails the projection error is higher for $k = 2$ than $k = 4$. This shows that while the choice of k has a significant impact on the projection error of the normal traffic, when it comes to detecting the anomalous part the method is quite robust for different choices of k. In fact this is one of the key strengths of SSA that we will exploit in the analysis of real network traffic data.

3.4 Network Anomaly Types

A key contribution of our paper is that the approach based on M-SSA is able to detect almost all known types of network anomalies. In this section we describe the different types of common anomalies and explain why M-SSA provides subsumes other anomaly detectors. Table 3.1 lists the common anomalies defined using the flow as a 5-tuple (source IP address, destination IP address, source port number, destination port number, transport protocol). More details can be found in [8, 9, 32, 48].

A **Denial of Service (DoS)** attack occurs when the attacking hosts send a large number of small packets - typically TCP SYN segments - to the attacked host and service, i.e. a single IP address and port number, in order to deplete the system resources in the target host. The resulting traffic from DoS attack consists of a relatively small number of flows with large packet counts as DoS attack tools often forge the source port number. Note that the specific case of Distributed Denial of Service (DDoS) attacks is effectively the same attack, but with several source IP addresses. The number of attacking hosts however, is typically much smaller than the packet count. We thus consider DDoS to be a special case of a DoS attack, and label as such.

Table 3.1: Network anomalies considered

Anomalies	Description (flow is defined as one 5-tuple)
DoS attack	a few flows with a large increase in packet count
port scan	large increase in number of flows with a small packet count
large file transfer	a few flows with a large increase in packet count, (but typically less than DoS attack)
prefix outages	drop in number of flows (from one IP prefix)
link outages	time intervals where all traffic disappear.

Port scans are typically used by attackers to discover open ports on the target host. This is accomplished by sending small packets as connections requests to a large number of different ports on a single destination IP address. At the flow level, they are therefore characterized as an increase in the number of flows, each with a small packet count.

Large file transfers are characterized by a few flows with packet counts which are significantly larger than what common applications use.

Prefix outages occurs when part of the network becomes unreachable, they can be identified when traffic from one or more IP prefixes disappears, which translates in a drop in the number of flows.

Link outage is in a way a more severe version of Prefix outage, where the number of flows on the link drop close to zero.

3.5 Experimental evaluation

We have evaluated our proposed approach using both real and synthetic data sets. For comparison we have implemented well known network anomaly techniques based on wavelets, kalman filtering, fourier analysis and the more recent ASTUTE method. The

Table 3.2: Alternative methods used in the experiments

Techniques are implemented by adjusting parameters as proposed in the literature.
Fourier analysis [13] We use fast Fourier transform (FFT) algorithm and set the cut off frequency to one cycle per 2 hours.
Wavelet analysis [6, 13] We use a multi-level, 1-dimension wavelet algorithm, with Daubechies mother wavelet of order 6 and set the cut off frequency to 3.
Kalman Filter [12] The target false positive rate of 2×10^{-5} is applied to the innovation process.
ASTUTE [8, 9] The target false positive rate of 2×10^{-5} is applied to the *AAV* process.

use of synthetic data sets and simulation is a prerequisite for a rigorous evaluation strategy for network anomaly detection ([8, 12, 83]).

3.5.1 Detection Capability

We evaluate the detection capability of M-SSA using two real network traces which we now describe.

3.5.1.1 Datasets

The first traffic trace if from the Abilene network[1] and has been used previously for network anomaly detection (see [8, 9, 32, 48]). The data set consists of a one month traffic trace from a backbone router in New York during August 2007. The Juniper router used to collect the data generated sampled J-flow statistics at the rate of 1/100. The flows were aggregated at five minute intervals. The key attributes of the flow are: number of packets, number of distinct source IP addresses, number of distinct destination IP addresses, number of distinct source port numbers and number of distinct

[1] Internet2 - http://www.internet2.edu/

destination ports numbers.

The second, and more recent, traffic trace is from the MAWI (Measurement and Analysis on the WIDE Internet) archive project in Japan[1]. Here the data was sampled from a 150Mbps trans-pacific link between Japan and the United States for 63-hours in April 2012.

Labelling traffic traces with anomalies is notoriously difficult. The commonly accepted method is to combine algorithmic detection with manual inspection of the data. We have followed the URCA (*Unsupervised Root Cause Analysis*) method proposed by [49] with a false positive rate of 2×10^{-9}, followed by a thorough manual inspection of the data set.

3.5.1.2 Results

Table 3.3 and Fig. 3.9 show the results of the different methods including M-SSA. The following are the key take aways.

1. M-SSA is capable of detecting a much wider range of anomalies regardless of their types. For the Abilene data, M-SSA was able to identify 100% of DoS attacks and over 95% port scans. Similarly on the MAWI data set the detection rate was 100% for DoS attacks and over 90% for port scans.

2. All other techniques (which were compared) can be placed in two groups: Wavelets, Kalman and Fourier have high detection rates only for DoS attacks while ASTUTE performs exceedingly well only for port scan anomalies.

3. In the Abilene data, around 7% of the anomalies are related to link outages. Here again, M-SSA has a 100% detection rate and except for Fourier, other techniques also have a high detection rate with Wavelets doing the best.

To understand the results better we have carried out a deeper analysis by examining the characteristic features of the anomalies. In Fig. 4.11 we plot the known Abilene anomalies using two features. The x-axis represents the change in packet counts between two consecutive time bins. The y-axis represents the number of distinct flows (5 tuples) in the time bin.

[1] http://www.wide.ad.jp/project/wg/mawi.html

Table 3.3: Number of anomalies per type found by each technique in two traffic traces from Abilene and WIDE networks. M-SSA is able to discover both DoS and port scan in both networks.

Trace: Internet2, from Abilene backbone

Period: August 2007

		ASTUTE	Kalman	Wavelet	Fourier	**M-SSA**	Hybrid[†]
Anomalies class	Labeled						
DoS attacks	44	1	37	41	17	**44**	44
port scans	221	198	0	18	0	**211**	216
large-file transfer	2	2	0	0	0	**2**	2
link outage	18	12	12	17	6	**18**	18
prefix outage	1	1	0	0	0	**1**	1
Total found	276	214	51	76		**265**	271

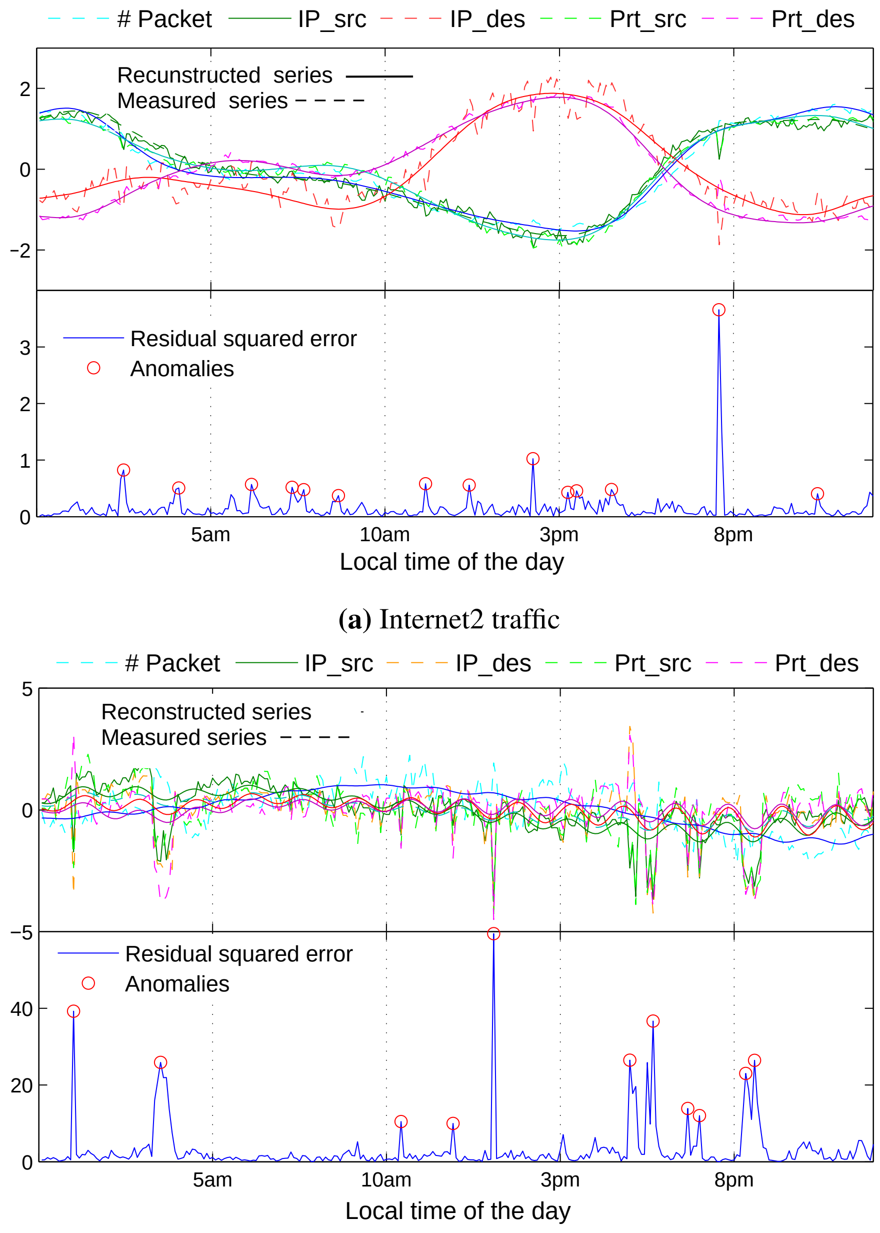
Trace: MAWI, from WIDE backbone

Period: April 2012[‡]

		ASTUTE	Kalman	Wavelet	Fourier	**M-SSA**	Hybrid[†]
Anomalies class	Labeled						
DoS attacks	9	1	7	8	4	**9**	9
port scans	98	89	11	19	0	**89**	89
large-file transfer	1	1	1	0	0	**1**	1
link outage	2	2	1	1	0	**2**	2
Total found	111	93	20	28	4	**101**	101

[†] Hybrid refers to ASTUTE ∪ Kalman ∪ Wavelet

[‡] This a 63-hours trace in the early days of the month.



(a) Internet2 traffic

(b) MAWI traffic

Figure 3.4: Timeseries plots of measured and reconstructed data along with related residual vector squared magnitude; for one day of both traffic traces from Abilene and WIDE networks. Triggered alarms shown as red circles.

The first observation is that the set of anomalies are clustered in distinct groups, with the set of anomalies detected by Wavelet and Kalman approximately common (Wavelet is slightly better in detecting some port scans). Secondly, the Kalman filter and Wavelet techniques are not able to find anomalies caused by large number of flows with small packet counts. These includes anomalies where the rate of change in packet count in individual flows over time is small, e.g. port scans, prefix outages and file transfers. Wavelet as a time-frequency technique is able to flag sudden changes in traffic, but will miss any small variations such as port scans and absorb them in the main trend.

The Kalman filter technique is effective at detecting anomalies when the packet count variation over time is significant, such as DoS attacks. This is expected, as Kalman Filtering is essentially a forecasting technique in the time dimension. Another observation is that ASTUTE is not able to detect anomalies involving a few large flows (bottom right hand corner of Fig. 4.11), such as DoS attacks. This is also expected, as ASTUTE is not able to detect large volume change in a few number of flows, because the *AAV* process threshold is not violated (as the denominator of *AAV* is the standard deviation which will be large) as mentioned by [9, 49].

The results and analysis clearly suggest, as has been noted before by [8], that a hybrid approach consisting of ASTUTE and Kalman (or Wavelet) will capture most of the anomalies. Importantly, Fig. 4.11 shows that the proposed M-SSA based approach is able to detect anomalies regardless of their location on the feature properties map. M-SSA is able to detect significant temporal changes in traffic as well as changes in the number of flows. M-SSA searches for correlation across flows properties (ASTUTE applies the same search concept between flows), while at the same time looking for temporal variation in a lag window dimension of ℓ. ASTUTE is limited to two consecutive time bins.

3.5.2 Detection Performance

In order to evaluate the robustness and sensitivity of M-SSA we have designed a simulation set up where we inject artificial anomalies in real traces and measure the trade-off between the true positive and false positives using ROC curves. One of the biggest challenges in network anomaly detection systems, and which has limited their

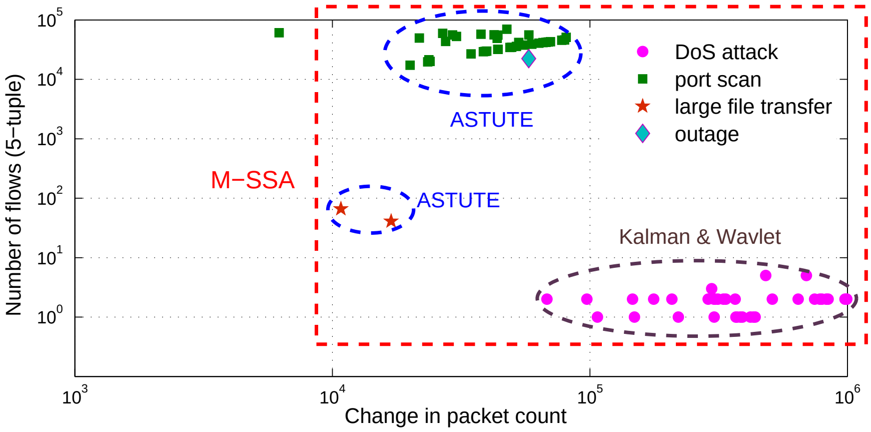


Figure 3.5: Anomalies feature map shows DoS attacks are associated with a small number of flows with large number of packets, while port scans are a larger number of flows correlated in same time. The coverage of M-SSA subsumes all the techniques.

widespread adoption, is the high false positive rate exhibited by most existing techniques (see [83, 84]).

3.5.2.1 Simulation

Our simulation is based on real trace data augmented with anomalous traffic injected in a similar fashion as in [8, 83, 84]. However and in addition to previous work, we build a simulation model which captures several distinctive characteristics of anomalies. We consider the distribution of time between anomalies, duration, magnitude (packet count for DoS attacks, number of flows for port scans, etc), and the anomaly type distribution (DoS, port scan, etc).

We first estimate the above parameters based on available observations in traffic traces. For example Fig. 3.6 and Fig. 3.7 show the histograms of these property values for DoS attacks and port scans respectively, as observed in the Abilene trace. We start the simulation assuming a non-anomalous time bin and choose the next attack time, by sampling from the empirical probability distribution of the time between anomalies. The anomaly type is then also chosen by sampling from the anomaly type distribution. At this point, a synthetic anomaly is generated by sampling from the anomaly duration and magnitude distribution, and injected into the synthetic trace. This process is repeated until the end of the simulation. The resulting trace therefore inherit the most

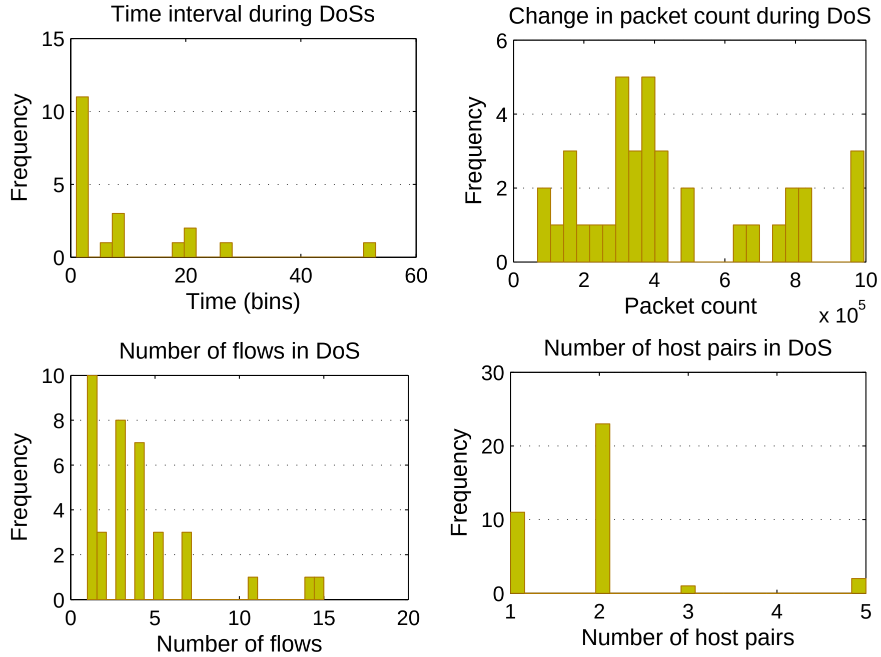


Figure 3.6: Illustration of the distribution histograms used to simulate DoS attacks. Distribution histograms characterize the duration of attacks and size of attack (e.g. number of flows involved in the attack plus the change in the packet volume)

significant statistical properties of the real data, e.g. the frequency of attacks and their magnitude.

3.5.2.2 Results

The trade-off between false positive and true positive rate using the simulation data are captured using the ROC curve and are shown in Fig. 3.8. The simulation parameters for all algorithms are set as per Table 3.2. The ROC curves depicted in Fig. 3.8 show that M-SSA has higher true positive rate for a given false positive rate, compared with all other techniques. For example, for a false positive rate of 0.01%, M-SSA detects 90% of anomalies, whereas Wavelet and ASTUTE only detect 77% and 81% respectively. A Hybrid detector including Wavelets, Kalman and ASTUTE shows slightly better trade-off for a false positive rate less than 10^{-5} but M-SSA is better for the rest of interval. The Area Under Curve (AUC) which measures the overall performance of the detector has been shown in Fig. 3.8 (left).

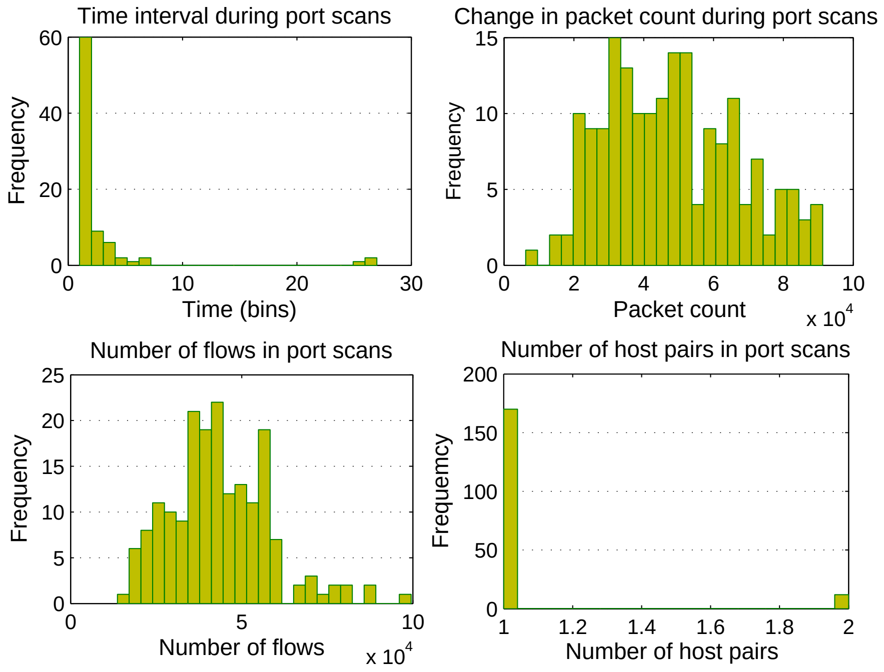


Figure 3.7: Illustration of the distribution histograms used to simulate port scans. Distribution histograms characterize the duration of attacks and size of attack (e.g. number of flows involved in the attack plus the change in the packet volume)

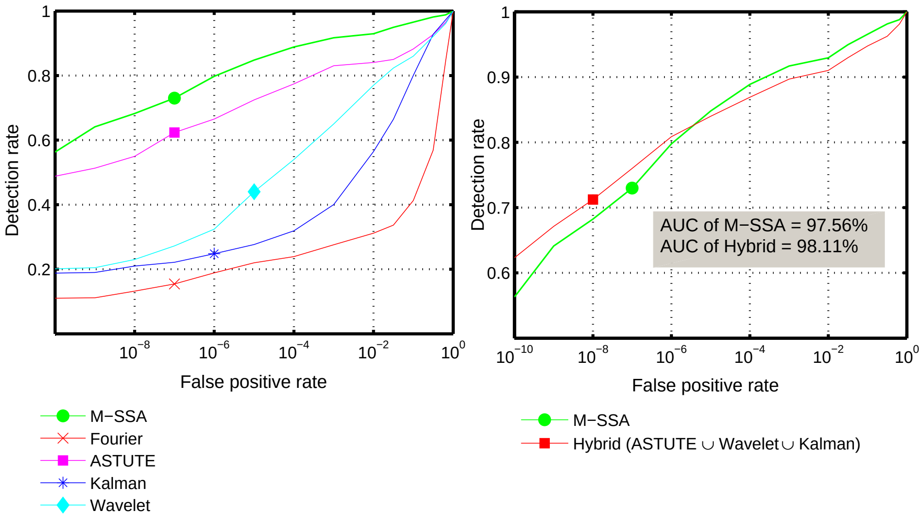


Figure 3.8: ROC curves: M-SSA has a better detection rate than alternative techniques. A Hybrid system shows slightly better trade-off for a false positive rate less than 10^{-5}.

3.5.3 Configuration of Parameters

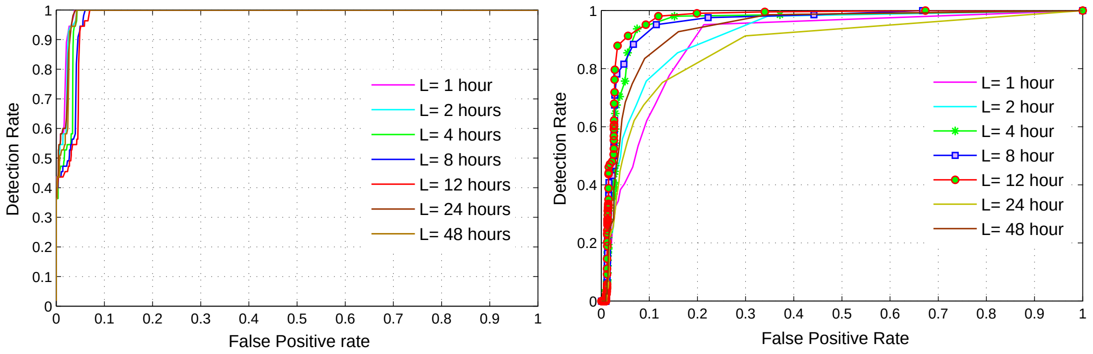
We now evaluate the impact of the parameters: Lag Window Length (ℓ), the dimensionality k of the projected space and the detection threshold q_β.

3.5.3.1 Lag window length (ℓ)

The key concept to take away from the theoretical literature is that for an appropriate choice of ℓ, the Hankel matrix will capture the appropriate dynamics of the underlying system (see [5]. According to [5, 14, 15] and [80], in choosing ℓ one must consider the trade-off between the maximum period (frequency) resolved and the statistical confidence of the result. A large value of ℓ will potentially better capture the long range trends but the size of the covariance matrix will be larger which will have to be estimated from a time series of effective length $n-\ell+1$.

The choice of ℓ has a significant impact on detection performance of different anomalies. DoS attacks and port scans are emblematic of two types of deviations in network traffic. DoS attacks are characterized by large changes in a (relatively) small number of flows as the attacking hosts send a large number of small packets to deplete system resources in the attacked host (see Fig. 3.6 and Fig. 4.11). Thus DoS like anomalies cause high temporal variation (within flows correlation) in the responsible flows and can be detected using techniques based on time series analysis. Port scans, on the other hand, are characterized by small increases in a large number of flows (see Fig. 3.7 and Fig. 4.11). Thus, we need to detect spatial correlation across flows (i.e. correlation between the flows) in order to find port scans. We run an experiment to test the impact of window length on capturing within and between flows correlation of the traffic data. ROC curves in Fig. 3.9a and Fig. 3.9b present DoS and port scan detection performance (separately) for varying window length. We describe the main findings learned from this experiment as follows.

- It is clear that the detection of DoS is almost independent of the window length (Fig. 3.9a). This is expected as DoS attacks cause high correlation within flows (temporal variations) and this can be always captured even if the window length



(a) DoS detection performance for different window length.

(b) Port scan detection performance for different window length.

Figure 3.9: The impact of window length ℓ on detecting DoS attacks and port scans. Notice that ℓ has almost no impact on on DoS detection but significant impact on port scan detection.

is zero, i.e. even the common PCA is able to report DoS attacks.

- Across flows correlation is crucially dependent on window length as shown in Fig. 3.9b. Thus the choice of window size has significant impact on detection of port scans. When the window length is zero the correlation across the flows can not be captured. When the window length is large across flows correlation is suppressed. What is required is a localized window where deviation from normal correlation can be detected. According to the experiment, detecting port scans is improved for window lengths of $\ell = \{4, 8, 12\}$ (hours) and worsened for smaller or larger window lengths.

3.5.3.2 Grouping indices (k)

Another important parameter of M-SSA affecting results is the grouping indices, i.e. which components are grouped to provide the reconstructed data. The aim of our technique is to make a decomposition of the observed traffic into the sum of underlying traffic system (can be a number of interpretable components such as a slowly varying trend, oscillatory components) and a structureless noise, as $Y = X + E$. The decomposition of the series Y into these two part is viable if the resulting additive components X

and E are approximately separable from each other. Suppose the the full reconstructed components are denoted by $V_i = Mz$ for $i = \{1, 2, ..., m \times \ell\}$. To select which components to group, we compute the weighted correlation matrix (w-corr), where each element of the matrix ρ_{ij} is defined as:

$$\rho_{ij} = \frac{cov_w(V_i, V_j)}{\sigma_w(V_i)\sigma_w(V_j)}$$

using:

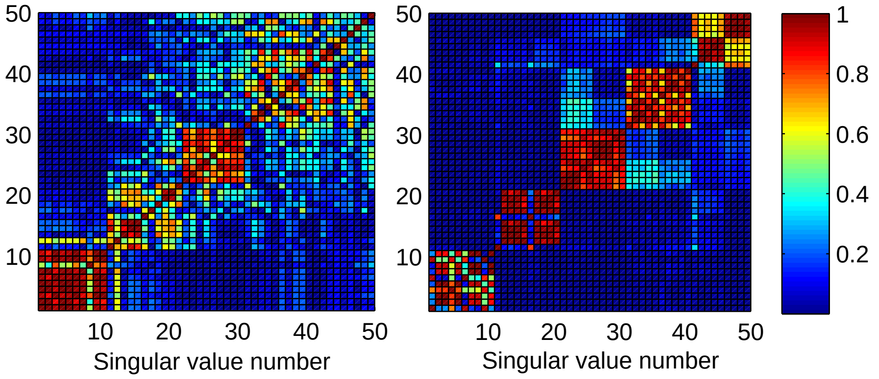
$$\sigma_w^2(V_i) = W'V_i^{'}V_i \quad , \quad cov_w(V_i, V_j) = W'V_i'V_j$$

where $w_t = min\{t, \ell, n-\ell\}$ for $t = \{1 : n\}$ is the weighting vector. If the absolute value of the w-correlations for two V_i and V_j is small (ideally zero), so the corresponding series are almost w-orthogonal and well separable. Fig. 3.10 shows the absolute values of w-correlation for the first 50 reconstructed components. This is a grade matrix plot from red (corresponding to 1) to blue (corresponding to 0), which shows both the separability and dominance of components with highest eigenvalues values. This plot is useful to select how many components to select in the reconstruction phase, as we only need to select the first k components with the largest w-corr values. From Fig. 3.10, we observe that the absolute value of the w-correlation for first 10 components are naturally grouped, a property that is observed for both the Abilene and MAWI data sets. We therefore suggest to use the first 10 components for the reconstruction when using M-SSA. So the $X = \sum_{i=1}^{i=10} V_i$ and residual space $E = Y - X$. In next section we will see that how the values of w-correlation can also be checked for adjusting the decision parameter (q_β) so that a false positive rate can be met.

3.5.3.3 Decision Variable (q_β)

For the decision threshold value (i.e., when to raise an alarm for any anomaly investigating E space), we use the variables proposed in previous studies (see [47, 70, 72])in network anomaly detection but we address the problem associated with this criteria as discussed by [58]. The threshold q_β is defined as

$$\begin{aligned} q_\beta &= Q(\lambda_{k+1} : \lambda_{\ell \times m}, \beta) \\ &= \phi_1 [\tfrac{(1-\beta)\sqrt{(2\phi_2 h^2)}}{\phi_1} + 1 + \tfrac{\phi_2 h(h-1)}{\phi_1^2}]^{1/h} \end{aligned}$$



(a) Internet2 (b) MAWI

Figure 3.10: Absolute values of w-correlation matrix plotted for the first 50 reconstructed components. The gaps in the scatter plot indicates how many components to select.

denotes the threshold for the $1-\beta$ confidence level, corresponds to a false alarm rate of β, and

$$h = 1 - \frac{2\phi_1\phi_3}{3\phi_2^2}, \quad \phi_i = \sum_{j=k+1}^{\ell m} \lambda_i \quad for \quad i = 1,2,3.$$

Based on [72] the Q in the above equation follows a gaussian distribution, and this convergence is robust even when the original data deviates from a gaussian distribution. [58] had questioned the robustness of the Q metric - especially in the low false positive regime. [85] have shown that the main reason the metric is not robust is because the use of standard PCA results in a residual which exhibits temporal correlation. In principle the residual should correspond to noise and be completely uncorrelated. Thus by ensuring that temporal correlation (in the case of KL transform) and spatio-temporal correlation (in the case of M-SSA) is captured by the model, the Q metric is robust.

The w-correlation matrix computed above can help verify if the residual space, given by $E = Y - X$ where X is the reconstructed space, contains correlated elements or not. For example, the w-correlation plot in Fig. 3.10 clearly shows that that when X is the space spanned by V_i for $i > 10$, the reconstructed elements are strongly w-orthogonal in both Abilene and MAWI traffic, resulting in uncorrelated residuals.

3.6 Related Work

Current network infrastructure is protected against malicious attacks by signature-based Intrusion Detection Systems (IDS) ([86, 87]). However, it is well known that attackers can circumvent these systems by generating small modifications of known signatures.

In principle, anomaly-based detection systems (ADS) offer an attractive alterative to signature-based systems. ADS are based on the notion of ””statistical normality”, and malicious events are those that cause deviations from normal behavior. The major challenge is to characterize normal traffic subject to the constraint that network traffic exhibits non-stationary behavior.

Existing techniques for ADS are based on decomposition methods of network time series. For example [32, 47, 48] has proposed the use of Principal Component Analysis (PCA) for detection of network wide anomalies. [13] has compared the use of Fourier, Wavelets and ARIMA methods for detection of link anomalies and then have used ℓ_1 optimization to recover the origin-destination pairs which may have caused the link anomalies to appear. Further refinements on PCA and state methods like Kalman Filtering have been extensively investigated for first extracting the normal behavior and then reporting deviations from normality as potential anomalies (see [6, 7, 10, 11, 12, 13]).

The mathematical basis of Singular Spectrum Analysis (SSA) is the celebrated result in nonlinear dynamics due to [5]. Taken’s theorem asserts that the latent non-linear dynamics governing can be recovered using a delayed time embedding of the observable time series. The first practical use of Taken’s theorem for time series analysis and the connection with spectral methods like singular value decomposition (SVD) was first proposed by [14, 15]. Further application of the technique in climate and geophysical time series analysis has been extensively investigated in [79, 80, 81, 88, 89, 90, 91, 92].

3.7 Summary

In this chapter we have proposed a unified and robust method for network anomaly detection based on Multivariate Singular Spectrum Analysis (M-SSA). As M-SSA can detect deviations from both spatial and temporal correlation present in the data, it allows for the detection of both DoS and port scan attacks. A DoS attack is an example of temporal deviation while a port scan attack violates spatial correlation. Besides the use of M-SSA for network anomaly detection, we have carried out a comprehensive evaluation and compared M-SSA with other approaches based on wavelets, fourier analysis, kalman filtering and the recently introduced ASTUTE method. We have also carried out a rigorous analysis of the parameter configurations that accompany the use of M-SSA and address some of the important issues that have been raised in the networks community. Finally we have introduced a new labeled dataset from a large backbone link between Japan and the United States.

4

OSAD: *Online Selective Anomaly Detection*

IN this chapter, we introduce a new computational problem, the ”Online Selective Anomaly Detection” (OSAD) problem, to model the situation where the objective is to report new anomalies in the system and suppress known faults.

In order to solve OSAD we first have to model the underlying system and learn its parameters using techniques from machine learning. In order to selectively report anomalies we have to design a residual system which can suppress certain forms of behavior of the underlying system. We use control theoretic ideas to accomplish the design of the residual system. Experiments on synthetic and real data sets confirm that the OSAD problem captures a general scenario and tightly integrates machine learning and control theory to solve a practical problem.

The rest of the chapter is as follows. Section 4.1 provides the motivations for the problem posed. In Section 4.2, we rigorously define the OSAD problem. In Section 4.3 we present our methodology to infer the parameters of the LDS and use control theory to design a new residual system. In Section 4.4, we apply our approach to real sleep data and evaluate our results. We overview related work in Section 4.6.

4.1 Motivations for OSAD

The OSAD problem arose while analyzing network traffic data and sleep EEG data.

Sleep EEG data two significant anomalies are Sleep Spindle (SS) and K-Complexes (KC). Around 100 sleep spindles will occur during the course of a night. The number of K-Complexes is much fewer. For some experiments scientists are interested in identifying both sleep spindles and K-Complexes but only want to be notified with an alert when a non-spindle anomaly occurs (for example K-Complexes).

Another application of OSAD can be found in network traffic data analysis. Network traffic consists of a flow of packets between nodes in the system. Thus over time, we can assume that the flow settles into an equilibrium pattern. Periodically, there are perturbations in the equilibrium due to various network conditions including malicious attacks like Denial of Service (DoS) and Port Scans. If we model the flow as a time series and learn the model, then in many cases, attacks show up as deviations (residual) between the learnt model and the observations. Often the interest is to identify new anomalies while suppressing known deviations or faults in the system. For example, in some networks it may be more important to identify Port Scans than DoS as the former are harbingers of future attacks.

4.1.1 Human Sleep EEG

Research in human sleep condition has emerged as a rapidly growing area within medicine, biology and physics. A defining aspect of sleep research is the large amount of data that is generated in a typical sleep experiment.

A sleep experiment consists of a human subject, in a state of sleep, whose neural activity is being recorded with Electroencephalography (EEG) [2, 3]. A typical full night EEG time-series, recorded between 4-64 locations on the scalp, at 200 Hz, for eight hours, will generate approximately 300MB of data. A typical clinical study will have between ten and fifty subjects. Surprisingly vast majority of sleep clinics still use a manual process to analyze the recorded EEG time-series. Hence there is considerable interest in automating the analysis of EEG generated from sleep experiments.

Scientists have segmented sleep into several stages based on the responsiveness of the subject and other physiological features. Of particular important is what is termed as stage 2 (moderately deep sleep). This stage is characterized by two phenomenon that occur in the EEG time series. These are *sleep spindles*, which are transient bursts of neural activity with a characteristic frequency of 12–14 Hz, and *K-Complexes*, which

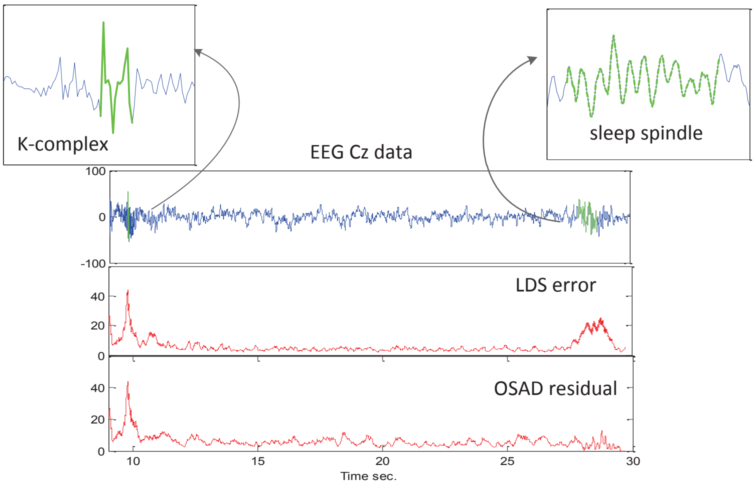


Figure 4.1: Sleep spindles (SS) along with K-Complexes (KC) are defining characteristics of stage 2 sleep. Both SS and KC will show up as residuals in an LDS system. The OSAD problem will lead to a *new* residual time-series where SS will be automatically supressed but KC will remain unaffected. Due to relatively high frequency of SS, there are certain situations where sleep scientists only want to be alerted when a non-SS anomaly occurs

are short, large-amplitude voltage spikes. Both phenomena are implicated in memory consolidation and learning, but the physiology and mechanisms by which they occur are not yet fully understood, see [2, 3, 93, 94].

In order to study these phenomena, they anomalies must be first located and identified in the EEG data. This can be challenging because they occur for an extremely short duration and irregularly. For example, sleep spindles and K-Complexes typically last less than 1sec and there are only on the order of 100 of these events over the course of an entire night. Identification of these events is further complicated by the presence of artifacts in the data, often caused by movement of the subject, but which can also occur due to electrical noise or loose electrodes connections. These artifacts must be ignored when attempting to identify sleep spindles and K-Complexes. Because the electric fields produced by the brain are quite weak (the induced electrical potential is on the order of 50 μV), the signals also contain a significant noise component.

In this paper we introduce the Online Selective Anomaly Detection (OSAD) problem which captures a particular scenario in sleep research. As noted above, around 100 sleep spindles will occur during the course of a night. The number of K-Complexes is much fewer. For some experiments scientists are interested in identifying both sleep spindles and K-Complexes but only want to be notified with an alert when a non-spindle anomaly occurs (for example K-Complexes).

The solution of the OSAD problem combines techniques form both data mining and control theory. Data Mining is used to model and infer the normal EEG pattern per subject. Experiments have shown that model parameters do not transfer accurately across to other subjects. In our case we will use a Linear Dynamical System (LDS) to model the EEG time series. Then based on frequency analysis, we infer the sleep spindle (SS) pattern and integrate the pattern as a disturbance into the LDS. The control theory part is used to *design* a new residual which suppresses SS signals but faithfully represents other errors generated by the LDS model. Thus by selectively suppressing SS pattern, the objectives of the OSAD problem are achieved.

For example, consider Figure 4.1. The top frame shows a typical EEG time series with both the SS and KC highlighted. The middle frame shows a typical residual time series based on an LDS model. The bottom frame shows a new residual designed to solve the OSAD problem. Notice that the error due to the presence of SS is suppressed but the residual due to the appearance of KC remains unaffected.

The main contributions of this chapter are:

- We introduce the Online Selective Anomaly Detection(OSAD) to address the requirement of selectively reporting sleep anomalies based on specifications by domain experts.

- In order to solve OSAD, we combine techniques from data mining and control theory. In particular we will use a Linear Dynamic System (LDS) to model the underlying data generating process and use control theory techniques to design an appropriate residual system.

4.2 Problem Definition

In this section we present our problem statement for selective anomaly detection.

The starting point is an observed time series of N points $y = \{y_i\}_{i=1}^{N}$ where each $y_i \in \mathbb{R}^m$. Furthermore, we assume that the y measures the output of a system which is generated from a latent variable $x \in \mathbb{R}^n$. The relationship between x and y is governed by a standard Linear Dynamic system (LDS) model [95] which is specified as

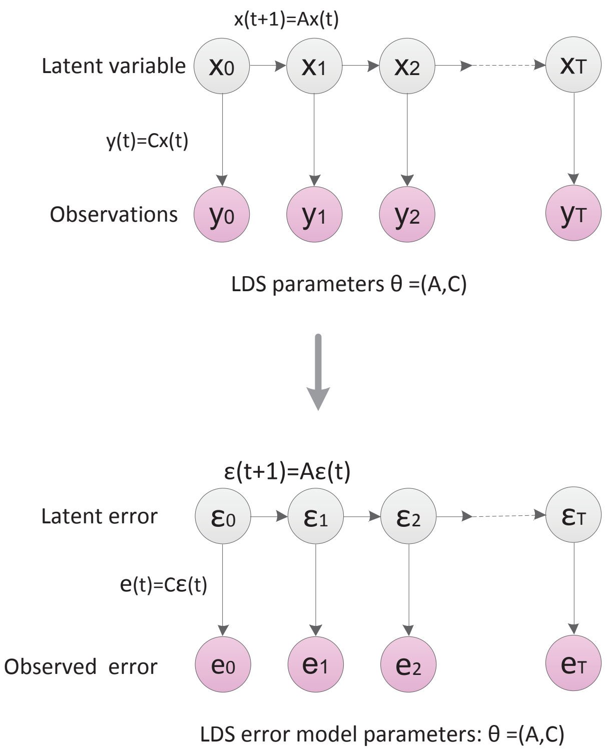


Figure 4.2: A linear dynamic system is a model which defines a linear relationship between the latent (or hidden) state of the model and observed outputs. The LDS parameters **A** and **C** need to be estimated from data. The LDS can also be used to model the relationship between the latent and the observed residuals (right figure).

$$\begin{aligned} x(t+1) = &\ \mathbf{A}x(t) \\ y(t) = &\ \mathbf{C}x(t) \end{aligned}$$

Here **A** is an $n \times n$ state matrix which governs the dynamics of the LDS while **C** is an $m \times n$ observation matrix. The modern convention is to represent the LDS as graphical model as shown in Figure 4.2. The state of the system, x, evolves according to LDS beginning at time $t = 0$, with value x_0. The standard learning problem is as follows.

Problem 2 (Learning Problem) *Given an observable time series $\{y_i\}_{i=1}^{N}$ and assuming that the observed y and the latent x are governed by an LDS, infer* **A** *and* **C**.

The standard LDS inference problem has been extensively studied in both the machine learning and control theory literature. Several algorithms have been proposed including those based on gradient descent, expectation maximization, subspace identification and spectral approaches [16, 17, 18, 19]. Several extensions of LDS to include non-linear relationships as well as to include stochastic disturbances have been proposed. However, for sleep analysis, the above LDS will suffice. For the sake of completeness, in the Appendix we will describe a simple but effective approach for inferring **A** and **C** based on a spectral method [19].

The standard approach to detect outliers using an LDS is to use the inferred **A** and **C** matrices to compute the latent and observed error variables as:

$$\begin{aligned} \varepsilon(t) :=& \; x(t) - \hat{x}(t) \\ e(t) :=& \; y(t) - \hat{y}(t) \end{aligned}$$

where $\hat{x}$ and $\hat{y}$ are estimated using LDS. Then given a threshold parameter δ, an anomaly is reported whenever, $e(t) > \delta$. However, our objective is not to report all anomalies but suppress some known user-defined patterns or even known anomalous pattern. We now formalize the notion of pattern.

Definition 1 *A* **pattern P** *is a user-defined matrix which operates in the latent space.*

In our context, we will design a specific matrix **P** for a sleep spindle. The matrix **P** is integrated into the LDS as

$$\begin{aligned} x(t+1) =& \; \mathbf{A}x(t) + \mathbf{P}\zeta(t) \\ y(t) =& \; \mathbf{C}x(t) \end{aligned}$$

We are now ready to define the design part of the OSAD problem.

Problem 3 (Design Problem) *Given an LDS, a pattern* **P** *in the latent space, design a residual $r(t)$ such that*

$$r(t) = \begin{cases} 0 & \textit{if}\, \varepsilon(t) = \mathbf{P}\zeta(t) \\ \mathbf{S}e(t) & \textit{otherwise} \end{cases}$$

Here **S** is suitably defined linear transformation on $e(t)$. Notice that the residual $r(t)$ depends both on the latent error $\varepsilon(t)$ and the observed error $e(t)$. In practice, $r(t)$ will never be exactly zero when the pattern **P** is active but will have small absolute values.

4.3 The OSAD Method

In this section we propose a method based on statistical inference and control theory to provide a solution of the OSAD problem. Using the LDS, we first develop a `Dynamic Residue Model` (DRM). Then we will show how to adjust the DRM parameters in order to design a residual $r(t)$ which will satisfy the constraints of the problem, i.e. the selected anomalous pattern will be canceled (or projected out) in the generated residual space.

4.3.1 DRM Formulation

Assume data is generated by an LDS. Any deviation of the state from its expected value can be captured by a structured error model. Intuitively, the discrepancy between the observed error $e(t)$ and latent error $\varepsilon(t)$ is modeled by the same LDS (because of linearity):

$$\begin{aligned} \varepsilon(t+1) = &\ \mathbf{A}\varepsilon(t) + \mathbf{P}\xi(t) \\ e(t) = &\ \mathbf{C}\varepsilon(t) \end{aligned}$$

The above error model can be used to detect changes occurring in the latent space.

We design a feedback loop (as shown in Figure 4.3) to effect the output of the error model. In particular a function of the residual will be used to manipulate the changes in the error. The design objective will be to map the anomalies generated by the **P** pattern into the null space of the new residual. The DRM based on this feedback design is developed as follows:

To design the feedback we define two transformation matrices **W** and **F** for error values to be weighted as:

$$\begin{aligned} r(t) := &\ \mathbf{W}e(t) \\ u(t) := &\ \mathbf{F}e(t) \end{aligned}$$

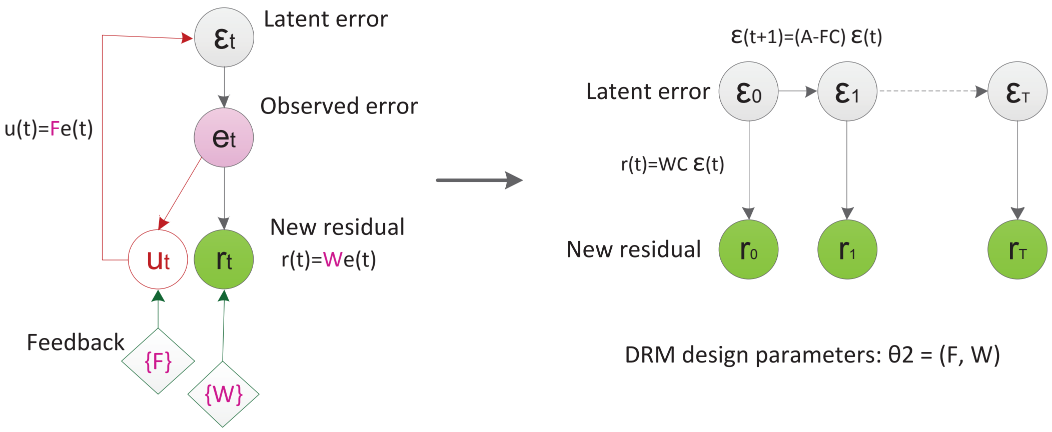


Figure 4.3: Using parameter **F** a virtual input $u(t)$ is generated to feed the error back to the latent space. The error $e(t)$ is is then calibrated by **W** to generate a new residual space $r(t)$.

F will be used as the feedback gain matrix and maps the error to the feedback vector $u(t)$, and **W** is the residual weighting matrix that generates the new residual $r(t)$. Now feeding back $u(t)$ into the LDS (as shown in Figure 4.2), with $u(t)$, the residual dynamic model will be:

$$\begin{aligned}\hat{x}(t+1) = &\ \mathbf{A}\hat{x}(t)+u(t)\\ = &\ \mathbf{A}\hat{x}(t)+\mathbf{F}e(t)\\ = &\ \mathbf{A}\hat{x}(t)+\mathbf{F}(y(t)-\hat{y}(t))\\ = &\ \mathbf{A}\hat{x}(t)+\mathbf{F}(\mathbf{C}x(t)-\mathbf{C}\hat{x}(t))\\ = &\ \mathbf{A}\hat{x}(t)+\mathbf{FC}x(t)-\mathbf{FC}\hat{x}(t)\\ = &\ (\mathbf{A}-\mathbf{FC})\hat{x}(t)+\mathbf{FC}x(t)\\ = &\ (\mathbf{A}-\mathbf{FC})\hat{x}(t)+\mathbf{F}y(t)\end{aligned}$$

Notice that since the residual is a linear transformation of the error, its rank (suppose $r(t)\in\mathbb{R}^p$) can not be larger than the observation dimension, i.e., $p\leq m$.

We are now able to define the dynamic of the latent error as:

$$\begin{aligned}\varepsilon(t+1) = &\ x(t+1)-\hat{x}(t+1)\\ = &\ \mathbf{A}x(t)-(\mathbf{A}-\mathbf{FC})\hat{x}(t)-\mathbf{F}y(t)\\ = &\ \mathbf{A}x(t)-\mathbf{A}\hat{x}(t)-\mathbf{FC})\hat{x}(t)+\mathbf{FC}x(t)\\ = &\ (\mathbf{A}-\mathbf{FC})(x(t)-\hat{x}(t))\\ = &\ (\mathbf{A}-\mathbf{FC})\varepsilon(t)\end{aligned}$$

and the residue $r(t)$ is obtained as:

$$\begin{aligned} r(t) = &\ \mathbf{W}(y(t) - \hat{y}(t)) \\ = &\ \mathbf{W}(\mathbf{C}x(t) - \mathbf{C}\hat{x}(t)) \\ = &\ \mathbf{WC}(x(t) - \hat{x}(t)) \\ = &\ \mathbf{WC}\varepsilon(t) \end{aligned}$$

We therefore have the following dynamic model for the latent error:

$$\begin{aligned} \varepsilon(t+1) = &\ (\mathbf{A} - \mathbf{FC})\varepsilon(t) \\ r(t) = &\ \mathbf{WC}\varepsilon(t) \end{aligned}$$

Notice that the observed residue $r(t)$ is governed by state error $\varepsilon(t)$ through matrix $\mathbf{WC}$ while it evolves in time through $\mathbf{A} - \mathbf{FC}$.

To simplify the notation, denote $\mathbf{C}_f = \mathbf{WC}$ and $\mathbf{A}_f = \mathbf{A} - \mathbf{FC}$. The DRM is then defined as:

$$\begin{aligned} \varepsilon(t+1) = &\ \mathbf{A}_f\varepsilon(t) \\ r(t) = &\ \mathbf{C}_f\varepsilon(t) \end{aligned}$$

The graphical diagram for this error model is shown in Figure 4.3.

4.3.2 OSAD Parameter Design

In this section we address the problem of designing the **F** and **W** matrix with objective of making the DRM insensitive to anomalies generated by **P**. The overarching design is shown in Figure 4.4 and is related to the use of control theory for fault diagnosis [96, 97, 98]. A typical LDS model will output the observed error $e(t)$. However, the OSAD model has a feedback loop which takes **W** and **F** matrices as input and return a variable $u(t)$ which is fed back into the model. The observed error is also transformed by a **W** matrix. The **F** and the **W** matrices satisfy the constraints which involve the **A**, **C** and the **P** matrices.

Since the model is time-dependent, we follow a standard approach and map the model into the frequency domain using a $\mathcal{Z}$-transform to design the **W** and **F** matrices. In the frequency domain, it will be easier to design matrices **W** and **F** such that $\mathbf{WC}(\mathbf{A} - \mathbf{FC}) = 0$ and $\mathbf{WCP} = 0$.

Definition 2 *The Z-transform of a discrete-time sequence $x(k)$ is the series $X(z)$ defined as*

$$X(z) = \mathcal{Z}\{x(k)\} = \sum_{0}^{\infty} x(k)z^{-k}.$$

Observation 1 *Two important (and well known) properties of the Z-transform are linearity and time shifting:*

$$ax(k) + by(t) \overset{Z}{\longleftrightarrow} aX(z) + bY(Z)$$

$$x(k+b) \overset{Z}{\longleftrightarrow} z^b X(z)$$

Applying Z-transform $\mathcal{Z}()$ to the DRM yields:

$$\begin{aligned} z\mathcal{E}(z) = & \quad \mathbf{A}_f \mathcal{E}(z) + \mathbf{P}\boldsymbol{\xi}(z) \\ \mathcal{E}(z) = & \quad (z\mathbf{I} - \mathbf{A}_f)^{-1}\mathbf{P}\boldsymbol{\xi}(z) \end{aligned}$$

and:

$$\begin{aligned} R(z) = & \quad \mathbf{C}_f \mathcal{E}(z) \\ = & \quad [\mathbf{C}_f (z\mathbf{I} - \mathbf{A}_f)^{-1}\mathbf{P}]\boldsymbol{\xi}(z) \end{aligned}$$

in which $\boldsymbol{\xi}(z) = \mathcal{Z}(\xi(t))$, $\boldsymbol{\vartheta} = \mathcal{Z}(\vartheta(t))$, $R(z) = \mathcal{Z}(r(t))$. The transfer gain between $\boldsymbol{\xi}$ and R:

$$\mathbf{G}_\xi(z) := \quad \mathbf{C}_f (z\mathbf{I} - \mathbf{A}_f)^{-1}\mathbf{P}$$

Thus if $\mathbf{G}_\xi$ would be zero, the residual $R(z)$ is independent of the $\boldsymbol{\xi}(z)$. In the other word, to make $R(z)$ independent of $\boldsymbol{\xi}(z)$, one must null the space of $\mathbf{G}_\xi(z)$. Then whenever $\mathcal{P}$ occurs it is transferred by a zero gain to the residual space. To find the null space $\mathbf{G}_\xi(z) = 0$, we expand it as:

$$\begin{aligned} \mathbf{G}_\xi(z) = & \quad z^{-1}\mathbf{C}_f(\mathbf{I} + \mathbf{A}_f z^{-1} + \mathbf{A}_f^2 z^{-2} + ...)\mathbf{P} \\ = & \quad 0 \end{aligned}$$

The sufficient conditions for $\mathbf{G}_\xi(z)$ to be nulled are $\mathbf{C}_f\mathbf{P} = 0$ and either $\mathbf{C}_f\mathbf{A}_f = 0$ or $\mathbf{A}_f\mathbf{P} = 0$. Thus we have the following result.

Theorem 4 *For a DRM, a sufficient condition for* $\mathbf{G}_\xi(z) = 0$ *is*

$$\mathbf{C}_f\mathbf{P} = 0 \textit{ and } \{\mathbf{C}_f\mathbf{A}_f = 0 \textit{ or } \mathbf{A}_f\mathbf{P} = 0\}$$

Now as $\mathbf{C}_f = \mathbf{WC}$, for $\mathbf{C}_f\mathbf{P} = 0$ it is sufficient that $\mathbf{WC}$ be orthogonal to $\mathbf{P}$. Furthermore for $\mathbf{C}_f\mathbf{A}_f = 0$, it is sufficient to design a matrix $\mathbf{A}_f$ such that its left eigenvectors corresponding to the zero eigenvalue are orthogonal to $\mathbf{P}$. Similarly, for $\mathbf{A}_f\mathbf{P} = 0$, it is sufficient to design a matrix $\mathbf{A}_f$, such that the right eigenvectors corresponding to the zero eigenvalues are orthogonal to $\mathbf{P}$. See Appendix D.

Now, to design a system which operates in an online fashion we proceed as follows. From the definition of residue:

$$r(t) = \mathbf{W}[y(t) - \hat{y}(t)]$$

Using the Z-transform, the computational form of the residual will be:

$$R(z) = [\mathbf{W} - \mathbf{C}_f(z\mathbf{I} - \mathbf{A}_f)^{-1}\mathbf{F}]Y(z)$$

Since $\mathbf{C}_f\mathbf{A}_f = 0$:

$$\mathbf{C}_f(z\mathbf{I} - \mathbf{A}_f)^{-1}\mathbf{F} = z^{-1}\mathbf{C}_f$$

Replacing this result to the above $R(z)$ equation:

$$R(z) = (\mathbf{W} - z^{-1}\mathbf{C}_f\mathbf{F})Y(z)$$

Applying the inverse Z-transform, the equation will be:

$$r(t) = \begin{bmatrix}\mathbf{W} & -\mathbf{C}_f\mathbf{F}\end{bmatrix}\begin{bmatrix}y(t) \\ y(t-1)\end{bmatrix}$$

This clearly says that the residual can be represented directly in terms of the observations. This property is crucial to make the anomaly detection system operate in near real-time.

4.3.3 Eigenpair Assignment and the F Matrix

In this section we explain the eigenpair assignment problem and its solution which is used for designing the matrix $\mathbf{F}$. Recall from Theorem 1, that we require either $\mathbf{C}_f\mathbf{A}_f = 0$ or $\mathbf{A}_f\mathbf{P} = 0$.

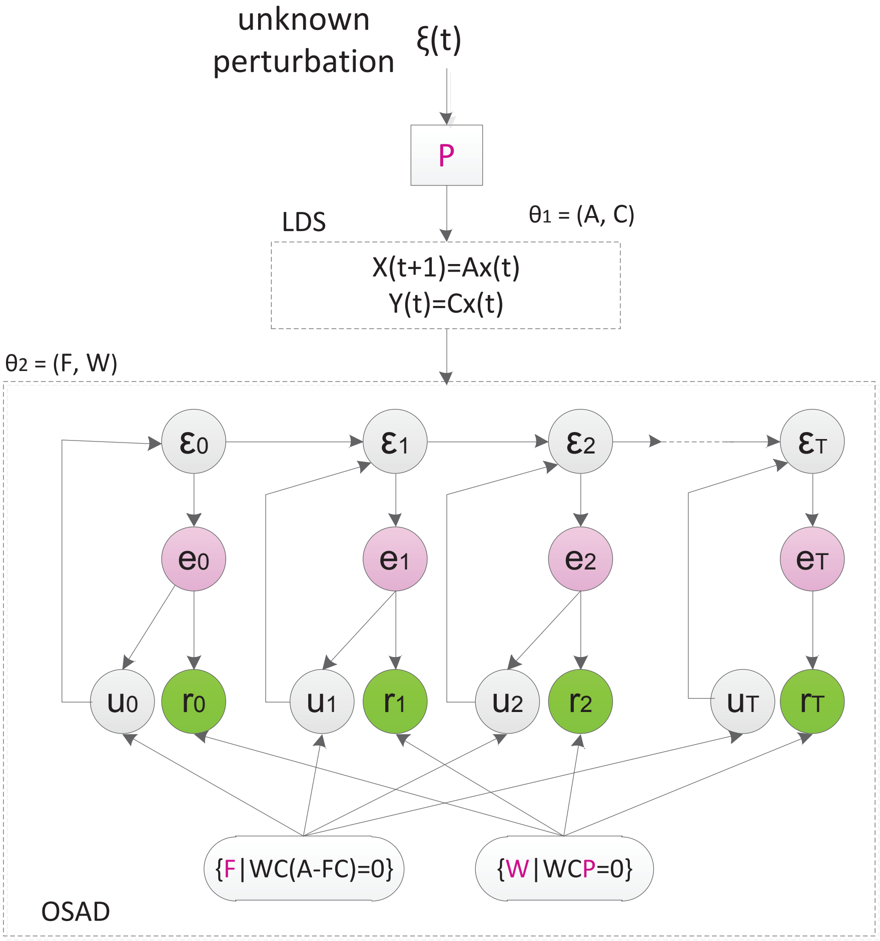


Figure 4.4: The complete diagram of OSAD. Using parameters **W** and **F** the residue space $r(t)$ is calibrated to cancel the impact of $\mathbf{P}\xi(t)$.

Problem 4 *Given a set of scalars $\{\lambda_i\}$ and a set of n-vectors $\{v_i\}$ (for $i = 1,2,...,n$), find a real matrix $\mathbf{A}_o$ ($m \times n$) such that the eigenvalues of $\mathbf{A}_o$ are precisely those of the set of scalars $\{\lambda_i\}$ with corresponding eigenvectors the set $\{v_i\}$.*

Given the residue model transition matrix $\mathbf{A}_f = \mathbf{A} - \mathbf{FC}$, the problem is to find a matrix $\mathbf{F}$ such that this matrix has the eigenvalues $\{\lambda_i\}$ corresponding to eigenvectors $\{v_i\}$,i.e.,:

$$(\mathbf{A} - \mathbf{FC})v_i = \lambda_i v_i$$

or:

$$\begin{bmatrix} \mathbf{A} - \lambda_i \mathbf{I} & \mathbf{C}' \end{bmatrix} \begin{bmatrix} v_i \\ -\mathbf{F}v_i \end{bmatrix} = 0$$

Define $q_i := -\mathbf{F}v_i$, then:

$$\begin{bmatrix} \mathbf{A} - \lambda_i \mathbf{I} & \mathbf{C}' \end{bmatrix} \begin{bmatrix} v_i \\ q_i \end{bmatrix} = 0$$

The implication of the above statement is of great importance: The vectors $\begin{bmatrix} v_i & q_i \end{bmatrix}'$ must be in the *kernel space* of $\begin{bmatrix} \mathbf{A} - \lambda_i \mathbf{I} & \mathbf{C}' \end{bmatrix}$, meaning, for $i = 1,2,...,n$:

$$\begin{bmatrix} q_1 & q_2 & ... & q_n \end{bmatrix} = \begin{bmatrix} -\mathbf{F}v_1 & -\mathbf{F}v_2 & ... & -\mathbf{F}v_n \end{bmatrix}$$

The matrix $\mathbf{F}$ now can be obtained as:

$$\mathbf{F} = -\begin{bmatrix} q_1 & q_2 & ... & q_n \end{bmatrix} \begin{bmatrix} v_1 & v_2 & ... & v_n \end{bmatrix}^+$$

where '+' stands for pseudoinverse. The whole procedure is summarized in algorithm 1.

4.3.4 Degrees of Freedom of P

There is an an important constraint that the matrix **P** must satisfy for the DRM approach to be valid solution of the OSAD problem. As the $\mathbf{WCP} = 0$, a necessary condition is that

$$\text{rank}(\mathbf{P}) \leq \text{rank}(\mathbf{C})$$

In the other word, the effective number of independent perturbations generated by the matrix **P** is bounded by the effective number of independent measurements

Algorithm 1 Find **F** such that the set $\{\lambda_i, v_i\}$ be the eigenpairs of $\mathbf{A}-\mathbf{FC}$

```
Input A,C, $\lambda_i = 0\ \forall i$ and $v_i = P(:,i)$.
Output F such that $(\mathbf{A}-\mathbf{FC})\mathbf{P} = \mathbf{0}$.
for $i = 1:n$ do
    $\phi_i = null\begin{bmatrix}\mathbf{A}-\lambda_i\mathbf{I} & \mathbf{C}'\end{bmatrix}$
    Find an element $[v_i\ q_i]' \in \phi_i$
end for
$\mathbf{F} = -\begin{pmatrix}q_1 & q_2 & \dots & q_n\end{pmatrix}\begin{pmatrix}v_1 & v_2 & \dots & v_n\end{pmatrix}^+$
```

governed by the observation matrix **C**, see [97]. For example, if **C** is the independent matrix on an LDS where the state vector has dimensionality n, then the rank of the **P** matrix must be less than $(n-1)$.

4.3.5 Inferring the Matrix P

The OSAD model is predicated on the existence of a **P** matrix. This matrix can be provided by a domain expert or can sometimes be inferred from data. For example, in the case of sleep spindle, frequency analysis shows that sleep spindles occur in the interval twelve to fourteen Hz. The exact frequency can change from one subject to another. The signature for K-Complexes is more a function of the amplitude of the signal rather than the frequency.

We now show how to construct a **P** matrix from data. For example, suppose there exists a frequency/peridicity $\mathcal{T} = f^{-1}$ in the EEG time series or:

$$x(t+\mathcal{T}) = x(t)$$

Replace this in linear dynamics:

$$\begin{aligned} x(t+1) &= \mathbf{A}x(t) \\ &= \mathbf{A}x(t+\mathcal{T}) \end{aligned}$$

Applying z-transform:

$$zX(z) = \mathbf{A}z^{\mathcal{T}}X(z)$$

Using Tailor expansion we expand $z^{\mathfrak{T}}$ around $z=1$:

$$z^{\mathfrak{T}} \approx 1+\alpha+\beta z+\gamma z^2$$

where $\alpha = 0.5\mathfrak{T}(\mathfrak{T}-3)$, $\beta = 0.5\mathfrak{T}(\mathfrak{T}-1)$ and $\gamma = -\mathfrak{T}(\mathfrak{T}-2)$. An approximation by this expansion will be:

$$zX(z) \quad \approx \mathbf{A}X(z)+\alpha\mathbf{A}X(z)+\beta z\mathbf{A}X(z)+\gamma z^2X(z)$$

Returning to the time-domain, we obtain

$$\begin{aligned} x(t+1) \quad &\approx \mathbf{A}x(t)+\alpha\mathbf{A}x(t)+\beta\mathbf{A}x(t+1)+\gamma\mathbf{A}x(t+2) \\ &\approx \mathbf{A}x(t)+[\alpha\mathbf{A} \quad \beta\mathbf{A} \quad \gamma\mathbf{A}][x(t) \quad x(t+1) \quad x(t+2)]' \end{aligned}$$

4.3.6 Summary Example

To summarize, the solution of the OSAD problem requires the availability of the following matrices:

Table 4.1: Parameters for learning and design

Matrix	Description	Source
A	The State Matrix	Inferred from data
C	The Observation Matrix	Inferred from data
P	The Pattern Matrix	Given by domain-expert
F	Feedback Gain Matrix	Designed using Theorem 1
W	Error Weighting Matrix	Designed using Theorem 1

We will now give a concrete example. Assume we have an LDS system given as

$$\begin{aligned} \varepsilon(t+1) = &\quad \mathbf{A}\varepsilon(t)+\mathbf{P}\xi(t) \\ e(t) = &\quad \mathbf{C}\varepsilon(t) \end{aligned}$$

Assume have identified the **A** and **C** matrices as

$$\mathbf{A} = \begin{pmatrix} 0.5 & 0.3 \\ 0.3 & 0.2 \end{pmatrix} \text{ and } \mathbf{C} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \text{ and } \mathbf{P} = \begin{pmatrix} 1 & 1 \\ 2 & 2 \end{pmatrix}$$

Now, to form the OSAD model, we have to identify **W** and **F** such that:

1. **W** is in the null space of **CP** and
2. $\mathbf{A} - \mathbf{FC}$ has its left eigenvectors (corresponding to the **0** eigenvalue), the rows of **WC**.

Since **C** is the identity matrix, an example of **W** is

$$\mathbf{W} = \begin{pmatrix} 2 & -1 \\ 2 & -1 \end{pmatrix}$$

Similarly, an example of **F** matrix is

$$\mathbf{F} = \begin{pmatrix} 0.0 & 0.2 \\ -0.7 & 0 \end{pmatrix}$$

As mentioned, the residual matrix is given by

$$r(t) = \begin{pmatrix} 1.3 & -1.4 \\ 1.3 & -1.4 \end{pmatrix} \begin{pmatrix} y(t) \\ y(t-1) \end{pmatrix}$$

4.4 Experimental Result

We now report on the experiments that have been carried out to test the effective of the proposed OSAD solution on sleep data. Our particular focus will be determining if OSAD can recognize sleep spindle and K-Complex anomalies and selectively raise an alert for non-Spindle anomalies.

4.4.1 Sleep Data Set

Our data set consists of EEG time series from four health controls (age 25-36) as described in [99]. Recordings were made with an Alice-4 system (Respironics, Murraysville PA, USA) at the Woolcock Institute of Medical Research, at Sydney University, using 6 EEG channels with a sampling rate of 200 Hz, and electrodes positioned according to the International 10-20 system [2, 3], see Figure 4.5. The International

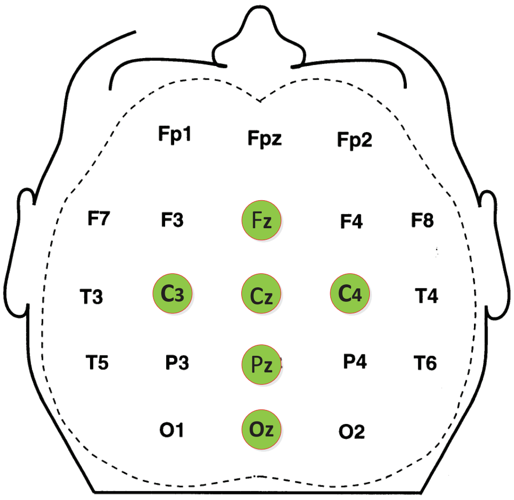
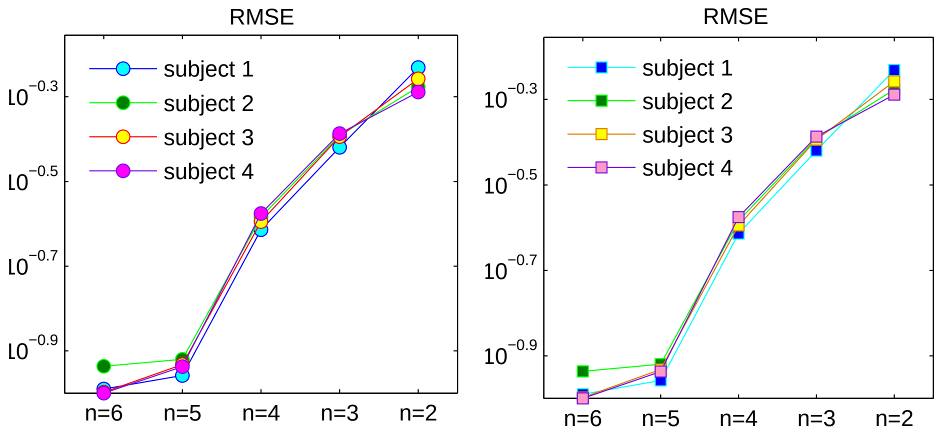


Figure 4.5: The position of scalp electrodes for EEG experiment follows the International 10-20 system [2, 3].

10/20 system is an internationally recognized method to describe the location of scalp electrodes. In this study we only examine the Cz electrode. A notch filter at 50 Hz (as provided by the Alice-4 system) was used to remove mains voltage interference. No other hardware filters were used. Spindles and K-Complexes were labeled using another automation program and then manually evaluated. As previously noted, while data from only four subjects were used, a typical EEG session generates a large amount of personal data.

4.4.2 Inference of A and C Matrices

Our first task is to learn the **A** and **C** matrices from the LDS for each subject. Others have reported, and our experiments confirm, that EEG of each subject tends to different and separate models need to learnt per subject. For each subject we took a sample of size 2000 (10 seconds) of EEG time series which did not contain either sleep spindle or K-Complex. We then formed a 2000×6 data matrix, **O**. The columns of the **O** matrix are time series associated with the six channels of EEG. We used both subspace and spectral methods to infer the matrices **A** and **C**. Both these methods are based on SVD decomposition of the **O** matrix and require as input the rank required of the



(a) Subspace method

(b) Spectral method

Figure 4.6: The RMSE error obtained from both methods are comparable. Notice the RMSE increases as the rank of LDS is reduced.

inferred matrices. We evaluated the inferred matrices using RMSE and the results are shown in Figure 4.6a and Figure 4.6b. Both the subspace and spectral methods have similar performance and RMSE goes up significantly when the rank falls below five. We selected a rank six matrix (maximum possible rank) for both **A** and **C**. In terms of running time, the two methods are comparable as we have to carry out an SVD of a relatively small 6×6 matrix.

4.4.3 Detection of SS and K-Complex

For each of the four subjects, statistics of the labeled sleep spindles and K-Complexes and those detected by the LDS are shown in Table 4.2. For LDS detection, we used a threshold derived from CUSUM which automatically adjusts for mean and standard deviation of the observed residual time series $e(t)$. To specify a CUSUM threshold we applied the alpha and beta approach in [100] and we set the probabilities of a false positive and a false negative to 10^{-4} and the change detection parameter to 1 sigma, in all subjects.

In all four subjects, the LDS residual slightly under predicts the number of spindles and K-Complexes. Since each labeled and predicted SS and K-Complex spans a time-

Table 4.2: Summary statistics of results. LDS is quite accurate but tends to over-predict the number of anomalies.

	No. of Labeled Anomalies		No. of Detected Anomalies	
	Spindle	K-Complex	Spindle	K-Complex
subject 1	170	277	164	251
subject 2	6	13	6	11
subject 3	23	38	21	37
subject 4	141	205	132	186

interval, we have modified the definitions of `precision` and `recall` to account for the intervals. For a given subject, let $\{[a_i, b_i]\}_{i=1}^{n}$ be the intervals of the labeled anomalies (spindles or K-Complex). Let $\{[a_j^{'}, b_j^{'}]\}_{j=1}^{m}$ be the predicted spindles. Then

$$\texttt{precision} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} |[a_i, b_i] \cap [a_j^{'}, b_j^{'}]|}{\sum_{j=1}^{m} |[a_j^{'}, b_j^{'}]|}$$

and

$$\texttt{recall} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} |[a_i, b_i] \cap [a_j^{'}, b_j^{'}]|}{\sum_{j=1}^{m} |[a_j, b_j]|}$$

Here, $|[a_i, b_i]|$, is the number of points in the time interval $[a_i, b_i]$. With these definitions in place, Table 4.3 and Table 4.4 show the precision and recall SS and K-Complex across alls the subjects. In general both precision and recall are high across subjects, but precision is significantly more higher than recall. For SS, the recall varies more than precision ranging for 71.24% to 97.18%. Also notice that the length of detection of both SS and K-Complex is higher compared to their labeled lengths.

4.4.4 Evaluation across Subjects

We now investigate the transfer properties of the inferred LDS across subjects. That is, we learn the **A** an **C** matrices on one subject and evaluate it against an another. We

Table 4.3: Summary statistics for spindles. LDS has higher precision than recall and total length of predicted interval is higher than the length of labeled intervals.

	Total time of spindles		Performance	
	Labeled in sec	Detected in sec	Recall	Precision
subject 1	129.8	168.74	95.53%	71.24%
subject 2	3.45	3.55	97.38%	97.18%
subject 3	15.15	16.23	95.66%	83.88%
subject 4	93.5	103.2	95.42%	79.15%

Table 4.4: Summary statistics for K-Complex. Both precision and recall are high. Total length of predicted interval is higher than labeled intervals.

	Total time of K-Complex		Performance	
	Labeled in sec	Detected in sec	Recall	Precision
subject 1	198.23	216.35	93.43%	90.45%
subject 2	11.48	11.25	94.12%	92.76%
subject 3	21.39	24.56	92.06%	91.01%
subject 4	147.68	160.49	93.73%	91.28%

Table 4.5: Recall across subjects. A substantial reduction in accuracy when model of one subject is evaluated against the EEG of another.

	A_1, C_1, W_1, F_1	A_2, C_2, W_2, F_2	A_3, C_3, W_3, F_3	A_4, C_4, W_4, F_4
subject 1	71.24%	38.13%	44.13%	41.29%
subject 2	41.32%	97.18%	35.26%	37.85%
subject 3	48.74%	43.21%	83.88%	44.43%
subject 4	51.26%	43.81%	35.36%	79.15%

Table 4.6: Precision across the subjects. Again, a substantial reduction in accuracy when model of one subjected is evaluated against another.

	A_1, C_1, W_1, F_1	A_2, C_2, W_2, F_2	A_3, C_3, W_3, F_3	A_4, C_4, W_4, F_4
subject 1	95.53%	41.11%	47.19%	43.67%
subject 2	39.54%	97.38%	37.82%	39.21%
subject 3	48.21%	41.29%	95.77%	41.83%
subject 4	51.77%	53.34%	33.49%	95.42%

just focus on the anomaly. The `recall` and `precision` results are shown in Table 4.5 and Table 4.6 respectively. The diagonal of the table corresponds to the results in Table 4.3 and Table 4.4. It is clear that there is a substantial reduction in accuracy and that indeed the EEG of subjects varies substantially. We have also computed the ”average” **A** and **C** matrix and evaluated against all the four subjects. The results are shown in Table 4.7. While there is an improvement compared to results in Table 4.5 and Table 4.6, the absolute performance is still quite low compared to the situation where the learning was customized per individual subject.

Table 4.7: Recall and Precision on each subject evaluated against an averaged model. Again, a substantial reduction in accuracy compared to individual models.

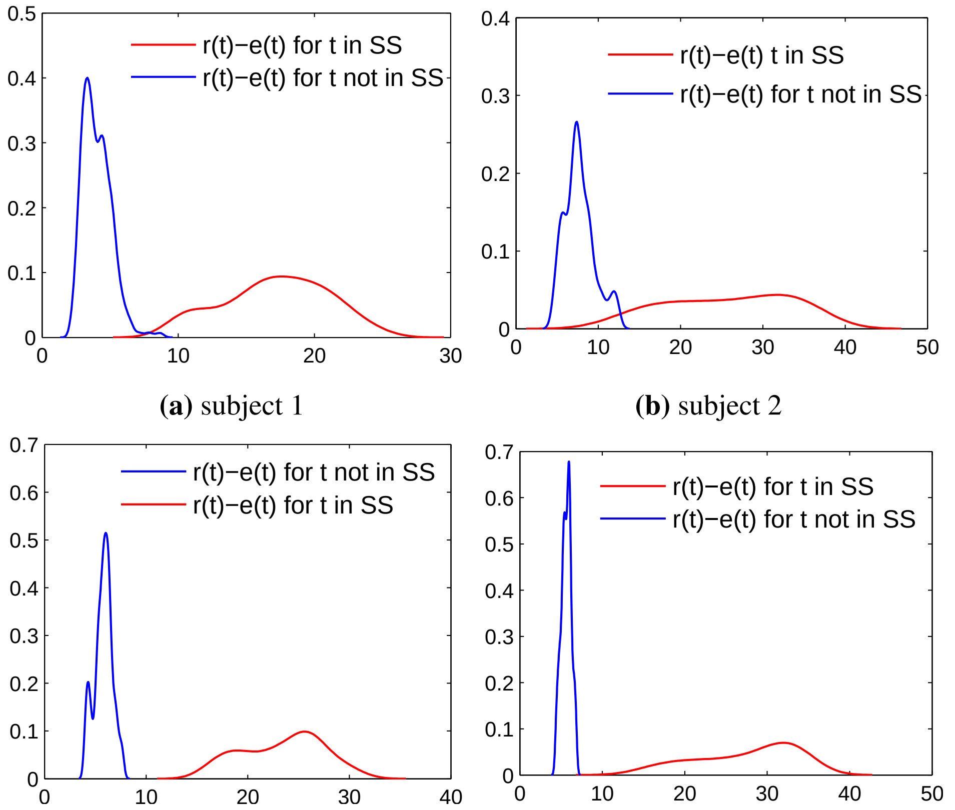
	Recall	Precision
subject 1	69.35%	51.39%
subject 2	65.43%	57.22%
subject 3	61.77%	61.47%
subject 4	68.12%	53.92%

4.4.5 Performance of Designed Residual

In this section we evaluate whether the new residual $r(t)$ satisfies the design criterion. Recall, $r(t)$ was designed to suppress the signal whenever a sleep spindle (SS) appears and behave like the observed error $\mathbf{e}(\mathbf{t})$ in otherwise. Figure 4.7 shows the distribution for $|r(t) - e(t)|_2$ for values of t when t is in (and not in) the predicted SS interval $[a^{'}_{j}, b^{'}_{j}]$ for some j. It is clear that the distribution when t is in a predicted SS interval is towards the right compared to when it is not in the interval. This is because in an SS interval, $r(t)$ will have a small absolute value (by design). In a non-SS interval, $r(t)$ will be a linear function of $e(t)$, as $r(t) = \mathbf{W}e(t)$. This behavior is observed across subjects suggesting that in all cases that $r(t)$ is behaving as designed. Furthermore in Figure 4.8, we plot the $|r(t)|$ against $|e(t)|$ when t is not in a spindle interval. Again we observe a straight line behavior, providing further confirmation that $r(t)$ is behaving according to specifications.

4.4.6 Delay in Detection of Anomalies

OSAD detects anomalies in near real time. We now discuss the lag between the appearance of a SS and before it is reported by the LDS. Figure 4.9 presents the delay distributions for subject 1 and subject 4 who experienced 164 and 132 labeled sleep spindles, respectively. In general, the predicted SS interval are longer and contain the actual intervals. This is confirmed in Figure 4.10 which shows one specific example of the location of the labeled sleep spindle and the predicted interval. In this case (which



(a) subject 1

(b) subject 2

(c) subject 3

(d) subject 4

Figure 4.7: Comparison of the distribution of the norm of $r(t) - e(t)$ for SS and non-SS intervals. In all four subjects the designed residual suppresses spindles as designed as the norm is higher for SS intervals.

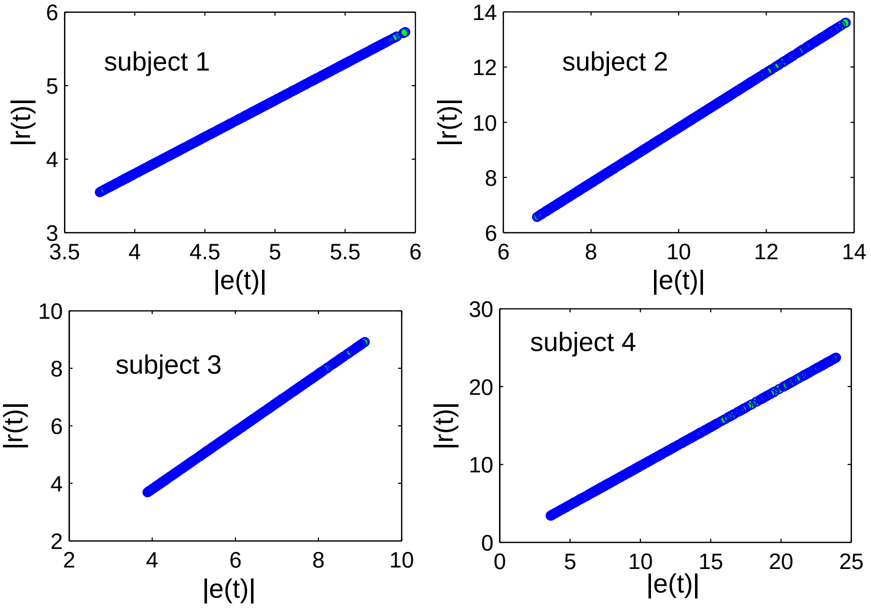
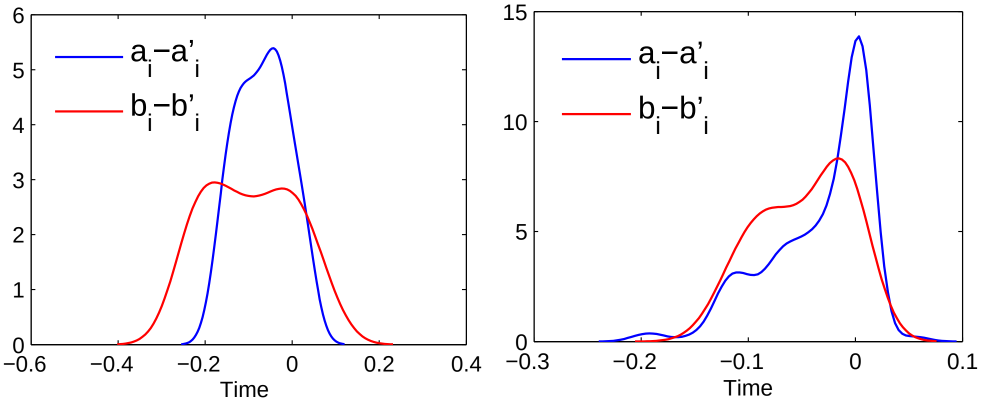


Figure 4.8: the $|r(t)|$ against $|e(t)|$ when t is not in a spindle interval as $r(t) = \mathbf{W}e(t)$

is typical), the prediction of SS begins before and ends later than the labeled spindle. Table 4.8 shows the results of the mean delay between matched intervals. Thus a mean of (a_i, a_i') equal to -0.0678 implies that on average, there was a delay of 1/200 second before LDS reported an anomaly. On the other hand for subject 2 there the SS was, on average, reported before it showed up in the labeled sequence. As noted in [99], this is consistent with the observation (and confirmed by double-blind scoring) that the labeling of SS is more conservative i.e., SS are labeled for a shorter duration than what they should be.

4.5 OSAD for Network Traffic Data

The OSAD problem is quite general and not limited to sleep analytics. Here we summarize its application for the detection of anomalies in communication network traffic. DoS and port scan attacks are emblematic of two types of deviations in network traffic. DoS attacks are characterized by large changes in a (relatively) small number of flows as the attacking hosts send a large number of small packets, typically TCP SYN segments, to deplete system resources in the attacked host. Thus DoS like anomalies



(a) subject 1 **(b)** subject 4

Figure 4.9: OSAD provides near real time detection. The delay between the actual appearance of a spindle and the predicted appearance is a fraction of a second. Similarly the lag between when the actual spindle disappears and it is reported to disappear is very small too. The x-axis is in seconds.

cause high temporal variation in the flows packet volume and can be detected using techniques based on time series analysis [6, 10, 11, 12, 13]. A port scan attack is typically accomplished by sending small packets as connections requests to a large number of different ports on a single destination IP address. At the flow level, they are therefore characterized as small increases in a large number of flows. Thus time series approaches often fail to detect port scans. Thus we can design perturbation matrices **P** which are specific to both DoS and port scans.

Table 4.8: Delay statistics. The lag between appearance and prediction of SS is, on average, a fraction of a second.

	$\mathtt{Mean}(a_i - a'_i)$	$\mathtt{Mean}(b_i - b'_i)$"	$\mathtt{Std}(a_i - a'_i)$	$\mathtt{Std}(b_i - b'_i)$
subject 1	-0.0678	- 0.0961	+0.0589	+0.0995
subject 2	+0.0041	+0.0016	+0.0113	+0.0089
subject 3	-0.0426	+0.0663	+0.0550	+0.0478
subject 4	-0.0340	-0.0480	+0.0474	+0.0407

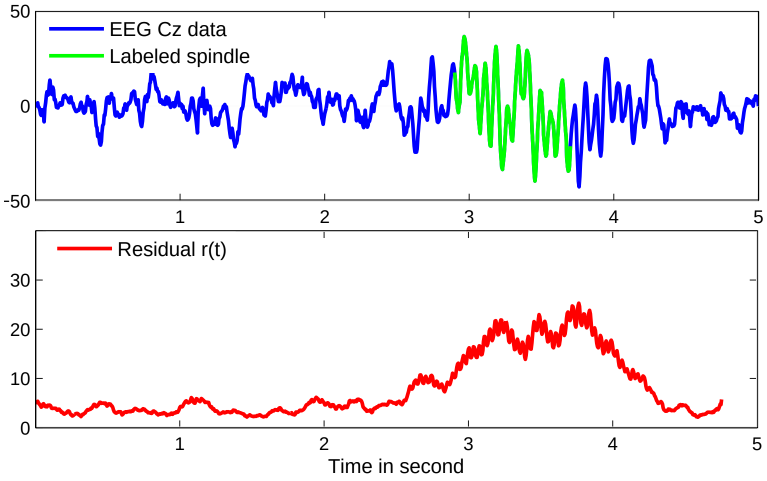


Figure 4.10: Top: Cz data and a typical sleep spindle labeled. Bottom: Residual and detected sleep spindle. In general the predicted spindle interval is longer than the labeled interval. The predicted interval tends to include the labeled interval, i.e., it begins earlier and finishes later. The EEG shows that the labeled intervals are actually quite conservative.

We use a widely known network traffic trace from the Abilene network[1] which has been widely used in network research [8, 12, 32, 47, 49]. The data set consists of a one month traffic trace from a backbone router in New York during August 2007. The data is aggregated at the five minute time intervals based on flows. Each five minute interval was either labeled as non-anomalous or with the specific anomaly observed during the interval.

As before we used both a subspace method [17] to estimate the LDS parameters, $\theta = \{\mathbf{A}, \mathbf{C}, \mathbf{Q}\}$, the entropy of each five minute interval based on source IPs, (2) destination IPs, (3) source ports, and (4) destination ports. The observation is a vector of $y(t) = (IP_{src} \quad IP_{des} \quad port_{src} \quad port_{des})'$ and $y(t) \in \mathbb{R}^4$, e.g $m = 4$. We chose $n = 4$ leading to $x(t) \in \mathbb{R}^4$, , and $A \in \mathbb{R}^{4\times4}$, and $C \in \mathbb{R}^{4\times4}$ are the state transition and the observation matrices.

$$\begin{aligned} x(t+1) = & \quad \mathbf{A}x(t) + \mathbf{P}\xi(t) \\ y(t) = & \quad \mathbf{C}x(t) + \xi(t) \end{aligned}$$

[1] Internet2 - http://www.internet2.edu/

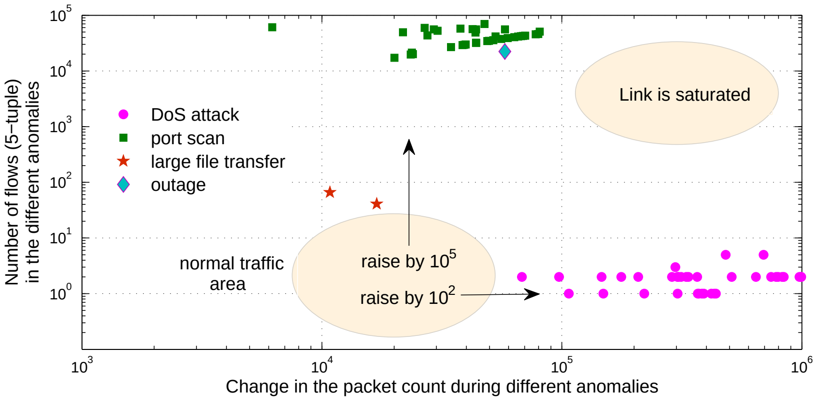


Figure 4.11: Characterization of DoS attacks and port scans by the number of flows and the change in packet volume for a traffic trace observed on a link in Abilene network April 2007.

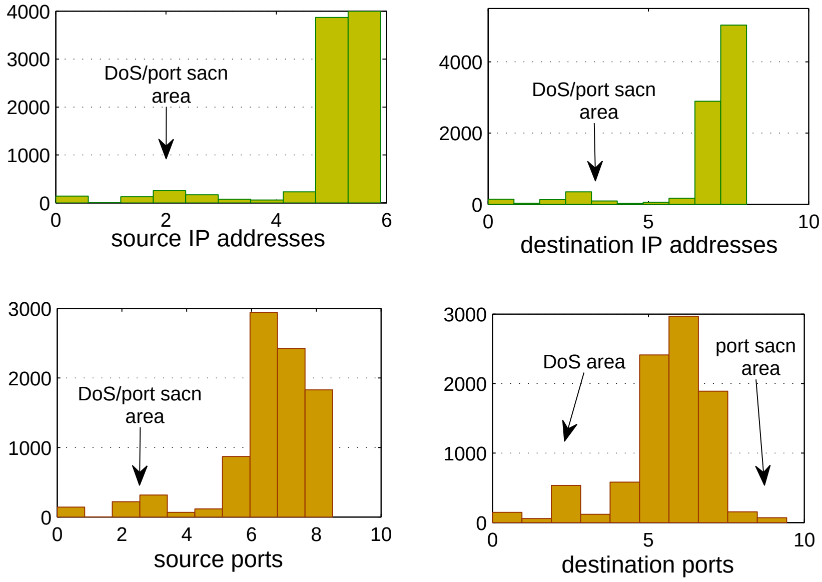


Figure 4.12: DoS attacks cause large changes within a few flows: the entropies of source/destination IP addresses and source/destinations ports significantly decrease. A port scan attack cause small changes across many flows: the entropies of source/destination IP addresses and source ports decrease while the entropies of destination ports increase dramatically.

A DoS attack by at time t_d is characterized by a fall in the entropies. Thus:

$$\begin{aligned} \Delta x(t_d) = &\ x(t_d) - x(t_d - 1) \\ = &\ \begin{pmatrix} -\alpha_1 & -\alpha_2 & -\alpha_3 & -\alpha_4 \end{pmatrix}' \end{aligned}$$

where $\{\alpha_i\}$ are positive scalars. Now we can use the DRM to suppress DoS attacks, by defining a **P** matrix as:

$$\mathbf{P} = \begin{pmatrix} -\alpha_1 & -\alpha_2 & -\alpha_3 & -\alpha_4 \end{pmatrix}'$$

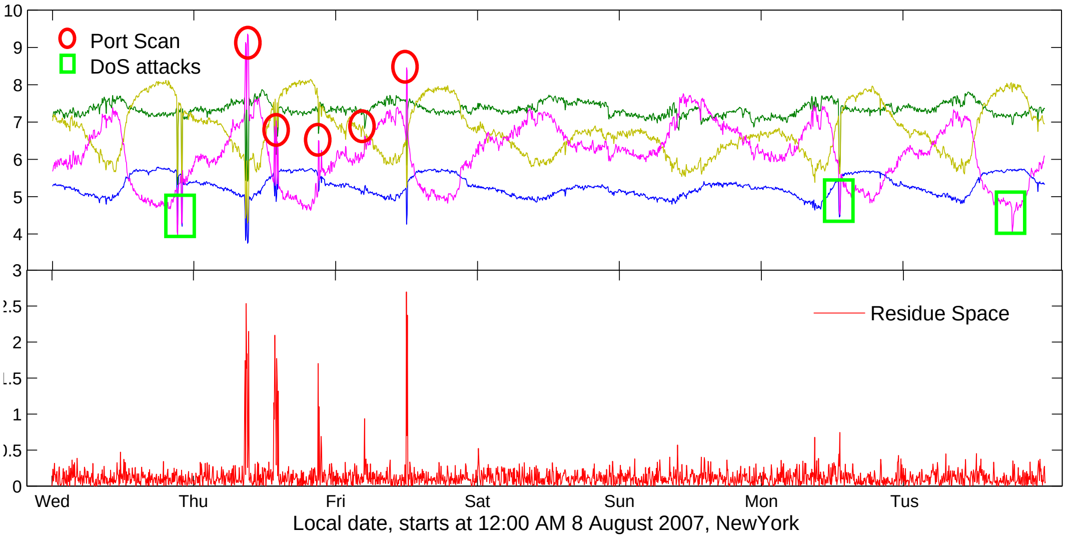
By looking at specific instances of DoS attacks, we noticed that the entropies approximately fell down by rate of -5 for the New York link in Abilene network. Therefore, we consider $\mathbf{P}_d = \begin{pmatrix} -5 & -5 & -5 & -5 \end{pmatrix}'$.

Similarly, port scans are characterized by source entropies falling and destination entropies increasing. Again, by observing real instances of port scans we set the **P** matrix as
$\mathbf{P}_p = \begin{pmatrix} -3 & -3 & -3 & +5 \end{pmatrix}'$.

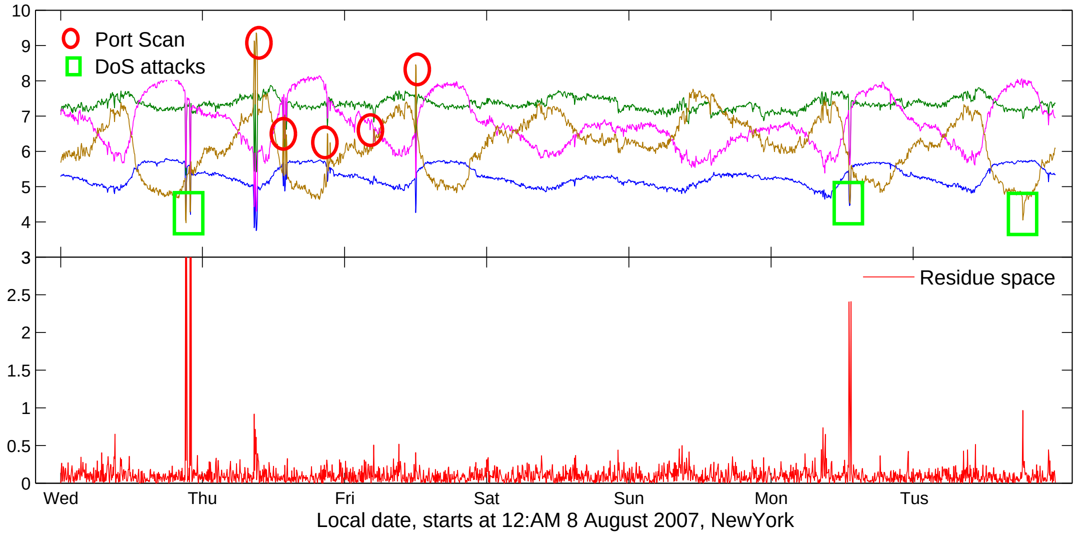
4.6 Related work

Automatic detection of sleep spindles is now an important topic in biomedical research. Different techniques including FFTs, wavelet analysis and autoregressive time series modeling have been applied for sleep spindle detection [101, 102, 103]. Attempts to integrate SVM to detect sleep spindles have also been explored [104]. There seems to be a large variability between sleep EEG across subjects. In our experiments we have also observed this phenomenon. This combined with the large amount of EEG noise has resulted in low level of agreement on the exact profile of sleep spindle [105].

The use of ”Linear Dynamical Systems” (LDS) to model time series is ubiquitous both in computer science [95] and control theory [17, 18, 106, 107, 108]. Expressing LDS in the language of graphical models and connections with HMM have been extensively examined in machine learning. The use of LDS for anomaly detection has also been investigated in network anomaly detection, among other areas [12]. The use of subspace identification methods for inferring the parameters of LDS have been discussed by Overschee [17]. Subspace methods estimate LDS parameters through a spectral decomposition of a matrix of observations to yield an estimate of the underlying state space. Subspace methods have low computational cost, are robust to perturbations and are relatively easy to implement. The recently introduced spectral learning methods are variations of the subspace method [19, 77]



(a) Top: Traffic measured in second week of August 2007 in NewYork, DoS attacks and port scans are labeled manually. Button: the residue based on $\mathbf{P}_d$ discards DoS attacks as anomaly.



(b) Top: Traffic measured in second week of August 2007 in NewYork, DoS attacks and port scans are labeled manually. Button: the residue based on $\mathbf{P}_p$ discards port scans as anomaly.

The use of eigenstructure assignment to alter the residual of an LDS has been investigated in the control theory literature especially in the context of fault diagnosis [109].Our approach closely follows the work Patton et. al. [110] who have used eigenstructure assignment for altering the LDS model using feedback. Other variations of LDS and fault diagnosis are discussed in [96, 97, 98, 111].

4.7 Summary

In this chapter we have introduced a new problem, the Online Selective Anomaly Detection (OSAD) to capture a specific scenario in sleep research. Scientists working on sleep EEG data required an alert system, which trigger alerts on selected anomalies. For example, sleep stage two is characterized by two known anomalies: sleep spindle and K-complex. The requirement was to design a system which detected both anomalies but only generated an alert when a non sleep spindle anomaly appeared. We combined methods from data mining, machine learning and control theory to design such a system. Experiments on real data set demonstrate that our approach is accurate and produces the required results and is potentially applicable to many other situations. We also note that data from sleep EEG provides a fertile ground to apply existing data mining methodologies and potentially design new computational problems and algorithms.

5

Conclusion

THIS thesis has addressed the anomaly detection problem in the context of complex time series using both learning and control theory.

The study has first presented some of the current practical challenges for anomaly detection in traffic time series by providing updated and detailed information on problem framework, traffic metrics, anomalies characterization, extractions techniques and solutions. Furthermore, we also showed how the state-of-the-art detection schemes vary in result and reported the strengths and shortcomings found. Specifically, we analysed a testbed traffic data set for anomaly detection and provided an empirical comparison about the type and characteristics of the threats every technique is able to flag.

Second, we have proposed a unified and robust method for network anomaly detection based on Multivariate Singular Spectrum Analysis (M-SSA). As M-SSA can detect deviations from both spatial and temporal correlation present in the data, it allows for the detection of both DoS and port scan attacks. A DoS attack is an example of temporal deviation while a port scan attack violates spatial correlation. Besides the use of M-SSA for network anomaly detection, we have carried out a comprehensive evaluation and compared M-SSA with other approaches based on wavelets, Fourier analysis, Kalman filtering and the recently introduced ASTUTE method. We have also carried out a rigorous analysis of the parameter configurations that accompany the use of M-SSA and address some of the important issues that have been raised in the networks community. Finally we have introduced a new labeled dataset from a large backbone link between Japan and the United States. Moreover, we showed that the proposed method, achieves the best overall performance for scan detection.

Third, we have introduced a new problem, the Online Selective Anomaly Detection (OSAD) to capture a specific scenario in sleep research. Scientists working on sleep EEG data required an alert system, which trigger alerts on selected anomalies. For example, sleep stage two is characterized by two known anomalies: sleep spindle and K-complex. The requirement was to design a system which detected both anomalies but only generated an alert when a non sleep spindle anomaly appeared. We combined methods from data mining, machine learning and control theory to design such a system. Experiments on real data set demonstrate that our approach is accurate and produces the required results and is potentially applicable to many other situations. We also note that data from sleep EEG provides a fertile ground to apply existing data mining methodologies and potentially design new computational problems and algorithms.

Appendix A

Pseudocodes

A.1 PCA-Based Subspace Method

Here we present the pseudocode of subspace method using PCA.

Algorithm 2 Subspace method based on PCA

```
Input: Y, alert_threshold
Output: anomalous_time
Covar = (YY')/(M-1); {% calculate covariance matrix of link load measurements Y}
[PC,Landa]=eig(Covar); {% determine the PCs of Y by the singular value decomposition of covariance matrix}
[Landa,i]=sort(-diag(Landa)); {% order the eigenvalues by higher properties}
Landa=-Landa;
PC=PC(:,i);  {% order the PCs by higher variance properties}
U = PC(:,1:k); {% choose the top k of PCs with the highest eigenvalues and construct normal subspace PCs U(M $\times$ k)}
$\bar{Y}$= UU'Y; {% project Y on these k axes to determine normal traffic subspace Y}
$\tilde{Y}$= (I-UU')Y; {% map Y on residual axes to determine anomalous traffic subspace $\tilde{Y}$}
for t = 1 $\rightarrow$ T do
    SPE = norm($\tilde{Y}(:,t)$)$^2$; {% calculate the Euclidian norm of each vector in $\tilde{Y}$}
    if SPE > alert-threshold then
        anomalous_time = [anomalous_time  t];
    end if
end for
```

A.2 Kalman Filter for Anomaly Detection

Here we present the pseudocode of the method of Kalman filtering for anomaly detection.

Algorithm 3 Kalman Filter

```
Input: A, B, Q, R {Input the state space computed matrices and covariances}
Input: $\hat{X}_0, \hat{P}_0$ {}Initialize the state and the variance of the estimation
Input: alert\_threshold {}Input the threshold for residual analysis for $(1-\beta)$ confidence interval
for $t = 0 \to T$ do
    $\hat{X}_{t+1} = B\hat{X}_t$ {}estimate the state
    $\hat{P}_{t+1} = BP_tB' + Q$ {}calculate the variance of the estimation
    $\varepsilon_{t+1} = Y_{t+1} - A\hat{X}_{t+1}$ {}calculate error of the prediction
    $S_{t+1} = A\hat{P}_{t+1}A' + R$ {}calculate Variance of the prediction error
    $K_{t+1} = \hat{P}_{t+1}A'S_{t+1}^{-1}$ {}calculate Kalman gain
    $\tilde{X}_{t+1} = \hat{X}_{t+1} + \hat{P}_{t+1}A'R^{-1}\varepsilon_{t+1}$ {}update the estimated state
    $\tilde{P}_{t+1} = (I - K_{t+1}A)\hat{P}_{t+1}$ {}update the variance of estimation
    $\eta_{t+1} = \tilde{X}_{t+1} - \hat{X}_{t+1}$ {}calculate the residuals which is a Gaussian zero-mean process
    if $\eta_{t+1}$ violates alert-threshold then
        anomalous_time= [anomalous_time t]; {if residual violates the alert-threshold, an anomaly has occurred at t}
    end if
end for
```

A.3 ASTUTE for Network Anomaly Detection

Here we present the pseudocode of ASTUTE for anomaly detection.

Algorithm 4 ASTUTE

```
Input: alert_threshold {input the state space computed matrices and covariances}
Input: $\mathcal{F}_0 = \{\mathrm{f}_{1,0}, \mathrm{f}_{2,0}, \dots, \mathrm{f}_{\mathrm{N},0}\}$ { input the initial flow matrix}
Require: $1 \leq$ time bins $\leq 5$ {flows must be measured in less than 5 minute time bins to hold the second assumption.}
Require: $\forall t : N \geq 100$ {only for at least 100 flows in a time bin, the AVV distribution is close to the Gaussian distribution.}
for $t = 0 \rightarrow T$ do
    Input : $\mathcal{F}_{t+1} = \{f_{1,t+1}, f_{2,t+1}, \dots, f_{N,t+1}\}$ {input the flow matrix}
    $\mathrm{N} = \mathrm{Size}(\mathcal{F}_{t+1})$
    for $\mathrm{i} = 1 \rightarrow \mathrm{N}$ do
        $\delta_{\mathrm{f}_{\mathrm{i,t}}} = \mathrm{f}_{\mathrm{i,t+1}} - \mathrm{f}_{\mathrm{i,t}}$ {compute the change of flow $\mathrm{f}_{\mathrm{i}}$ from time bin t to $\mathrm{t}+1$}
    end for
    $\Delta_{\mathrm{t}} = \{\delta_{\mathrm{f}_{1,\mathrm{t}}}, \delta_{\mathrm{f}_{2,\mathrm{t}}}, \dots, \delta_{\mathrm{f}_{\mathrm{N,t}}}\}$ {construct the et of changes for all the active flows}
    $\hat{\delta}_{\mathrm{t}} = \mathrm{mean}(\Delta_{\mathrm{t}})$ {calculate the sample mean of the set of changes}
    $\hat{\sigma}_{\mathrm{t}} =$ standard deviation $(\Delta_{\mathrm{t}})$ {calculate the sample standard deviation of the set of changes}
    $\mathrm{AAV} = \frac{\hat{\delta}_{\mathrm{t}}}{\hat{\sigma}_{\mathrm{t}}} \sqrt{\mathrm{N}}$ {calculate the ASTUTE assessment value}
    if AAV violates alert_threshold then
        anomalous_time= [anomalous_time t]; {if AAV violates the alert_threshold, an anomaly has occurred at t}
    end if
end for
```

Appendix B

Hankelization

The purpose of Hankelization is to transform an arbitrary matrix to the form of a Hankel matrix, which subsequently corresponds to a time series. Hankelization operator $\mathcal{H}$ act on any arbitrary matrix **M** to turn it into a Hankle matrix, $\mathbb{M} = \mathcal{H}\mathbf{M}$ such that it is the nearest one to M in respect to the matrix norm.

Problem 5 *Find the nearest Hankel matrix* $\mathbb{M}$ *to the matrix* $\boldsymbol{M}$ *(with respect to the matrix norm), i.e.*

$$\underset{\mathbb{M}}{arg\,min} \quad ||\mathbb{M} - \boldsymbol{M}||^2 \tag{B.1}$$

$$\textit{subject to}$$

$$\mathbb{M}_{i,j} = \mathbb{M}_{i-1,j+1}$$

Solution: Assume $\mathbf{M}_{i,j}$ as the entries of the matrix **M**, then the n-th element of the resulting series is obtained by averaging $\mathbf{M}_{i,j}$ over all i and j such that $\mathrm{i}+\mathrm{j} = \mathrm{n}+2$ (see [81] and [78] for details).

Appendix C

DRM and Kalman Filter

If there is no constrained on the anomaly detection problem the but the latent variables and observations are governed by a linear stochastic difference model, then the DRM is reduced to a Kalman filter problem.

Suppose $\omega(t) := \mathbf{P}\xi(t) + \eta(t)$, then:

$$\begin{aligned} x(t+1) = &\quad \mathbf{A}x(t) + \omega(t) \\ y(t) = &\quad \mathbf{C}x(t) + \upsilon(t) \end{aligned}$$

$$\begin{aligned} \omega \sim &\quad \mathcal{N}(0, \mathbf{Q}) \\ \nu \sim &\quad \mathcal{N}(0, \mathbf{R}) \end{aligned}$$

By defining a white Gaussian distribution as $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$, then the error model can be obtained:

$$\begin{aligned} \varepsilon(t+1) = &\quad \mathbf{A}\varepsilon(t) + \mathbf{P} \\ e(t) = &\quad \tfrac{\mathbf{CPC}' + \mathbf{R}}{\mathbf{PC}'}\varepsilon(t) \end{aligned}$$

Where $\mathbf{P} = E(\varepsilon\varepsilon^T)$ is the latent error covariance. Based on the Kalman theory [18, 106, 112, 113] the error model can be obtained as:

$$\begin{aligned} \varepsilon(t+1) = &\quad \mathbf{A}\varepsilon(t) + \mathbf{P} \\ e(t) = &\quad \mathbf{K}^{-1}\varepsilon(t) \end{aligned}$$

where $\mathbf{K}$ called *Kalman gain* is given by:

$$\mathbf{K} = \frac{\mathbf{PC}'}{\mathbf{CPC}' + \mathbf{R}}$$

Appendix D

Proof of Theorem 1

Theorem 1. For a DRM, a sufficient condition for $\mathbf{G}_{\xi}(z)$ *to be zero, is*

$$\mathbf{C}_f\mathbf{P} = 0 \text{ and } \{\mathbf{C}_f\mathbf{A}_f = 0 \text{ or } \mathbf{A}_f\mathbf{P} = 0\}$$

Proof: Let the set $\{\lambda_i = 0, v_i\}$, for $i = 1:n$, be the left eigenvectors and corresponding eigenvalues of $\mathbf{A}_f$, i.e.

$$\begin{aligned} v_i\mathbf{A}_f &= \lambda_i v_i \\ &= 0 \end{aligned}$$

If one chooses v_1 as the rows of matrix $[\mathbf{WC}]$, then:

$$\begin{bmatrix} v_1 & \ldots & v_n \end{bmatrix}' \mathbf{A}_f = 0 \quad \Rightarrow \quad \mathbf{WCA}_f = 0$$

The matrix $\mathbf{A}_f = \mathbf{A} - \mathbf{FC}$, so it is sufficient to chose $\mathbf{F}$ so that the set $\{\lambda_i = 0, v_i = [\mathbf{CW}]'\}$ to be assigned as left eigenpairs of $(\mathbf{A} - \mathbf{FC})$.

In the other side, suppose If the columns of $\mathbf{P}$ are the right eigenvectors of $\mathbf{A}_f$ corresponding to zero-values eigenvectors, then

$$\mathbf{A}_f v_i = 0 \quad \Rightarrow \quad \mathbf{A}_f\mathbf{P} = 0$$

So it is sufficient to chose $\mathbf{F}$ so that the set $\{\lambda_i = 0, v_i = \mathbf{P}\}$ to be assigned as right eigenpairs of $(\mathbf{A} - \mathbf{FC})$.

References

[1] Tahereh Babaie, Sanjay Chawla, and Sebastien Ardon. **Network Traffic Decomposition for Anomaly Detection**. *CoRR*, **abs/1403.0157**, 2014.

[2] Ernst Niedermeyer and FH Lopes da Silva. *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005. xii, 86, 87, 100, 101

[3] Gyorgy Buzsaki. *Rhythms of the Brain*. Oxford University Press, 2006. xii, 86, 87, 100, 101

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. **Anomaly detection: A survey**. *ACM Computing Surveys (CSUR)*, **41**(3):15, 2009. 1

[5] F. Takens, D.A. Rand, and L. S. Young. **Detecting strange attractors in turbulence**. In *Lecture Notes in Mathematics*, **898**, pages 366 – 381, 1981. 2, 60, 68, 79, 83

[6] Paul Barford, Jeffery Kline, David Plonka, and Amos Ron. **A signal analysis of network traffic anomalies**. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 71–82. ACM, 2002. 2, 3, 15, 31, 32, 71, 83, 109

[7] Wei Lu and Ali A. Ghorbani. **Network anomaly detection based on wavelet analysis**. *EURASIP Journal on Advances in Signal Processing - Special issue on signal processing applications in network intrusion detection systems*, **2009**:4:1–4:16, 2009. 2, 32, 83

[8] Fernando Silveira, Christophe Diot, Nina Taft, and Ramesh Govindan. **ASTUTE: detecting a different class of traffic anomalies**. In *Proceedings of the ACM SIGCOMM 2010 conference*, pages 267–278, 2010. 2, 3, 19, 32, 42, 43, 44, 55, 69, 71, 75, 76, 110

[9] Fernando Silveira, Christophe Diot, Nina Taft, and Ramesh Govindan. **Detecting traffic anomalies using an equilibrium property**. In *Proceedings of the ACM SIGMETRICS 2010 conference*, pages 377–378, 2010. 2, 3, 42, 43, 69, 71, 75

[10] Jake D. Brutlag. **Aberrant Behavior Detection in Time Series for Network Monitoring**. In *LISA '00: Proceedings of the 14th USENIX conference on System administration*, pages 139–146, Berkeley, CA, USA, 2000. USENIX Association. 3, 15, 30, 83, 109

[11] Balachander Krishnamurthy, Subhabrata Sen, Yin Zhang, and Yan Chen. **Sketch-based change detection: methods, evaluation, and applications**. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, IMC '03, pages 234–247, New York, NY, USA, 2003. ACM. 3, 15, 30, 83, 109

[12] Augustin Soule, Kavé Salamatian, Antonio Nucci, and Nina Taft. **Traffic matrix tracking using Kalman filters**. *SIGMETRICS Perform. Eval. Rev.*, **33**(3):24–31, December 2005. 3, 15, 19, 21, 29, 35, 39, 51, 71, 83, 109, 110, 112

[13] Yin Zhang, Zihui Ge, Albert Greenberg, and Matthew Roughan. **Network anomography**. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, IMC '05, pages 30–30. USENIX Association, 2005. 3, 15, 21, 22, 23, 31, 32, 47, 71, 83, 109

[14] D.S. Broomhead and G. P. King. **Extracting qualitative dynamics from experimental data**. In *Physica D, Nonlinear Phenomena*, **20**, pages 217 – 236, 1986. 4, 60, 68, 79, 83

[15] D. S. Broomhead and G. P. King. **On the qualitative analysis of experimental dynamical systems**. *Nonlinear Phenomena and Chaos*, pages 113–144, 1986. 4, 60, 68, 79, 83

[16] R. H. Shumway and D. S. Stoffer. **AN APPROACH TO TIME SERIES SMOOTHING AND FORECASTING USING THE EM ALGORITHM**. *Journal of Time Series Analysis*, **3**(4):253–264, 1982. 4, 90

[17] P. van Overschee and L.R. de Moor. *Subspace identification for linear systems: theory, implementation, applications*, **1**. Kluwer Academic Publishers, 1996. 4, 60, 90, 110, 112

[18] LENNART LJUNG. *System Identification*. John Wiley & Sons, Inc., 2001. 4, 60, 90, 112, 123

[19] B. BOOTS, S. SIDDIQI, AND G. GORDON. **Closing the Learning-Planning Loop with Predictive State Representations**. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010. 4, 60, 90, 112

[20] DEBRA ANDERSON, TERESA F. LUNT, HAROLD JAVITZ, ANN TAMARU, AND ALFONSO VALDES. **Detecting Unusual Program Behavior Using the Statistical Components of Next-generation Intrusion Detection Expert System NIDES**. Technical report, SRI-CSL-95-07, Computer Science Laboratory, SRI International, 1995. 11

[21] MICHAEL E. WHITMAN AND HERBERT J. MATTORD. *Principles of Information Security*. Course Technology Press, Boston, MA, United States, 4rd edition, 2011. 11, 13

[22] DOROTHY E. DENNING. *Information warfare and security*. Addison-Wesley Longman Ltd., 1999. 11

[23] FRANK FEATHER, DAN SIEWIOREK, AND ROY MAXION. **Fault detection in an Ethernet network using anomaly signature matching**. *SIGCOMM Comput. Commun. Rev.*, **23**(4):279–288, 1993. 11, 15

[24] CYNTHIA S. HOOD AND CHUANYI JI. **Proactive Network Fault Detection**. In *INFOCOM '97: Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*, page 1147. IEEE Computer Society, 1997. 11

[25] IRENE KATZELA AND MISCHA SCHWARTZ. **Schemes for fault identification in communication networks**. *IEEE/ACM Trans. Netw.*, **3**(6):753–764, 1995. 11

[26] M. THOTTAN AND C. JI. **Anomaly detection in IP networks**. *IEEE Transactions in Signal Processing*, **51**(8):2191–2204, 2003. 11

[27] JAMES P. ANDERSON. **Computer security threat monitoring and surveillance**. *Technical report, James P. Anderson Company, Fort Washington, Pennsylvania*, 1980. 13

[28] DOROTHY E. DENNING. **An Intrusion-Detection Model**. *IEEE Trans. Softw. Eng.*, **13**(2):222–232, 1987. 13

[29] Anukool Lakhina, John W Byers, Mark Crovella, and Ibrahim Matta. **On the geographic location of Internet resources**. *Selected Areas in Communications, IEEE Journal on*, **21**(6):934–948, 2003. 15

[30] Matthew Roughan, Tim Griffin, Morley Mao, Albert Greenberg, and Brian Freeman. **Combining routing and traffic data for detection of IP forwarding anomalies**. In *SIGMETRICS '04/Performance '04: Proceedings of the joint international conference on Measurement and modeling of computer systems*, pages 416–417. ACM, 2004. 15

[31] Amy Ward, Peter Glynn, and Kathy Richardson. **Internet service performance failure detection**. *SIGMETRICS Perform. Eval. Rev.*, **26**(3):38–43, December 1998. 15

[32] Anukool Lakhina, Mark Crovella, and Christophe Diot. **Mining anomalies using traffic feature distributions**. *SIGCOMM Comput. Commun. Rev.*, **35**(4):217–228, August 2005. 15, 19, 21, 29, 39, 41, 46, 69, 71, 83, 110

[33] Yu Gu, Andrew McCallum, and Don Towsley. **Detecting anomalies in network traffic using maximum entropy estimation**. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, IMC '05, pages 32–32, Berkeley, CA, USA, 2005. USENIX Association. 16

[34] George Nychis, Vyas Sekar, David G. Andersen, Hyong Kim, and Hui Zhang. **An empirical evaluation of entropy-based traffic anomaly detection**. In *Internet Measurement Comference*, pages 151–156, 2008. 16, 27

[35] Cristian Estan, Ken Keys, David Moore, and George Varghese. **Building a better NetFlow.** In *SIGCOMM*, pages 245–256. ACM, 2004. 16

[36] Ramana Rao Kompella and Cristian Estan. **The Power of Slicing in Internet Flow Measurement.** In *Internet Measurment Conference*, pages 105–118. USENIX Association, 2005. 16

[37] Lihua Yuan, Chen-Nee Chuah, and Prasant Mohapatra. **ProgME: towards programmable network measurement.** In Jun Murai and Kenjiro Cho, editors, *SIGCOMM*, pages 97–108. ACM, 2007. 16

[38] Ken Keys, David Moore, and Cristian Estan. **A robust system for accurate real-time summaries of internet traffic.** In *SIGMETRICS*, pages 85–96. ACM, 2005. 16

[39] JAEYEON JUNG, BALACHANDER KRISHNAMURTHY, AND MICHAEL RABINOVICH. **Flash crowds and denial of service attacks: characterization and implications for CDNs and web sites**. In *Proceedings of the 11th international conference on World Wide Web*, WWW '02, pages 293–304. ACM, 2002. 17, 30

[40] ALEFIYA HUSSAIN, JOHN HEIDEMANN, AND CHRISTOS PAPADOPOULOS. **A framework for classifying denial of service attacks**. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, pages 99–110, 2003. 17

[41] ANUKOOL LAKHINA, MARK CROVELLA, AND CHRISTIPHE DIOT. **Characterization of network-wide anomalies in traffic flows**. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, IMC '04, pages 201–206. ACM, 2004. 17, 18, 20

[42] PAUL BARFORD AND DAVID PLONKA. **Characteristics of network traffic flow anomalies**. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, IMW '01, pages 69–73. ACM, 2001. 18

[43] VERN PAXSON. **Measurements and analysis of end-to-end Internet dynamics**. Technical report, Computer Science Division, University of California, Berkeley, 1997. 18

[44] K. CLAFFY, H. BRAUN, AND G. POLYZOS. **Internet traffic flow profiling**. Technical report, Applied Network Research, San Diego Supercomputer Center, Mar 1994. 19

[45] ANJA FELDMANN, ALBERT GREENBERG, CARSTEN LUND, NICK REINGOLD, JENNIFER REXFORD, AND FRED TRUE. **Deriving traffic demands for operational IP networks: methodology and experience**. *IEEE/ACM Trans. Netw.*, **9**(3):265–280, June 2001. 19

[46] YIN ZHANG, MATTHEW ROUGHAN, CARSTEN LUND, AND DAVID DONOHO. **An information-theoretic approach to traffic matrix estimation**. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, pages 301–312. ACM, 2003. 19

[47] ANUKOOL LAKHINA, MARK CROVELLA, AND CHRISTOPHE DIOT. **Diagnosing network-wide traffic anomalies**. *SIGCOMM Comput. Commun. Rev.*, **34**(4):219–230, 2004. 19, 21, 46, 81, 83, 110

[48] Anukool Lakhina, Mark Crovella, and Christophe Diot. **Characterization of network-wide anomalies in traffic flows**. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, **35**, pages 201–206, 2004. 19, 46, 69, 71, 83

[49] Fernando Silveira and Christophe Diot. **URCA: pulling out anomalies by their root causes**. In *Proceedings of the 29th IEEE INFOCOM 2010 conference*, pages 722–730, 2010. 19, 42, 72, 75, 110

[50] Anukool Lakhina, Konstantina Papagiannaki, Mark Crovella, Christophe Diot, Eric D. Kolaczyk, and Nina Taft. **Structural analysis of network traffic flows**. *SIGMETRICS Perform. Eval. Rev.*, **32**(1):61–72, June 2004. 20, 33, 52

[51] Anukool Lakhina, Mark Crovella, and Christophe Diot. **Diagnosing network-wide traffic anomalies**. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 219–230, 2004. 20, 32, 33, 34, 52

[52] Augustin Soule, Kavé Salamatian, Antonio Nucci, and Nina Taft. **Traffic matrix tracking using Kalman filters**. *SIGMETRICS Perform. Eval. Rev.*, **33**(3):24–31, 2005. 21, 39

[53] Y Vardi. **Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data**. *Journal of the American Statistical Association*, **91**(433):365, 1996. 21

[54] David L. Donoho. **For most large underdetermined systems of equations, the minimal l1-norm near-solution approximates the sparsest near-solution**. Technical report, Comm. Pure Appl. Math, 2004. 23

[55] David L. Donoho. **For Most Large Underdetermined Systems of Linear Equations the Minimal l1-norm Solution is also the Sparsest Solution**. *Comm. Pure Appl. Math*, **59**:797–829, 2004. 23

[56] Fernando Silveira, Christophe Diot, Nina Taft, and Ramesh Govindan. **Empirical Evaluation of Network-Wide Anomaly Detection**. Technical report, Technicolor Paris Research & Innovation Center, 2008. 24

[57] Inc. Cisco Systems. **Cisco NetFlow. At**. 27

[58] HAAKON RINGBERG, AUGUSTIN SOULE, JENNIFER REXFORD, AND CHRISTOPHE DIOT. **Sensitivity of PCA for traffic anomaly detection**. *SIGMETRICS Perform. Eval. Rev.*, **35**(1):109–120, June 2007. 27, 51, 81, 82

[59] FERNANDO SILVEIRA, CHRISTOPHE DIOT, NINA TAFT, AND RAMESH GOVINDAN. **ASTUTE: detecting a different class of traffic anomalies**. In *Proceedings of the ACM SIGCOMM 2010 conference*, SIGCOMM '10, pages 267–278, 2010. 29

[60] FERNANDO SILVEIRA, CHRISTOPHE DIOT, NINA TAFT, AND RAMESH GOVINDAN. **Detecting traffic anomalies using an equilibrium property**. In *Proceedings of the ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '10, pages 377–378, 2010. 29

[61] FERNANDO SILVEIRA AND CHRISTOPHE DIOT. **URCA: pulling out anomalies by their root causes**. In *Proceedings of the 29th conference on Information communications*, INFOCOM'10, pages 722–730, 2010. 29

[62] PATRICE ABRY AND DARRYL VEITCH. **Wavelet Analysis of Long-Range-Dependent Traffic**. *IEEE Transactions on Information Theory*, **44**(1):2–15, 1998. 31

[63] INGRID DAUBECHIES, BIN HAN, AMOS RON, AND ZUOWEI SHEN. **Framelets: MRA-based constructions of wavelet frames**. *Applied and Computational Harmonic Analysis*, **14**:1–46, 2003. 32

[64] H HOTELLING. **Analysis of a complex of statistical variables into principal components**. *Journal of Educational Psychology*, **24**(6):417–441, 1933. 33

[65] I.T. JOLLIFFE. *Principal Component Analysis*. Springer, second edition, 2002. 33

[66] J.E. JACKSON. *A User's Guide to Principal Components*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley, 1991. 33

[67] H. ABDI AND L. J. WILLIAMS. **Principal Component Analysis**. *WIREs Comp Stat*, **2**:433459, 2010. 33

[68] RICARDO DUNIA AND S. JOE QIN. **Multi-dimensional Fault Diagnosis Using a Subspace Approach**. In *In American Control Conference*, 1997. 33

[69] RICARDO DUNIA AND S. JOE QIN. **Subspace approach to multidimensional fault identification and reconstruction**. *American Institute of Chemical Engineers Journal*, **44**:18131831, 1998. 33

[70] J. Edward Jackson and Govind S. Mudholkar. **Control Procedures for Residuals Associated with Principal Component Analysis**. *Technometrics*, **21**(3):pp. 341–349, 1979. 34, 46, 81

[71] Anukool Lakhina, Mark Crovella, and Christiphe Diot. **Characterization of network-wide anomalies in traffic flows**. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, IMC '04, pages 201–206. ACM, 2004. 35

[72] D. R. Jensen and Herbert Solomon. **A Gaussian Approximation to the Distribution of a Definite Quadratic Form**. *Journal of the American Statistical Association*, **67**(340):898–902, 1972. 46, 47, 81, 82

[73] Stefan Axelsson. **The base-rate fallacy and its implications for the difficulty of intrusion detection**. In *Proceedings of the 6th ACM conference on Computer and communications security*, CCS '99, pages 1–7, 1999. 47

[74] Kun chan Lan, Zhe Wang, R. Berriman, T. Moors, M. Hassan, L. Libman, M. Ott, B. Landfeldt, Z. Zaidit, and A. Senevirante. **Implementation of a Wireless Mesh Network Testbed for Traffic Control**. In *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, pages 1022 –1027, aug. 2007. 49

[75] Sara Hakami, Zainab Zaidi, Bjorn Landfeldt, and Tim Moors. **Detection and Identification of Anomalies in Wireless Mesh Networks Using Principal Component Analysis (PCA)**. In *Proceedings of the The International Symposium on Parallel Architectures, Algorithms, and Networks*, ISPAN '08, pages 266–271. IEEE Computer Society, 2008. 49

[76] Michael E. Tipping and Christopher M. Bishop. **Mixtures of probabilistic principal component analyzers**. *Neural Comput.*, **11**(2):443–482, February 1999. 51

[77] Daniel Hsu, Sham M. Kakade, and Tong Zhang. **A Spectral Algorithm for Learning Hidden Markov Models**. *CoRR*, **abs/0811.4413**, 2008. 60, 112

[78] Iosif Semenovich Iokhvidov, Jack Nicholson, Diane Keaton, Warren Beatty, Edward Herrman, Trevor Griffith, Jerry Kosinski, and Paramount Pictures. *Hankel and Toeplitz matrices and forms: algebraic theory*. Birkhäuser Boston, 1982. 63, 121

[79] R. Vautard and M. Ghil. **Singular spectrum analysis in nonlinear dynamics, with applications to paleoclimatic time series**. In *Physica D, Nonlinear Phenomena*, **35**, pages 395 – 424, 1989. 67, 83

[80] M. Ghil, M. R. Allen, M. D. Dettinger, K. Ide, D. Kondrashov, M. E. Mann, A. W. Robertson, A. Saunders, Y. Tian, F. Varadi, and P. Yiou. **ADVANCED SPECTRAL METHODS FOR CLIMATIC TIME SERIES**. *Reviews of Geophysics*, **40**(1), 2002. 67, 79, 83

[81] N. Golyandina, V. Nekrutkin, and A.A. Zhigljavsky. *Analysis of Time Series Structure: SSA and Related Techniques*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 2010. 67, 83, 121

[82] Pascal Yiou, Didier Sornette, and Michael Ghil. **Data-adaptive Wavelets and Multi-scale Singular-spectrum Analysis**. *Phys. D*, **142**(3-4):254–290, 2000. 67, 68

[83] Haakon Ringberg, Matthew Roughan, and Jennifer Rexford. **The need for simulation in evaluating anomaly detectors**. *SIGCOMM Comput. Commun. Rev.*, **38**(1):55–59, 2008. 71, 76

[84] Stefan Axelsson. **The base-rate fallacy and the difficulty of intrusion detection**. *ACM Trans. Inf. Syst. Secur.*, **3**(3):186–205, 2000. 76

[85] Daniela Brauckhoff, Kavé Salamatian, and Martin May. **Applying PCA for Traffic Anomaly Detection: Problems and Solutions**. In *INFOCOM*, pages 2866–2870, 2009. 82

[86] Martin Roesch. **Snort - Lightweight Intrusion Detection for Networks**. In *Proceedings of the 13th USENIX conference on System administration*, LISA '99, pages 229–238, Berkeley, CA, USA, 1999. USENIX Association. 83

[87] Vern Paxson. **Bro: a system for detecting network intruders in real-time**. In *Proceedings of the 7th conference on USENIX Security Symposium - Volume 7*, SSYM'98, pages 3–3. USENIX Association, 1998. 83

[88] Robert Vautard, Pascal Yiou, and Michael Ghil. **Singular-spectrum analysis: a toolkit for short, noisy chaotic signals**. *Phys. D*, **58**(1-4):95–126, 1992. 83

[89] Myles R. Allen and L. A. Smith. **Monte Carlo SSA: Detecting irregular oscillations in the Presence of Colored Noise**. *J. Climate*, **9**(12):3373–3404, 1996. 83

[90] M. Ghil and R. Vautard. **Interdecadal oscillations and the warming trend in global temperature time series**. *Nature*, **350**(6316):324–327, 1991. 83

[91] P. Yiou, E. Baert, and M.F. Loutre. **Spectral analysis of climate data**. *Surveys in Geophysics*, **17**(6):619–663, 1996. 83

[92] J.B. Elsner and A.A. Tsonis. *Singular Spectrum Analysis: A New Tool in Time Series Analysis*. The language of science. Springer, 1996. 83

[93] Susanne Diekelmann and Jan Born. **The memory function of sleep**. *Nature Reviews Neuroscience*, **11**(2):114–126, 2010. 87

[94] Thien Thanh Dang-Vu, Scott M McKinney, Orfeu M Buxton, Jo M Solet, and Jeffrey M Ellenbogen. **Spontaneous brain rhythms predict sleep stability in the face of noise**. *Current Biology*, **20**(15):R626–R627, 2010. 87

[95] Sam Roweis and Zoubin Ghahramani. **A Unifying Review of Linear Gaussian Models**. *Neural Comput.*, **11**(2):305–345, 1999. 88, 112

[96] R.J. Patton and J. Chen. **Observer-based fault detection and isolation: Robustness and applications**. *Control Engineering Practice*, **5**(5):671 – 682, 1997. 93, 114

[97] Ron J. Patton and Jie Chen. **On eigenstructure assignment for robust fault diagnosis**. *International Journal of Robust and Nonlinear Control*, **10**(14):1193–1208, 2000. 93, 98, 114

[98] C.J. Chen and R.J. Patton. *Robust Model-Based Fault Diagnosis For Dynamic Systems*. Kluwer International Series on Asian Studies in Computer and Information Science, 3. Kluwer, 1999. 93, 114

[99] Angela L. DRozario, Jong Won. Kim, Keith K.H. Wong, Delwyn J. Bartlett, Nathaniel S. Marshall, Derk-Jan Dijk, Peter A. Robinson, and Ronald R. Grunstein. **A new {EEG} biomarker of neurobehavioural impairment and sleepiness in sleep apnea patients and controls during extended wakefulness**. *Clinical Neurophysiology*, **124**(8):1605 – 1614, 2013. 100, 108

[100] D.C. Montgomery. *Introduction to Statistical Quality Control*. Wiley, 2004. 102

[101] Laura B Ray, Stuart M Fogel, Carlyle T Smith, and Kevin R Peters. **Validating an automated sleep spindle detection algorithm using an individualized approach**. *Journal of sleep research*, **19**(2):374–378, 2010. 112

[102] Fazil Duman, Aykut Erdamar, Osman Erogul, Ziya Telatar, and Sinan Yetkin. **Efficient sleep spindle detection algorithm with decision tree**. *Expert Systems with Applications*, **36**(6):9980–9985, 2009. 112

[103] Eero Huupponen, Germán Gómez-Herrero, Antti Saastamoinen, Alpo Värri, Joel Hasan, and Sari-Leena Himanen. **Development and comparison of four sleep spindle detection methods**. *Artificial intelligence in medicine*, **40**(3):157–170, 2007. 112

[104] Nurettin Acir and Cüneyt Güzeliş. **Automatic recognition of sleep spindles in EEG via radial basis support vector machine based on a modified feature selection algorithm**. *Neural Computing & Applications*, **14**(1):56–65, 2005. 112

[105] Antoine Nonclercq, Charline Urbain, Denis Verheulpen, Christine Decaestecker, Patrick Van Bogaert, and Philippe Peigneux. **Sleep spindle detection through amplitude–frequency normal modelling**. *Journal of neuroscience methods*, **214**(2):192–203, 2013. 112

[106] O. Nelles. *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Engineering Online Library. Springer, 2001. 112, 123

[107] John H. Cochrane. *Asset Pricing*. Princeton University Press, 2001. 112

[108] Genshiro Kitagawa. **Non-Gaussian StateSpace Modeling of Nonstationary Time Series**. *Journal of the American statistical association*, **82**(400):1032–1041, 1987. 112

[109] A.N. Andry, E.Y. Shapiro, and J. C. Chung. **Eigenstructure Assignment for Linear Systems**. *Aerospace and Electronic Systems, IEEE Transactions on*, **AES-19**(5):711–729, 1983. 114

[110] R.J. Patton and J. Chen. **Robust fault detection using eigenstructure assignment: a tutorial consideration and some new results**. In *Decision and Control, 1991., Proceedings of the 30th IEEE Conference on*, pages 2242–2247 vol.3, 1991. 114

[111] J. Chen and R.J. Patton. **Optimal filtering and robust fault diagnosis of stochastic systems with unknown disturbances**. *Control Theory and Applications, IEE Proceedings -*, **143**(1):31–36, 1996. 114

[112] Rudolph Emil Kalman. **A new approach to linear filtering and prediction problems**. *Journal of basic Engineering*, **82**(1):35–45, 1960. 123

[113] Greg Welch and Gary Bishop. **An introduction to the Kalman filter**, 1995. 123