BasisDetect : A Model-based Network Event Detection Framework

Brian Eriksson
UW-Madison
bceriksson@wisc.edu

Paul Barford
UW-Madison
and Nemean Networks
pb@cs.wisc.edu

Rhys Bowden
University of Adelaide
rhysbowden@gmail.com

Nick Duffield
AT&T Research
duffield@research.att.com

Joel Sommers
Colgate University
jsommers@colgate.edu

Matthew Roughan
University of Adelaide
matthew.roughan@adelaide.edu.au

ABSTRACT

The ability to detect unexpected events in large networks can be a significant benefit to daily network operations. A great deal of work has been done over the past decade to develop effective anomaly detection tools, but they remain virtually unused in live network operations due to an unacceptably high false alarm rate. In this paper, we seek to improve the ability to accurately detect unexpected network events through the use of BasisDetect, a flexible but precise modeling framework. Using a small dataset with labeled anomalies, the BasisDetect framework allows us to define large classes of anomalies and detect them in different types of network data, both from single sources and from multiple, potentially diverse sources. Network anomaly signal characteristics are learned via a novel basis pursuit based methodology. We demonstrate the feasibility of our BasisDetect framework method and compare it to previous detection methods using a combination of synthetic and real-world data. In comparison with previous anomaly detection methods, our BasisDetect methodology results show a 50% reduction in the number of false alarms in a single node dataset, and over 65% reduction in false alarms for synthetic network-wide data.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations—*Network monitoring*

General Terms

Measurement, Performance

Keywords

Anomaly Detection

IMC'10, November 1–3, 2010, Melbourne, Australia.
Copyright 2010 ACM 978-1-4503-0057-5/10/11 ...$10.00.

1. INTRODUCTION

Networks are complex, dynamic and subject to external factors outside of their operators' control. Network operators must therefore vigilantly monitor their networks for faults and other events that could jeopardize their contractual commitments to customers. The problem is relatively easy when the type of fault is well understood (*e.g.,* link failures). There are standard protocols for alerting an operator to such faults, and although extending these methods particularly in the context of security is an ongoing effort, they are not the specific focus of this paper. Instead, this paper considers *unforeseen faults*. These faults are intrinsically more challenging to detect because we do not *a priori* know what we are looking for. These faults often manifest in unusual measurements that are commonly referred to as *anomalies*. Being able to find anomalies, and use them to diagnose network problems quickly and effectively would significantly enhance network operations. Developing a framework for effective and practical anomaly detection is the objective of our work.

A large number of studies over the past decade have been focused on developing methods to detect anomalous events in networks. The typical approach begins by measuring network traffic (*e.g.,* flow-export records) and then establishing a profile for "normal" behavior. Next, a method for detecting deviations from normality is applied. Most prior studies have largely taken a one-size-fits-all approach that has ultimately resulted in problems with accuracy and false alarm rate.

It is critically important in any anomaly detection system to have a very low false alarm rate. False alarms waste operator time and discredit results, leading to a "cry wolf" syndrome, where the anomaly detection system is quickly ignored. Most existing systems suffer from unduly high false-alarm rates. This is exacerbated by anomalies polluting the data used in determining the normal profile. In this paper, we seek to improve the accuracy of network event detection to the point where it becomes an effective tool for network operators.

To approach this problem of anomaly detection, we introduce the *BasisDetect* framework. The primary intuition behind the BasisDetect framework is that both normal traffic and anomalies have features that we can model and exploit for the purpose of automated detection. For instance, it is well known the traffic has strong diurnal and weekly cycles. Our hypothesis is that by considering traffic as a superposition of waveforms and then breaking these down into their component parts, we can build detection models that offer the opportunity to separate bundles of energy that can be semantically divided into to normal and anomalous traffic. The BasisDetect framework is divided into three components. The first step learns potential anomaly signal features from a small set of labeled network data provided to the algorithm. The second step uses a novel *basis pursuit* methodology to simultaneously decompose traffic into components of both non-anomalous behavior representing expected network traffic, and anomalous behavior learned from the previous step. This simultaneous estimation avoids the problem of anomalies polluting our normal profile data. The final step of the algorithm exploits known network structure to intelligently merge together the detected anomalous behavior using state-of-the-art statistical techniques.

Further objectives of our framework include developing an anomaly detection method that can be applied *(i)* to different data types since critical anomalies may be entirely invisible in some data, and *(ii)* in both a single node and network-wide context. Prior work has typically fallen into one or the other category due to detection methods that are primarily spatial or temporal. While our initial signal decomposition approach is temporal, we combine anomalies across the network using a *higher reasoning* framework. This combined, best-of-both-worlds approach offers a significant opportunity to improve detection accuracy. Intuitively, we treat network wide detection as a data fusion problem, where one can significantly reduce false alarms through the use of multiple time-series signals. It has the secondary advantage that it naturally incorporates different data types, without the need for strong relationships between the different time-series such as required, for example by PCA [1].

We use both synthetic and real world data to rigorously assess the capabilities of our model-based detection methodology. The first part of our evaluation considers NetFlow data collected at a single router along with a set of labeled anomalies that include DoS attacks, outages, scans, etc. We isolate a subset of the anomalies in the data and then apply the BasisDetect framework to learn anomalous models using a combination of signal components that isolate key elements of the events. We find that our BasisDetect methodology identifies all the labeled anomalies with 50% improvement in the false alarm rate when compared with the best competing methodology.

Next, we use a set of carefully generated synthetic data to assess the sensitivity of our model-based detection methodology. The data is designed to capture the key low and high frequency and spatial characteristics of non-anomalous traffic flows in a network-wide setting. We insert simple volume anomalies into this data and modulate the relative amplitude and frequency of these anomalies versus the non-anomalous traffic in order to assess sensitivity. While this synthetic data is not as rich as measurements collected in situ, we argue that it provides a powerful and meaningful starting point for assessing detection sensitivity. The results of our analysis show that the BasisDetect methodology detects all of the injected anomalies with false alarm rate over 65% less than the current state-of-the-art network-wide anomaly detection methodology.

Finally, we consider a set of Internet2 byte count data collected simultaneously across 11 PoPs. While this dataset does not have labeled anomalous events, we can compare the ability of the BasisDetect methodology and a state-of-the-art distributed method [2] to detect the most dominant anomalies detected by the standard PCA [1] anomaly detection methodology. Our results show that BasisDetect method will identify the PCA anomaly locations with 40% fewer false alarms than the competing state-of-the-art network-wide anomaly detection method. We believe that these results along with the results from single node and network-wide labeled data sets make a strong case for the utility of our model-based approach.

The remainder of this paper is as follows. In Section 2, we describe the background of detecting network anomalies and relevant related work. In Section 3, we describe the datasets used to test our method and competing anomaly detection algorithms. In Section 4 the BasisDetect framework is introduced. Then in Section 5, our temporal signal decomposition methodology is described with applications of anomaly extraction from a small training set, and anomaly detection. Then in Section 6, an intelligent data fusion methodology is described to localize anomalies given the results of our basis pursuit algorithm. Combining both methods, we summarize the BasisDetect methodology in Section 7. Finally, in Section 8 we evaluate the results of applying our method and several other well known methods to the given data sets. We summarize our work and discuss future directions in Section 9.

2. BACKGROUND AND RELATED WORK

Anomaly detection is now a large field, and we cannot hope to survey all papers within the field. We will focus on those of direct relevance to our work and describe them in detail since they help to highlight the uniqueness and potential benefits of our method. Our specific focus is on studies that consider anomaly detection on vector time-series data. This type of work gained its initial impetus with the consideration of how Principle Component Analysis (PCA) would perform in a network-wide setting [1, 3, 4]. Prior work relied primarily on performing some kind of temporal transform of the data, and assumed that anomalies will stand out against the traffic in the transformed space (examples transforms include wavelets in [5, 6], the Exponentially Weighted Moving Average or EWMA in [7, 8], and Fourier filtering in [6]). Anomalies are then generated for every individual set of measurements. The key benefits of the PCA methodology was that it took direct advantage of the non-scalar nature of network data, and that it sought to find an optimal linear transform of the data in order to reveal inconsistent data points. Following the initial work on PCA, Zhang *et al.* showed how much of the prior work on anomaly detection (including PCA) could be seen in a single framework [6], but more notably, that paper showed that the "sparseness" of anomalies could be exploited in aiding their discovery.

In more detail, the PCA framework described in [1] decomposes a traffic matrix into a set of vector components that capture the variance across all links or flows of the network. The components that resolve the highest variance across all links (*e.g.,* the most standard components) are considered to represent standard operating characteristics of the network observed in the link data matrix, the *"modeled traffic"*. Meanwhile, the less dominant components represent the *"residual traffic"* that is abnormal to the links in general. The amount of traffic energy in this residual component determines whether or not an anomaly has occurred in the observed traffic on each link.

The limitations of this PCA approach are well documented in [9]. In addition to having high sensitivity to tuning parameters, large anomalies in the network can corrupt the "modeled traffic" components and therefore cause obvious events to be ignored by the methodology. Also, detected anomalies found by PCA can not be localized to the specific anomalous link or router, and the PCA methodology can lead to *masking*, where one anomaly hides another. Finally, in PCA the "residual traffic" does not necessarily represent signal components specifying anomalies (possibly it is normal behavior found only on a single link), and therefore detecting events based on residual energy is prone to false alarms. Furthermore, the work in [10] shows how standard PCA-based anomaly detection methods are vulnerable to attacks. While the technique introduced in this paper builds on the idea of dividing the signal into modeled and residual components, the underlying methodologies used are completely different (PCA in the prior work, basis pursuit in this paper).

The authors of the Distributed Spatial Anomaly Detection technique described in [2] recognize that one of the main limitations of the PCA approach was the necessity of communicating all flow information back to some centralized computation point. Using non-parametric statistics and False Discovery Rate techniques (FDR) [11], each router in the network generates just a small test statistic that is communicated for anomaly detection. The use of more sophisticated multiple hypothesis detection techniques, like FDR thresholding, allows for a better statistical detection rate than more naive methodologies, such as Bonferroni Correction [12]. The biggest limitation of this approach is the complete decoupling of the measurements in the time domain. Therefore, any temporal correlation between network anomaly events (the measurements at time t helping inform the events from measurements at $t+1$) are ignored. In addition, the measurements considered are with respect to traffic volume only, with no discussion on how other link characteristic information (bytes, unique IP address, etc.) could be intelligently fused into the framework. Finally, the detected anomalies are not necessarily points of interest to a network administrator or anything that might represent the known structure of anomalies in networks. These detected anomalies are simply events of traffic volume that are abnormal compared with the remaining observed set of network data. A situation may occur when events are unlike the other observed network data and yet uninteresting from a network administration prospective. Other distributed approaches to anomaly detection exist [13, 14]. Although it should be noted that the BasisDetect framework is amenable to distribution, the focus of this paper will be to carefully treat the false alarm problem.

Our anomaly detection methodology will exploit the same non-parametric statistical techniques as [2] (originally developed in [15]). However, our methodology differs in that we use an estimated feature vector of detected anomaly energy instead of the raw packet counts. Data fusion from different data sources was shown some time ago to reduce false alarm rates (*e.g.,* [7]). In contrast, this paper develops an approach which can flexibly incorporate various different sources of data. By considering a general feature vector, we can potentially fuse a wide range of link characteristics, thereby improving results.

For the detection of anomalies in time-series data, our methodology will leverage the significant prior work on basis decomposition of signals [16, 17, 18]. This prior work focused on creating methodologies to exactly represent a signal given a sparse linear combination of components from the signal dictionary (*i.e.,* a matrix of signal components). In this paper, our goal is to resolve the gross characteristics of the signal, allowing for non-exact signal representation by our basis dictionary signals. In addition, our novel methodology will allow for the penalization of choosing selected dictionary signals, an application previously unexplored in the basis pursuit literature.

3. DATASETS

We use three different data sets to evaluate our model-based detection methodology. The intent of our analysis is to assess the capability of our approach as thoroughly as possible. To that end, we use empirical data sets for both single node with labeled anomalies and network-wide settings without labeled anomalies. We also use a synthetic data set in which we can precisely control both the normal and anomalous traffic in order to carefully assess the sensitivity of our method. Each of the data sets is described in detail below.

3.1 Synthetic Traffic Data

In order to accurately test anomaly detection algorithms, we need to be able to simulate reasonable datasets in a controlled way. Ringberg *et al.* [19] explain in detail why simulation must be used for accurate comparisons of anomaly detection techniques. In brief the reasons are: (i) accurate and complete ground truth information is needed to form both false-alarm and detection probability estimates; (ii) many more results are needed (than one can obtain from any realistic real dataset) to form accurate estimates of probabilities, and (iii) simulation allows one to vary parameters (say the anomaly size) in a controlled way in order to see the effect this has on anomaly detection.

Our approach to simulation is intended to highlight the features of the different techniques. We make no claim that the simulation is completely realistic, only that it illustrates clearly the properties of the different anomaly detection techniques. The simulations used here were generated in a similar manner to those in [20]. In particular, a spatial traffic matrix is generated using a gravity model and then extended into the temporal domain using a matrix product with a simple periodic signal. The resulting traffic is then enhanced by Gaussian noise with variance that is proportional to the traffic mean. The only differences with the previous study are that (i) we consider a range of sizes of networks, and (ii) consider a range of length of anomalies.

We should stress that the goal of these simulations is not to produce the most realistic test possible for the algorithms. However, the simulations allow us to obtain exact quantitative comparisons of algorithms in completely controlled circumstances, so we can explore the properties of the different approaches.

3.2 GEANT Data

The second set of data will be a collection of time-series data obtained from a GEANT network backbone router [21] located in Vienna, Austria. Collection of data began on January 14th 2009 and ended on February 24th 2009, for a total of 42 days of data acquisition. The dataset contains packet counts, byte counts, and IP entropy measured along this single link extracted using Juniper J-Flow records, sampled in aggregation bins of 1 minute for a total of 60,480 data samples observed. This dataset contains labeled anomalies, including Denial of Service (DoS) attacks, portscan events, and Distributed Denial of Service (dDoS) attacks. These events were found, validated, and annotated by network engineers.[1]

A limitation of this single node time-series is that it cannot show the power of strictly network-wide techniques (such as PCA or Distributed Spatial). Although we are restricted in the comparison methodologies available for this dataset, the single link information has the advantage that a great deal of effort has gone into classifying the anomalies in this data, so that we are closer to having ground truth than we are in any almost any other setting.

3.3 Abliene Real-World Data

The final set of data consists of byte counts recorded from the Abliene Internet2 backbone network.[2] Across 11 PoPs in the continental United States with 41 network links, byte counts were sampled into 10 minute time intervals from April 7th 2003 to April 13th 2003, resulting in 1008 byte count samples across each of the 41 links. Unfortunately, this dataset is completely unlabeled with no prior annotation of possible anomaly locations. To compensate for this deficiency in the dataset, we will use this real world network data to study how the new BasisDetect framework detects anomalies that are found by previous network-wide anomaly detection algorithms.

4. BASISDETECT OVERVIEW

Our automated BasisDetect framework for detecting network anomalies is divided into three distinct components. Practically speaking, these components are predicated on having a small initial set of network data with labeled anomalies from which event characteristics can be learned and the algorithm parameters are optimized against. The components of the BasisDetect framework are:

1. *Anomalous Dictionary Construction from Labeled Set* - Using a training set of labeled anomalies, we extract signal characteristics that have been pre-established as anomalous.

2. *Anomaly Decomposition using Penalized Basis Pursuit* - Using our novel Penalized Basis Pursuit methodology and the learned anomaly dictionary signals from the previous step, the BasisDetect methodology extracts anomaly energy from temporal network data for each data signal observed in the network.

3. *Network-wide Data Fusion* - Using knowledge of the network topology structure and the estimated anomaly energy for each link, our methodology classifies anomalous behavior at each router.

A visual description of the BasisDetect framework can be seen in Figure 1.

5. BASIS DECOMPOSITION OF NETWORK DATA

To begin, we establish an anomaly detection methodology on a single observed time-series signal (denoted $\mathbf{y}$). To detect anomalies on this signal, consider decomposing the signal into its anomalous and non-anomalous components. In contrast to previous methods, like PCA [1], we will not consider other concurrently observed signals as potentially non-anomalous behavior. Instead, we will decompose this signal by specifying characteristics that represent both anomalous and non-anomalous behavior for that link with respect to an established set of signal components. This avoids the drawback of PCA-related methods where anomalies pollute the representations of non-anomalous behavior.

In order to perform this decomposition, we introduce the idea of a signal *dictionary*, Φ, a matrix of signal components that will represent our observed data. The signal dictionary considered here will contain both anomalous and non-anomalous signal components,

$$\Phi = \left[\begin{array}{cc} \Phi_{non-anomaly} & \Phi_{anomaly} \end{array} \right] \qquad (1)$$

While these anomalous dictionary components will not be known a priori, to begin we will assume they are known (with later discussion in Section 5.1 describing methodologies to extract these signals). We state that the observed traffic signal can be approximated by a linear combination of dictionary components. Therefore, the observed traffic signal can be stated as,

$$\begin{aligned} \mathbf{y} &\approx \Phi\mathbf{x} \qquad (2) \\ &= \left[\begin{array}{cc} \Phi_{non-anomaly} & \Phi_{anomaly} \end{array} \right] \left[\begin{array}{c} \mathbf{x}_{non-anomaly} \\ \mathbf{x}_{anomaly} \end{array} \right] \end{aligned}$$

Where the coefficient $x_i \in \mathbf{x}$ is the contribution of dictionary element $\phi_i \in \Phi$ to the observed signal $\mathbf{y}$.

The amount of anomalous energy in the signal as a function of time is defined as the anomaly feature vector,

$$\mathbf{y}_{anomaly} = \Phi_{anomaly}\mathbf{x}_{anomaly} \qquad (3)$$

It should be intuitive that if more coefficient energy is placed in the anomaly dictionary, then the more likely an anomaly has occurred. If there is little energy in the anomaly domain, then the non-anomalous, standard operating environment signal components are accurately approximating the signal, and therefore an anomaly is unlikely to have occurred. To discover this level of anomalous energy for each signal, it is necessary to resolve the unknown coefficient vector $\mathbf{x}$.

Given a dictionary of signal components Φ and the observed signal $\mathbf{y}$, we must determine which components are used to represent our observed flow record, specified by the coefficients in the vector $\mathbf{x}$ (such that $\mathbf{y} \approx \Phi\mathbf{x}$). In addition to representing the signal, we wish to also restrict the coefficient vector $\mathbf{x}$ to be sparse. This sparsity constraint will require as few dictionary elements as possible be used to represent the observed signal.

[1] We thank Fernando Silveira from Technicolor Research for supplying us with this dataset.

[2] We thank Mark Crovella for supplying us with this dataset.

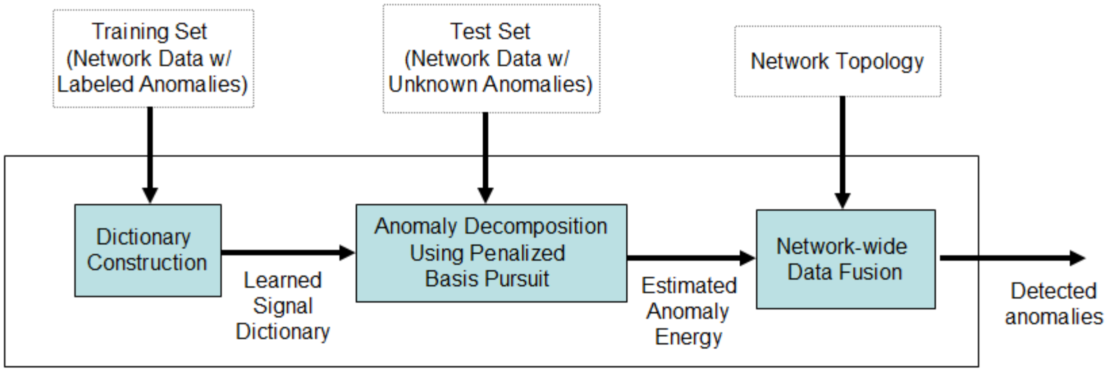
Training Set (Network Data w/ Labeled Anomalies)
Test Set (Network Data w/ Unknown Anomalies)
Network Topology
Dictionary Construction
Learned Signal Dictionary
Anomaly Decomposition Using Penalized Basis Pursuit
Estimated Anomaly Energy
Network-wide Data Fusion
Detected anomalies

Figure 1: The BasisDetect Framework

The motivation for sparsity in the signal comes from two directions. First, as noted in [6], anomalies are sparse in a signal. Similar motivation comes from studies of the underlying causes of faults in networks [22, 23]. Motivation for sparsity of the normal traffic comes from simple consideration of Fourier analysis of traffic. Figure 2 shows a simple example of the excellent degree of approximation to traffic we can obtain using only a very small number of Fourier coefficients corresponding to daily periods. The figure shows the important components of the power-spectrum of one week of GEANT data, clearly highlighting the importance of the daily cycles. The approximation curve in Figure 2-(right) is generated using only the largest 30 terms from the Discrete Cosine Transform (DCT, which contains around 10,080 coefficients in total). The remaining coefficients of the power-spectrum (including those not shown in the figure) contain little power, so modeling the few critical components will enable us to obtain a reasonable initial model for the data.

While standard inverse problem techniques (linear least squares, etc.) could be used to resolve the coefficient vector, these standard techniques do not require the resulting coefficient vector to be sparse. This sparsity constraint is considered in the theory of *Basis Pursuit* [16], which defines the sparse optimization problem,

$$\min \|\mathbf{x}\|_0 \quad \text{such that} \quad \mathbf{y} = \Phi\mathbf{x} \tag{4}$$

Where we try to find the set of coefficients $\mathbf{x}$ such that the observed vector is reconstructed using the fewest number of dictionary components possible (where $\|\mathbf{x}\|_0 = \#$ of non-zero components in $\mathbf{x}$).

Unfortunately, solving the optimization problem in Equation 4 is combinatorial and therefore computationally intractable for signals of any practical size. In [16] it was shown that this optimization is equivalent to the ℓ_1 relaxation,

$$\min \|\mathbf{x}\|_1 \quad \text{such that} \quad \mathbf{y} = \Phi\mathbf{x} \tag{5}$$

(Where $\|\mathbf{x}\|_1 = \sum_{i=1}^{N} |x_i|$.) As a result of this relaxation, the problem can be restated as a simple linear optimization problem.

While this may be the best approach when we are trying to exactly reconstruct the observed signal, however, in the case of our anomaly detection problem, we are not interested in reconstructing every perturbation of the signal. The problem becomes how to approximate the general characteristics of the current system behavior. Considering the case where we wish to approximate the observed signal using relatively few elements of a signal dictionary, Orthogonal Matching Pursuit (OMP) [17] will offer a simple greedy approximation of the dictionary coefficients.

The Orthogonal Matching Pursuit algorithm starts with an all-zeros signal approximation ($\widehat{\mathbf{y}} = [0, 0, ..., 0]$), the signal component dictionary (Φ), and a null estimated dictionary space ($\widehat{\Phi} = []$). At each iteration of the OMP algorithm the residual signal ($\mathbf{r} = \mathbf{y} - \widehat{\mathbf{y}}$, the observed data minus the current approximation) is calculated. This residual can be considered the signal that is *orthogonal* to the current estimated dictionary space, $\widehat{\Phi}$. The best dictionary component signal, ϕ, not currently in the estimated dictionary space ($\phi \in \Phi$ and $\phi \notin \widehat{\Phi}$) is found. This component is then added to the current estimated dictionary space, $\widehat{\Phi}$. Finally, the best signal approximation is found given the specified current dictionary space, $\widehat{\mathbf{y}} = \widehat{\Phi}\mathbf{x}$ (for some vector $\mathbf{x}$). The process is repeated until either a specified number of components are found or the error of the signal approximation is below some threshold. In addition to the non-exact approximation of the signal, due to the greedy approach, the algorithm is significantly faster than the standard Basis Pursuit algorithm and offers considerable memory savings.

Using the orthogonal matching pursuit (OMP) algorithm in conjunction with a signal dictionary derived from the Discrete Cosine Transform (DCT), consider the decomposition of a vector of packet counts in Figure 2-(Left). The performance of describing the flow signal with this reconstructed signal is shown in Figure 3 for 24 hours of observed of packet counts on the GEANT network. Note that by representing only the gross characteristics of the signal, the anomalous parts of the signal are becoming apparent in the residual between the observed data and the approximated signal.

5.1 Anomalous Dictionary Construction from Labeled Set

While the Orthogonal Matching Pursuit methodology will find the coefficient vector $\mathbf{x}$, it requires knowledge of the

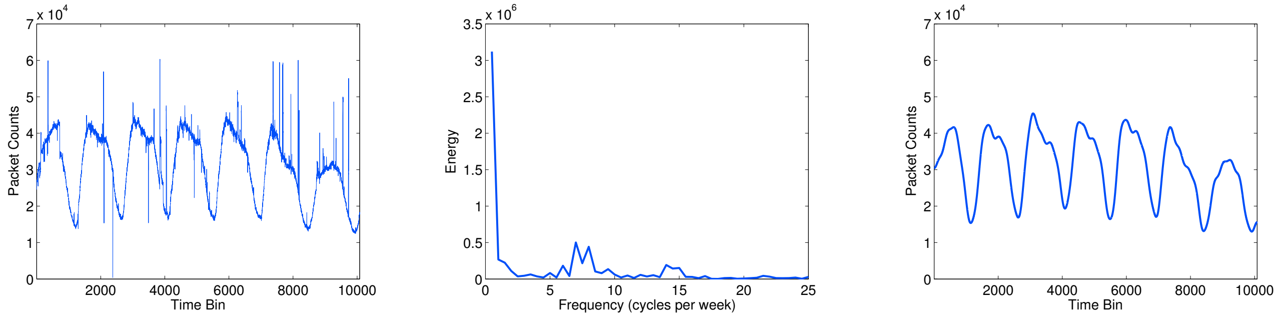

Figure 2: Fourier analysis of GEANT data (Left) - Observed one week of packet counts across a single link in the GEANT network. (Center) - Important region of Fourier power spectrum found using a Discrete Cosine Transformation (DCT). (Right) - Signal approximation using 30 largest DCT coefficients.

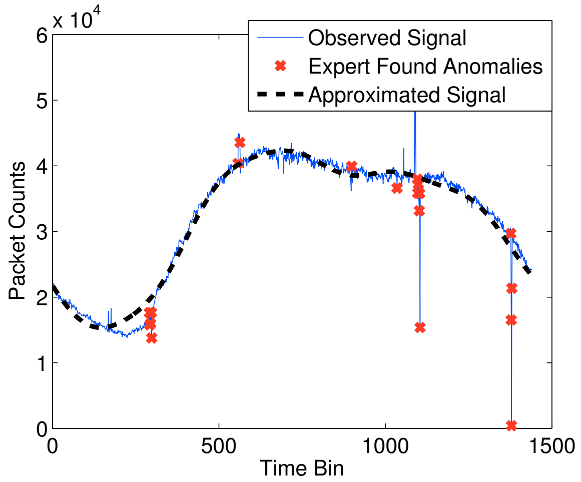

Figure 3: Approximation of 24 hours of GEANT data - Comparison of the observed signal with an approximation using the 30 largest Discrete Cosine Transform (DCT) coefficients. Expert annotated anomalies are marked with 'x'.

anomalous signal dictionary $\Phi_{anomaly}$, which is not known a priori. Through examining Figure 3, one can see that while the rough approximation from the matching pursuit algorithm is fitting most of the signal, the anomalous portions of the signal are not well represented by this simple approximation. If we knew for a subset of network data where anomalies were temporally located and had simple approximations of the signal based on expected signal behavior, we could extract examples of anomalous signal characteristics. Using this intuition and a small training set of labeled anomalies, we can estimated the anomaly dictionary $\widehat{\Phi}_{anomaly}$ and construct the complete signal dictionary $\Phi = \left[\ \Phi_{non-anomaly} \quad \widehat{\Phi}_{anomaly} \ \right]$.

First, consider knowledge of the non-anomalous signal dictionary ($\Phi_{non-anomaly}$). Given the Fourier decomposition example from Figure 2, the most obvious non-anomalous signal type to represent the network data would be a set of sinusoids. This signal type can be created via a Discrete Cosine Transformation (DCT). Second, consider local variation that may not be represented by the global sinusoidal wave based representation of the Discrete Cosine Transform. Due to the need to represent non-anomalous localized variation characteristics in the network data, we will also add a discrete wavelet transform filter set to our non-anomalous signal dictionary (with motivation discussed in Section 8.2 as to the exact type of wavelet decomposition considered). Note that while the DCT/wavelet basis will be used as the non-anomalous signal dictionary in this paper, the BasisDetect framework is agnostic to the choice of non-anomalous signals and can be designed to operate with any chosen basis. We leave discovery of the optimal set of non-anomalous signal components as future work.

Given the constructed non-anomalous signal dictionary, we finally look to determine the anomalous signal dictionary $\Phi_{anomaly}$. Consider a single time-series signal $\mathbf{y}$ with known anomaly locations. In order to isolate anomalous signal characteristics, we first obtain a signal approximation $\widehat{\mathbf{y}}$ using Orthogonal Matching Pursuit, the observed signal $\mathbf{y}$, and the non-anomalous signal dictionary $\Phi_{non-anomaly}$. By examining the residual signal $\mathbf{r} = \widehat{\mathbf{y}} - \mathbf{y}$ (the difference between the approximated signal and the observed signal), we can see where the non-anomalous signal characteristics fail at representing the observed network data. By windowing the residual signal around areas of known anomalies, we can extract anomalous signal characteristics from the training set. A step-by-step description of this methodology can be seen in Algorithm 1.

5.2 Anomaly Decomposition using Penalized Basis Pursuit

In standard OMP all dictionary component signals are weighted equally, therefore there is no preference towards choosing one dictionary signal or another (with the exception of the contribution towards describing the original observed signal). In our anomaly detection problem, specific dictionary component signals may be more preferential than others. Generally, we want to use an anomaly dictionary signal component (and thereby classify that area of the observed data as anomalous) only if that anomaly signal is the sole dictionary component that can properly decompose that area of the signal. Therefore, we want to penalize choosing an anomaly dictionary element. This changes the OMP algorithm to a modified Penalized Basis Pursuit approach by choosing the next element in the signal dictionary $\phi_{\widehat{i}}$ as,

Algorithm 1 - Dictionary Construction Algorithm

```
Given:
  • $\mathbf{Y} = \{\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_N}\}$ - Training set of network time-series data
  • $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_N\}$ - Index of known anomalies in each of the training set signals.
  • $\Phi_{non-anomaly}$, Non-anomalous signal dictionary
  • $w$, anomaly window size
Main Body
  • Set $\widehat{\Phi}_{anomaly} = []$, the estimated anomalous signal dictionary
  • For each network time-series signal, $\mathbf{y_i}$
    1. Obtain signal estimate $\widehat{\mathbf{y_i}}$ by using Orthogonal Matching Pursuit [17] with respect to time-series signal $\mathbf{y_i}$ and signal dictionary $\Phi_{non-anomaly}$.
    2. Find residual signal, $\mathbf{r_i} = \widehat{\mathbf{y_i}} - \mathbf{y_i}$.
    3. For each anomaly in the current signal, $j \in \mathcal{I}_i$
      (a) Set a new anomalous vector to the windowed component of the residual signal at the known anomaly, $\mathbf{a_j} = \left[\begin{array}{cccc} r_i(j-w) & r_i(j-w+1) & ... & r_i(j+w) \end{array}\right]$
      (b) Add $\mathbf{a_j}$ to anomalous signal dictionary $\widehat{\Phi}_{anomaly}$
Return:
  • Return $\widehat{\Phi}_{anomaly}$
```

$$\widehat{i} = \arg \max_{i=\{1,2,...\}} \left(|\langle \phi_i, \mathbf{r} \rangle| - \lambda_i \right) \tag{6}$$

Where $\mathbf{r} = \widehat{\mathbf{y}} - \mathbf{y}$ the current residual signal with respect to the current signal dictionary, and defining the penalty vector $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_X\}$ (for a dictionary of X number of signal components), such that

$$\lambda_i = \begin{cases} \gamma & \text{if } \phi_i \in \widehat{\Phi}_{anomaly} \\ 0 & \text{if } \phi_i \notin \widehat{\Phi}_{anomaly} \end{cases} \tag{7}$$

For a large enough $\gamma > 0$, this prevents the algorithm from over-representing the signal from the anomalous dictionary. The full methodology is described in Algorithm 2.

Finally, using the Penalized Basis Pursuit methodology, we obtain the anomaly feature vector for the observed signal using the anomalous chosen dictionary signal components, $\mathbf{y}_{anomaly} = \widehat{\Phi}_{anomaly}\widehat{\mathbf{x}}_{anomaly}$. While we expect that a majority of anomalies will be detected using the anomaly dictionary representation, we also want to avoid the situation where limited training set anomalies result in missing true anomalies. To avoid missing anomalies, we also incorporate knowledge of the residual signal $\mathbf{r}$ into the anomaly energy.

Algorithm 2 - Penalized Basis Pursuit Algorithm

```
Given:
  • $\mathbf{y}$ = observed $N$-length network data vector
  • $\Phi = \left[\begin{array}{cc} \Phi_{non-anomaly} & \Phi_{anomaly} \end{array}\right]$, the signal dictionary matrix
  • $\gamma$, penalty for representing the signal using an anomalous signal component.
  • $N_{coef}$, the specified number of coefficients used to represent the signal $\mathbf{y}$.
  • $\rho$, the weight of the residual signal in the final anomaly energy output.
Main Body
  1. Construct $\mathbf{\Lambda} = \{\lambda_1, \lambda_2, ..., \lambda_M\}$, the vector of penalty terms for each dictionary signal component using Equation 7.
  2. Set $\widehat{\Phi} = []$, matrix of chosen dictionary signal components
  3. Set $\mathbf{r} = \mathbf{y}$, current residual signal given chosen dictionary
  4. For $\ell = \{1, 2, ..., N_{coef}\}$.
    (a) Find the most dominant dictionary component not yet considered,
          $k = \arg \max_{i=\{1,2,...\}} \left( |\langle \phi_i, \mathbf{r} \rangle| - \lambda_i \right)$
    (b) Remove $\phi_k$ from $\Phi$, $\lambda_k$ from $\Lambda$.
    (c) Add $\phi_k$ to $\widehat{\Phi}$
    (d) Find the dictionary coefficient for the current chosen dictionary signal components. Solve the least squares problem, setting $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\widehat{\Phi}\mathbf{x} - \mathbf{y}\|_2$
    (e) Recalculate the residual signal : $\mathbf{r} = \mathbf{y} - \widehat{\Phi}\widehat{\mathbf{x}}$
Return:
  • Return the estimated anomaly energy, $\widehat{\mathbf{y}}_{energy} = \widehat{\Phi}_{anomaly}\widehat{\mathbf{x}}_{anomaly} + \rho\mathbf{r}$
```

$$\widehat{\mathbf{y}}_{energy} = \widehat{\Phi}_{anomaly}\widehat{\mathbf{x}}_{anomaly} + \rho\mathbf{r} \tag{8}$$

Where $\rho \in [0, 1]$.

6. NETWORK-WIDE DATA FUSION

While the penalized basis pursuit methodology will extract anomaly energy for a single time-series signal, we must add to our framework to detect network-wide anomalies. Given a network with known topology, containing N routers and M links, consider multiple observed time-series characteristics for each link (represented here by $\{c_1, c_2, ..., c_C\}$ for packet count, byte count, etc.). We observe time-series signal $\mathbf{y}^{(\ell,c)}$ for the data at link ℓ and for characteristic c.

After performing the Penalized Basis Pursuit Algorithm on the observed time-series network signal, $\mathbf{y}^{(\ell,c)}$, we obtain the estimated signal energy relating to the anomaly energy from Equation 8, denoted by $\widehat{\mathbf{y}}^{(\ell,c)}_{energy}$. Therefore, for each link ℓ, we would have an estimated anomaly energy vector for each observed characteristic of the link. For example,

$$\widehat{\mathbf{Y}}^{(\ell)}_{energy} = \left[\begin{array}{cccc} | & | & | & | \\ \widehat{\mathbf{y}}^{(\ell,c_1)}_{energy} & \widehat{\mathbf{y}}^{(\ell,c_2)}_{energy} & ... & \widehat{\mathbf{y}}^{(\ell,c_C)}_{energy} \\ | & | & | & | \end{array} \right] \quad (9)$$

One could imagine aggregating the anomaly energy across all characteristics on this link and performing a simple energy thresholding mechanism for determining anomalies. One problem with this methodology is that it would ignore the spatial correlation between links. Given a router with links ℓ_A and ℓ_B, the appearance of anomaly energy on both links (at the same time) will strengthen our confidence that an anomaly has actually occurred. Meanwhile, the lack of anomaly energy on one of the links should weaken our belief in an abnormal event occurring at that time period.

Motivated by the work on distributed anomaly detection in [2], consider the multidimensional space where each dimension represents a link connected to the router. For a router r, with m links $\{\ell_1, \ell_2, ..., \ell_m\}$, and time-series observations at T time steps, we can form the $T \times Cm$ sized estimated router anomaly energy matrix,

$$\widehat{\mathbf{Y}}^{(r)}_{energy} = \left[\begin{array}{cccc} \widehat{\mathbf{Y}}^{(\ell_1)}_{energy} & \widehat{\mathbf{Y}}^{(\ell_2)}_{energy} & ... & \widehat{\mathbf{Y}}^{(\ell_m)}_{energy} \end{array} \right] \quad (10)$$

Where row t' represents a specific time bin with the estimated anomaly energy on each of the m links connected to the router with respect to each observed link characteristic (*e.g.,* packet count, byte count, etc.). This Cm-length estimated anomaly energy vector, $\widehat{\mathbf{Y}}^{(r)}_{energy}(t')$ could be considered a point in some $\mathbb{R}^{Cm}$ estimated anomaly energy feature space. The anomaly energy vector's placement in this Cm-dimensional space will inform us whether or not an anomaly actually has occurred at router r at time t'. Intuitively, if every link and observed characteristic has very little anomaly energy estimated, there is likely not an anomaly at this router. Conversely, if several observed characteristics of this router have very large estimated anomaly energy values, then it is likely some anomaly is occurring at this time step. This reduces to a hypothesis testing problem of detecting whether or not the estimated anomaly energy vector $\widehat{\mathbf{Y}}^{(r)}_{energy}(t')$ is anomalous compared with the other estimated anomaly energy vectors.

Using the matrix of estimated anomaly energy for router r, $\widehat{\mathbf{Y}}^{(r)}_{energy}$, we can use the minimum volume level set methodology of [15] to assess which rows of the matrix are anomalous[3]. This methodology uses a nonparametric statistical technique to output the False Discovery Rate (FDR) of each vector, $p_{r,t}$, for each vector $\widehat{\mathbf{Y}}^{(r)}_{energy}(t)$. This False Discovery Rate value is the probability of observing a vector of estimated anomaly energy more extreme given the remaining matrix of estimated energy vectors. If the probability is very low that a more extreme vector would be observed, this vector is likely anomalous. The False Discovery Rate has been found [11] to be a more accurate metric than standard multiple hypothesis testing techniques (*e.g.,* Bonferroni Correction [12]). By thresholding based on these FDR $p_{r,t}$ values, we classify which entries are anomalous, and therefore where anomalies occur in the network, localized by both the specific router and the specific time of the anomaly.

7. BASISDETECT ALGORITHM

Combining the novel basis pursuit methodology from Section 5 and the nonparametric statistical methodology described in Section 6, we can summarize our full BasisDetect framework in Algorithm 3. The non-anomalous signal dictionary, $\Phi_{non-anomaly}$, will be taken as the collection of waveforms from a combination of the Discrete Cosine Transform (DCT) and a Discrete Wavelet Transform using the discrete Meyer wavelet (as motivated in Section 8.2). Additionally, the BasisDetect methodology requires the detection parameter, ν, which determines the detection/false alarm rate of the anomaly detection. Throughout the experiments, this parameter is adjusted to present the spectrum of detection/false alarm rate for our BasisDetect methodology. Finally, the use of Algorithm 1 requires the tuning parameter w to adjust the time-series window for extracting the learned anomaly characteristics. While examination of our labeled anomalies resulted in setting this parameter to $w = 5$ (for consideration of 10 time bins around the labeled anomalies), this value will depend on link sampling rates and requires careful consideration through inspection of the training data.

The reliance on the Penalized Basis Pursuit Methodology of Algorithm 2 requires the additional input of three tuning parameters (N_{coef}, γ, ρ) into the BasisDetect algorithm. In order to find the optimal parameter values with respect to the number of false alarms declared, consider some initial estimate of these three parameters and running the BasisDetect algorithm using the training data as the test data. By using the training data as test, we are given a priori knowledge of where the anomalies are located and the performance of the BasisDetect methodology (in terms of the number of false alarms declared) with respect to the given input tuning parameters. Using a grid search of parameter values over feasible possible values, we can optimize the choice of tuning parameter values by choosing the set of parameter values (N_{coef}, γ, ρ) that minimize the total number of false alarms declared to detect all the anomalies in the training data.

8. RESULTS

We perform an extensive set of tests using three different data sources, which were described in Section 3. The intent of our experiments is to assess the capabilities of our model-based detection methods in both a single node and network-wide setting. We also compare and contrast our method with standard detection methods that have been described in prior studies. Finally, we assess the sensitivity of our detection method using synthetic traffic traces in which ground truth is intrinsic. For all experiments described, the tuning parameters required by the BasisDetect methodology and the training anomalies are discovered by hold out cross validation [24], where 20% of the data is held out as training data while the remaining 80% is used as test data. The results of our experiments are described below.

[3] We thank the authors of this code for making the program readily available at http://www.eecs.umich.edu/∼cscott/code/mnscann.zip

Algorithm 3 - BasisDetect Algorithm

Given:

- $\mathbf{y}^{(\ell,c)}$ = the observed link data from link ℓ and link characteristic c. Known for all network links $\ell = \{1, 2, ..., M\}$ and link characteristics $c = \{1, 2, ..., C\}$.
- Set of routers $r = \{1, 2, ..., N\}$.
- Router-level topology of the network.
- Training set of network data with labeled anomalies
- Test set of network data with unknown anomalies
- ν = detection threshold for anomaly FDR values
- $\Phi_{non-anomaly}$ = dictionary of non-anomalous signal components
- γ = penalty for representing the signal using an anomalous signal component.
- N_{coef} = the specified number of coefficients used to represent the signal $\mathbf{y}$.
- ρ = the weight of the residual signal in the final anomaly energy output.

Main Body

1. Using the training set with labeled anomalies, apply the Dictionary Construction approach in Algorithm 1 to learn the anomalous dictionary components, $\widehat{\Phi}_{anomaly}$. Construct the full dictionary array $\widehat{\Phi} = \left[\begin{array}{cc} \Phi_{non-anomaly} & \widehat{\Phi}_{anomaly} \end{array} \right]$.
2. Perform Penalized Basis Pursuit Method from Algorithm 2 to estimate the anomalous signal energy $\widehat{\mathbf{y}}^{(l,c)}_{energy}$ for each link $l = \{1, 2, ..., M\}$ and link characteristic $c = \{1, 2, ..., C\}$ using the learned dictionary $\widehat{\Phi}$.
3. Using knowledge of the network topology, construct the router anomaly energy matrix, $\widehat{\mathbf{Y}}^{(r)}_{energy}$ for each router $r = \{1, 2, .., N\}$ using Equation 10
4. Apply the nonparametric technique from [15], finding the False Discovery Rate value, $p_{r,t}$ for each router $r = \{1, 2, .., N\}$ at time index $t = \{1, 2, ..., T\}$.
5. For each element of $p_{r,t} < \nu$, label the router r as having an anomaly at time t.

8.1 GEANT Time-series Network Data

The first experiment will be on a collection of time-series data obtained from a Juniper J-Flow records from a GEANT router as described in Section 3.2. The dataset contains packet counts, byte counts, and IP entropy measured along the single link, sampled in aggregation bins of 1 minute evaluated for 42 days, resulting in a time-series signal of length 60,480. This dataset contains labeled anomalies (DoS, dDoS, portscan) found by network engineers.

In addition to our new BasisDetect methodology, we will compare against two time-series based forecasting techniques. Due to the availability of only single link data, the network-wide approaches of PCA and the Distributed Spatial methodology are not applicable here. Instead we will focus on two basic time-series anomaly detection methodologies, the Exponentially Weighted Moving Average (EWMA) Filter and Fourier thresholding.

8.1.1 EWMA Filter

The EWMA filter is a simple smoothing methodology that uses the previous observed values ($\{y_{t-1}, y_{t-2}, ...\}$) to forecast what the next observed value (y_t) should be.

$$\widehat{y}_{t+1} = \alpha_{ewma} y_t + (1 - \alpha_{ewma})\, \widehat{y}_t \tag{11}$$

An anomaly is detected if the forecasted value deviates significantly compared with the observed value,

$$r^{ewma}_t = |y_t - \widehat{y}_t| \tag{12}$$

Where the threshold value α_{ewma} is found by the value that minimizes the false alarms declared to find all the anomalies in the training set.

8.1.2 Fourier Thresholding

The second time-series anomaly detection methodology consists of resolving the residual energy from Fourier analysis on the time-series signal. Given a Discrete Cosine Transformation of the time-series signal, we determine the vector of discrete cosine coefficients, α_{dct}, such that our observed time-series signal $\mathbf{y} = \Phi_{dct}\alpha_{dct}$. In order to generate a residual signal, we threshold the small components of the discrete cosine coefficients, such that,

$$\alpha'_{dct}(i) = \left\{ \begin{array}{cl} \alpha_{dct}(i) & : \text{if } |\alpha_{dct}(i)| \geq \alpha_{fourier} \\ 0 & : \text{if } |\alpha_{dct}(i)| < \alpha_{fourier} \end{array} \right. \tag{13}$$

Finally, the energy in the residual Fourier signal indicates whether or not an anomaly has occurred at each time step,

$$\mathbf{r}^{fourier} = \left| \mathbf{y} - \Phi_{dct}\alpha'_{dct} \right| \tag{14}$$

Where the threshold value $\alpha_{fourier}$ is found by the value that minimizes the false alarms declared to find all the anomalies in the training set.

8.1.3 GEANT Results

After performing anomaly detection using both the new BasisDetect algorithm and the two comparison time-series anomaly detection methodologies (EWMA and Fourier) on the GEANT dataset, the false alarm results for detecting the labeled true anomalies can be found in Figure 4. As seen in the figure, our BasisDetect framework consistently performs better than the two comparison methodologies with respect to the number of false alarms declared. The number of false alarms declared for specific percentages of true anomalies found can be seen in Table 1. The table shows that to find all of the labeled anomalies in our time-series signal, the BasisDetect method declares almost 50% fewer false alarms than the best competing methodology (the exponentially weighted moving average filter methodology (EWMA)), and over 75% fewer false alarms than the Fourier methodology.

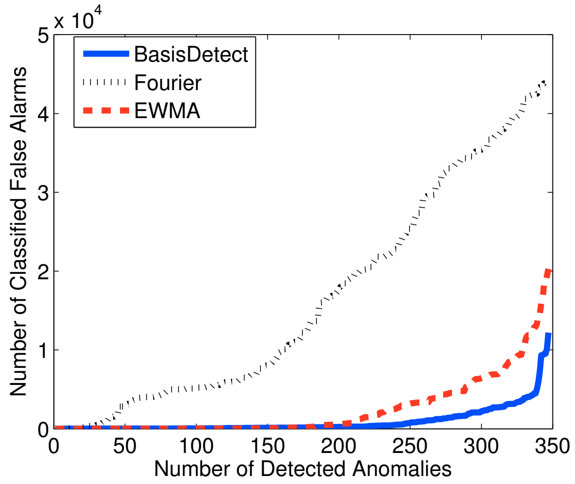

Figure 4: GEANT Network Data - False Alarm anomalies found for a specified level of true anomaly detection for the three time-series detection methodologies (Fourier, EWMA, BasisDetect).

Table 1: GEANT Network Data - Number of false alarms declared for a percentage of the true anomalies detected.

Methodology	Percentage of True Anomalies Found			
	70%	80%	90%	100%
BasisDetect	550	1,428	2,363	10,474
EWMA	2,825	4,558	7,138	20,275
Fourier	23,978	33,048	37,568	43,919

8.2 Wavelet Performance Analysis

To represent localized non-anomalous components of the observed data we use wavelets in the non-anomalous dictionary ($\Phi_{non-anomaly}$). The choice of which type of discrete wavelet signals to use is non-straightforward. The decision to use discrete Meyer wavelets in the signal dictionary was due to its performance against other wavelet types on the GEANT dataset. The results can be seen in Table 2, where in order to detect all the anomalies in the GEANT dataset, using the discrete Meyer wavelet results in almost 1,500 fewer false alarms declared against the next competing wavelet type (Haar wavelets). Intuitively, the Meyer wavelet represents localized sinusoidal behavior, which can commonly be found in non-anomalous network data. Meanwhile, Haar wavelets represent sharp signal discontinuities and Daubechies wavelets represent signal polynomial structure, neither of which should be expected to represent non-anomalous signal behavior well. The performance of the Meyer wavelet transform here motivates its use throughout the remainder of the experiments.

8.3 Tuning Parameter Performance Analysis

The BasisDetect framework uses a series of tuning parameters (γ, ρ) to optimize performance. In order to assess improvements in BasisDetect's false alarm rate due to these parameters, experiments were run on the GEANT dataset setting each of the tuning parameters to zero and observing performance. The results can be seen in Figure 5. As seen in the figure, the full BasisDetect methodology including both tuning parameters optimized by the training set significantly outperforms both parameter eliminated methods with respect to the number of false alarms declared. In the case of the eliminated anomaly signal component penalty ($\gamma = 0$) we revert to a standard greedy basis pursuit methodology, with this modified methodology consistently outperformed by the full BasisDetect methodology for the detection of every anomaly in the GEANT dataset. This indicates that our penalized methodology of BasisDetect offers a clear performance advantage over standard Basis Pursuit methodologies. In the case where the residual signal is ignored ($\rho = 0$), the results indicate better performance than the non-penalized BasisDetect method for a majority of the detected anomalies (while still worse than the full BasisDetect methodology), the method then fails at detecting the last fraction of anomalies in terms of the large number of false alarms declared to find that last fraction of anomalies. This represents a regime where the basis pursuit methodology is failing at fitting the anomalies to our estimated anomalous signal dictionary, likely due to limited training data for our learning-based methodology. These results motivate the use of both tuning parameters in our full BasisDetect framework.

Table 2: GEANT Network Data - Number of false alarms declared in order to detect every anomaly in the GEANT dataset (with respect to various wavelet types).

Wavelet Type	Number of False Alarms
Discrete Meyer	10,474
Haar	12,096
Daubechies	14,210

8.4 Synthesized Network-wide Data

Using the synthetic network-wide traffic matrix technique described Section 3.1, we generate network-wide data with injected anomalies. To compare performance of our BasisDetect methodology, we will use two state-of-the-art network-wide anomaly detection techniques. The first methodology is the Principle Component Analysis (PCA) technique from [1], and the second will be the Distributed Spatial anomaly detection technique from [2]. In order to test the detection capabilities of all methods, the injected anomalies vary in both amplitude and length, while the level of added noise in the network data varies between network snapshots. Our detection methodologies will be tested to detect both the beginning and ending of each injected anomaly.

For the initial synthetic experiment, we evaluate performance of the three network-wide anomaly detection methodologies using data with constant injected anomaly amplitudes, $a_{anomaly}$ across all network snapshots. Using 20 synthesized network snapshots each with a single injected anomaly, we modify both the length of the anomaly injected (between 2 and 8 time bins) and the size of the network (between 3 and 5 fully connected routers, relating to 9 and 25 observed links respectively). Each network snapshot simulates 2.5 days of packet out link observations aggregated into 5 minute time bins, resulting in an observed 1024-length

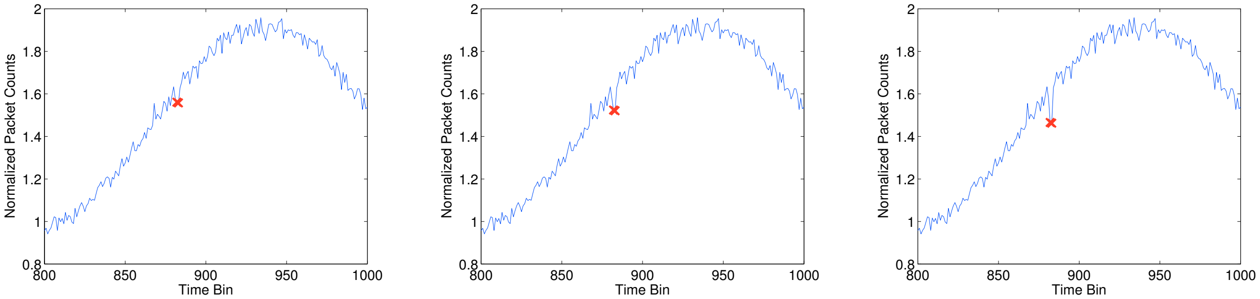

Figure 6: Three examples of injected anomalies with varying anomaly amplitudes. (Left) - $a_{anomaly} = 0.063$, (Center) - $a_{anomaly} = 0.1$, (Right) - $a_{anomaly} = 0.158$

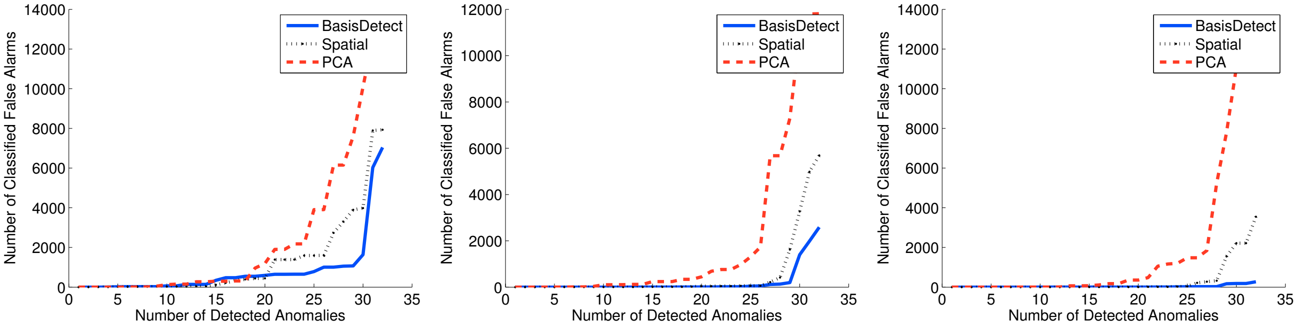

Figure 7: Synthetic Traffic Matrices with Constant Anomaly Amplitude - False Alarms declared for a specified level of true anomaly detection for the three network-wide detection methodologies (PCA, Distributed Spatial, BasisDetect). (Left) - $a_{anomaly} = 0.063$, (Center) - $a_{anomaly} = 0.1$, (Right) - $a_{anomaly} = 0.158$.

Table 3: Synthetic Traffic Matrices with Constant Anomaly Amplitude - Number of false alarms declared for a given number of the true anomalies detected.

Anomaly amplitude ($a_{anomaly}$)	0.063				0.1				0.158			
Number of true anomalies found	8	16	24	32	8	16	24	32	8	16	24	32
PCA	11	298	2,179	12,579	9	239	974	11,812	3	110	1,202	12,598
Spatial	21	216	1,594	7,932	2	10	55	5,690	0	1	24	3,545
BasisDetect	27	473	653	7,041	0	2	18	2,587	0	2	15	272

time-series vector at each synthetic link. Using hold-out cross validation, we use 4 of the network snapshots to train our BasisDetect methodology and then test detection performance across the remaining 16 networks. We consider three anomaly amplitude regimes, (low - $a_{anomaly} = 0.063$, medium - $a_{anomaly} = 0.1$, and high - $a_{anomaly} = 0.158$), with examples of these injected anomaly regimes for the observed network data shown in Figure 6.

In the experiment results in Figure 7, we see that for all three amplitude regimes our BasisDetect methodology outperforms the current state-of-the-art detection techniques in terms of the number of false alarms anomalies declared across all 16 test networks. In Table 3 we find a breakdown of the false alarms declared for various detection levels. As seen in the table, the BasisDetect methodology performs significantly better than both the PCA and Distributed Spatial methodology. In terms of the medium amplitude anomaly ($a_{anomaly} = 1$), BasisDetect finds all anomalies with over 54% fewer false alarms than Distributed Spatial and over 75% fewer false alarms than the PCA methodology. In terms of the high anomaly amplitude experiment ($a_{anomaly} = 0.158$), BasisDetect declares 90% fewer false alarms compared with the Distributed Spatial methodology and almost 99% fewer false alarms than the PCA approach, in order to find all the injected anomalies.

The second synthetic experiment tested anomaly detection performance on 30 network snapshots for a single anomaly injected into each snapshot, with each having varying anomaly amplitude levels ranging from $a_{anomaly} = 0.0316$ to $a_{anomaly} = 1$. In Figure 8 we see the results of detecting injected anomalies across all 24 test network snapshots with anomalies of various amplitude (holding out 6 of the network snapshots as training data for BasisDetect). Again, our BasisDetect methodology has uniformly better performance than both the Distributed Spatial and PCA anomaly detection methodologies. Selected results

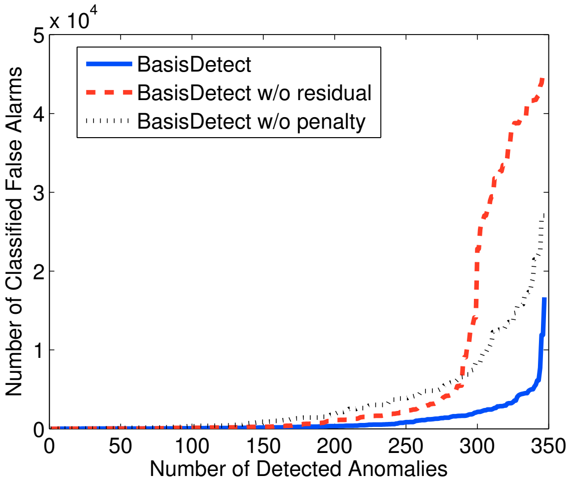

Figure 5: Tuning parameter performance experiment, examination of how well the BasisDetect algorithm performs as each of the tuning parameters are removed. Using the full BasisDetect algorithm (γ, ρ learned from training set), BasisDetect w/o residual (γ learned from training set, $\rho = 0$), and BasisDetect w/o penalty (ρ learned from training set, $\gamma = 0$)

highlighted in Table 4 show that with respect to the best competing methodology (the Distributed Spatial methodology), our BasisDetect algorithm declares over 65% fewer false alarms in order to detect all of the true anomalies. In comparison with the PCA algorithm, we find over 80% fewer false alarms in order to discover all of the true anomalies.

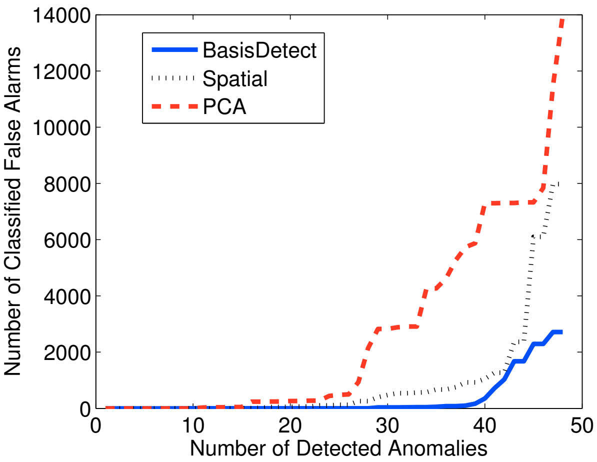

Figure 8: Synthetic Traffic Matrices with Varying Anomaly Amplitude - False Alarms declared for a specified level of true anomaly detection for the three network-wide detection methodologies (PCA, Distributed Spatial, BasisDetect).

One surprising result of both synthetic experiments is the very large number of false alarms declared in contrast with prior published results in [1, 2]. Note that our experiments consider a significantly larger dataset. While prior experiments examined network data with 1,008 time-series samples, we examined data with 16,384 and 24,576 time-series samples for the first and second synthetic experiments respectively. While the increase in the experiment size results in a significantly greater number of absolute false alarms declared, we feel that the larger experiments offer a greater understanding as to the performance characteristics of the three anomaly detection methodologies.

Table 4: Synthetic Traffic Matrices for Varying Anomaly Amplitude - Number of false alarms declared for a percentage of the true anomalies detected.

Methodology	Percentage of True Anomalies Found			
	70%	80%	90%	100%
PCA	4,266	5,874	7,325	13,966
Spatial	577	924	2,365	7,977
BasisDetect	43	166	1,673	2,716

Table 5: Abilene Network Data - Number of false alarms declared for a percentage of the PCA anomalies detected.

Methodology	Percentage of PCA Anomalies Found			
	70%	80%	90%	100%
Spatial	563	733	1,287	4,564
BasisDetect	327	495	747	2,746

8.5 Abilene Real-World Network Data

Finally, we test the performance of the BasisDetect framework on the Abilene real-world dataset. As mentioned in Section 2, there is no ground truth labeling of anomalies for this dataset. Instead, here we will use one of the two prior network-wide anomaly detection methodologies (PCA) to classify the most obvious anomalies for that methodology. Using these classified anomalies as the ground truth, we compare how the new BasisDetect methodology performs in comparison with the other competing network-wide anomaly detection methodology (*e.g.*, the Distributed Spatial methodology).

Using the PCA network-wide anomaly detection technique, we classify the 15 most dominant anomalies in the Abilene data. The performance of both the BasisDetect framework and the Distributed Spatial approach can be seen in Figure 9 on detecting these PCA classified anomalies using 5-way Cross Validation (thus expanding the total anomalies considered to 70 in the dataset). As seen in the figure, our BasisDetect methodology is detecting these PCA classified anomalies with lower false alarm rate than the Distributed Spatial approach, with almost 40% fewer false alarms declared to detect all the PCA anomalies. A specific breakdown of the false alarm rate can be seen in Table 5.

9. CONCLUSIONS AND FUTURE WORK

The ability to detect anomalies accurately and in a timely fashion in large networks would be a significant benefit in day to day network operations. It can be argued that cur-

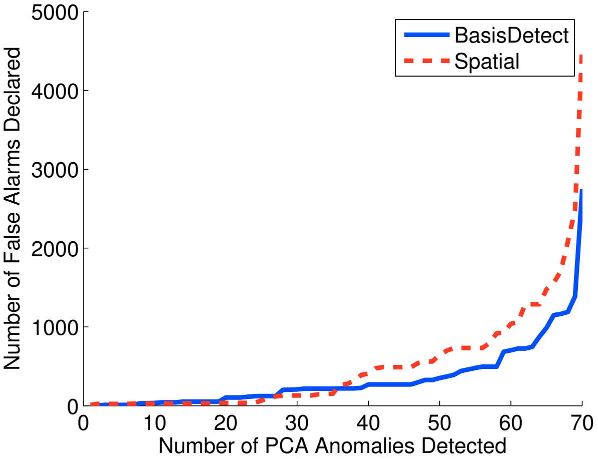

Figure 9: Abilene Real-World Network Data - Using 15 anomalies found by the PCA methodology, the false alarm rates are displayed for both BasisDetect and the Distributed Spatial methodology.

rent methods are not yet sufficiently capable based on the fact that they are not widely used today. The objective of our work is to develop an anomaly detection capability that is sufficiently accurate to be considered practical for operational deployment and use.

In this paper we present the BasisDetect framework, a model-based methodology for anomaly detection. We build temporal models by applying a novel basis pursuit algorithm to the key components of learned anomalies from a small training set. These temporal models are extended from anomaly detection on a single node to the network-wide context by applying a higher reasoning framework. This combined approach has additional benefits, such as the ability to be applied flexibly to a wide variety of data, extensibility to include a variety of filter functions, and low computational complexity.

We test and evaluate the BasisDetect methodology using both empirical and synthetic network data with labeled anomalies. In the single node case, when we compare with standard time-series based methods, our method identifies all of the labeled anomalies with over 50% fewer false alarms declared compared with the competing methodologies. In the case of unlabeled real-world network wide data, we show considerable improvements in detecting anomalies declared by previous network-wide anomaly detection methodologies. Finally, we use synthetic traces to examine in detail the sensitivity of our method over a range of anomalies to show that to find all of the labeled anomalies, our methodology will declare 65% fewer false anomalies than the best competing methodology. The results show that the model-based method is highly effective even when large amounts of noise are present.

The implication of our results is that our model-based methodology is indeed feasible for event detection in an operational environment. While we show that models from a small set of filters can be effective, we have not investigated optimizations of the filters that could enhance their ability. Furthermore, there are many practical issues revolving around where and how data is gathered in a network that must be systematically addressed to facilitate application of our methods. Ultimately, the best test of an anomaly detector is in a live environment. In the future, we plan to work closely with several network operations groups to deploy and test our techniques.

10. ACKNOWLEDGMENTS

This work was supported in part by NSF grants CNS-0716460, CNS-0831427 and CNS-0905186. Also support was provided by ARC Grant DP0665427. Any opinions, findings, conclusions or other recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the NSF or ARC. We would also AT&T, Technicolor, Abilene and GÉANT networks for providing data.

11. REFERENCES

[1] A. Lakhina, M. Crovella, and C. Diot, "Diagnoising Network-Wide Traffic Anomalies," in *Proceedings of ACM SIGCOMM Conference*, Portland, OR, August 2004.

[2] P. Chhabra, C. Scott, E. Kolaczyk, and M. Crovella, "Distributed Spatial Anomaly Detection," in *Proceedings of IEEE INFOCOM Conference*, Phoenix, AZ, March 2008.

[3] A. Lakhina, M. Crovella, and C. Diot, "Characterization of Network-Wide Anomalies in Traffic Flows," in *Proceedings of ACM SIGCOMM Internet Measurement Conference*, Taormina, Sicily, Italy, October 2004.

[4] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural Analysis of Network Traffic Flows," in *ACM SIGMETRICS / Performance*, 2004.

[5] P. Barford, J. Kline, D. Plonka, and A. Ron, "A Signal Analysis of Network Traffic Anomalies," in *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, Marseilles, France, November 2002.

[6] Y. Zhang, Z. Ge, M. Roughan, and A. Greenberg, "Network Anomography," in *Proceedings of the Internet Measurement Conference*, Berkeley, CA, USA, October 2005.

[7] M. Roughan, T. Griffin, M. Mao, A. Greenberg, and B. Freeman, "IP forwarding anomalies and improving their detection using multiple data sources," in *ACM SIGCOMM Workshop on Network Troubleshooting (NetTS)*, Portland, OR, September 2004, pp. 307–312.

[8] S. H. Steiner, "Grouped Data Exponentially Weighted Moving Average Control Charts," *Applied Statistics*, vol. 47, no. 2, 1998.

[9] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for Traffic Anomaly Detection," in *Proceedings of ACM SIGMETRICS Conference*, San Diego, CA, June 2007.

[10] B. Rubinstein, B. Nelson, L. Huang, A. Joseph, S. Lau, S. Rao, N. Taft, and J. Tygar, "ANTIDOTE: Understanding and Defending Against Poisoning of Anomaly Detectors," in *Proceedings of ACM SIGCOMM Internet Measurements Conference*, Chicago, Illinois, November 2009.

[11] Y. Benjamini and Y. Hochberg, "Controlling the False Discovery Rate," in *Journal of the Royal Statistical Society B*, vol. 57, no. 1, 1995, pp. 289–300.

[12] R. Miller, in *Simultaneous Statistical Inference*. Springer-Verlag, 1991.

[13] L. Huang, X. Nguyen, M. Garofalakis, J. Hellerstein, M. Jordan, M.Joseph, and N.Taft., "Communication-Efficient Online Detection of Network-Wide Anomalies," in *Proceedings of IEEE INFOCOM Conference*, Anchorage, Alaska, May 2007.

[14] Y. Liu, L. Zhang, and Y. Guan, "A Distributed Data Streaming Algorithm for Network-Wide Traffic Anomaly Detection," *SIGMETRICS Performance Evaluation Review*, vol. 37, no. 2, pp. 81–82, 2009.

[15] C. Scott and E. Kolaczyk, "Nonparametric Assessment of Contamination in Multivariate Data using Generalized Quantile Sets and FDR," in *Accepted for Publication in J. Computational and Graphical Statistics*, 2007.

[16] S. Chen, D. Donoho, and M. Saunders, "Atomic Decomposition by Basis Pursuit," in *SIAM Journal of Scientific Computing*, vol. 20, 1998, pp. 33–61.

[17] G. Davis, S. Mallat, and M. Avellaneda, "Greedy Adaptive Approximation," in *Journal of Constructive Approximation*, vol. 13, 1997, pp. 57–98.

[18] P. Huggins and S. Zucker, "Greedy Basis Pursuit," in *IEEE Transactions on Signal Processing*, vol. 55, no. 7, July 2007, pp. 3760–3771.

[19] H. Ringberg, M. Roughan, and J. Rexford, "The Need For Simulation In Evaluating Anomaly Detectors," *ACM SIGCOMM CCR*, vol. 38, no. 1, pp. 55–59, January 2008.

[20] Y. Zhang, M. Roughan, W. Willinger, and L. Qui, "Spatio-Temporal Compressive Sensing and Internet Traffic Matrices," in *Proceedings of ACM SIGCOMM Conference*, Barcellona, Spain, August 2009, pp. 267–278.

[21] "Geant Project Website," http://www.geant.net/.

[22] A. Markopoulou, G. Iannaccone, S. Bhattacharrya, C.-N. Chuah, and C. Diot, "Characterization of failures in an IP backbone," in *Proceedings of IEEE INFOCOM Conference*, Hong Kong, China, March 2004.

[23] D. Oppenheimer, A. Ganapathi, and D. A. Patterson, "Why do Internet services fail, and what can be done about it?" in *4th Usenix Symposium on Internet Technologies and Systems (USITS'03)*, 2003.

[24] L. Wasserman, "All of Nonparametric Statistics," in *Springer Texts*, 2006.