Detection of Internet Traffic Anomalies using Sparse Laplacian Component Analysis

Manas Khatua, Seyed Hamid Safavi, and Ngai-Man Cheung
Singapore University of Technology and Design, Singapore
Email:{manaskhatua,hamid.safavy}@gmail.com, ngaiman_cheung@sutd.edu.sg

***Abstract*—We consider the problem of anomaly detection in network traffic. It is a challenging problem because of high-dimensional and noisy nature of network traffic. A popularly used technique is *subspace analysis*. Principal component analysis (PCA) and its improvements have been applied for subspace analysis. In this work, we take a different approach to determine the subspace, and propose to capture the essence of the traffic using the eigenvectors of graph Laplacian, which we refer as Laplacian components (LCs). Our main contribution is to propose a regression framework to compute LCs followed by its application in anomaly detection. This framework provides much flexibility in incorporating different properties into the LCs, notably LCs with sparse loadings, which we exploit in detail. Furthermore, different from previous work that uses sample graphs to preserve local structure, we advocate modelling with a dual-input feature graph that encodes the correlation of the time series data and prior information. Therefore, the proposed model can readily incorporate the 'physics' of some applications as prior information to improve the analysis. We perform experiments on volume anomaly detection using only link-based traffic measurements. We demonstrate that the proposed model can correctly uncover the essential low-dimensional principal subspace containing the normal Internet traffic and achieve outstanding detection performance.**

***Index Terms*—Network Anomaly Detection, Dimensionality Analysis, Graph Laplacian, Regression, Sparse Loadings**

I. Introduction

Network traffic anomalies are considered as unusual but significant changes present in network traffic [4]. Example includes both the legitimate activities such as flash crowds, sudden changes in customer demand, and illegitimate activities such as port scans, distributed denial-of-service (DDoS), link flooding attack, [5]. Very often, the collected data used for network anomaly detection is huge, high-dimensional, noisy and grossly distorted. Therefore, processing and analyzing such massive data in time-critical environment poses unprecedented challenges. We propose to address anomaly detection in massive data traffic by exploiting recent discoveries in high-dimensional graph signal analysis [6].

Determining anomalies in network data streams has attracted a significant amount of research efforts [7]–[10]. Mainly, there exist two paths of work for network anomaly detection: signature-based and non-signature-based detection. The second approach is useful for unknown threat and anomaly as it does not require any prior knowledge about the anomalies. In the domain of non-signature-based anomaly detection, *subspace analysis* is a popular approach that aims to partition the high-dimensional traffic signal space into disjoint subspaces corresponding to the normal and anomalous network conditions [7], [8], [11]. One of the most critical requirements of subspace analysis is to uncover the essential low-dimensional normal traffic subspace from the noisy and high-dimensional traffic. Many spectral techniques such as PCA have been proposed to address this dimensionality analysis problem. Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{p\times n}$ be the high-dimensional traffic measurement. In particular, $\mathbf{X}$ consists of n measurements of dimensionality p (In our experiment, $\mathbf{X}$ consists of measurements in n successive time intervals; each measurement consists of traffic statistics of p links/nodes). The *classical PCA* (Model 1 in Table I) finds the projection $\mathbf{Q}^T \in \mathbb{R}^{k\times n}$ of $\mathbf{X}$ on a k-dimensional ($k \leq p$) linear space characterized by an orthogonal basis $\mathbf{V} \in \mathbb{R}^{p\times k}$. The product $\mathbf{VQ}^T$ is known as the low-rank approximation $\mathbf{L} \in \mathbb{R}^{p\times n}$ of $\mathbf{X}$. The clustering or subspace analysis is performed on $\mathbf{L}$.

It is observed that, in many cases, low-dimensional data follows certain structures which are hidden in the original data. The performance of different applications such as dimensionality reduction, clustering, and anomaly detection could be improved if we leverage that structures in the models. Therefore, there is a trend to improve the performance of PCA by utilizing the hidden structure in a form of graph [1]–[3], [12]–[16]. These works mainly consider the graph structure based on *sample similarity*. Few of them (e.g., [13]) considered *feature similarity* along with sample similarity.

A. Focus of this work

The assumption in most of the graph-based models is that the data is "smooth" on the underlying graph, and they use the corresponding graph Laplacian to impose the smoothness constraints during the recovery of the low-rank approximation of data. Specifically, they use spectral graph regularization in the optimization problems to impose graph smoothness [1]–[3]. On the other hand, inspired by the recent graph signal processing [6], [17], our work takes a different approach and imposes graph smoothness using the first k eigenvectors of the *feature graph Laplacian*, where $k < p$ for p-dimensional input data. Our approach takes the view of signal reconstruction of the normal traffic, which is assumed to be graph-smooth. Our main contribution is to propose a regression framework to compute these eigenvectors. As will be discussed, the regression framework provides flexibility to introduce sparse loadings in the components, leading to improved detection accuracy. In addition, in many cases, the low-dimensional data follows the

TABLE I: A comparison on the properties of classical PCA, RPCA, and various recent graph-based PCA models [1]–[3]. $\|.\|_1, \|.\|, \|.\|_F$ and $\|.\|_*$ denote the l_1, l_2, Frobenious, and nuclear norm, respectively. $\delta, \gamma, \gamma_1, \gamma_2$ are the weighting constants. $\mathbf{M}$ is the sparse matrix. In our proposed LCA and SLCA, $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_m\}^T$ is a $m \times p$ matrix where $\mathbf{s}_i = \{s_{i1}, \ldots, s_{ip}\}$, and $\mathbf{S}^T\mathbf{S}$ is the $p \times p$ Laplacian matrix of dual-input feature graph. G is the graph structure representation of prior information. $\mathcal{F}_2$ represents the weight matrix computation function. $\mathbf{b}_j$ is the j^{th} column vector, and the Laplacian components $\mathbf{V} = \mathbf{B}_{p\times k} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\}, k \leq p$.

#	Model	Objective	Constraints	Parameter	Graph on		
					samples	features	'physics'
1	PCA	$\min_{\mathbf{V},\mathbf{Q}} \|\mathbf{X} - \mathbf{V}\mathbf{Q}^T\|_F^2$	$\mathbf{V}^T\mathbf{V} = \mathbf{I}$	k	×	×	×
2	RPCA	$\min_{\mathbf{L},\mathbf{M}} \|\mathbf{L}\|_* + \delta\|\mathbf{M}\|_1$	$\mathbf{X} = \mathbf{L} + \mathbf{M}$	δ	×	×	×
3	RPCAG	$\min_{\mathbf{L},\mathbf{M}} \|\mathbf{L}\|_* + \delta\|\mathbf{M}\|_1 + \gamma_1\ tr(\mathbf{L}\Phi_s\mathbf{L}^T)$	$\mathbf{X} = \mathbf{L} + \mathbf{M}$	δ, γ_1	✓	×	×
4	FRPCAG	$\min_{\mathbf{L}} \|\mathbf{X} - \mathbf{L}\|_1 + \gamma_1\ tr(\mathbf{L}\Phi_s\mathbf{L}^T) + \gamma_2\ tr(\mathbf{L}^T\Phi_f\mathbf{L})$		γ_1, γ_2	✓	✓	×
5	GLPCA	$\min_{\mathbf{V},\mathbf{Q}} \|\mathbf{X} - \mathbf{V}\mathbf{Q}^T\|_F^2 + \gamma_1\ tr(\mathbf{Q}^T\Phi_s\mathbf{Q})$	$\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$	k, γ_1	✓	×	×
6	LCA (this work)	$\min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^m \|\mathbf{s}_i - \mathbf{A}\mathbf{B}^T\mathbf{s}_i\|^2 + \gamma\sum_{j=1}^k \|\mathbf{b}_j\|^2;\ \mathcal{F}_2(\mathbf{X}, G) \Rightarrow \Phi_{df}$	$\mathbf{A}^T\mathbf{A} = \mathbf{I}_{k\times k}$	k, γ	×	✓	✓
7	SLCA(this work)	$\min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^m \|\mathbf{s}_i - \mathbf{A}\mathbf{B}^T\mathbf{s}_i\|^2 + \gamma\sum_{j=1}^k \|\mathbf{b}_j\|^2 + \sum_{j=1}^k \delta_j\|\mathbf{b}_j\|_1$	$\mathbf{S}^T\mathbf{S} = \Phi_{df}$	k, γ, δ			

"physics" of the problem at hand which is known a priori [6]. Examples of such prior knowledge could be the physical proximity of the sensors in wireless sensor networks (WSNs) [18], the magnetometers in Magnetoencephalography (MEG) or the network routing topology in Internet traffic analysis [7]. A standard way to incorporate the knowledge of such information is by using a graph. Such prior information could be useful when the data samples are noisy. Thus, we investigate the mechanism to incorporate prior information in the process of recovering low-rank data matrix. In brief, we focus on an improved PCA using graph and prior information for more accurate subspace identification followed by its application in detection of network traffic anomaly such as volume anomaly. Note that, as in previous work (e.g., [7], [11]), we use link-level measurements to detect volume anomaly, as this is scalable for network-wide monitoring. However, the anomalous change in a flow is rather small and difficult to be identified in the link-level measurements, and volume anomaly detection is a challenging problem [7].

B. Contribution

Let $\Phi_s \in \mathbb{R}^{n\times n}$ and $\Phi_f \in \mathbb{R}^{p\times p}$ be the graph Laplacian on the sample graph G_s and the feature graph G_f of $\mathbf{X}$, respectively. The graph G_s connects the different samples of $\mathbf{X}$ (columns of $\mathbf{X}$) and the graph G_f connects the features of $\mathbf{X}$ (rows of $\mathbf{X}$). In this paper, we propose a subspace analysis based spectral method for detecting network traffic anomaly. We apply the eigenvectors of the *graph Laplacian* Φ_{df} in which both the data matrix and the prior information are incorporated together. Note that the subscripts 'f' and 'df' with Φ indicate default feature graph and *dual-input feature graph*, respectively (more discussion in Sections III and IV). In particular, for p-dimensional data that can be defined on the vertices of a graph $\mathcal{G}$ ($|\mathcal{V}| = p$), we use the first k smooth eigenvectors $\{\mathbf{v}_1, ..., \mathbf{v}_k\}$ of Φ_{df} to define the intrinsic k-dimensional subspace corresponding to normal network conditions. Our approach uses a different mechanism to impose the graph smoothness constraint: instead of using spectral graph regularization and relying on γ_1, γ_2 to control the graph smoothness (as in Model 3 to 5 in Table I), we control the graph smoothness directly via the selection of k smooth eigenvectors. Note that the eigenvectors of a graph Laplacian with small eigenvalues are smooth with respect to the underlying graph, i.e., changes are small between the connected vertices. It is an application of Courant-Fischer Formula for Laplacian [19] and is an important result for the recent graph signal processing [6]. Therefore, the first k eigenvectors model a class of signals that are smooth with respect to underlying graph. Reconstruction using k eigenvectors imposes graph smoothness constraint. Our approach is more intuitive and direct in controlling and adjusting the smoothness of the low-rank data matrix on the graph $\mathcal{G}$.

Moreover, as our main contribution, we propose to use a regression-type optimization framework to compute the orthogonal bases $\{\mathbf{v}_l\}$ from the normalized graph Laplacian Φ_{df}. We name $\{\mathbf{v}_l\}$ as Laplacian components (LCs) and the regression-based approach as *Laplacian component analysis* (LCA). LCA produces the same resulting vectors as direct eigendecomposition of Φ_{df}. On the other hand, LCA provides flexibility to achieve different properties of the resulting LCs. In particular, in this work, we exploit the use of lasso penalty in the baseline LCA to obtain LCs with sparse loadings. We summarize our contributions as follows:

- We propose a new approach namely Laplacian Component Analysis (LCA). LCA performs regression on the structure of the graph Laplacian and is different from existing work based on eigendecomposition [6].
- Based on LCA, we present a general framework to include additional attributes (e.g. sparse loading) in the Laplacian components.
- We design a network anomaly detection scheme by introducing dual-input feature graph (i.e. source graph) with (sparse) LCA.

II. Related Work

The most critical step for subspace analysis-based detection is to identify the low-dimensional normal traffic subspace. Many spectral techniques such as PCA have been proposed to address the dimensionality analysis problem [20]. The classical PCA suffers from few disadvantages [13], [18], [21]. Therefore, many improvements over the classical PCA have been proposed. Candes *et al.* [21] proposed *Robust PCA* (RPCA, Model 2 in Table I) which is robust to outliers by directly recovering the low-rank matrix $\mathbf{L}$ from the grossly corrupted $\mathbf{X}$. Recently, there is a trend to improve PCA by leveraging the hidden structure of data matrix in the form of a graph for improving dimensionality analysis. These works consider structure of feature similarity, sample similarity, and combination of both in the form of graphs. The *graph Laplacian PCA* (GLPCA) [1](Model 5 in Table I) considers implicit structure among data samples for improving the accuracy of clustering. Shahid *et al.* [2] proposed *robust PCA on graph* (RPCAG) (Model 3 in Table I) which can accurately learn $\mathbf{L}$ in the presence of occlusion and missing pixel. Note that GLPCA [1] assumes the graph smoothness of the projected data $\mathbf{Q}$ while RPCAG [2] assumes the graph smoothness of the low-rank approximation $\mathbf{L}$. These smoothness constraints are used as the regularization. Considering the feature similarity graph, *Laplacian Lasso* (LLasso) [13] proposed a network-constrained regularization procedure in which the lasso penalty is combined with the network penalty induced by Laplacian matrix of the graph. Inspired by the two-way graph regularization scheme [22], Shahid *et al.* [3] proposed *Fast RPCAG* (FRPCAG) (Model 4 in Table I) for faster and better clustering.

Several other techniques for dimensionality analysis, such as [12], [23] and [24], involve eigen-decomposition of the graph Laplacian. In particular, LPP [12] is a linear mapping that involves graph Laplacian and can be seen as an alternative to PCA, similar to our work. However, the framework and mechanism of our work are different from LPP and previous graph based works. Specifically, the LPP is geometrically motivated. Given a set of n p-dimensional points $\mathbf{x}_i$, LPP aims to preserve local structure of the n points. For LPP, the solution is the set of eigenvectors of $X\Phi_s X^T$. On the other hand, our approach focuses on the $p \times p$ graph Laplacian Φ_f, or Φ_{df} if prior information is incorporated. Besides, the LCs are equivalent to the eigenvectors of Φ_{df}. We do not perform eigen-decomposition to compute LCs. Instead, we propose to regress on the structure of Φ_{df} to compute the components. Note that our work outperforms LPP, as will be discussed in our experiment. In addition, Spectral Regression [14] has been proposed as a two-step more efficient approach to solve the eigen-problem of a *specific* form. It is not applicable for our eigen-problem. Also, our algorithm performs regression directly on the structure of the Laplacian in a single step.

III. Methodology

We focus on network traffic anomaly detection using subspace analysis, and the most critical step is to determine the low-dimensional subspace corresponding to the normal traffic. In this work, we propose a new method to determine the normal subspace more accurately, and, therefore, the accuracy of anomaly detection increases. This section describes the steps to determine the LCs followed by their sparse approximation to obtain more accurate normal subspace.

A. Overview of Proposed Method

In brief, **our proposed model** is as follows:

$$G(\mathcal{V},\mathcal{E}) \Leftarrow \mathcal{F}_1(\text{Prior Information}) \tag{1}$$

$$\mathcal{G}(\mathcal{V},\mathcal{E},\mathbf{W},\mathbf{D}) \Leftarrow \mathcal{F}_2(G,\mathbf{X});\ \mathbf{X} \in \mathbb{R}^{p\times n} \tag{2}$$

$$\Phi_{df} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}} \tag{3}$$

$$\min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^{m} \|\mathbf{s}_i - \mathbf{A}\mathbf{B}^T\mathbf{s}_i\|^2 + \gamma \sum_{j=1}^{k} \|\mathbf{b}_j\|^2 + \sum_{j=1}^{k} \delta_j \|\mathbf{b}_j\|_1$$
$$s.t.\ \mathbf{A}^T\mathbf{A} = I_{k\times k},\ \mathbf{S}^T\mathbf{S} = \Phi_{df} \tag{4}$$

where $\mathbf{B}_{p\times k} = [\mathbf{b}_1,\ldots,\mathbf{b}_k]$, and γ and δ are the tuning parameters. The steps in (1) and (2) describe the conversion from the traffic measurements $\mathbf{X}$ and prior information to an undirected graph $\mathcal{G}$, using the conversion functions $\mathcal{F}_1(.)$ and $\mathcal{F}_2(.)$. This is problem specific and we will discuss this in Section IV. With the result from (1) and (2), which is a simple, undirected, connected and weighted graph $\mathcal{G}(\mathcal{V},\mathcal{E},\mathbf{W},\mathbf{D})$, the step in (3) shows the computation method of normalized graph Laplacian of graph $\mathcal{G}$ followed by its decomposition into a matrix $\mathbf{S}$, which is, then, used as input in the optimization framework shown in Equation (4). Finally, the computed k sparse LCs are $\{\mathbf{v}_1,...,\mathbf{v}_k\} = \mathbf{B}$, which are used to define the low-rank approximation of $\mathbf{X}$ corresponds to normal subspace.

B. Regression Framework to Compute LCs

Unlike eigendecomposition, we propose to use a regression-type optimization framework to compute the eigenvectors from the graph Laplacian Φ_{df}. As will be discussed later, this allows us to impose different types of constraints to achieve different properties of the resulting LCs, e.g., sparse loadings. Specifically, since Φ_{df} is real, positive semi-definite and diagonalizable, we can write $\Phi_{df} = \mathbf{S}^T\mathbf{S}$ for some $m \times p$ matrix $\mathbf{S}$ (Note that we introduce $\mathbf{S}$ to ease our discussion. In practice, computing $\mathbf{S}$ is not needed, as will be discussed.). We apply the theorem from [25] to convert the computation of eigenvectors into a regression problem. Specifically, suppose we are considering the first k LCs ($\mathbf{v}_1,\ldots,\mathbf{v}_k$) of $\Phi_{df} = \mathbf{S}^T\mathbf{S}$, with $\mathbf{S} = [\mathbf{s}_1,\ldots,\mathbf{s}_m]^T$. Let $\mathbf{A}_{p\times k} = [\mathbf{a}_1,\ldots,\mathbf{a}_k]$ and $\mathbf{B}_{p\times k} = [\mathbf{b}_1,\ldots,\mathbf{b}_k]$. For any $\gamma > 0$, let

$$(\hat{\mathbf{A}},\hat{\mathbf{B}}) = \arg\min_{\mathbf{A},\mathbf{B}} \sum\nolimits_{i=1}^{m} \|\mathbf{s}_i - \mathbf{A}\mathbf{B}^T\mathbf{s}_i\|^2 + \gamma \sum\nolimits_{j=1}^{k} \|\mathbf{b}_j\|^2$$
$$s.t.\ \mathbf{A}^T\mathbf{A} = I_{k\times k}, \mathbf{S}^T\mathbf{S} = \Phi_{df} \tag{5}$$

where γ is a ridge penalty factor for each principal component (PC). Then it can be shown that $\hat{\mathbf{b}_j} \propto \mathbf{v}_j$ for $j = 1,2,\ldots,k$. Note that we are interested in the first k principal components of the Laplacian matrix (i.e. LCs). In particular, the interested Laplacian components correspond to the small eigenvalues of Φ_{df} and include the LC with the zero eigenvalue. This is different from [25]. It is noteworthy that (5) is an application of the theorem from [25]. To illustrate the link between (5) and regression analysis, note that when $\mathbf{A}$ is given, it can be shown that (5) becomes:

$$\arg\min_{\mathbf{B}} \sum\nolimits_{j=1}^{k} \|\mathbf{S}\mathbf{a}_j - \mathbf{S}\mathbf{b}_j\|^2 + \gamma \sum\nolimits_{j=1}^{k} \|\mathbf{b}_j\|^2. \tag{6}$$

Therefore, with $\mathbf{S}\mathbf{a}_j$ being viewed as the (known) response vector and $\mathbf{b}_j$ the regression coefficients, (6) is equivalent to k independent ridge regression problems. We name the formulation in (5) as Laplacian component analysis (LCA).

C. Sparse LCA (SLCA)

The regression formulation in (5) is a flexible framework that allows various enhancement of the baseline LCA. In particular, to achieve LCs with sparse loadings, we add the lasso penalty in (5):

$$(\hat{\mathbf{A}}, \hat{\mathbf{B}}) = \arg\min_{\mathbf{A},\mathbf{B}} \sum\nolimits_{i=1}^{m} \|\mathbf{s}_i - \mathbf{A}\mathbf{B}^T\mathbf{s}_i\|^2 + \gamma \sum\nolimits_{j=1}^{k} \|\mathbf{b}_j\|^2 + \sum\nolimits_{j=1}^{k} \delta_j \|\mathbf{b}_j\|_1 \quad s.t.\ \mathbf{A}^T\mathbf{A} = I_{k\times k}, \mathbf{S}^T\mathbf{S} = \Phi_{df} \tag{7}$$

Note that different δ_j can be used for different LCs to promote sparse loadings. However, because of the special behavior of the PC corresponding to the zero eigenvalue, in the experiments part, we use two sparse penalty factor; one for the PC corresponding to the zero eigenvalue and one for the other PCs. To solve (7), we use the alternating algorithm proposed in [25] and solve: (i) $\mathbf{B}$ given $\mathbf{A}$ and (ii) $\mathbf{A}$ given $\mathbf{B}$. On the other hand, we re-formulate the problem to apply the popular fast iterative shrinkage thresholding algorithm (FISTA) to attack $\mathbf{B}$ given $\mathbf{A}$ efficiently. Following the method of conversion from (5) to (6), it can be shown that (7) is equivalent to:

$$\min_{\mathbf{A},\mathbf{B}} \sum_{j=1}^{k} \|\mathbf{S}\mathbf{a}_j - \mathbf{S}\mathbf{b}_j\|^2 + \gamma \sum_{j=1}^{k} \|\mathbf{b}_j\|^2 + \sum_{j=1}^{k} \delta_j \|\mathbf{b}_j\|_1 \quad s.t.\ \mathbf{A}^T\mathbf{A} = I_{k\times k}, \mathbf{S}^T\mathbf{S} = \Phi_{df} \tag{8}$$

Moreover, if the first two terms are combined together, we reach a simple version:

$$\min_{\mathbf{A},\mathbf{B}} \sum_{j=1}^{k} \|\tilde{\mathbf{S}}\mathbf{a}_j - \bar{\mathbf{S}}\mathbf{b}_j\|^2 + \sum_{j=1}^{k} \delta_j \|\mathbf{b}_j\|_1 \quad s.t.\ \mathbf{A}^T\mathbf{A} = I_{k\times k}, \mathbf{S}^T\mathbf{S} = \Phi_{df} \tag{9}$$

where $\bar{\mathbf{S}} = \begin{bmatrix} \mathbf{S} \\ \sqrt{\gamma_j}\mathbf{I}_{p\times p} \end{bmatrix}$, $\tilde{\mathbf{S}} = \begin{bmatrix} \mathbf{S} \\ \mathbf{0}_{p\times p} \end{bmatrix}$.

We describe the $\mathbf{B}$ given $\mathbf{A}$ step using the FISTA in Algorithm 1, and the overall SLCA method in Algorithm 2.

We initialize matrix $\mathbf{A}$ using the first k eigenvectors of Φ_{df}. In Algorithm 1, $\tau_\gamma(.)_i$ is a shrinkage operator which is defined for any $\mathbf{q} \in \mathbb{R}^n$ as follows: $\tau_\gamma(\mathbf{q})_i = (|q_i| - \gamma)_+ \operatorname{sgn}(q_i)$. The parameter ϵ is used as a convergence threshold and η_{max} is used as a maximum iteration number for convergence. Moreover, the parameter $\mu_j = \frac{1}{\psi_j}$ in which $\psi_j = 2\lambda_{max}\left(\bar{\mathbf{S}}_j^T\bar{\mathbf{S}}_j\right)$ is the Lipschitz constant of the first two terms in the objective function in (7), where $\lambda_{max}(.)$ denotes the maximum eigenvalue of a matrix. Note that, in the Step 5 of the Algorithm 1, we need the following parameters: $\bar{\mathbf{S}}_j^T\bar{\mathbf{S}}_j = \mathbf{S}^T\mathbf{S} + \gamma\mathbf{I}_{p\times p}$ and $\bar{\mathbf{S}}^T\tilde{\mathbf{S}} = \mathbf{S}^T\mathbf{S}$. Therefore, for solving the optimization problem, we just need the $\mathbf{S}^T\mathbf{S} = \Phi_{df}$. That is, *there is no need to decompose Φ_{df} in practice: Φ_{df} can be used directly in the optimization.* Note that the ridge penalty factor γ should be chosen as a small positive number and large values of δ_j gives sparser solution. Finally, we obtain the k sparse LCs $\{\mathbf{v}_1, ..., \mathbf{v}_k\}$ from $\mathbf{B}$.

Algorithm 1 FISTA Algorithm for SLCA: ($\mathbf{B}$ given $\mathbf{A}$)

```
1: Input: $\mathbf{b}_j^1$: initial random solution, $t_0 = t_1 = 1$, $\mu$, $\epsilon$, $\eta_{max}$, $keep = 1$.
2: while $(\eta \leq \eta_{max})$ and $(keep == 1)$ do
3:     Step $\eta$: $(\eta \geq 1)$ Compute
4:         $y^\eta = \mathbf{b}_j^\eta + \left(\frac{t_{\eta-1}-1}{t_\eta}\right)\left(\mathbf{b}_j^\eta - \mathbf{b}_j^{\eta-1}\right)$,
5:         $\mathbf{b}_j^{\eta+1} = \tau_{\delta_j\mu}\left(y^\eta - 2\mu\bar{\mathbf{S}}^T\left(\bar{\mathbf{S}}y^\eta - \tilde{\mathbf{S}}\mathbf{a}_j\right)\right)$
6:         $t_{\eta+1} = \frac{1+\sqrt{1+4t_\eta^2}}{2}$
7:     if $\frac{\|\mathbf{b}_j^{\eta+1}-\mathbf{b}_j^\eta\|}{\|\mathbf{b}_j^{\eta+1}\|} \leq \epsilon$ then
8:         $keep = 0$
9: Output: $\mathbf{b}_j^{\eta+1}$
```

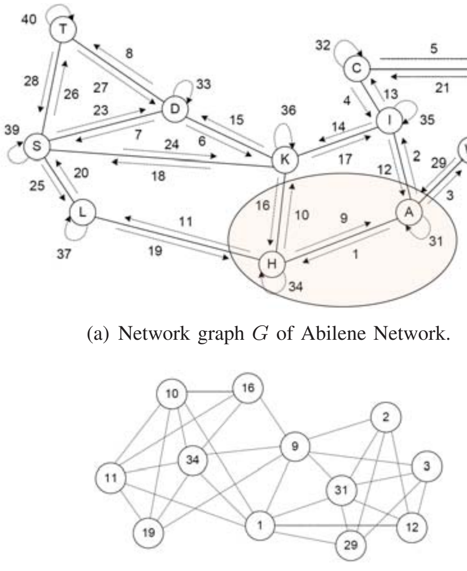
Algorithm 2 Regression Framework of SLCA

```
1: Input: $\mathbf{A}^1 = \left[\mathbf{a}_1^1, \ldots, \mathbf{a}_k^1\right]$: Ordinary principal components, $\epsilon$, $Itr_{max}$, $keep = 1$.
2: while $(\ell \leq Itr_{max})$ and $(keep == 1)$ do
3:     Step $\ell$: $(\ell \geq 1)$ Compute
4:         $\mathbf{b}_j^\ell = FISTA(\mathbf{a}_j^\ell)$, $j = 1, \ldots, k$
5:         $\mathbf{B}^\ell = \left[\mathbf{b}_1^\ell, \ldots, \mathbf{b}_k^\ell\right]$
6:         Take SVD of $\mathbf{S}^T\mathbf{S}\mathbf{B}^\ell = \mathbf{U}\mathbf{D}\mathbf{V}^T$.
7:         Then $\mathbf{A}^{\ell+1} = \mathbf{U}\mathbf{V}^T$
8:     if $\left(\frac{\|\mathbf{A}^{\ell+1}-\mathbf{A}^\ell\|}{\|\mathbf{A}^{\ell+1}\|} \leq \epsilon\right)$ and $\left(\frac{\|\mathbf{B}^\ell-\mathbf{B}^{\ell-1}\|}{\|\mathbf{B}^\ell\|} \leq \epsilon\right)$ then
9:         $keep = 0$
10: Output: $\hat{\mathbf{V}} = [\hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_k]$, where $\hat{\mathbf{v}}_j = \frac{\hat{\mathbf{b}}_j}{\|\hat{\mathbf{b}}_j\|}$, $j = 1, \ldots, k$
```

IV. Graph Construction

A. Graph Structure Representation of Prior Information

In our experiment, we use a standard benchmark dataset: Abilene network dataset [7]. We consider a graph isomorphic to the Abilene network representing the network routers as vertices and the communication links as edges in the graph, respectively. Figure 1(a) shows the *network graph G* of the Abilene network. In this work, we need a graph describing the relationship among the links in G. In particular, the input dataset for the SLCA method is link-level data. Therefore, we compute another graph called *network link graph* from the directed *network graph* using the method described as follows. For a directed and connected network graph $G = (\mathcal{V}, \mathcal{E})$ with self-loop, the corresponding link graph $G_L = (\mathcal{V}', \mathcal{E}')$ is defined such that $|\mathcal{V}'| = |\mathcal{E}|$, and there exists an undirected edge in G_L for each pair of edges in G that shares a common end point which makes it possible to flow data from one edge to other; i.e., $\{(i,j) \in \mathcal{E}'\} \leftrightarrow \{\exists v \in \mathcal{V} | [v_s^i = v_e^j = v] \vee [v_s^j = v_e^i = v]\}$ where $i = (v_s^i, v_e^i) \in \mathcal{E}$, and $j = (v_s^j, v_e^j) \in \mathcal{E}$, $v_s^i \in \mathcal{V}$, $v_e^i \in \mathcal{V}$, $v_s^j \in \mathcal{V}$, $v_e^j \in \mathcal{V}$. An example of corresponding link graph for



(a) Network graph G of Abilene Network.

(b) Link graph G_L for the highlighted part.

Fig. 1: Network graph G of the Abilene Network with link id (numbers written beside the links) and corresponding direction of data flow, and Link graph G_L corresponding to the marked (by highlighted eclipse) portion of G only.

the specific portion marked by a highlighted eclipse in Figure 1(a) is shown in Figure 1(b). We have shown a portion of the link graph because the size of the full link graph is big. Note that, the full step described here corresponds to $\mathcal{F}_1$ in (1).

B. Dual-Input Feature Graph

In this paper, we construct the dual-input feature graph i.e. source-graph using two characteristic parameters. The first one is related to the *Pearson correlation coefficient* (ρ) between the $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$ where $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$ are the row vector of $\mathbf{X} \in \mathbb{R}^{p\times n}$. The second characteristic parameter is related to *minimum distance* (h) between two vertices in the graph G_L. For the network dataset, we propose to consider 'hop count' as distance measure in G_L. For computing the smallest hop count from one vertex to other in G_L, we use the Bellman-Ford shortest path algorithm, and the weight of each edge in graph G_L is unity. We compute the source-graph $\mathcal{G}$ as follows: (i) The set of vertices in source-graph is same as in G_L. (ii) Source-graph has all possible set of edges except self loop. (iii) The weight matrix $\mathbf{W}$ of the source-graph is computed following the equation of Gaussian kernel on both the parameters - data and hop count (i.e. prior information). With respect to our objective of finding anomaly in dataset, the correlation coefficient carries proportional relationship between the vertices whereas the distance metric carries reciprocal relationship. This is because an increase in minimum distance decreases the possibility of exposing similar behavior by the vertices in G_L. Therefore, the weight matrix of the source-graph is computed as follows:

TABLE II: Comparison of average AUC score, Standard Deviation, reduced dimension k (i.e., size of the normal subspace) and optimal setting of model parameters for the different models using 10-fold cross validation procedure on Abilene dataset. Note that k is chosen for the optimal AUC for each method.

Model	Avg. AUC	S.D. AUC	k	Optimal parameter values
PCA	76.19	0.169	4	$k = 4$
LPP	65.23	0.246	5	$k = 5$
RPCA	72.35	0.221	10	$\delta = 0.45$
RPCAG	73.51	0.245	6	$\delta = 0.0445, \gamma = 2^7$
FRPCAG	76.54	0.145	20	$\gamma_1 = 0.0625, \gamma_2 = 0.5$
GLPCA	71.68	0.218	15	$k = 15, \beta = 0.5$
LCA	**87.88**	0.09	22	$k = 22, \theta_c = 0.4, \theta_h = 2, \gamma = 0.016$
SLCA	**88.39**	0.068	19	$k = 19, \theta_c = 0.2, \theta_h = 2, \gamma = 0.004, \delta_j = 0.01, \delta_1 = 1e - 17$

$$\begin{aligned}[w_{i,j}] =& \exp\left(- \frac{(1 - [\|\rho(i,j)\|_1]_+)^2}{\Delta_c^2}\right) \\ & \times \exp\left(- \frac{([\widehat{h}(i,j)]_+)^2}{\Delta_h^2}\right) \end{aligned} \qquad (10)$$

where $[\|\rho(i,j)\|_1]_+$ equals $\|\rho(i,j)\|_1$ if $\|\rho(i,j)\|_1 \geq \theta_c$, and equals unity otherwise; $[\widehat{h}(i,j)]_+$ equals $\widehat{h}(i,j)$ if $\widehat{h}(i,j) \leq \theta_h$, and equals zero otherwise; θ_c and θ_h are constant thresholds; Δ_c and Δ_h are the parameters determining the rate of exponential decay; and $\widehat{h}$ represents the normalized h. Note that this step corresponds to $\mathcal{F}_2$ in (2).

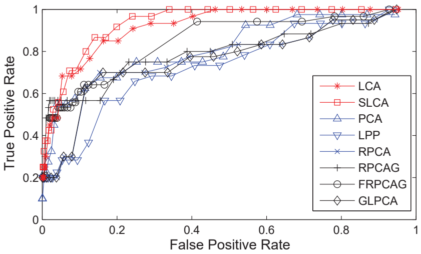


Fig. 2: Comparison of ROC curves under the different models using 10-fold cross validation on Abilene dataset.

V. Experiment and Validation

We consider the real dataset of traffic collected from Abilene network, which is the standard dataset for the volume anomaly detection problem [7], [26]. Details of the dataset can be found in [7]. Following previous work, we use the *link-level* data. The Abilene network has 41 directed links and the dataset has measurements at 2016 time instants. Thus, the input dataset is $\mathbf{X} \in \mathbb{R}^{41\times 2016}$. We follow the same procedure as in [7] to construct the ground truth by using the exponential weighted moving average (EWMA). We compare with the following standard and state-of-art subspace analysis methods: PCA, Locality Preserving Projections(LPP) [12], Robust PCA (RPCA) [21], Robust PCA on Graph (RPCAG) [2], Fast Robust PCA on Graph (FRPCAG) [3], and Graph Laplacian PCA (GLPCA) [1]. Note that RPCAG and FRPCAG are state-of-the-art improvements of PCA using spectral graph regularization.

TABLE III: AUC score comparison using the default feature graph and our dual-input feature graph i.e. source-graph

Dataset	Graph Type	LCA	SLCA
Abilene	Standard Feature Graph [3]	83.21	77.09
	Source-Graph	**87.88**	**88.39**

Using these subspace analysis methods, we obtain different normal and abnormal subspaces. We follow previous work (e.g. [7], [18]) to compute the anomaly score by projecting the normalized link traffic (i.e., $\mathbf{z}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|_2}$) onto the normal and abnormal subspaces, and use the difference in projections as the anomaly score. Using the anomaly score we compute the receiver operating characteristic (ROC) curve and the area under the curve (AUC) for comparing different methods. Table II shows a comparative analysis on the average value of AUC scores and their corresponding standard deviation. We capture the average AUC score using 10-fold cross validation process. We observe that both the versions (LCA and SLCA) of the proposed model outperforms all the existing methods for the real datasets of Abilene network. Figure 2 shows the comparison on ROC curve for the real dataset of Abilene network. The figure once again demonstrates better accuracy on true positive rate with respect to a certain false positive rate.

To understand the effect of the proposed source-graph, we compare the AUC score computed using the proposed models under the standard feature graph [3] and our proposed source-graph. The result is shown in Table III. The result suggests that the traffic measurements are smoother with respect to the source graph compared to the default feature graph. Importantly, when comparing Table II and III, we observe that LCA / SLCA can outperform previous work when using the standard feature graph, demonstrating that our proposed LCA and SLCA are superior in subspace analysis. The use of source graph leads to further improvements.

VI. Conclusion

Within the subspace-analysis anomaly detection framework, we propose to use a regression-based optimization framework to compute the eigenvectors of the normalized graph Laplacian (Laplacian components, LCs), to uncover the normal traffic subspace. The framework allows us to add the lasso penalty and achieve LCs with sparse loadings. Furthermore, we exploit the inclusion of prior information in computing the LCs within the framework. Experiment results suggest that the proposed method is superior in identifying the essential low-dimensional normal traffic subspace from a real Internet traffic dataset, compared to other state-of-the-art. Future work exploits the regression framework for incorporating other regularization.

References

[1] B. Jiang, C. Ding, B. Luo, and J. Tang, "Graph-Laplacian PCA: Closed-form solution and robustness," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3490–34 968.

[2] N. Shahid, V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, "Robust principal component analysis on graphs," in *Proceedings of International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 2812–2820.

[3] N. Shahid, N. Perraudin, V. Kalofolias, G. Puy, and P. Vandergheynst, "Fast robust PCA on graphs," *arXiv:1507.08173v2 [cs.CV] 25 Jan 2016*, pp. 1–17, 2016.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Survey*, vol. 41, no. 3, pp. 15:1–15:58, 2009.

[5] H. Huang, H. Al-Azzawi, and H. Brani, "Network traffic anomaly detection," New Mexico State University, Las Cruces, USA, Tech. Rep., 2014.

[6] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, pp. 83–98, May 2013.

[7] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *Proceedings of ACM SIGCOMM*, 2004, pp. 219–230.

[8] Y.-J. Lee, Y.-R. Yeh, and Y.-C. F. Wang, "Anomaly detection via online oversampling principal component analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 7, pp. 1460–1470, July 2013.

[9] T. Huang, H. Sethu, and N. Kandasamy, "A fast algorithm for detecting anomalous changes in network traffic," in *Proceedings of International Conference on Network and Service Management*, 2015.

[10] S. Gajjar, M. Kulahci, and A. Palazoglu, "Use of sparse principal component analysis (SPCA) for fault detection," in 11^{th} *IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems*, NTNU, Trondheim, Norway, June 2016, pp. 693–698.

[11] D. Brauckhoff, K. Salamatian, and M. May, "Applying pca for traffic anomaly detection: Problems and solutions," in *Proceedings of INFOCOM*, 2009.

[12] X. He and P. Niyogi, "Locality preserving projections," in *Proceedings of NIPS*, 2003, pp. 153–160.

[13] C. Li and H. Li, "Network-constrained regularization and variable selection for analysis of genomic data," *Bioinformatics*, vol. 24, no. 9, pp. 1175–1182, 2008.

[14] D. Cai and J. Han, "Spectral regression: a regression framework for efficient regularized subspace learning," Ph.D. dissertation, University of Illinois at Urbana-Champaign Champaign, IL, USA, 2009, iSBN: 978-1-109-22296-8.

[15] Z. Zhang and K. Zhao, "Low-rank matrix approximation with manifold regularization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1717–1729, 2013.

[16] T. Jin, J. Yu, J. You, K. Zeng, C. Li, and Z. Yu, "Low-rank matrix factorization with multiple hypergraph regularizers," *Pattern Recognition*, vol. 48, no. 3, pp. 1011–1022, March 2015.

[17] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[18] H. E. Egilmez and A. Ortega, "Spectral anomaly detection using graph-based filtering for wireless sensor networks," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 1085–1089.

[19] F. R. K. Chung, *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*, 2nd ed. American Mathematical Society, 1997, vol. 92.

[20] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *Journal of Machine Learning Research*, vol. 16, pp. 2859–2900, 2015.

[21] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, no. 3, pp. 11:1–11:37, May 2011.

[22] V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, "Matrix completion on graphs," *arXiv:1408.1717*, 2014.

[23] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proceedings of NIPS*, vol. 14, 2001, pp. 585–591.

[24] S. Yan, D. Xu, B. Zhang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

[25] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 265–286, 2006.

[26] L. Fillatre, I. Nikiforov, P. Casas, and S. Vaton, "Optimal volume anomaly detection in network traffic flows," in *Proceedings of 16th European Signal Processing Conference*, 2008.