Fast Tensor Factorization for Accurate Internet Anomaly Detection

Kun Xie, *Member, IEEE*, Xiaocan Li, Xin Wang, *Member, IEEE, Member, ACM*, Gaogang Xie, *Member, IEEE*, Jigang Wen, Jiannong Cao, *Fellow, IEEE*, and Dafang Zhang, *Member, IEEE*

***Abstract*— Detecting anomalous traffic is a critical task for advanced Internet management. Many anomaly detection algorithms have been proposed recently. However, constrained by their matrix-based traffic data model, existing algorithms often suffer from low accuracy in anomaly detection. To fully utilize the multi-dimensional information hidden in the traffic data, this paper takes the initiative to investigate the potential and methodologies of performing tensor factorization for more accurate Internet anomaly detection. More specifically, we model the traffic data as a three-way tensor and formulate the anomaly detection problem as a robust tensor recovery problem with the constraints on the rank of the tensor and the cardinality of the anomaly set. These constraints, however, make the problem extremely hard to solve. Rather than resorting to the convex relaxation at the cost of low detection performance, we propose TensorDet to solve the problem directly and efficiently. To improve the anomaly detection accuracy and tensor factorization speed, TensorDet exploits the factorization structure with two novel techniques, sequential tensor truncation and two-phase anomaly detection. We have conducted extensive experiments using Internet traffic trace data Abilene and GÈANT. Compared with the state of art algorithms for tensor recovery and matrix-based anomaly detection, TensorDet can achieve significantly lower false positive rate and higher true positive rate. Particularly, benefiting from our well designed algorithm to reduce the computation cost of tensor factorization, the tensor factorization process in TensorDet is 5 (Abilene) and 13 (GÈANT) times faster than that of the traditional Tucker decomposition solution.**

***Index Terms*— Internet traffic anomaly detection, tensor recovery, tensor completion.**

Manuscript received November 25, 2016; revised June 26, 2017; accepted September 17, 2017; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor C. Joo. Date of publication October 30, 2017; date of current version December 15, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61572184, Grant 61472130, Grant 61472131, and Grant 61725206, and in part by the open project funding of the CAS Key Laboratory of Network Data Science and Technology (Institute of Computing Technology, Chinese Academy of Sciences) under Grant CASNDST201704. *(Corresponding author: Xiaocan Li.)*

K. Xie is with the College of Computer Science and Electronics Engineering, Hunan University, Changsha 410012, China, and also with the Department of Electrical and Computer Engineering, The State University of New York at Stony Brook, Stony Brook, NY 11794 USA (e-mail: xiekun@hnu.edu.cn).

X. Li and D. Zhang are with the College of Computer Science and Electronics Engineering, Hunan University, Changsha 410012, China (e-mail: hnulxc@163.com; dfzhang@hnu.edu.cn).

X. Wang is with the Department of Electrical and Computer Engineering, The State University of New York at Stony Brook, Stony Brook, NY 11794 USA (e-mail: x.wang@stonybrook.edu).

G. Xie and J. Wen are with the CAS Key Laboratory of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: xie@ict.ac.cn; wenjigang@ict.ac.cn).

J. Cao is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: csjcao@comp.polyu.edu.hk).

Digital Object Identifier 10.1109/TNET.2017.2761704

I. Introduction

AN ANOMALY in a data set is defined by Barnett and Lewis as “an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data” [1]. Anomaly detection aims to identify data that do not conform to the patterns exhibited by the data set [2]. Traffic anomalies, such as flash crowds, denial-of-service attacks, port scans, and the spreading of worms, can have detrimental effects on network services. These anomalies often lead to unusual and significant changes of network traffic levels, and the changes can often span multiple links. Detecting and diagnosing these anomalies are critical to both network operators and end users.

Different from traffic characterization which has a large body of literatures, traffic anomaly detection is a difficult problem because one must extract and interpret anomalous patterns from a large amount of traffic data. Recently, many efforts [3]–[11] have been made to develop various anomaly detection algorithms. They usually model the traffic data as a traffic matrix and design the anomaly detection algorithms based on the matrix data. As a matrix can only record two-dimensional information and is not enough to capture the comprehensive correlations hidden in the traffic data, the accuracy of the anomaly detection is often low.

Rather than being constrained by the matrix-based approaches, we propose to model the traffic monitoring data with a multi-way tensor, and investigate the possibility and methodology of exploiting the correlations in a higher dimensional tensor to more robustly detect the network anomaly. Tensors are the higher-order generalization of vectors and matrices. Tensor-based multilinear data analysis has shown that tensor models can take full advantage of the multilinear structures to provide better data understanding and information precision. Tensor-based methods have proven to be good analytical tools for dealing with the multi-dimensional data in a variety of fields.

A popular assumption in anomaly detection is that the normal data have close values, while outliers are far away from the others and lie in the low-density region of the data distribution [12], [13]. In our recent study [14], based on the experiments of real traffic trace, we reveal that traffic data have the features of temporal stability, spatial correlation, and periodicity. With the structure feature and similarity, normal traffic data will reside in a low-dimensional linear subspace and form a low-rank tensor, while the anomalies (outliers) will stay outside this subspace. Based on these observations, we propose a novel *tensor-based anomaly detection* approach with two steps: We first decompose the noisy traffic data into two parts, low-rank normal data and outlier data; We then detect the anomaly by finding the outlier data.

We term the decomposition problem as the robust tensor recovery problem.

Current literature studies on tensor mainly focus on tensor completion to fill in missing data [14]–[17] instead of tensor recovery to obtain complete and normal tensor data from noisy measurements with some corrupted by outliers. Some recent efforts [18]–[20] are made to investigate the tensor recovery problem. They directly extend either RPCA (Robust Principal Component Analysis) of matrix [21] or PCA (Principal Component Analysis) of matrix [4] to the tensor field by unfolding a tensor into matrices, and then utilize the information of different mode in a tensor individually. Thus these approaches are fundamentally still matrix-based and would suffer from the low anomaly detection performance without fully exploiting the tensor pattern and the multilinear information inherent in the data.

For more accurate data recovery and anomaly detection, we directly formulate the robust tensor recovery problem as a tensor approximation problem with constraints on the rank of the tensor and the cardinality of the set of outliers. Unlike [18], [19] that utilize the trace norm of matrix to relax the low-rank tensor constraint, our problem formulation can take advantage of the tensor pattern and correlations among multiple modes to better recover data. Although promising, this method is very challenging to apply for practical anomaly-detection in Internet for several reasons:

- Directly solving the tensor recovery problem is made extremely hard under the constraints of the tensor rank and the cardinality of the outlier set.
- It would involve a high computation overhead thus slower processing speed to complete tensor factorization which is needed to approximate the noisy tensor with a low-rank tensor.
- It is a challenge to decompose the original noisy tensor into a low-rank normal tensor and locate the anomaly from the noisy tensor data as the tensor decomposition is very sensitive to outliers.

In light of the above challenges, we propose a tensor recovery scheme, **TensorDet**, for accurate and fast anomaly detection. To the best of our knowledge, this is the first work that demonstrates the capability of applying tensor factorization to enable robust tensor data recovery for fast and accurate Internet anomaly detection. The main contributions in TensorDet are as follows:

- Despite the difficulty of handling the constraints on tensor rank and set cardinality, we propose a block coordinate descent scheme to solve the tensor recovery problem directly in its original form by iteratively solving two sub problems, a tensor factorization subproblem and an anomaly detection subproblem.
- To find the low rank normal data with much lower computation cost for tensor factorization, we propose a sequential tensor truncating algorithm through the finding of best processing order and the dimension reduction in each tensor truncation step.
- To solve the anomaly detection subproblem and locate the anomaly, we propose a highly efficient two-phase anomaly detection algorithm with a theoretical proof of its ability to accurately detect the anomaly despite the cardinality constraint of the anomaly set.
- Using traffic trace data Abilene [22] and GÈANT [23], we compare our TensorDet with the state of art tensor recovery algorithms and matrix-based outlier detection algorithms. Our results demonstrate that TensorDet can achieve significantly better accuracy performance in terms of False Positive Rate and True Positive Rate. Specifically, benefiting from our sequential tensor truncating algorithm, the tensor factorization process in TensorDet is 5 (Abilene) and 13 (GÈANT) times faster compared with the traditional tensor factorization methods based on Tucker decomposition.

The rest of the paper is organized as follows. Section II presents the related work. The preliminaries of tensor are presented in Section III. We present our system model and problem formulation in Section IV. We describe our sequential tensor truncating algorithm and our two phase anomaly detection algorithm in Section V and Section VI, respectively. Finally, we implement the proposed TensorDet and evaluate the performance using real traffic trace data in Section VII, and conclude the work in Section VIII.

II. Related Work

We are not aware of any other work that provide accurately anomaly detection based on tensor factorization. Following we review some literature work.

A. Traffic Anomaly Detection

Despite a large body of literature on traffic characterization, anomaly detection remains a challenge, and Principal Component Analysis (PCA) [3] is perhaps the best-known statistical-analysis technique. PCA uses an orthogonal transformation to convert possibly correlated observed variables into a set of linearly uncorrelated variables called principal components [24]. PCA is a dimensionality-reduction technique that returns a compact representation of a multi-dimensional dataset by reducing the data to a lower dimensional subspace [25]. Some recent papers that apply PCA to the traffic anomaly detection have shown some promising initial results [4]–[6], [26]–[29]. PCA has also been combined with sketches [7], [8] and distributed monitors [9] to provide more efficient traffic anomaly detection.

To make PCA more robust, Candès *et. al.* [21] proposed to approach Robust PCA (RPCA) via Principal Component Pursuit (PCP), which decomposes a given observation (noisy) matrix $\mathbf{X}$ into a low-rank component $\mathbf{X}'$ and a sparse component $\mathbf{E}$. To make the problem solvable, the work in [30] replaces the matrix rank and the cardinality ($\|\|_0$) functions with their convex surrogates, the nuclear norm and the L_1 norm, and solves the following convex optimization problem

$$\begin{aligned}&\min_{\mathbf{X}',\mathbf{E}} \ \{\|\mathbf{X}'\|_* + \lambda\|\mathbf{E}\|_1\}\\ &st.\ \mathbf{X}' + \mathbf{E} = \mathbf{X} \qquad (1)\end{aligned}$$

where $\|\|_*$ denotes the nuclear norm of a matrix (i.e., the sum of its singular values), $\|\|_1$ denotes the sum of the absolute values of matrix entries, and λ is a positive weighting parameter. Recently, work in [31] proposes a direct robust matrix factorization which aims at minimizing the L_2 error of the low-rank approximation subject to that the number of ignored outliers is small.

Although promising, current anomaly detection techniques are mainly designed based on the matrix data. As a matrix is not enough to capture the comprehensive correlations among the traffic data, the accuracy of the anomaly detection is often low.

Different from current matrix-pattern based anomaly detection which only utilize two-dimensional information, in this work, we propose to investigate the possibility and methodology of exploiting the correlations in a higher dimensional tensor to more robustly detect the network anomaly.

B. Tensor Factorization

Current literature studies on tensor mainly focus on tensor completion instead of tensor recovery. Several Tensor completion algorithms [14]–[16], [32]–[34] are proposed to capture the global structure of data for recovering the missing data in a tensor. Among which, our own studies [14], [16] propose to apply the tensor completion to infer the full Internet traffic data from partial measurements and loss. These previous studies demonstrate that, by exploiting the inherent relationship among higher dimensional data, tensor models can take full advantage of the multilinear structures to provide better data understanding and information precision.

Different from missing data inferring, the aim of tensor recovery is to recover the low-rank tensor from noisy tensor data with some entries corrupted by outliers. Recently, some initial efforts are made [18]–[20] to investigate the tensor recovery problem, where a multi-dimensional tensor is first matricizationed into multiple unfolding matrices. In [18] and [19], a convex relaxation is given by applying the trace norm (the sum of singular values of the optimization matrix) to the unfolding matrices, and finally RPCA [21] is extended to solve the tensor recovery problem. In [20], a PCA is applied to the unfolded matrices to solve the tensor recovery problem. Relying on matricization and the relaxation technique based on matrix, these methods are still matrix-based, and the information of different modes in a tensor is individually utilized. Thus these approaches can not fully utilize the multidimentional information hidden in the tensor pattern to robustly recover the tensor and accurately detect anomaly.

Different from current tensor-based techniques, this paper focuses on robust tensor recovery problem and proposes TensorDet as a simple and effective way for faster low-rank tensor factorization and more accurate outlier detection. We start from the fundamental notion of outliers and use a direct formulation (with constraints in the direct form of low-rank tensor and the cardinality of outliers) to address the problem. Our evaluation results demonstrate that the performance of TensorDet is significantly better than that of the state-of the art peer algorithms based on relaxation. We also propose some novel techniques to speed up the processes of tensor factorization and anomaly detection.

III. Preliminaries of Tensor

In this section, we introduce some basic concepts related to the tensor. The notations used in this paper are described as follows. Scalars are denoted by lowercase letters $(a, b, \cdots)$, vectors are written in boldface lowercase $(\mathbf{a}, \mathbf{b}, \cdots)$, and matrices are represented with boldface capitals $(\mathbf{A}, \mathbf{B}, \cdots)$. Higher-order tensors are written as calligraphic letters $(\mathcal{X}, \mathcal{Y}, \cdots)$. The elements of a tensor are denoted by the symbolic name of the tensor with indexes as subscripts. For example, the ith entry of a vector $\mathbf{a}$ is denoted by a_i, element (i, j) of a matrix $\mathbf{A}$ is denoted by a_{ij}, and element (i, j, k) of a third-order tensor $\mathcal{X}$ is denoted by x_{ijk}.

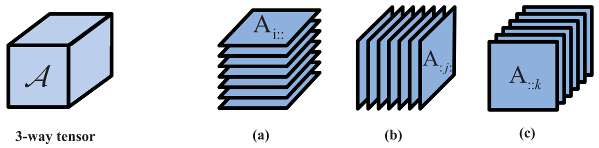


Fig. 1. Tensor slices- (a) The horizontal $(\mathbf{A}_{i::})$, (b) lateral $(\mathbf{A}_{:j:})$ and (c) frontal $(\mathbf{A}_{::k})$ slices of a 3-way tensor respectively.

Definition 1: A *tensor* is a multidimensional array, and is a higher-order generalization of a vector (first-order tensor) and a matrix (second-order tensor). A d-way or dth-order tensor (denoted as $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_d}$) is an element of the tensor product of d vector spaces, where d is the order of $\mathcal{A}$, also called way or mode.

The element of $\mathcal{A}$ is denoted by $a_{i_1, i_2, \cdots, i_d}$, $i_n \in \{1, 2, \cdots, I_n\}$ with $1 \leq n \leq d$.

Definition 2: Slices are two-dimensional sections of a tensor, and are defined by fixing all but two indexes.

A 3-way tensor $\mathcal{A}$ has horizontal, lateral and frontal slices shown in Fig.1, which are denoted by $\mathbf{A}_{i::}$, $\mathbf{A}_{:j:}$ and $\mathbf{A}_{::k}$, respectively.

Definition 3: Given a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_d}$, a mode-k vector $\mathbf{v}$ is defined as the vector that is obtained by fixing all indices of $\mathcal{A}$ but varying the mode-k index:$\mathbf{v} = \mathcal{A}_{\mathbf{i_1}, \cdots, \mathbf{i_{k-1}}, :, \mathbf{i_{k+1}}, \cdots, \mathbf{i_d}}$ with $i_j (j \neq k)$ a fixed value. We refer to the set of all mode-k vectors of $\mathcal{A}$ as the mode-k vector space. The mode-k unfolding, or matricization [15], of $\mathcal{A}$, denoted by $\mathbf{A}_{(k)}$, is an $I_k \times \prod_{i \neq k} I_i$ matrix whose columns are all possible mode-k vectors.

For a dth-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_d}$, the mode-$k$ unfolded matrix $\mathbf{A}_{(k)} \in R^{I_k \times \prod_{i \neq k} I_i}$ contains the tensor element $a_{i_1, i_2, \cdots, i_d}$, $i_n \in \{1, 2, \cdots, I_n\}$ at the position in the unfolding matrix with its row index i_k and column index j equal to

$$j = 1 + \sum_{n=1, n \neq k}^{d} \left[(i_n - 1) \prod_{m=1, m \neq k}^{n-1} I_m \right] \tag{2}$$

Fig. 2 shows an unfolding procedure of a 3rd-order tensor, which involves the tensor dimensions I_1, I_2, I_3 in a cyclic way. Fig. 3 shows an example of a tensor $\mathcal{A} \in \mathbb{R}^{3 \times 4 \times 2}$, in which the matrix unfolding $\mathbf{A}_{(2)}$ is given.

Definition 4 (Multilinear-Rank [35]): The multilinear-rank of a dth-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_d}$ is d-tuple of the ranks of unfolded matrices, that is,

$$\left(rank \left(\mathbf{A}_{(1)} \right), rank \left(\mathbf{A}_{(2)} \right), \ldots, rank \left(\mathbf{A}_{(d)} \right) \right). \tag{3}$$

Definition 5: The k-mode (matrix) product of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_d}$ with a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_k}$ is a tensor denoted by $\mathcal{A} \times_k \mathbf{U}$, whose mode-$k$ unfolded matrix is equal to the matrix $\mathbf{U}$ times the mode-k unfolded matrix $\mathbf{A}_{(k)}$:

$$\mathcal{B} = \mathcal{A} \times_k \mathbf{U} \Leftrightarrow \mathbf{B}_{(k)} = \mathbf{U} \mathbf{A}_{(k)} \tag{4}$$

The dimension of the k-mode product is $I_1 \times \cdots \times I_{k-1} \times J \times I_{k+1} \times \cdots \times I_d$, and the element is

$$\begin{aligned} &(\mathcal{A} \times_k \mathbf{U})_{i_1, \cdots, i_{k-1}, j, i_{k+1} \cdots i_d} \\ &\qquad = \sum_{i_k=1}^{I_k} a_{i_1, \cdots, i_{k-1}, i_k, i_{k+1}, \cdots, i_d} u_{j, i_k}. \end{aligned} \tag{5}$$

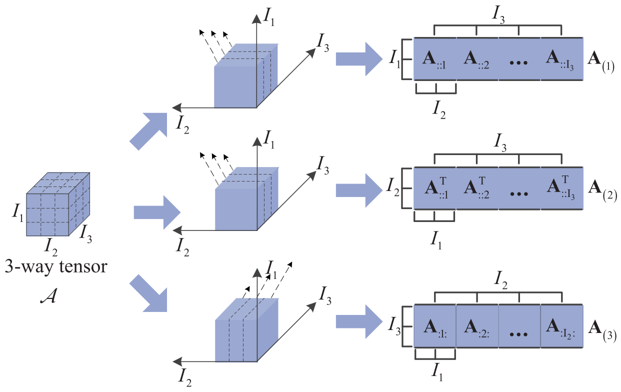

I_1
I_2
I_3
3-way tensor
$\mathcal{A}$
$\mathbf{A}_{::1}$ $\mathbf{A}_{::2}$ ... $\mathbf{A}_{::I_3}$ $\mathbf{A}_{(1)}$
$\mathbf{A}^T_{::1}$ $\mathbf{A}^T_{::2}$ ... $\mathbf{A}^T_{::I_3}$ $\mathbf{A}_{(2)}$
$\mathbf{A}_{:1:}$ $\mathbf{A}_{:2:}$ ... $\mathbf{A}_{:I_2:}$ $\mathbf{A}_{(3)}$

Fig. 2. Unfolding of the $(I_1 \times I_2 \times I_3)-$ tensor $\mathcal{A}$ to the $(I_1 \times I_2 I_3)-$ matrix $\mathbf{A}_{(1)}$, the $(I_2 \times I_3 I_1)-$ matrix $\mathbf{A}_{(2)}$, and the $(I_3 \times I_1 I_2)-$ matrix $\mathbf{A}_{(3)}$.

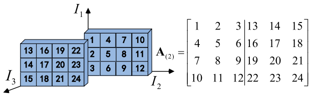

I_1
I_2
I_3
1 4 7 10
2 5 8 11
3 6 9 12
13 16 19 22
14 17 20 23
15 18 21 24
$\mathbf{A}_{(2)} = \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{array}\right]$

Fig. 3. A tensor $\mathcal{A} \in R^{3\times4\times2}$.

As an example, let $\mathcal{A}$ be the tensor defined in Fig.3 and let $\mathbf{U} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{bmatrix}$. Then, the product $\mathcal{B} = \mathcal{A} \times_2 \mathbf{U} \in \mathbb{R}^{3\times2\times2}$ is a tensor with its mode-2 unfolding matrix being

$$\mathbf{B}_{(2)} = \left[\begin{array}{ccc|ccc} 118 & 134 & 150 & 310 & 326 & 342 \\ 140 & 160 & 180 & 380 & 400 & 420 \end{array}\right] \tag{6}$$

The k-mode matrix product satisfies the following properties.

$$\mathcal{A} \times_m \mathbf{X} \times_n \mathbf{Y} = \mathcal{A} \times_n \mathbf{Y} \times_m \mathbf{X}\,(m \neq n) \tag{7}$$

and

$$\mathcal{A} \times_n \mathbf{X} \times_n \mathbf{Y} = \mathcal{A} \times_n (\mathbf{YX}) \tag{8}$$

Definition 6 (Tucker Decomposition [35]): As shown in Fig. 4, a third-order tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times I_3}$ can be decomposed into three factor matrices $\mathbf{X} \in \mathbb{R}^{I_1\times I_1}$, $\mathbf{Y} \in \mathbb{R}^{I_2\times I_2}$ and $\mathbf{Z} \in \mathbb{R}^{I_3\times I_3}$ as well as a third order core tensor $\mathcal{S} \in \mathbb{R}^{I_1\times I_2\times I_3}$ as follows

$$\mathcal{A} = \mathcal{S} \times_1 \mathbf{X} \times_2 \mathbf{Y} \times_3 \mathbf{Z} \tag{9}$$

That is

$$a_{i,j,k} = \sum_{i'=1}^{I_1} \sum_{j'=1}^{I_2} \sum_{k'=1}^{I_3} s_{i',j',k'} x_{i,i'} y_{j,j'} z_{k.k'} \tag{10}$$

The Tucker decomposition can be built from several SVDs, as follows [36], [37]:

Given a tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times I_3}$, construct the mode-k unfolding $\mathbf{A}_{(k)}$.

Compute the singular value decomposition $\mathbf{A}_{(k)} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$, let $\mathbf{X} = \mathbf{U}_1$, $\mathbf{Y} = \mathbf{U}_2$, $\mathbf{Z} = \mathbf{U}_3$.

The core tensor $\mathcal{S}$ is then the projection of $\mathcal{A}$ onto the tensor basis formed by the factor matrices $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$, i.e., $\mathcal{S} = \mathcal{A} \times_1 \mathbf{X}^T \times_2 \mathbf{Y}^T \times_3 \mathbf{Z}^T$.

Definition 7 (Multilinear Orthogonal Projections): An orthogonal projector [38], [39] is a linear transformation $\mathbf{P}$ that projects a vector $\mathbf{u} \in \mathbb{R}^n$ into a vector space $\mathcal{U} \subseteq \mathbb{R}^n$ such that the residual $\mathbf{u} - \mathbf{Pu}$ is orthogonal to $\mathcal{U}$. Such a projector can always be represented in the matrix form as $\mathbf{P} = \mathbf{UU^T}$, assuming that the columns of U form an orthonormal basis for the vector space $\mathcal{U}$. De Silva and Lim [40] state that $\phi = (\phi_1, \phi_2, \ldots, \phi_d)$ is a multilinear orthogonal projection from the tensor space $\mathcal{V} = \mathcal{V}_1 \times \mathcal{V}_2 \times \cdots \times \mathcal{V}_d$ onto the tensor subspace $\mathcal{U} = \mathcal{U}_1 \times \mathcal{U}_2 \times \cdots \times \mathcal{U}_d \subseteq \mathcal{V}$. In this paper, we deal with an orthogonal projector from $\mathbb{R}^{I_1} \times \cdots \times \mathbb{R}^{I_{k-1}} \times \mathbb{R}^{I_k} \times \mathbb{R}^{I_{k+1}} \times \cdots \times \mathbb{R}^{I_d}$ into the subspace $\mathbb{R}^{I_1} \times \cdots \times \mathbb{R}^{I_{k-1}} \times \mathcal{U}_k \times \mathbb{R}^{I_{k+1}} \times \cdots \times \mathbb{R}^{I_d}$ exclusively. This multilinear orthogonal projection is given by

$$\pi_k \mathcal{A} = \mathcal{A} \times_k \left(\mathbf{U}_k \mathbf{U}_k^T\right) \tag{11}$$

with $\mathcal{A} \in \mathbb{R}^{I_1\times\cdots\times I_d}$. In (11), the columns of U_k form an orthonomal basis of the mode-k vector space of $\mathcal{A}$. The subscript of the projector π_k indicates that it projects orthogonally along mode k. The projector satisfies the following properties. Every projector π_k is idempotent, $\pi_k \pi_k \mathcal{A} = \pi_k \mathcal{A}$, and any two projectors commute, $\pi_i \pi_j \mathcal{A} = \pi_j \pi_i \mathcal{A}$. The orthogonal complement of π_k can be characterized explicitly by

$$\pi_k^{\perp} \mathcal{A} = (1 - \pi_k)\,\mathcal{A} = \mathcal{A} \times_k \left(I - \mathbf{U}_k \mathbf{U}_k^T\right) \tag{12}$$

where I is the identity matrix of a suitable dimension.

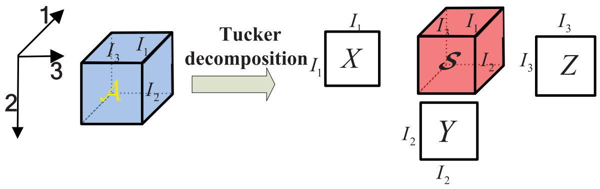


Fig. 4. Tucker decomposition: $\mathcal{A} = \mathcal{S} \times_1 \mathbf{X} \times_2 \mathbf{Y} \times_3 \mathbf{Z}$.

IV. Models and Problems

In this section, we first introduce our system model, then the problem.

A. Traffic Tensor Model

In our recent study [14], based on the experiments of real traffic trace, we reveal that traffic data have the features of temporal stability, spatial correlation, and periodicity. To fully exploit these traffic features for accurate anomaly detection, we model the traffic data as a 3-way tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times I_3}$, where I_3 corresponds to the number of OD pairs with $I_3 = N \times N$ (N is the number of nodes in the network), and there are I_1 days to consider with each day having I_2 time slots. Fig.5 uses Abilene trace data [22] as an example to illustrate this model. The traffic data are collected between 144 OD pairs in 168 days, and the measurements are made every 5 minutes which corresponds to 288 time slots every day. Therefore, the trace data can be modeled as a 3-way tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times I_3}$ with $I_1 = 168$, $I_2 = 288$, and $I_3 = 144$.

B. Problem Formulation

The data captured by a traffic tensor tend to be noisy and are subject to outliers and arbitrary corruptions. For more accurate detection of the outliers ad corruptions, we will exploit the structure and correlation of traffic data in all the dimensions. Specifically, we propose to design robust detection algorithms by applying the PCA technique to the tensor data for more accurate traffic anomaly detection.

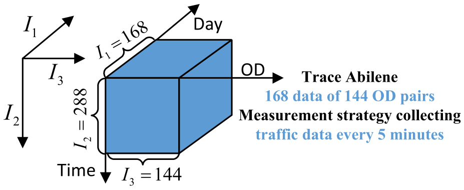


Fig. 5. Traffic tensor model.

The extension of the PCA technique to the higher dimensional tensor data, however, is highly no trivial. Intuitively, the problem in (1) can be directly extended to formulate the PCA problem for the tensor data:

$$\begin{aligned} &\min_{\mathcal{X}',\mathcal{E}} \ \{\|\mathcal{X}'\|_* + \lambda\|\mathcal{E}\|_1\} \\ &st.\ \mathcal{X}' + \mathcal{E} = \mathcal{X} \end{aligned} \tag{13}$$

where $\mathcal{X}', \mathcal{E}, \mathcal{X} \in \mathbb{R}^{I_1\times I_2\times I_3}$, $\|\|_*$ denotes the nuclear norm of a tensor, and $\|\|_1$ denotes the sum of the absolute values of tensor entries. However, the nuclear norm of a higher dimensional tensor is not well defined.

To achieve highly accurate anomaly detection, we propose to solve the Robust tensor PCA problem in its direct form. In (14), we decompose a given observation tensor $\mathcal{X}$ into a low-rank component $\mathcal{X}'$ and a sparse component $\mathcal{E}$ by solving the optimization problem below:

$$\begin{aligned} &\min_{\mathcal{X}',\mathcal{E}} \ \|(\mathcal{X}-\mathcal{E})-\mathcal{X}'\|_F \\ &s.t.\ rank(\mathcal{X}') \le rank(r_1,r_2,r_3) \\ &\qquad \|\mathcal{E}\|_0 \le \varepsilon, \end{aligned} \tag{14}$$

where $rank(\mathcal{X}') \le rank(r_1,r_2,r_3)$ means $rank(X'_{(1)}) \le r_1$, $rank(X'_{(2)}) \le r_2$ and $rank(X'_{(3)}) \le r_3$, and $\mathcal{E}$ is the tensor of outliers, whose number of non-zero entries (i.e., anomalies) is smaller than ε. The $rank(r_1,r_2,r_3)$ is the maximum rank of $\mathcal{X}'$, and can be set based on rank from the recent recovered data, plus some extra value to increase the accuracy of recovering $\mathcal{X}'$.

With $(\mathcal{X}-\mathcal{E})$, we exclude the outliers $\mathcal{E}$ from the observation tensor as long as $\mathcal{E}$ is sufficiently sparse with the number of outliers limited to be not too large, and we do not need to know the actual number of outliers.

C. Solution Overview

The problem (14) involves the tensor rank and the L_0-norm (i.e., the cardinality of the outlier set), and is very difficult to solve. As the constraints of $\mathcal{X}'$ and $\mathcal{E}$ are independent, the problem is decomposable. Taking advantage of this property, we propose to adopt the block coordinate descent strategy, and divide the original problem into two subproblems: a tensor factorization subproblem (15) and an anomaly detection subproblem (41), which can be alternately solved until the solution converges as shown in Algorithm 1.

In the tensor factorization subproblem (Eq.(15)), we first fix the current estimate of outliers $\mathcal{E}$ and exclude them from $\mathcal{X}$ to obtain the "clean" data $\mathcal{C}$, and then approximate $\mathcal{C}$ using $\mathcal{X}'$. In the anomaly detection subproblem (Eq.(41)), we update the outliers $\mathcal{E}$ based on the error $\mathcal{B} = \mathcal{X} - \mathcal{X}'$.

In following two Sections, we provides our key techniques to solve these two subproblems.

Algorithm 1 Robust Tensor PCA

```
Input: $\mathcal{X}$ the traffic tensor
    rank$(r_1, r_2, r_3)$ the maximal rank of the traffic tensor
    $\varepsilon$ the maximal number of outliers $\mathcal{E}$ the initial outliers
Output:
while not converged do
    solve the tensor factorization subproblem
        $\mathcal{X}' = \arg\min_{\mathcal{X}'} \|\mathcal{C}-\mathcal{X}'\|_F$
        $s.t.\ \mathcal{C} = \mathcal{X} - \mathcal{E}$
        $rank(\mathcal{X}') \le rank(r_1,r_2,r_3)$    (15)
    solve the anomaly detection subproblem
        $\mathcal{E} = \arg\min_{\mathcal{E}} \|\mathcal{B}-\mathcal{E}\|_F$
        $s.t.\ \mathcal{B} = \mathcal{X} - \mathcal{X}'$
        $\|\mathcal{E}\|_0 \le \varepsilon$    (16)
end while
```

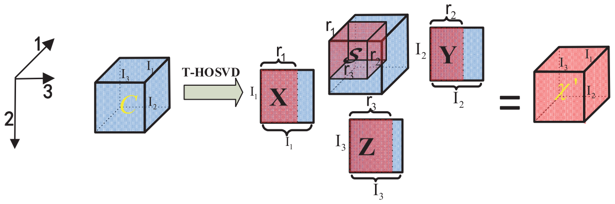


Fig. 6. T-HOSVD problem: to approximate the tensor $\mathcal{C}$ with a tensor $\mathcal{X}'$ of $rank(r_1, r_2, r_3)$.

V. Tensor Factorization

The subproblem in (15) is defined to approximate the tensor $\mathcal{C}$ with a tensor $\mathcal{X}'$ of $rank(r_1,r_2,r_3)$, and we call it *truncating the higher-order singular value decomposition problem (T-HOSVD), which can be explained through Fig.6.*

A straight-forward way to solve T-HOSVD is to first calculate the tucker decomposition (Definition 6) to obtain the factor matrices $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$, and then restrict tensor $\mathcal{C}$'s factor matrices $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ to their first r_1, r_2, and r_3 columns, and restrict the core tensor to $\mathcal{S}' = [\![s_{ijk}]\!]_{i,j,k=1}^{r_1,r_2,r_3}$. An important step in the above procedure is to obtain the factor matrices. When the size of tensor becomes large, however, the computation cost to derive the factor matrices is huge.

In this paper, we seek a novel way to solve the T-HOSVD problem through sequential tensor truncation thus sequential size reduction to largely reduce the computation cost.

A. Problem Transformation

Given a tensor $\mathcal{C}$ and the truncation rank (r_1,r_2,r_3), the approximation $\mathcal{X}'$ can be obtained by an orthogonal projection onto the tensor basis of $\mathcal{C}$, represented by the truncated factor matrices of the tensor $\mathcal{C}$:

$$\begin{aligned} \mathcal{X}' &= \pi_1\pi_2\pi_3\mathcal{C} \\ &= \mathcal{C} \times_1 \left(\mathbf{U}_1\mathbf{U}_1^T\right) \times_2 \left(\mathbf{U}_2\mathbf{U}_2^T\right) \times_3 \left(\mathbf{U}_3\mathbf{U}_3^T\right) \\ &= \mathcal{S} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3 \end{aligned} \tag{17}$$

where the truncated factor matrix $\mathbf{U}_k \in \mathbb{R}^{I_k \times r_k}$ is a column orthonormal matrix. In (17), the core tensor $\mathcal{S}$ is defined as

$$\mathcal{S} = \mathcal{C} \times {}_1\mathbf{U}_1^T \times {}_2\mathbf{U}_2^T \times {}_3\mathbf{U}_3^T \tag{18}$$

where $\mathcal{S} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$.

Based on (17), the problem (15) can be rewritten as

$$\min_{\pi_1,\pi_2,\pi_3} \|\mathcal{C} - \pi_1\pi_2\pi_3\mathcal{C}\|_F^2 \tag{19}$$

To solve the T-HOSVD problem, we need to calculate the optimal π_1, π_2, and π_3.

According to the properties of multilinear orthogonal projections, we know that

$$\begin{aligned}\pi_1\pi_2\pi_3\mathcal{C} &= \pi_1\pi_3\pi_2\mathcal{C} = \pi_2\pi_1\pi_3\mathcal{C} = \pi_2\pi_3\pi_1\mathcal{C} \\ &= \pi_3\pi_1\pi_2\mathcal{C} = \pi_3\pi_2\pi_1\mathcal{C}\end{aligned} \tag{20}$$

In this paper, we denote the order in which the modes are processed as a sequence P, which is a permutation of $\{1,2,3\}$. For example, with $\pi_1\pi_2\pi_3\mathcal{C}$, $P = [1,2,3]$. We have

$$\begin{aligned}&\mathcal{C} - \pi_1\pi_2\pi_3\mathcal{C} \\ &\quad = \mathcal{C} - \pi_{P_1}\pi_{P_2}\pi_{P_3}\mathcal{C} \\ &\quad = (\mathcal{C} - \pi_{P_1}\mathcal{C}) + (\pi_{P_1}\mathcal{C} - \pi_{P_2}\pi_{P_1}\mathcal{C}) \\ &\qquad + (\pi_{P_2}\pi_{P_1}\mathcal{C} - \pi_{P_3}\pi_{P_2}\pi_{P_1}\mathcal{C}) \\ &\quad = \pi_{P_1}^{\perp}\mathcal{C} + \pi_{P_2}^{\perp}\pi_{P_1}\mathcal{C} + \pi_{P_3}^{\perp}\pi_{P_1}\pi_{P_2}\mathcal{C}\end{aligned} \tag{21}$$

Theorem 1: Applying the decomposition in (21) to any permutation P of $\{1,2,3\}$, the approximation error can be represented by

$$\begin{aligned}\|\mathcal{C} - \pi_{P_1}\pi_{P_2}\pi_{P_3}\mathcal{C}\|_F^2 &= \left\|\pi_{P_1}^{\perp}\mathcal{C}\right\|_F^2 + \left\|\pi_{P_2}^{\perp}\pi_{P_1}\mathcal{C}\right\|_F^2 \\ &\quad + \left\|\pi_{P_3}^{\perp}\pi_{P_1}\pi_{P_2}\mathcal{C}\right\|_F^2\end{aligned} \tag{22}$$

Proof: According to (21), we have

$$\begin{aligned}&\|\mathcal{C} - \pi_{P_1}\pi_{P_2}\pi_{P_3}\mathcal{C}\|_F^2 \\ &\qquad = \left\|\pi_{P_1}^{\perp}\mathcal{C} + \pi_{P_2}^{\perp}\pi_{P_1}\mathcal{C} + \pi_{P_3}^{\perp}\pi_{P_1}\pi_{P_2}\mathcal{C}\right\|_F^2\end{aligned} \tag{23}$$

which can be further deduced as

$$\begin{aligned}&\|\mathcal{C} - \pi_{P_1}\pi_{P_2}\pi_{P_3}\mathcal{C}\|_F^2 \\ &\quad = \left\|\pi_{P_1}^{\perp}\mathcal{C}\right\|_F^2 + \left\|\pi_{P_2}^{\perp}\pi_{P_1}\mathcal{C}\right\|_F^2 + \left\|\pi_{P_3}^{\perp}\pi_{P_1}\pi_{P_2}\mathcal{C}\right\|_F^2 \\ &\qquad + 2\left\|\pi_{P_1}^{\perp}\mathcal{C}\pi_{P_2}^{\perp}\pi_{P_1}\mathcal{C}\right\|_F + 2\left\|\pi_{P_1}^{\perp}\mathcal{C}\pi_{P_3}^{\perp}\pi_{P_1}\pi_{P_2}\mathcal{C}\right\|_F \\ &\qquad + 2\left\|\pi_{P_2}^{\perp}\pi_{P_1}\mathcal{C}\pi_{P_3}^{\perp}\pi_{P_1}\pi_{P_2}\mathcal{C}\right\|_F.\end{aligned} \tag{24}$$

Moreover, according to the definition of multilinear orthogonal projection, $\pi_{P_1}^{\perp}\mathcal{C}$, $\pi_{P_2}^{\perp}\pi_{P_1}\mathcal{C}$, and $\pi_{P_3}^{\perp}\pi_{P_1}\pi_{P_2}\mathcal{C}$ are orthogonal with each other, and we have $\left\|\pi_{P_1}^{\perp}\mathcal{C}\pi_{P_2}^{\perp}\pi_{P_1}\mathcal{C}\right\|_F = 0$, $\left\|\pi_{P_1}^{\perp}\mathcal{C}\pi_{P_3}^{\perp}\pi_{P_1}\pi_{P_2}\mathcal{C}\right\|_F = 0$, and $\left\|\pi_{P_2}^{\perp}\pi_{P_1}\mathcal{C}\pi_{P_3}^{\perp}\pi_{P_1}\pi_{P_2}\mathcal{C}\right\|_F = 0$. With this orthogonality, the approximation error can be represented by (22). ■

Given a tensor $\mathcal{C} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, the singular value decomposition (SVD) of mode-k unfolded matrix $C_{(k)} = U_k\Sigma_k V_k$, where Σ_k is a diagonal matrix with the diagonal elements (i.e. the singular values) organized in the decreasing order (i.e. $\Sigma_k = diag\left(\sigma_1, \sigma_2, \cdots, \sigma_{r_{(k)}}, 0, \cdots, 0\right)$). If the truncation rank (r_1, r_2, r_3) is set to be larger than the ranks of the corresponding unfolded matrices $r_{(1)}$, $r_{(2)}$ and $r_{(3)}$, respectively, the truncated tensor using the truncation rank (r_1, r_2, r_3) will preserve all data variability. Thus we have $\|\mathcal{C} - \pi_1\pi_2\pi_3\mathcal{C}\|_F^2 = 0$.

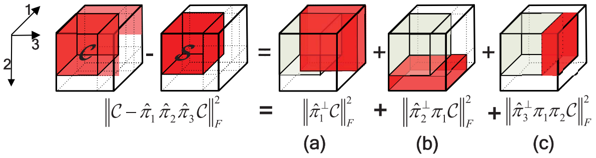

1
2
3
$\mathcal{C}$
$\mathcal{S}$
-
=
+
+
$\left\|\mathcal{C} - \hat{\pi}_1\hat{\pi}_2\hat{\pi}_3\mathcal{C}\right\|_F^2$
=
$\left\|\hat{\pi}_1^{\perp}\mathcal{C}\right\|_F^2$
+
$\left\|\hat{\pi}_2^{\perp}\pi_1\mathcal{C}\right\|_F^2$
+
$\left\|\hat{\pi}_3^{\perp}\pi_1\pi_2\mathcal{C}\right\|_F^2$
(a)
(b)
(c)

Fig. 7. Approximation error $\|\mathcal{C} - \pi_1\pi_2\pi_3\mathcal{C}\|_F^2$ for a third-order tensor with $P = [1,2,3]$. Red shaded area in Figure Fig.7(a) corresponds to $\left\|\pi_1^{\perp}\mathcal{C}\right\|_F^2$, in Fig.7(b) it corresponds to $\left\|\pi_2^{\perp}\pi_1\mathcal{C}\right\|_F^2$, and in Fig.7(c) to $\left\|\pi_3^{\perp}\pi_1\pi_2\mathcal{C}\right\|_F^2$.

From Fig.7, we can visualize the approximation error for a third-order tensor with $P = [1,2,3]$. The cube is partitioned into octants.

From Eq(7), we can observe that any processing order P may correspond to the summation of different octants, but the resulting approximation error is clearly the same. However, as we will analyze in Section V-C, different processing orders may lead to big differences in the computation cost.

B. Sequentially Truncated HOSVD

From this section, we present our techniques to minimize Eq(22) and find the optimal processing order to largely reduce the computation cost.

According to Eq(22), the problem (19) can be further expressed as

$$\begin{aligned}&\min_{\pi_1,\pi_2,\pi_3} \|\mathcal{C} - \pi_1\pi_2\pi_3\mathcal{C}\|_F^2 \\ &= \min_{\pi_1,\pi_2,\pi_3} \left(\left\|\pi_{P_1}^{\perp}\mathcal{C}\right\|_F^2 + \left\|\pi_{P_2}^{\perp}\pi_{P_1}\mathcal{C}\right\|_F^2 + \left\|\pi_{P_3}^{\perp}\pi_{P_1}\pi_{P_2}\mathcal{C}\right\|_F^2\right) \\ &= \min_{\pi_1}\left[\left\|\pi_{P_1}^{\perp}\mathcal{C}\right\|_F^2 + \min_{\pi_2}\left[\left\|\pi_{P_2}^{\perp}\pi_{P_1}\mathcal{C}\right\|_F^2 \right.\right. \\ &\qquad\qquad\qquad \left.\left. + \min_{\pi_3}\left\|\pi_{P_3}^{\perp}\pi_{P_1}\pi_{P_2}\mathcal{C}\right\|_F^2\right]\right]\end{aligned} \tag{25}$$

According to Eq(25), we propose a sequentially truncated HOSVD solution with the following steps:

$$\textit{Step 1: } \pi_{P_1}^* = \underset{\pi_{P_1}}{\operatorname{argmin}} \left\|\pi_{P_1}^{\perp}\mathcal{C}\right\|_F^2 \tag{26}$$

$$\textit{Step 2: } \pi_{P_2}^* = \underset{\pi_{P_2}}{\operatorname{argmin}} \left\|\pi_{P_2}^{\perp}\pi_{P_1}^*\mathcal{C}\right\|_F^2 \tag{27}$$

$$\textit{Step 3: } \pi_{P_3}^* = \underset{\pi_{P_3}}{\operatorname{argmin}} \left\|\pi_{P_3}^{\perp}\pi_{P_1}^*\pi_{P_2}^*\mathcal{C}\right\|_F^2 \tag{28}$$

The details of each step are as follows:

1) First Step: Based on the definition of multilinear orthogonal projection in Eq(11), the problem (26) can be transformed into

$$\mathbf{U}_{P_1} = \underset{\mathbf{U}_{P_1}}{\arg\min} \left\|\mathcal{C} \times {}_{P_1}\left(I - \mathbf{U}_{P_1}\mathbf{U}_{P_1}^T\right)\right\|_F^2 \tag{29}$$

The problem (29) can be solved by a truncated SVD of the mode-P_1 unfolding of the tensor $\mathcal{C}$ with $\mathbf{U}_{P_1} \in \mathbb{R}^{I_{P_1} \times r_{P_1}}$ obtained from following operations.

$$\mathbf{C}_{(P_1)} = \begin{bmatrix}\mathbf{U}_{P_1} & \tilde{\mathbf{U}}_{P_1}\end{bmatrix}\begin{bmatrix}\Sigma_{P_1} & \\ & \tilde{\Sigma}_{P_1}\end{bmatrix}\begin{bmatrix}\mathbf{V}_{P_1}^T \\ \tilde{\mathbf{V}}_{P_1}^T\end{bmatrix} \tag{30}$$

After we obtain $\mathbf{U}_{P_1}$, the solution of problem (26) is $\pi_{P_1}^* = \mathbf{U}_{P_1}\mathbf{U}_{P_1}^T$. Let $S^{(1)}$ denote the truncated core tensor of $\pi_{P_1}^*\mathcal{C}$.

We have

$$\mathcal{S}^{(1)} = \mathcal{C} \times_{P_1} \mathbf{U}_{P_1}^T \tag{31}$$

That is,

$$\begin{aligned}
\mathbf{S}_{(P_1)}^{(1)} &= \mathbf{U}_{P_1}^T \mathbf{C}_{(P_1)} \\
&= \mathbf{U}_{P_1}^T \left[\mathbf{U}_{P_1} \quad \tilde{\mathbf{U}}_{P_1} \right] \begin{bmatrix} \sum_{P_1} & \\ & \tilde{\sum}_{P_1} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{P_1}^T \\ \tilde{\mathbf{V}}_{P_1}^T \end{bmatrix} \\
&= [I \quad 0] \begin{bmatrix} \sum_{P_1} & \\ & \tilde{\sum}_{P_1} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{P_1}^T \\ \tilde{\mathbf{V}}_{P_1}^T \end{bmatrix} \\
&= \sum_{P_1} \mathbf{V}_{P_1}^T
\end{aligned} \tag{32}$$

where $\sum_{P_1} \in \mathbb{R}^{r_{P_1} \times r_{P_1}}$ and $\mathbf{V}_{P_1}^T \in \mathbb{R}^{r_{P_1} \times \left(I_{P_2} \times I_{P_3}\right)}$. Therefore, $\mathbf{S}_{(P_1)}^{(1)}$ can be obtained simply by scaling the right singular vectors with the corresponding singular values, that is $\mathbf{S}_{(P_1)}^{(1)} = \sum_{P_1} \mathbf{V}_{P_1}^T$.

2) Second Step: Replace $\mathcal{C}$ in (27) with $\mathcal{C} = \mathcal{S}^{(1)} \times_{P_1} \mathbf{U}_{P_1}$, the problem (27) can be transformed as follows.

$$\mathbf{U}_{P_2} = \underset{\mathbf{U}_{P_2}}{\arg\min} \left\| \mathcal{S}^{(1)} \times_{P_1} \mathbf{U}_{P_1} \times_{P_2} \left(I - \mathbf{U}_{P_2} \mathbf{U}_{P_2}^T \right) \right\|_F^2 \tag{33}$$

Similarly, the problem (33) can be computed by means of truncated SVD of the mode-P_2 of unfolding of $\mathcal{S}^{(1)}$. After we obtain $\mathbf{U}_{P_2}$, the solution of the original problem (27) is $\pi_{P_2}^* = \mathbf{U}_{P_2} \mathbf{U}_{P_2}^T$.

Let $\mathcal{S}^{(2)}$ denote the truncated core tensor of $\pi_{P_2}^* \pi_{P_1}^* \mathcal{C}$. We have

$$\mathcal{S}^{(2)} = \mathcal{S}^{(1)} \times_{P_2} \mathbf{U}_{P_2}^T = \mathcal{C} \times_{P_1} \mathbf{U}_{P_1}^T \times_{P_2} \mathbf{U}_{P_2}^T \tag{34}$$

and

$$\mathcal{C} = \mathcal{S}^{(2)} \times_{P_1} \mathbf{U}_{P_1} \times_{P_2} \mathbf{U}_{P_2} \tag{35}$$

3) Third Step: Replace $\mathcal{C} = \mathcal{S}^{(2)} \times_{P_1} \mathbf{U}_{P_1} \times_{P_2} \mathbf{U}_{P_2}$ to Eq.(28), problem (28) can be transformed as

$$\begin{aligned}
\mathbf{U}_{P_3} = \underset{\mathbf{U}_{P_3}}{\arg\min} \\
\times \left\| \mathcal{S}^{(2)} \times_{P_1} \mathbf{U}_{P_1} \times_{P_2} \mathbf{U}_{P_2} \times_{P_3} \left(I - \mathbf{U}_{P_3} \mathbf{U}_{P_3}^T \right) \right\|_F^2
\end{aligned} \tag{36}$$

Similar to the problem (33), the problem (36) can be translated to the truncated SVD of the mode-P_3 with the unfolding of $\mathcal{S}^{(2)}$. Obviously, $\pi_{P_3}^* = \mathbf{U}_{P_3} \mathbf{U}_{P_3}^T$ is the solution of problem (28). We denote the truncated core tensor of $\pi_{P_3}^* \pi_{P_2}^* \pi_{P_1}^* \mathcal{C}$ as $\mathcal{S}^{(3)}$ which can be calculated by

$$\mathcal{S}^{(3)} = \mathcal{S}^{(2)} \times_{P_3} \mathbf{U}_{P_3}^T = \mathcal{C} \times_{P_1} \mathbf{U}_{P_1}^T \times_{P_2} \mathbf{U}_{P_2}^T \times_{P_3} \mathbf{U}_{P_3}^T \tag{37}$$

According (17), the optimal rank (r_1, r_2, r_3) approximation tensor of tensor $\mathcal{C}$ can be calculated by

$$\mathcal{X}' = \mathcal{S}^{(3)} \times_{P_1} \mathbf{U}_{P_1} \times_{P_2} \mathbf{U}_{P_2} \times_{P_3} \mathbf{U}_{P_3} \tag{38}$$

where $\mathcal{S}^{(3)}$ is the core tensor of the resulted truncated tensor $\mathcal{X}'$.

Given a processing order $P = [2, 3, 1]$, Fig.8 shows the core tensor under sequential truncation. Obviously, the size of the core tensor is reduced after each truncation.

Rather than performing the calculation in Eq (15) directly with a tensor $\mathcal{C}$ of large size, the sequential truncation process allows for the calculation with a tensor $\mathcal{S}$ of much smaller size. This will significantly reduce the computation cost. In the next subsection, we will analyze the computation cost and find the methodology of further reducing the computation cost.

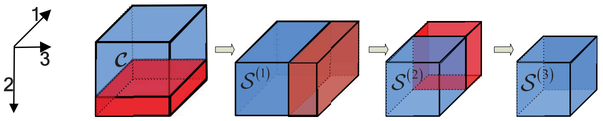


Fig. 8. A graphical illustration of the sequential truncation of a third-order tensor, corresponding to the processing order [2, 3, 1] of the modes.

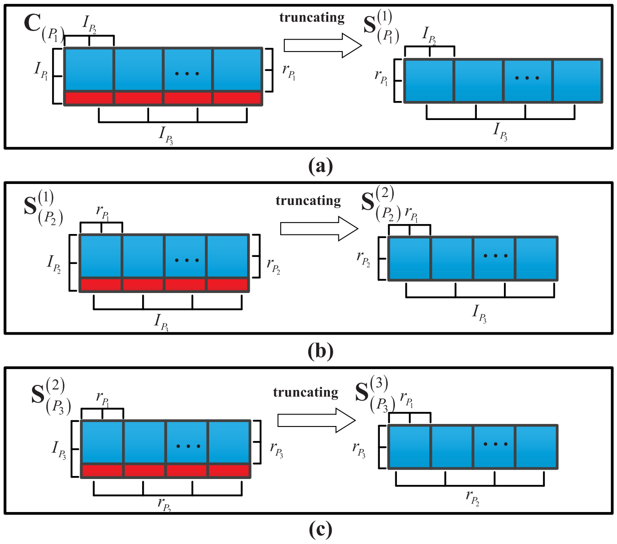


Fig. 9. The unfolding matrix of the core tensor before and after sequence truncation. (a) Mode-$P1$ truncation. (b) Mode-$P2$ truncation. (c) Mode-$P3$ truncation.

C. Cost Analysis

Generally, for an $m \times n$ matrix $(m \leq n)$, the cost to compute the SVD is $O\left(m^2 n\right)$. For an $m \times n$ matrix $\mathbf{X}$ and an $n \times p$ matrix $\mathbf{Y}$, the computation cost of matrix multiplication (i.e., $\mathbf{XY}$) is $O\left(mnp\right)$.

In the first step, the computation cost includes the cost on SVD of the mode-P_1 unfolding matrix and the cost of matrix multiplication $\mathbf{S}_{(P_1)}^{(1)} = \sum_{P_1} \mathbf{V}_{P_1}^T$. The size of mode-$P_1$ unfolding matrix of tensor $\mathcal{C}$ is $I_{P_1} \times (I_{P_2} I_{P_3})$, therefore, the cost on SVD is $O(I_{P_1}^2 I_{P_2} I_{P_3})$. As $\sum_{P_1} \in \mathbb{R}^{r_{P_1} \times r_{P_1}}$ and $\mathbf{V}_{P_1}^T \in \mathbb{R}^{r_{P_1} \times (I_{P_2} \times I_{P_3})}$, the multiplication cost of $\sum_{P_1} \mathbf{V}_{P_1}^T$ is $O(r_{P_1}^2 I_{P_2} I_{P_3})$. Thus, the total cost of the first step is $O_1 = O(I_{P_1}^2 I_{P_2} I_{P_3} + r_{P_1}^2 I_{P_2} I_{P_3})$.

Fig.9(a) shows the mode-P_1 unfolding matrix of tensor $\mathcal{C}$ and the truncated core tensor $\mathcal{S}^{(1)}$ after the mode-P_1 truncation. It is noticed that the number of rows in the unfolding matrix of $\mathcal{S}^{(1)}$ is r_{P_1} instead of I_{P_1}.

In the second step, the unfolding matrix of the truncated core tensor $\mathcal{S}^{(1)}$ is $I_{P_2} \times (r_{P_1} I_{P_3})$, which results in the SVD cost being $O\left(I_{P_2}^2 r_{P_1} I_{P_3}\right)$ and the matrix multiplication cost being $O\left(r_{P_2}^2 r_{P_1} I_{P_3}\right)$. Fig.9(b) further shows the mode-P_2 unfolding matrix of the truncated core tensor before (i.e.,$\mathcal{S}^{(1)}$)

TABLE I
COMPUTING COMPLEXITY

	Our algorithm	Tucker-Decomposition
$P = [1, 2, 3]$	$O(39366)$	$O(87120)$
$P = [1, 3, 2]$	$O(37588)$	$O(87120)$
$P = [2, 1, 3]$	$O(36666)$	$O(87120)$
$P = [2, 3, 1]$	$O(33450)$	$O(87120)$
$P = [3, 1, 2]$	$O(32280)$	$O(87120)$
$P = [3, 2, 1]$	$O(30910)$	$O(87120)$

and after (i.e.,$\mathcal{S}^{(2)}$) mode-P_2 truncation. The total cost of second step is $O_2 = O\left(I_{P_2}^2 r_{P_1} I_{P_3} + r_{P_2}^2 r_{P_1} I_{P_3}\right)$.

In the third step, the mode-P_3 unfolding matrix of the truncated core tensor $\mathcal{S}^{(2)}$ is $I_{P_3} \times (r_{P_1} r_{P_2})$. The SVD cost to compute the $\mathbf{U}_{P_3}$ is $O\left(I_{P_3}^2 r_{P_1} r_{P_2}\right)$ and the matrix multiplication cost is $O\left(r_{P_3}^2 r_{P_1} r_{P_2}\right)$. The total cost of the third step is $O_3 = O\left(I_{P_3}^2 r_{P_1} r_{P_2} + r_{P_3}^2 r_{P_1} r_{P_2}\right)$.

From Fig. 9, we can observe that for each truncation, the shape of the tensor before truncating is larger than the tensor after the truncating. The tensor after the truncating is the operating tensor for the next step. As the computation cost depends on the size of the tensor shape, therefore, the computation cost decreases with each truncating step, i.e., $O_1 > O_2 > O_3$. However, in the Tucker decomposition, every mode unfolding matrix used to calculate the factor matrices has same the shape size, which leads to the same large computation cost. As the shape size of unfolding matrices sequentially reduces in our solution, compared with the Tucker decomposition, our solution can largely reduce the computation cost.

D. Detailed Algorithm

From the discussion in the previous section, we can see the computational cost can be significantly reduced by translating the calculation in Eq (15) into a sequential truncation process with the calculation of a tensor $\mathcal{S}$ whose size is much smaller than that of the tensor $\mathcal{C}$. Different sequential truncation order would lead to different cost as can be seen from the following example.

The total computation cost of our solution includes the cost of each step is

$$\begin{aligned}\text{cost}(P) &= O(I_{P_1}^2 I_{P_2} I_{P_3} + I_{P_2}^2 r_{P_1} I_{P_3} + I_{P_3}^2 r_{P_1} r_{P_2}) \\ &\quad + O\left(r_{P_1}^2 I_{P_2} I_{P_3} + r_{P_2}^2 r_{P_1} I_{P_3} + r_{P_3}^2 r_{P_1} r_{P_2}\right) \quad (39)\end{aligned}$$

where $O(I_{P_1}^2 I_{P_2} I_{P_3} + I_{P_2}^2 r_{P_1} I_{P_3} + I_{P_3}^2 r_{P_1} r_{P_2})$ and $O\left(r_{P_1}^2 I_{P_2} I_{P_3} + r_{P_2}^2 r_{P_1} I_{P_3} + r_{P_3}^2 r_{P_1} r_{P_2}\right)$ are the total SVD cost and matrix multiplication cost, respectively.

To approximate a 3-way traffic tensor $I_1 \times I_2 \times I_3$ by 3-way tensor with a rank (r_1, r_2, r_3), a different processing order may lead to the big difference in the computation cost. Table I shows the computation cost to approximate a 3-way tensor $\mathcal{C} \in \mathbb{R}^{10\times 11\times 12}$ with the truncation rank $(7, 6, 5)$.

Therefore, to speed up our tensor factorization process, we propose Algorithm 2 that will look for the sequential truncation order with the minimum calculation cost. It first evaluates the computation cost of different processing orders, among which, the order with the least cost is selected, denoted by P. According to the selected processing order P, the original tensor $\mathcal{C}$ is sequentially truncated to obtain the optimal rank(r_1, r_2, r_3) tensor. On line 11, the truncated tensor is $\mathcal{X}'$ with its core tensor being $\mathcal{S}^{(3)}$.

Algorithm 2 Tensor Truncation

```
Input: Traffic tensor $\mathcal{C} \in \mathbb{R}^{I_1\times I_2\times I_3}$ and truncation rank $(r_1, r_2, r_3)$
Output: Truncated traffic tensor $\mathcal{X}'$ with $rank(r_1, r_2, r_3)$.
1:  For all the permutations of $\{1, 2, 3\}$, according to (39), calculate the cost of each processing sequence (each sequence corresponding to one permutation of $\{1, 2, 3\}$).
2:  Select the one with least cost as the optimal processing order, denoted as $P$.
3:  $\mathcal{S}^{(0)} = \mathcal{C}$
4:  $i = 0$
5:  for $k \leftarrow P_1, P_2, P_3$ do
6:      $i = i + 1$
7:      Compute the compact singular value decomposition of $\mathbf{S}_{(k)}^{(i-1)}$
            $\mathbf{S}_{(k)}^{(i-1)} = \begin{bmatrix} \bar{\mathbf{U}}_1 & \bar{\mathbf{U}}_2 \end{bmatrix} \begin{bmatrix} \bar{\sum}_1 & \\ & \bar{\sum}_2 \end{bmatrix} \begin{bmatrix} \bar{\mathbf{V}}_1^T \\ \bar{\mathbf{V}}_2^T \end{bmatrix}$    (40)
8:      $\mathbf{U}_k \leftarrow \bar{\mathbf{U}}_1$
9:      $\mathbf{S}_{(k)}^{(i)} \leftarrow \bar{\sum}_1 \bar{\mathbf{V}}_1^T$
10: end for
11: $\mathcal{X}' = S^{(3)} \times_{P_3} U_{P_3} \times_{P_2} U_{P_2} \times_{P_1} U_{P_1}$
```

VI. ANOMALY DETECTION

For a given sparse tensor approximation $\mathcal{X}'$, we can estimate the outliers $\mathcal{E}$ based on the difference $\mathcal{B} = \mathcal{X} - \mathcal{X}'$ as

$$\begin{aligned}&\min_{\mathcal{E}} \ \|\mathcal{B} - \mathcal{E}\|_F \\ &\text{s.t. } \mathcal{B} = \mathcal{X} - \mathcal{X}' \\ &\qquad \|\mathcal{E}\|_0 \leq \varepsilon \end{aligned} \quad (41)$$

This problem, however, is generously hard to solve with its use of L_0-norm to get the set cardinality. To make the problem trackable, we propose to investigate a solution consisting of two phases:

- **Phase 1**
  We relax the l_0 constraint and solve the following problem to identify the candidate entry values of $\mathcal{E}$:
  $$\begin{aligned}&\min_{\mathcal{E}} \ \|\mathcal{B} - \mathcal{E}\|_F \\ &\text{s.t. } \mathcal{B} = \mathcal{X} - \mathcal{X}' \end{aligned} \quad (42)$$
  Given $\mathcal{B}$, the solution to problem in (42) is $\mathcal{E}^* = \mathcal{B}$ with each entry $e_{ijk}^* = b_{ijk}$. With the l_0 constraint $\|\mathcal{E}\|_0 \leq \varepsilon$, the total number of non-zero entries in tensor $\mathcal{E}$ is restricted to be less than ε. Thus for the original problem, each entry e_{ijk} can have only two candidate values, b_{ijk} (the solution of the relaxed problem) or 0.
- **Phase 2**
  To solve the original anomaly detection problem in (41), we need to restrict the number of non-zero items to be less than ε while setting the remaining items to be 0.
  As $\|\mathcal{B} - \mathcal{E}\|_F = \sqrt{\sum_k \sum_j \sum_i (b_{ijk} - e_{ijk})^2}$, to solve the problem, we can define the function $f_{ijk} : \mathbb{R} \rightarrow \mathbb{R}$ as $f_{ijk}(e_{ijk}) = (b_{ijk} - e_{ijk})^2$.
  Obviously, we have $f_{ijk}(0) > f_{ijk}(b_{ijk})$. When setting an item e_{ijk} found in the relaxed problem (42) to 0, it

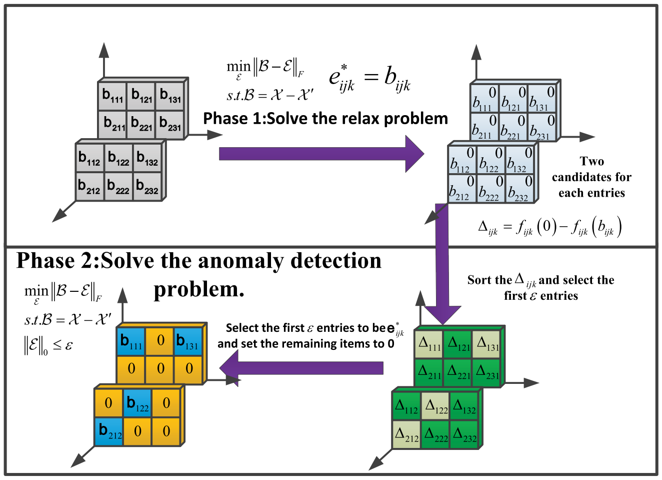

$\min_{\mathcal{E}} \|\mathcal{B}-\mathcal{E}\|_F$
$s.t. \mathcal{B} = \mathcal{X} - \mathcal{X}'$
$e^*_{ijk} = b_{ijk}$
Phase 1:Solve the relax problem
b_{111} b_{121} b_{131} b_{211} b_{221} b_{231} b_{112} b_{122} b_{132} b_{212} b_{222} b_{232}
Two candidates for each entries
$\Delta_{ijk} = f_{ijk}(0) - f_{ijk}(b_{ijk})$
Phase 2:Solve the anomaly detection problem.
$\min_{\mathcal{E}} \|\mathcal{B}-\mathcal{E}\|_F$
$s.t. \mathcal{B} = \mathcal{X} - \mathcal{X}'$
$\|\mathcal{E}\|_0 \le \varepsilon$
Sort the Δ_{ijk} and select the first ε entries
Select the first ε entries to be e^*_{ijk} and set the remaining items to 0
Δ_{111} Δ_{121} Δ_{131} Δ_{211} Δ_{221} Δ_{231} Δ_{112} Δ_{122} Δ_{132} Δ_{212} Δ_{222} Δ_{232}
0

Fig. 10. Two phase anomaly detection algorithm.

will increase the total summation $\sum_k \sum_j \sum_i (b_{ijk} - e_{ijk})^2$ with $\Delta_{ijk} = f_{ijk}(0) - f_{ijk}(b_{ijk})$, which is against the objective of minimizing $\|\mathcal{B} - \mathcal{E}\|_F$. To find the approximate $\mathcal{E}$ while minimizing the total increase of summation due to the enforcement of the l_0 constraint, we will sort the items Δ_{ijk} in the descending order. We select the first ε entries to be b_{ijk} and set the remaining items to 0 to better meet the problem objective after involving the l_0 constraint.

Fig.10 presents an example to illustrate our two-phase anomaly detection algorithm. We further provide a theoretical proof for our two-phase anomaly detection algorithm (Theorem 2).

Theorem 2: Let $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$ and $\phi_{ijk} : \mathbb{R} \to \mathbb{R}$. Suppose that ε is a positive integer and $0 \in \mathcal{A}$. Consider the following minimization problem:

$$\min \left\{ \phi(a) = \sum_{i=1,j=1,k=1}^{I,J,K} \phi_{ijk}(a_{ijk}) \right\} : \|\mathcal{A}\|_0 \le \varepsilon, a_{ijk} \in \mathcal{A} \tag{43}$$

Let $\tilde{a}^*_{ijk} \in \arg\min \{\phi_{ijk}(a_{ijk}) : a_{ijk} \in \mathcal{A}\}$ and $M^* \subseteq \{ijk | i = 1, \cdots, I, j = 1, \cdots, J, andk = 1, \cdots, K\}$ be the index set corresponding to ε largest values of $\{x^*_{ijk}\}_{i=1,j=1,k=1}^{I,J,K}$, where $x^*_{ijk} = \phi_{ijk}(0) - \phi_{ijk}(\tilde{a}^*_{ijk})$ for $i = 1, \cdots, I, j = 1, \cdots, J,$ and $k = 1, \cdots, K$. Then a^*_{ijk} is an optimal solution of problem (43) where a^*_{ijk} is defined as follows:

$$a^*_{ijk} = \begin{cases} \tilde{a}^*_{ijk} & if\ ijk \in M^* \\ 0 & otherwise \end{cases} \tag{44}$$

Proof: By the assumption $0 \in \mathcal{A}$ and the definitions of a^*_{ijk}, $\tilde{a}^*_{ijk}$ and M^*, we easily obtain that $\|\mathcal{A}^*\|_0 \le \varepsilon$, hence, $\mathcal{A}^*$ is a feasible solution of (43). It remains to show that $\phi_{ijk}(\mathcal{A}) \ge \phi_{ijk}(\mathcal{A}^*)$ for any feasible solution $\mathcal{A}$ of (43). Indeed, let $\mathcal{A}$ be arbitrarily chosen such that $\|\mathcal{A}\|_0 \le \varepsilon$, and let $N = \{ijk | a_{ijk} \ne 0\}$. Clearly, $|N| \le \varepsilon = |M^*|$. Let $\bar{M}^*$ and $\bar{N}$ denote the complement of M^* and N, respectively. It then follows that

$$\begin{aligned} \left|\bar{N} \cap M^*\right| &= |M^*| - |M^* \cap N| \ge |N| - |M^* \cap N| \\ &= \left|N \cap \bar{M}^*\right| \end{aligned} \tag{45}$$

Because $\left|\bar{N} \cap M^*\right| + |M^* \cap N| + \left|\bar{M}^* \cap \bar{N}\right| + \left|N \cap \bar{M}^*\right| = U$ (Universal Set), we have

$$\begin{aligned} &\phi(a) - \phi(a^*) \\ &= \left\{ \begin{array}{l} \sum_{ijk \in \bar{N} \cap M^*} (\phi_{ijk}(a_{ijk}) - \phi_{ijk}(a^*_{ijk})) \\ + \sum_{ijk \in M^* \cap N} (\phi_{ijk}(a_{ijk}) - \phi_{ijk}(a^*_{ijk})) \\ + \sum_{ijk \in \bar{M}^* \cap \bar{N}} (\phi_{ijk}(a_{ijk}) - \phi_{ijk}(a^*_{ijk})) \\ + \sum_{ijk \in \bar{M}^* \cap N} (\phi_{ijk}(a_{ijk}) - \phi_{ijk}(a^*_{ijk})) \end{array} \right\} \end{aligned} \tag{46}$$

As $U \ge \left|\bar{N} \cap M^*\right| + \left|N \cap \bar{M}^*\right|$, we further have

$$\begin{aligned} &\phi(a) - \phi(a^*) \\ &\ge \left\{ \begin{array}{l} \sum_{ijk \in M^* \cap \bar{N}} (\phi_{ijk}(0) - \phi_{ijk}(a^*_{ijk})) \\ + \sum_{ijk \in \bar{M}^* \cap N} (\phi_{ijk}(a^*_{ijk}) - \phi_{ijk}(0)) \end{array} \right\} \\ &= \left\{ \begin{array}{l} \sum_{ijk \in M^* \cap \bar{N}} (\phi_{ijk}(0) - \phi_{ijk}(a^*_{ijk})) \\ - \sum_{ijk \in \bar{M}^* \cap N} (\phi_{ijk}(0) - \phi_{ijk}(a^*_{ijk})) \end{array} \right\} \end{aligned} \tag{47}$$

As $\left|\bar{N} \cap M^*\right| \ge \left|N \cap \bar{M}^*\right|$ (according to (45)) and $\phi_{ijk}(0) - \phi_{ijk}(a^*_{ijk}) \ge 0$, we have $\phi(a) - \phi(a^*) \ge 0$ for any feasible point a of (43), which implies that the conclusion holds. ■

VII. Performance Evaluations

Before we present the experiment results, we first present the setting of our experiment.

A. Generation of Corrupted Synthesized Data

Although datasets such as 1999 DARPA IDS (named KDD99) contain some anomalous data, these datasets are not suitable for the study of end-to-end network connections which we focus on. Therefore, we synthetically generate anomalies by adding data outliers into the public traffic traces Abilene [22] and GÈANT [23]. As the two traces record the volume of traffic flows between all source and destination pairs, they allow us to form a network-wide traffic tensor. Abilene and GÈANT collect the monitoring data every 5 minutes and 15 minutes respectively, so they have different lengths for the time slot.

We denote the raw trace data as $\mathcal{M} \in R^{I_1 \times I_2 \times I_3}$. For more efficient data processing, data normalization [41] is often applied in the data preprocessing step to scale the variables or features of data, and the normalized values are often within the range [0,1]. In this paper, given $m_{i,j,k}$, we adopt the following equation to normalize the data:

$$m_{i,j,k} = \frac{m_{i,j,k} - \min\limits_{u,v,w} \{m_{u,v,w}\}}{\max\limits_{u,v,w} \{m_{u,v,w}\} - \min\limits_{u,v,w} \{m_{u,v,w}\}} \tag{48}$$

where $\max\limits_{u,v,w} \{m_{u,v,w}\}$ and $\min\limits_{u,v,w} \{m_{u,v,w}\}$ are the maximum value and minimum value of all the traffic data, respectively.

Following [4], [42]–[44], we inject the outliers to the normalized traffic data to generate the synthesized corrupted data with the following steps:

1) A outlier tensor $\mathcal{E}$ is generated as

$$e_{i,j,k} = \begin{cases} e_{i,j,k} & (i,j,k) \in \Omega \\ 0 & otherwise \end{cases} \tag{49}$$

where e_{ijk} is the generated outlier, and Ω is the outlier location set.

2) The synthesized data $\mathcal{X}$ is the sum of the outlier data $\mathcal{E}$ and the raw data $\mathcal{M}$, that is $x_{i,j,k} = m_{i,j,k} + e_{i,j,k}$ for all (i, j, k).

From [45], the intensity and locations of anomalies may have impact on the accuracy of anomaly detection. To make the injected outliers more close to real ones, we generate the outliers as follows.

We adopt two strategies to generate the location set Ω. 1) ***Random anomalies.*** To simulate anomalies that do not have fixed locations, we randomly select $\gamma \times (I_1 \times I_2 \times I_3)$ locations as the outlier locations with $|\Omega| = \gamma \times (I_1 \times I_2 \times I_3)$, where γ is the the outlier ratio. We set $\gamma = 0.1$ as the default one. 2) ***Week long anomalies.*** Under some special cases, such as week long attacks [42], [46], anomalous data have specific location patterns. Therefore, besides injecting the outlier data at random locations, we also generate the outlier location according to week long attack. Specially, for each week, we randomly select 10 OD flows in Abilene and 30 OD flows in GÈANT be the ones under attack. The locations corresponding to these OD pairs form the outlier location set Ω.

Following the [42], [47], we adopt two data distributions to generate the outlier data values. 1) ***Gaussian distribution***: $e_{ijk} \in \Omega$ is generated following $\mathcal{N}\left(\mu, \sigma^2\right)$ with the mean μ and the variance σ^2 2) ***Exponential distribution***: $e_{ijk} \in \Omega$ is generated following the Exponential distribution $E\left(\frac{1}{\mu}\right)$ with the mean μ. In this paper, we set $\mu = 0$, $\sigma = 1$ as the default values. To investigate how anomaly intensity impacts the detection performance, we vary σ^2 and mean μ of the injected outliers in Section VII-D. For each experiment setting, we run the experiments ten times with the random seeds and get the average of the results.

B. Performance Metric

Our TensorDet aims to decompose the noisy traffic data into the low-rank normal traffic data and the sparse outlier data. We denote the decomposed low-rank traffic data as $\hat{\mathcal{X}}$, and the decomposed outlier data as $\hat{\mathcal{E}}$. The outlier is further detected based on $\hat{\mathcal{E}}$. We use following four metrics to evaluate the performance of the proposed TensorDet.

False Positive Rate (FPR): It measures the proportion of non-outliers that are wrongly identified as outliers.

True Positive Rate (TPR): It measures the proportion of outliers that are correctly identified.

Speedup: Given the computation time under two different algorithms (alg_1 and alg_2), denote as T_1 and T_2. The speedup in computation time of the alg_2 with respect to the alg_1: $S_{1-2} = T_1/T_2$.

Smaller False Positive Rate and higher True Positive Rate mean better detection performance. To evaluate the performance of the proposed TensorDet, we implement six schemes for performance comparison.

Three tensor-based anomaly detection schemes: Based on our traffic tensor model, three tensor-based anomaly detection schemes are implemented. The first is our proposed TensorDet, in which the tensor factorization subproblem and the anomaly detection subproblem are iteratively solved as shown in Algorithm 1. The second one is TensorRPCA proposed in [18] which has three steps: 1) the multi-way tensor is matricizationed into multiple unfolding matrices, 2) the RPCA with Trace norm relaxation is utilized on the unfolding matrices to detect the anomaly, and 3) folding the result of each unfolding matrix to obtain the final result. The third one is RTD proposed in [48] which decomposes a tensor into low rank and sparse components through CP decomposition.

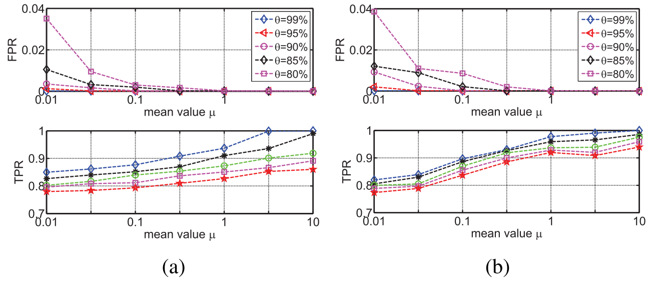


Fig. 11. Impact of different truncation rank. (a) Abilene. (b) GÈANT.

Three matrix-based anomaly detection schemes: Current traffic data analysis is usually based on a traffic matrix model with its row representing the origin and destination (OD) pair and the column representing the time interval. Following the traffic matrix, other 3 anomaly detection schemes are implemented. The first is MatrixDR where direct robust matrix factorization is applied for anomaly detection [31]. The second is MatrixPCA which applies a PCA-based anomaly detection algorithm [4] to identify the anomaly. The third is MatrixRPCA in which the robust PCA [21] is applied to the traffic matrix to detect the anomaly.

All these anomaly detection algorithms can decompose the noisy traffic data into the normal traffic data and the candidate outlier data. To fairly compare these algorithms, we adopt the same anomaly detection principle: among all the candidate outlier data, return the α data points with the largest α absolute values where α is the number of outliers injected.

In Section VII-D, we will show that TensorDet achieves significantly higher accuracy for anomaly detection compared with other peer algorithms. With a higher dimensional structure, tensor data processing involves a higher computational cost. As shown in Algorithm 1, the main computation complexity comes from the tensor factorization. In Section VII-E, we show the gain in computation cost reduction by comparing our sequential tensor truncating algorithm with other tensor factorization schemes.

C. Impact of the Parameters

Our TensorDet includes two parameters: truncation rank (r_1, r_2, r_3) and ε. The outlier number ε does not need to match the actual number of outliers, but is only used to prevent that too many data are regarded as outliers. Consequently, we set ε to equal 10% of the whole data.

We focus on detecting the anomalous data after the traffic data are collected. As we don't know the number and locations of anomalous data items thus the normal data pattern, it is important to appropriately set the truncation rank for anomalous data detection.

According to [49] and [50], the truncation rank (r_1, r_2, r_3) can be set to preserve certain amount of the tensor data variability to capture the main features of the normal data. In addition, we can use the ranks of past data as a reference to set up the current data rank, and vary the rank setting when the data are periodically collected. The study of rank variation is outside the scope of this work.

To investigate how the truncation rank (r_1, r_2, r_3) setting impacts the outlier detection performance, we let the rank to

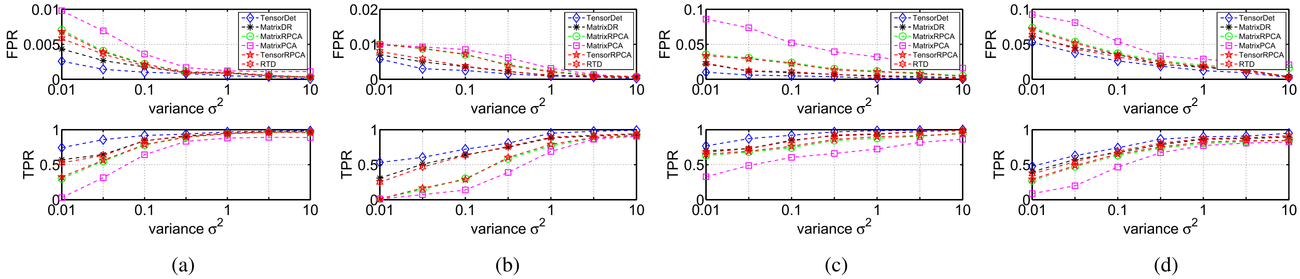


Fig. 12. Gaussian distribution (different variance σ^2). (a) Abilene: random anomaly. (b) GÈANT: random anomaly. (c) Abilene: week long anomaly. (d) GÈANT: week long anomaly.

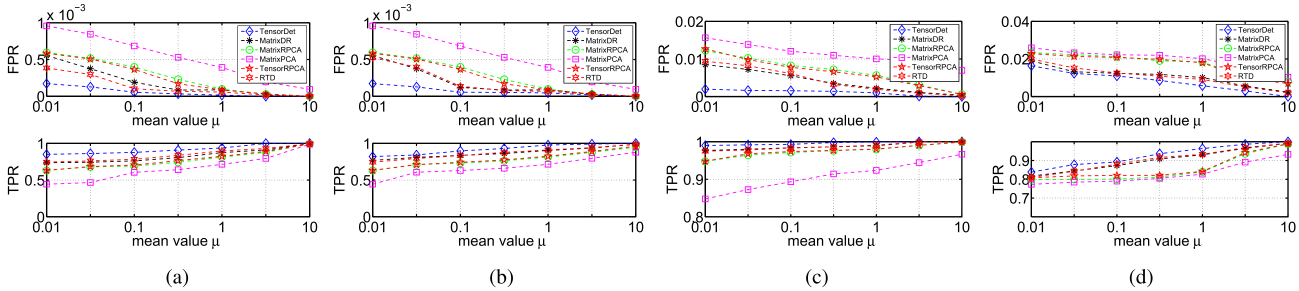


Fig. 13. Gaussian distribution (different mean value μ). (a) Abilene: random anomaly. (b) GÈANT: random anomaly. (c) Abilene: week long anomaly. (d) GÈANT: week long anomaly.

preserve different amount of the tensor data variability. Specially, we let each truncation rank r_k of the mode determined with $\sum_{i=1}^{k}\sigma_i^2/\sum_{i=1}^{r}\sigma_i^2 \geq \theta$ by varying θ, where θ is the ratio of data variability preservation and σ_i is the singular value of the corresponding unfolded matrix).

With outliers injected at random locations and the data values following the Gaussian distribution, we show the performance in Fig.11. When the outliers have a larger mean value μ, the detection performance becomes more accurate with lower False Positive Rate and high True Positive Rate. When the truncation rank is set too small with a small θ, the model cannot capture the normal variability of the data, leading to a worse detection performance. Even when the mean value μ is small, we find that setting the truncation rank (r_1, r_2, r_3) to preserve 99% of the data variability of the traffic tensor can achieve higher detection performance.

For more accurate detection or when the ranks of the data are not known in advance, the operator can set the rank to be more conservatively with a higher value at the cost of higher computation overhead. According to the results, we set the truncation rank (r_1, r_2, r_3) to preserve 99% of the data variability of the traffic tensor with $\theta = 99\%$. In other outlier injected scenarios (week long attacker and Exponential distribution), similar results can be obtained. Due to the space limitation, we do not show simulation results of other scenarios.

D. The Accuracy Comparison

1) Gaussian Distribution: To compare the performance of different anomaly detection algorithms, with other parameters fixed, we vary the variance σ^2, the average value μ, and the outlier ratio γ of the outliers.

From Fig.12, Fig.13, and Fig.14, among all the algorithms compared, our TensorDet achieves the best performance. It achieves the lowest False Positive Rate and Highest True Positive Rate, under all the simulation scenarios using different traffic traces (Abilene and GÈANT) with different anomaly strategies (random anomaly and week long anomaly). These results also demonstrate that TensorDet is a robust anomaly detection technique that can fully utilize the tensor's multi-dimensional information to more accurately detect the anomaly.

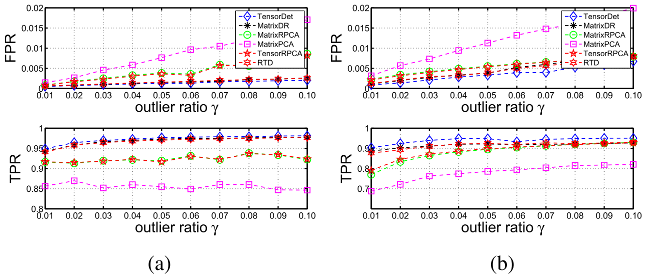
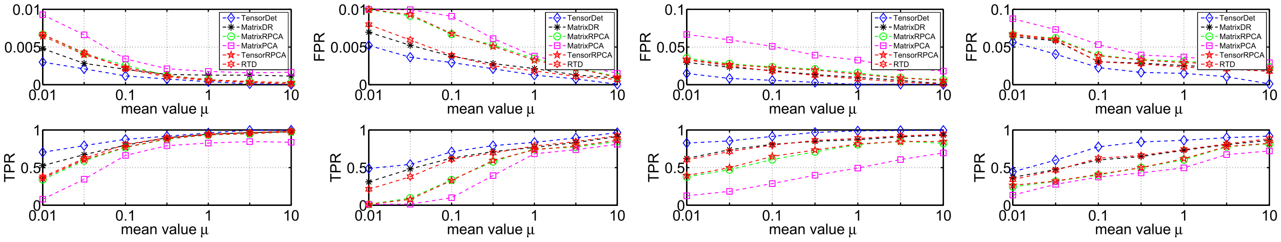


Fig. 14. Gaussian distribution (different outlier ratio γ). (a) Abilene: random anomaly. (b) GÈANT: random anomaly.

Compared with the tensor-based anomaly detection algorithms TensorRPCA and RTD, our TensorDet achieves much better overall performance. Designed based on unfolding matrices and using the trace norm to relax the low-rank feature of the unfolding matrices, TensorRPCA is fundamentally a matrix-based approach and cannot fully exploit the tensor pattern with the multilinear information to better detect the anomaly. As a result, compared with MatrixDR, the detection performance under TensorRPCA is even worse, as MatrixDR can model and solve the anomaly detection problem directly using low rank feature instead of using trace norm to relax the low rank feature like TensorRPCA.



(a) (b) (c) (d)

Fig. 15. Exponential distribution (different mean value μ). (a) Abilene: random anomaly. (b) GÈANT: random anomaly. (c) Abilene: week long anomaly. (d) GÈANT: week long anomaly.

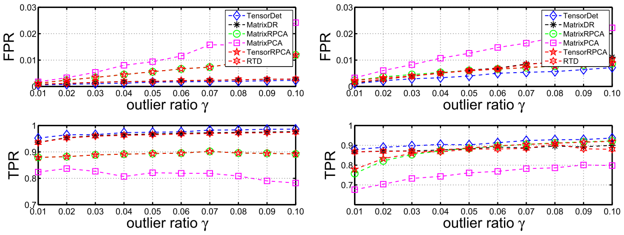
Compared with RTD, our TensorDet achieves better performance with higher True Positive Rate and lower False Positive Rate. They are designed based on different tensor decomposition methods. RTD is based on CP decomposition and TensorDet is based on Tucker decomposition. Tucker method decomposes a tensor into a core tensor multiplied (or transformed) by matrices each along a mode with the entries in the core tensor showing the level of interaction between components along different modes. CP decomposition can be viewed as a special case of Tucker where the core tensor is super-diagonal. The Tucker decomposition captures more hidden information in the tensor, thus our TensorDet achieves a better detection performance.

As shown in Fig.12 and Fig.13, with the increase of variance σ^2 and mean μ of the outliers, the True Positive Rate increases while the False Positive Rate decreases for all algorithms implemented. Obviously, when the variance and mean of outliers are smaller, synthesized outlier data have closer and smaller values, and are more difficult to be differentiated from the normal data. As shown in Fig.12(a), even when variance σ^2 is a small value with $\sigma = 0.01$, our TensorDet achieves the highest True Positive Rate 0.75, while the True Positive Rates under MatrixDR, MatrixRPCA, MatrixPCA, TensorRPCA, and RTD are 0.6, 0.3, 0.1, 0.3, and 0.5, respectively; our TensorDet achieves the lowest False Positive Rate 0.0025, while the False Positive Rates under MatrixDR, MatrixRPCA, MatrixPCA, TensorRPCA, and RTD are 0.004, 0.0075, 0.01, 0.007, and 0.006, respectively.

In the week long attack, we fix the number of ODs to be attacked, and therefore, in Fig.14, we only draw the simulation results under different outlier ratio when the outlier locations are randomly generated. Fig.14 shows the detection performance by varying the outlier ratio γ from 0.01 to 0.10. Even with a large outlier ratio at $\gamma = 0.1$, our TensorDet still achieves very large True Positive Rate and low False Positive Rate. The performance of other matrix-based detection algorithms such as MatrixPCA changes largely with the variation of γ, while TensorDet can capture the multi-dimensional information hidden in the 3-way tensor to more robustly detect the anomaly. The detection performance of TensorDet is much more stable compared to peer matrix-based detection algorithms and TensorRPCA algorithm.

2) Exponential Distribution: With the synthesized data generated following the exponential distribution, we vary the average value of the outlier μ (Fig.15) and the outlier ratio γ (Fig.16) to evaluate the performance of different anomaly detection algorithms.

Similar to the simulation results with the synthesized data generated following the Gaussian distribution, among all the detection schemes implemented, TensorDet achieves significantly better performance in terms of the False Positive Rate and True Positive Rate, and thus can provide more accurate and stable detection.



(a) (b)

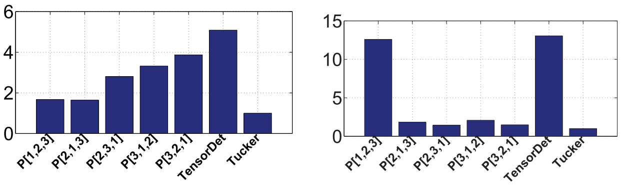
Fig. 16. Exponential distribution (different outlier ratio γ). (a) Abilene: random anomaly. (b) GÈANT: random anomaly.

E. The Speed Comparison With Other Tensor Factorization

A core technique used in TensorDet is to apply tensor factorization to obtain the approximation rank (r_1, r_2, r_3) tensor. In this paper, we propose a sequential truncation method which seeks the optimal processing order to obtain the approximated tensor as well as to minimize the computation cost. To evaluate the computation cost, besides our sequential truncation method, we implement another truncating tensor algorithm based on tucker decomposition. In this comparison experiment, we first implement tucker decomposition (denoted as $Tucker$), and then obtain the approximation tensor based on the method mentioned at the beginning of Section V. Moreover, besides the optimal processing order adopted in our algorithm, we also plot all other 5 processing order with our sequence truncating for comparisons.

We use speedup to compare the speed of different algorithms. We use $Tucker$ to denote the Tucker decomposition based tensor approximation algorithm. To calculate the speedup metric, we use $Tucker$ as the baseline algorithm and set $alg_1 = Tucker$.

Fig.17 shows the speedup of all the peer tensor factorization executions. Benefiting from our sequential tensor truncating algorithm with the good processing order in the sequential execution, the tensor factorization process in TensorDet is 5 (Abilene) and 13 (GÈANT) times faster compared with the traditional Tucker decomposition solution. Moreover, compared with other processing orders, although all these executions also follow the sequential truncation in this paper with the dimension reduction in each sequential truncation steps, the best order selected in our algorithm also brings significantly



(a) (b)

Fig. 17. Speed up comparison with other tensor factorization. (a) Abilene. (b) GÈANT.

better performance. These results also confirm that different processing orders may result in significantly different computation cost, which is our basic design principle applied to reduce the cost.

VIII. Conclusion

We present TensorDet, a tensor based algorithm for accurate and fast Internet anomaly detection. We formulate the anomaly detection problem as a tensor recovery problem which is further formulated as a tensor approximation problem with constraints on the rank of the tensor and the cardinality of the anomaly set. Although such a problem formulation can take advantage of the tensor pattern and correlations among multiple modes to better detect the anomaly, the two constraints bring a significant challenge to find the solution. Unlike existing methods which resorts to convex relaxation and consequently compromises the detection performance, TensorDet solves the problem efficiently with the support of two proposed techniques, *sequential tensor truncation* and *two-phase anomaly detection*. We have conducted extensive experiments using Internet traffic trace data to compare the proposed TensorDet with the state of art tensor recovery algorithms and matrix-based anomaly detection algorithms. Our results demonstrate the effectiveness and efficiency of TensorDet.

References

[1] V. Bamnett and T. Lewis, "Outliers in statistical data,"1994.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surveys*, vol. 41, no. 3, p. 15, 2009.

[3] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psychol.*, vol. 24, no. 6, p. 417, 1933.

[4] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun. (SIGCOMM)*, vol. 34. Oct. 2004, pp. 219–230.

[5] L. Huang *et al.*, "Communication-efficient online detection of network-wide anomalies," in *Proc. 26th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, May 2007, pp. 134–142.

[6] D. Brauckhoff, K. Salamatian, and M. May, "Applying PCA for traffic anomaly detection: Problems and solutions," in *Proc. INFOCOM*, 2009, pp. 2866–2870.

[7] X. Li *et al.*, "Detection and identification of network anomalies using sketch subspaces," in *Proc. 6th ACM SIGCOMM Conf. Internet Meas.*, 2006, pp. 147–152.

[8] Y. Liu, L. Zhang, and Y. Guan, "Sketch-based streaming PCA algorithm for network-wide traffic anomaly detection," in *Proc. IEEE 30th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2010, pp. 807–816.

[9] X. Li *et al.*, "MIND: A distributed multi-dimensional indexing system for network diagnosis," in *Proc. INFOCOM*, 2006, pp. 1–12.

[10] K. Xie *et al.*, "Sequential and adaptive sampling for matrix completion in network monitoring systems," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2015, pp. 2443–2451.

[11] G. Xie *et al.*, "Fast low-rank matrix approximation with locality sensitive hashing for quick anomaly detection," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.

[12] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, 2000.

[13] M. Zhao and V. Saligrama, "Anomaly detection with score functions based on nearest neighbor graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2250–2258.

[14] X. Kun *et al.*, "Accurate recovery of Internet traffic data: A tensor completion approach," in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.

[15] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.

[16] H. Zhou, D. Zhang, K. Xie, and Y. Chen, "Spatio-temporal tensor completion for imputing missing Internet traffic data," in *Proc. IEEE 34th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2015, pp. 1–7.

[17] X. Kun, P. Can, W. Xin, X. Gaogang, and W. Jigang, "Accurate recovery of Internet traffic data under dynamic measurements," in *Proc. IEEE INFOCOM*, May 2017, pp. 1–9.

[18] D. Goldfarb and Z. Qin, "Robust low-rank tensor recovery: Models and algorithms," *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 1, pp. 225–253, 2014.

[19] H. Tan, J. Feng, G. Feng, W. Wang, and Y.-J. Zhang, "Traffic volume data outlier recovery via tensor model," *Math. Problems Eng.*, vol. 2013, Feb. 2013, Art. no. 164810.

[20] J. Li, G. Han, J. Wen, and X. Gao, "Robust tensor subspace learning for anomaly detection," *Int. J. Mach. Learn. Cybern.*, vol. 2, no. 2, pp. 89–98, 2011.

[21] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, p. 11, May 2011.

[22] *The Abilene Observatory Data Collections*. Accessed: 2013. [Online]. Available: http://abilene.internet2.edu /observatory/data-collections.html

[23] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 83–86, 2006.

[24] B. Chen, J. Yang, B. Jeon, and X. Zhang, "Kernel quaternion principal component analysis and its application in RGB-D object recognition," *Neurocomputing*, vol. 266, pp. 293–303, Nov. 2017.

[25] C. Yuan, X. Sun, and R. Lv, "Fingerprint liveness detection based on multi-scale LPQ and PCA," *China Commun.*, vol. 13, no. 7, pp. 60–65, 2016.

[26] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 217–228, 2005.

[27] C. Callegari, L. Gazzarrini, S. Giordano, M. Pagano, and T. Pepe, "A novel PCA-based network anomaly detection," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2011, pp. 1–5.

[28] Z. Lin, M. Chen, and Y. Ma. (2010). "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices." [Online]. Available: https://arxiv.org/abs/1009.5055

[29] K. Xie *et al.*, "Recover corrupted data in sensor networks: A matrix completion solution," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1434–1448, May 2017.

[30] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[31] L. Xiong, X. Chen, and J. Schneider, "Direct robust matrix factorizatoin for anomaly detection," in *Proc. IEEE 11th Int. Conf. Data Mining (ICDM)*, Mar. 2011, pp. 844–853.

[32] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometrics Intell. Lab. Syst.*, vol. 106, no. 1, pp. 41–56, 2011.

[33] E. Acar, T. G. Dunlavy, and M. D. Kolda, "A scalable optimization approach for fitting canonical tensor decompositions," *J. Chemometrics*, vol. 25, no. 2, pp. 67–86, 2011.

[34] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Problems*, vol. 27, no. 2, p. 025010, 2011.

[35] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[36] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl*, vol. 21, no. 4, pp. 1253–1278, 2000.

[37] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear subspace analysis of image ensembles," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 2003, p. II-93.

[38] D. C. Sorensen, "Numerical methods for large eigenvalue problems," *Acta Numerica*, vol. 11, pp. 519–584, Mar. 2002.

[39] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Philadelphia, PA, USA: SIAM, 2003.

[40] V. de Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1084–1127, 2008.

[41] S. Aksoy and R. M. Haralick, "Feature normalization and likelihood-based similarity measures for image retrieval," *Pattern Recognit. Lett.*, vol. 22, no. 5, pp. 563–582, 2001.

[42] B. I. Rubinstein *et al.*, "Compromising PCA-based anomaly detectors for network-wide traffic," Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2008-73, 2008.

[43] R. A. Maxion and K. M. Tan, "Benchmarking anomaly-based detection systems," in *Proc. Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2000, pp. 623–630.

[44] A. Soule, K. Salamatian, and N. Taft, "Combining filtering and statistical methods for anomaly detection," in *Proc. 5th ACM SIGCOMM Conf. Internet Meas.*, 2005, p. 31.

[45] B. Zhang, J. Yang, J. Wu, D. Qin, and L. Gao, "PCA-subspace method's it good enough for network-wide anomaly detection," in *Proc. Netw. Oper. Manage. Symp. (NOMS)*, 2012, pp. 359–367.

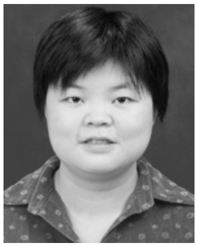
[46] L. Zonglin, H. Guangmin, Y. Xingmiao, and Y. Dan, "Detecting distributed network traffic anomaly with network-wide correlation analysis," *EURASIP J. Adv. Signal Process.*, vol. 2009, no. 1, p. 2, Jan. 2009.

[47] J. Jiang and S. Papavassiliou, "Detecting network attacks in the internet via statistical network traffic normality prediction," *J. Netw. Syst. Manage.*, vol. 12, no. 1, pp. 51–72, 2004.

[48] A. Anandkumar, P. Jain, Y. Shi, and U. N. Niranjan, "Tensor vs matrix methods: Robust tensor decomposition under block sparse perturbations," *CoRR*, vol. abs/1510.04747, 2015. [Online]. Available: http://arxiv.org/abs/1510.04747

[49] J. Ye, "Generalized low rank approximations of matrices," *Mach. Learn.*, vol. 61, nos. 1–3, pp. 167–191, 2005.

[50] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, Sep. 1936.



Kun Xie received the Ph.D. degree in computer application from Hunan University, Changsha, China, in 2007. She was a Post-Doctoral Fellow with the Department Of Computing, The Hong Kong Polytechnic University, from 2007 to 2010. She was a Visiting Researcher with the Department of Electrical and Computer Engineering, The State University of New York at Stony Brook, from 2012 to 2013. She is currently a Professor with Hunan University. She has authored over 60 papers in major journals and conference proceedings (including journals the IEEE/ACM TRANSACTIONS ON NETWORKING, the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS ON COMPUTERS, and the IEEE TRANSACTIONS ON WIRELESS COMPUTING, and conferences INFOCOM, ICDCS, SECON, and IWQoS). Her research interests include wireless network and mobile computing, network management and control, cloud computing and mobile cloud, and big data.

Xiaocan Li is currently pursuing the Ph.D. degree with the College of Computer Science and Electronics Engineering, Hunan University, Changsha, China.

Xin Wang (M'01) received the Ph.D. degree in electrical and computer engineering from Columbia University, New York, NY, USA. She is currently an Associate Professor with the Department of Electrical and Computer Engineering, The State University of New York at Stony Brook, Stony Brook, NY, USA. She was a member of Technical Staff in the area of mobile and wireless networking with Bell Labs Research, Lucent Technologies, NY, USA, and an Assistant Professor with the Department of Computer Science and Engineering, The State University of New York at Buffalo, Buffalo, NY, USA. Her research interests include algorithm and protocol design in wireless networks and communications, mobile and distributed computing, and also networked sensing and detection. She served as a member of the ACM in 2004. She has served in the executive committee and technical committee of numerous conferences and funding review panels, and served as an Associate Editor of the IEEE TRANSACTIONS ON MOBILE COMPUTING. She received the NSF Career Award in 2005 and the ONR Challenge Award in 2010.

Gaogang Xie received the B.S. degree in physics, and the M.S. and Ph.D. degrees in computer science from Hunan University, respectively, in 1996, 1999, and 2002. He is currently a Professor and the Director of the Network Technology Research Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His research interests include Internet architecture, packet processing and forwarding, and Internet measurement.

Jigang Wen received the Ph.D. degree in computer application from Hunan University, China, in 2011. He was a Research Assistant with the Department Of Computing, The Hong Kong Polytechnic University, from 2008 to 2010. He is currently a Post-Doctoral Fellow with the Institute of Computing Technology, Chinese Academy of Science, China. His research interests include wireless network and mobile computing, high-speed network measurement, and management.

Jiannong Cao (M'93–SM'05–F'14) received the Ph.D. degree in computer science from Washington State University, Pullman, WA, USA, in 1990. He is currently a Chair Professor and the Head of the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His research interests include parallel and distributed computing, computer networks, mobile and pervasive computing, fault tolerance, and middleware. He has served as an Associate Editor and a member of the editorial boards of many international journals, including the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE NETWORK, the *Pervasive and Mobile Computing Journal*, *Peer-to-Peer Networking and Applications*, and *Wireless Communications and Mobile Computing*.

Dafang Zhang received the Ph.D. degree in application mathematics from Hunan University, Changsha, China, in 1997. He is currently a Professor with Hunan University. His research interests include packet processing, Internet measurement, wireless network and mobile computing, and big data.