Accurate Recovery of Internet Traffic Data: A Sequential Tensor Completion Approach

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

***Abstract*— The inference of traffic volume of the whole network from partial traffic measurements becomes increasingly critical for various network engineering tasks, such as capacity planning and anomaly detection. Previous studies indicate that the matrix completion is a possible solution for this problem. However, as a 2-D matrix cannot sufficiently capture the spatial-temporal features of traffic data, these approaches fail to work when the data missing ratio is high. To fully exploit hidden spatial-temporal structures of the traffic data, this paper models the traffic data as a 3-way traffic tensor and formulates the traffic data recovery problem as a low-rank tensor completion problem. However, the high computation complexity incurred by the conventional tensor completion algorithms prevents its practical application for the traffic data recovery. To reduce the computation cost, we propose a novel sequential tensor completion algorithm, which can efficiently exploit the tensor decomposition result based on the previous traffic data to derive the tensor decomposition upon arriving of new data. Furthermore, to better capture the changes of data correlation over time, we propose a dynamic sequential tensor completion algorithm. To the best of our knowledge, we are the first to propose sequential tensor completion algorithms to significantly speed up the traffic data recovery process. This facilitates the modeling of Internet traffic with the tensor to well exploit the hidden structures of traffic data for more accurate missing data inference. We have done extensive simulations with the real traffic trace as the input. The simulation results demonstrate that our algorithms can achieve significantly better performance compared with the literature tensor and matrix completion algorithms even when the data missing ratio is high.**

***Index Terms*— Internet traffic data recovery, tensor completion.**

I. Introduction

GAINING a full knowledge of the traffic data volume between a set of origin and destination (OD) pairs in the networks becomes increasingly critical for a wide variety of network engineering tasks [2], including capacity planning, load balancing, path setup, dimensioning, provisioning, anomaly detection, and failure recovery.

Although important, it is impractical to collect measurement data from a very large number of points in a large network at the fine time-scales. To reduce the cost, an alternative measurement strategy usually adopted by the network monitoring system is to take random measurement samples from the full traffic data. The actual data collected can be even less when experiencing data loss under severe communication and system conditions, such as network congestion, node misbehavior, transmission interference [3]–[6], and monitor failure. As many network engineering tasks require the complete traffic information or they are highly sensitive to the missing data, the accurate reconstruction of missing values from partial traffic measurements becomes a key problem, and we refer this problem as the traffic data recovery problem.

Various studies have been made to handle missing traffic data. As most of the known approaches are designed based on purely spatial [7]–[9] or purely temporal [10], [11] information, their data recovery performance is low. To utilize both spatial and temporal information, several recent studies model the traffic data as traffic matrices and propose matrix-based algorithms to recover the missing traffic data [12]–[20]. Although these approaches present good performance when the data missing ratio is low, their performance suffers when the missing ratio is large, especially in the extreme case when the traffic data on several time intervals are all lost.

Tensors are the higher-order generalization of vectors and matrices. Tensor-based multilinear data analysis has shown

Manuscript received November 25, 2016; revised September 17, 2017 and January 14, 2018; accepted January 14, 2018; approved by IEEE/ACM Transactions on Networking Editor X. Wang. Date of publication February 23, 2018; date of current version April 16, 2018. This work was supported by the National Natural Science Foundation of China under Grant 61572184, Grant 61725206, Grant 61472130, Grant 61472131, and Grant 61772191, in part by the Hunan Provincial Natural Science Foundation of China under Grant 2017JJ1010, in part by the Beijing Natural Science Foundation under Grant 4162057, in part by the Science and Technology Key Projects of Hunan Province under Grant 2015TP1004, in part by the U.S. ONR under Grant N00014-17-1-2730, in part by the NSF under Grant ECCS 1408247, Grant CNS 1526843, and Grant ECCS 1731238, 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, and in part by the 2015 China Academy of Sciences Equipment Project (Open Efficient Network Security Test System Development). A preliminary version of this paper is published in IEEE INFOCOM 2016 [1]. *(Corresponding author: Kun Xie.)*

K. Xie is with the College of Computer Science and Electronics Engineering, Hunan University, Changsha 410006, China, with the CAS Key Laboratory of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100864, China, and also with the Department of Electrical and Computer Engineering, State University of New York at Stony Brook, Stony Brook, NY 11794 USA (e-mail: xiekun@hnu.edu.cn).

L. Wang and D. Zhang are with the College of Computer Science and Electronics Engineering, Hunan University, Changsha 410006, China (e-mail: cswangle89@gmail.com; dfzhang@hnu.edu.cn).

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

G. Xie, J. Wen, and G. Zhang are with the Network Research Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100864, China (e-mail: xie@ict.ac.cn; wenjigang@ict.ac.cn; guangxing@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.2018.2797094

that tensor models can take full advantage of the multilinear structures to provide better data understanding and information precision. Tensor-based analytical tools have seen applications for web graphs, knowledge bases, chemometrics, signal processing, computer vision, anomaly detection [21], [22] etc.

To overcome the shortcomings of the matrix-based methods, we propose to model the traffic data based on the multi-way tensor, and design an accurate traffic recovery algorithm. Our algorithm takes advantage of the tensor pattern to combine and exploit correlations among multi-dimensional spatial-temporal data to better preserve multiple features of the traffic data and extract the underlying factors in each feature.

Although promising, compared to matrices, tensors have additional data dimensions and it is more difficult to perform tensor completion. Several tensor completion algorithms [23]–[27] have been proposed for the data recovery with their core lying in the tensor decomposition. Requiring a large number of computations, it is difficult to adopt the existing tensor decomposition methods in the traffic data recovery. It is important and challenging to reduce the computation cost and speed up the tensor completion process.

To design an efficient and accurate traffic data recovery algorithm, we first analyze a large trace of real traffic data, and our studies reveal that there exist hidden structures in the data. To fully exploit theses hidden structures for the data recovery, we model the traffic data as a 3-way traffic tensor and formulate the traffic data recovery problem as a low-rank tensor completion problem. Furthermore, we propose two sequential tensor completion algorithms to quickly solve the problem with a low computation cost. To the best of our knowledge, this is the first time that the tensor pattern is introduced to model the Internet traffic data to well capture the hidden features in the data. Our model has good low-rank property, which helps to preserve the multi-way nature of the traffic data and extract the underlying multi-mode hidden structures in the traffic data. Our contributions are summarized as follows:

- Based on the analysis of real traffic trace, we reveal that traffic data have the features of temporal stability, spatial correlation, and periodicity.
- To fully exploit the hidden structures for the data recovery, we model the traffic data as a 3-way traffic tensor, which allows us to combine and utilize the multi-mode (i.e. OD pair-mode, time-mode, and day-mode) correlations of data to better infer the missing data.
- To reduce the computation cost of the traffic recovery, we propose a Sequential Tensor Completion algorithm (STC) so that the tensor can be decomposed for the current data based on the tensor decomposition result of the previous traffic data. To more accurately recover the data exploiting the feature of the dynamic data, we further propose a Dynamic Sequential Tensor Completion algorithm (DSTC) based on STC. Both algorithms do not need to involve a complete tensor decomposition procedure for the current data, so the computation cost can be significantly reduced.
- To evaluate the performance of our proposed algorithms, we have performed extensive simulations based on real traffic trace. Compared with existing tensor or matrix completion schemes, our algorithms can achieve significantly better performance in terms of several metrics, including the ratio of the recovery error, the ratio of the successful recovery, recovery loss, MAE, RMSE, and the computation time.

The rest of the paper is organized as follows. We introduce the related work in Section II. The preliminaries of tensor are presented in Section III. We present our analyses on the real traffic data, our system model and problem formulation, and our sequential tensor completion algorithm in Section IV, Section V, and Section VI, respectively. Finally, we evaluate the performance of the proposed algorithm through extensive simulations in Section VII, and conclude the work in Section VIII.

II. Related Work

In this section, we review the related work on the recovery of the missing Internet traffic data, and identify the differences of our work from the existing work.

A set of studies have been made to handle the missing traffic data. Designed based on purely spatial [7]–[9] or purely temporal [10], [11] information, most of the known approaches have a low data recovery performance.

To capture more spatial-temporal features in the traffic data, SRMF [12] proposes the first spatio-temporal model of traffic matrices (TMs). It finds sparse approximations to TMs, and recovers the missing data with the spatio-temporal operation and local interpolation. Following SRMF, several other matrix recovery algorithms [12]–[20], [28] including our paper [17], [20], [29] are proposed to recover the missing data from partial measurements. Compared with the vector-based recovery approaches, as a matrix could capture more information and correlation among traffic data, matrix-based approaches achieve much better recovery performance. However, a two-dimension matrix is still limited in capturing a large variety of correlation features hidden in the traffic data. For example, although the traffic matrix defined in [12] can represent the traffic flows in different time slots to catch the spatial correlation among flows and the small-scale temporal feature, it can not incorporate other temporal features such as the feature of the traffic periodicity. Therefore, a matrix is still not enough to capture the comprehensive correlations among the traffic data, and the data recovery performance under the matrix-based approaches is still low.

It is promising to apply the emerging higher-order tensors to model the data that intrinsically have many dimensions. Tensor-based missing data recovery methods can capture the global structure of the data via a high-order decomposition (named tensor decomposition), and tensor-based methods prove to be good analytical tools for dealing with the multi-dimensional data.

So far, tensor-based data recovery has been utilized in various fields (see an in-depth survey by Kolda and Bader [30]). Although tensor completion has proven to be effective in these applications, the features extracted from signal processing [31], deep neural networks [32], and road traffic [33], [34] are different from those of the Internet traffic. The modeling and solutions of existing tensor completion algorithms can not be directly applied for the efficient and accurate Internet traffic recovery. Our recent work in [35] made an attempt to apply the tensor completion to recover the traffic data from partial measurements and loss. Directly modeling the traffic data using 3-way tensor with each mode corresponding respectively to the origin, destination and the total number of time intervals to consider, this study can not exploit the traffic periodicity in the traffic data recovery, so the recovery accuracy is not high.

Several tensor completion algorithms [23]–[26], [36] are proposed for the data recovery. The core of the tensor completion lies in the tensor decomposition, which commonly takes two forms: CANDECOMP/PARAFAC (CP) decomposition [37], [38] and Tucker decomposition [39]. In multilinear algebra, the tensor decomposition may be regarded as a generalization of the matrix singular value decomposition (SVD) to tensors. In fact, Tucker decomposition is also known as a higher-order SVD (HOSVD) [40]. Haeffele and Vidal in [32] present a general framework to analyze a wide variety of factorization problems within a convex formulation (where tensor decomposition is one example), and show that the global minimum of the factorization problems can be achieved if they satisfy a simple condition. Some recent work [41]–[44] study the low-tubal-rank tensor model and low-tubal-rank tensor completion. Among which, the work in [41] provides the first theoretical guarantees on the global optimality for the low-tubal-rank tensor completion problem, and the Liu *et al.* in [42] provide an adaptive tubal sampling strategy to reduce the sampling budget. As the number of elements in a tensor increases exponentially with the number of dimensions, the computational and memory requirements increase quickly, which becomes the main challenge of applying the tensor decomposition in the practical applications.

To the best of our knowledge, we are the first to apply the tensor pattern to model the Internet traffic data to well exploit the hidden structures (temporal stability, spatial correlation feature, and traffic periodic pattern) of the traffic data, and propose sequential tensor completion algorithms to significantly speed up the traffic data recovering process. We have performed extensive simulations with the real traffic trace as the input. The simulation results show that our sequential tensor completion algorithms can achieve highly accurate recovery performance with a short computation time.

III. Preliminaries of Tensor

In this section, we introduce some basic concepts related to the tensor.

Definition 1 (Tensor): A tensor, also known as Nth-order or N-way tensor, multidimensional array, N-way or N-mode array, is a higher-order generalization of a vector (first-order tensor) and a matrix (second-order tensor), and denoted as $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ where N is the order of $\mathcal{A}$, also called way or mode. The element of $\mathcal{A}$ is denoted by $a_{i_1,i_2,\cdots,i_N}, i_n \in \{1,2,\cdots,I_n\}, 1 \le n \le N$.

Definition 2: Given a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, 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:$\mathbf{v} = \mathcal{A}_{\mathbf{i_1},\cdots,\mathbf{i_{k-1}},:,\mathbf{i_{k+1}},\cdots,\mathbf{i_n}}$ 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 [40], 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.

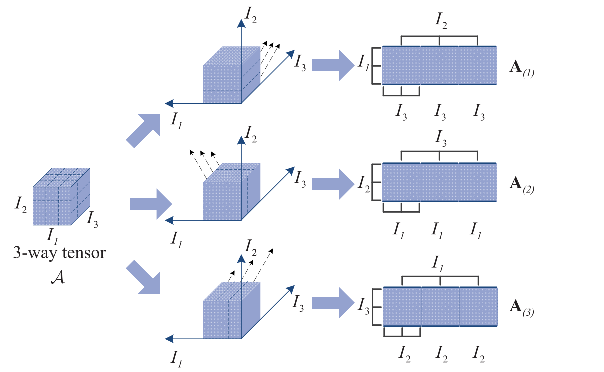


Fig. 1. Unfolding of the $(I_1 \times I_2 \times I_3)-$ tensor $\mathcal{A}$ to the $(I_1 \times I_2 I_3)-$ matrix $\mathbf{A_{(1)}}$, the $(I_2 \times I_3 I_1)-$ matrix $\mathbf{A_{(2)}}$, and the $(I_3 \times I_1 I_2)-$ matrix $\mathbf{A_{(3)}}$.

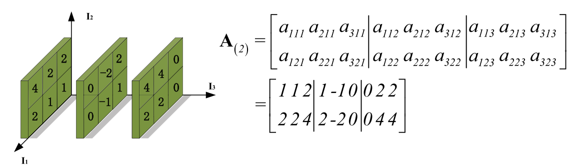


Fig. 2. A tensor $\mathcal{A} \in R^{3\times2\times3}$.

For a Nth-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$, the mode-$k$ unfolded matrix $\mathbf{A}_{(k)} \in \mathbb{R}^{I_k \times \prod_{i \neq k} I_i}$ contains the tensor element $a_{i_1,i_2,\cdots,i_n}$, $i_d \in \{1,2,\cdots,I_d\}$ at the position in the unfolding matrix with its row index i_k and column index j equal to $(i_{n+1}-1) I_{n+2} I_{n+3} \cdots I_N I_1 I_2 \cdots I_{n-1} + (i_{n+2}-1) I_{n+3} I_{n+4} \cdots I_N I_1 I_2 \cdots I_{n-1} + \cdots + (i_N - 1) I_1 I_2 \cdots I_{n-1} + (i_1 - 1) I_2 I_3 \cdots I_{n-1} + (i_2-1) I_3 I_4 \cdots I_{n-1} + \cdots + i_{n-1}$.

Fig. 1 shows an unfolding procedure of a 3rd-order tensor, which involves the tensor dimensions I_1, I_2, I_3 in a cyclic way. The dotted arrow in Fig. 1 shows how the mode-k unfolding matrix is formed. Fig. 2 gives an example of the unfolded matrix $\mathbf{A}_{(2)}$ for a tensor $\mathcal{A} \in \mathbb{R}^{3\times2\times3}$.

Definition 3 (Tensor Rank or CP-Rank [30], [45]): The rank of an arbitrary Nth-order tensor $\mathcal{A}$, denoted by $R = rank(\mathcal{A})$, is the minimal number of rank-1 tensors that yield $\mathcal{A}$ in a linear combination. In other words, this is the smallest number of components in an exact CP decomposition [37], [38].

One major difference between the matrix rank and the tensor CP-rank is that there is no straightforward algorithm to determine the CP-rank of a specific given tensor, which is proven to be NP-hard problem [45].

Definition 4 (n-rank [30]): The n-rank of an arbitrary Nth-order tensor $\mathcal{A}$, denoted by $R_n = rank_n(\mathcal{A})$, is the tuple of the ranks of the N unfolding matrices, that is, $R_n = \left(rank\left(\mathbf{A}_{(1)}\right), rank\left(\mathbf{A}_{(2)}\right), \ldots, rank\left(\mathbf{A}_{(N)}\right)\right)$.

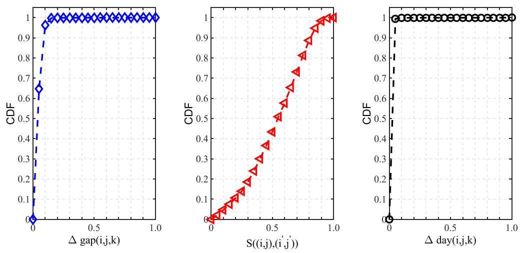


Fig. 3. Empirical study with real traffic data.

IV. Empirical Study With Real Traffic Data

The literature studies [33] have shown that the similarity is one of the factors that impact the interpolation performance for data recovery. In this section, we perform a set of experiments with the public traffic trace Abilene [46] to investigate and discover the Internet traffic features.

A. Temporal Stability

Let Υ denote the non-empty set of all origins and destinations in a network and let $|\Upsilon| = N$. Traffic data are typically measured over some time intervals, and the value reported is an average. Therefore, we can denote $z(i,j,k)$ to be the traffic from origin i to destination j averaged over the time duration $[k, k+\tau)$, where τ denotes the measurement interval.

Traffic data usually change slowly over time. To study the stability of traffic data, we calculate the difference between each pair of adjacent time measurements at a origin-destination (OD) pair. The difference for two consecutive time slots (k, and $k-1$) is equal to

$$gap(i,j,k) = |z(i,j,k) - z(i,j,k-1)| \tag{1}$$

where $1 \leqslant i,j \leqslant N$, $2 \leqslant k \leqslant \Gamma$ and Γ is the number of time intervals of interest. Obviously, $gap(i,j,k) = 0$ if the traffic data of OD pair (i,j) does not change from time slot $k-1$ to k. The smaller the $gap(i,j,k)$, the more stable the traffic data for OD pair (i,j) around time slot k.

By computing the normalized difference values between adjacent time slots, we measure the temporal stability at OD pair (i,j) and time slot k as

$$\Delta gap(i,j,k) = \frac{|z(i,j,k) - z(i,j,k-1)|}{\max\limits_{2\leqslant k\leqslant\Gamma} |z(i,j,k) - z(i,j,k-1)|} \tag{2}$$

where $\max\limits_{2\leqslant k\leqslant\Gamma} |z(i,j,k) - z(i,j,k-1)|$ is the maximal gap between any two consecutive time slots in the traffic data from origin i to destination j.

We plot the CDF of $\Delta gap(i,j,k)$ in Fig.3(a). The X-axis represents the normalized difference values between two consecutive time slots, i.e., $\Delta gap(i,j,k)$. The Y-axis represents the cumulative probability. We observe that more than 90% $\Delta gap(i,j,k)$ are very small (< 0.1). These results indicate that the temporal stability exists in the real traffic data.

B. Spatial Correlation Feature

A correlation coefficient is a quantitative measure of some type of correlation and dependence. Let $z(i,j), z(i',j') \in \mathbb{R}^T$ denote the traffic vectors of OD pair(i,j) and OD pair(i',j'). The spatial correlation between OD pair(i,j) and OD pair(i',j') can be calculated according to

$$S\left((i,j),(i',j')\right) = \frac{\sum\limits_{k=1}^{\Gamma}\left(\frac{|z(i,j,k)-\bar{z}(i,j)|}{\max\limits_{2\leqslant k\leqslant\Gamma}|z(i,j,k)-\bar{z}(i,j)|} \times \frac{|z(i',j',k)-\bar{z}(i',j')|}{\max\limits_{2\leqslant k\leqslant\Gamma}|z(i',j',k)-\bar{z}(i',j')|}\right)}{\sqrt{\sum\limits_{k=1}^{\Gamma}\frac{(z(i,j,k)-\bar{z}(i,j))^2}{\max\limits_{2\leqslant k\leqslant\Gamma}(z(i,j,k)-\bar{z}(i,j))^2}}\sqrt{\sum\limits_{k=1}^{\Gamma}\frac{(z(i',j',k)-\bar{z}(i',j'))^2}{\max\limits_{2\leqslant k\leqslant\Gamma}(z(i',j',k)-\bar{z}(i',j'))^2}}} \tag{3}$$

where $1 \leq i,j,i',j' \leq N$, $\bar{z}(i,j) = \frac{1}{\Gamma}\sum\limits_{k=1}^{\Gamma} z(i,j,k)$, $\bar{z}(i',j') = \frac{1}{\Gamma}\sum\limits_{k=1}^{\Gamma} z(i',j',k)$.

The CDF of $S\left((i,j),(i',j')\right)$ is plotted in Fig. 3(b). The X-axis represents value of $S\left((i,j),(i',j')\right)$, the Y-axis represents the cumulative probability. From the figure, we can see that the value $S\left((i,j),(i',j')\right) < 0.3$ is less than 20%, the value $S\left((i,j),(i',j')\right) > 0.5$ is nearly about 60%, which indicates that real Internet traffic data have strong spatial correlation.

C. Traffic Periodic Pattern

As we know, users usually have similar Internet visiting behaviors at the same time of different days, such as the similar traffic mode in working hours and sleeping hours. To study the traffic periodic pattern in a day, we calculate the gap between each pair of measurements in two consecutive days at an OD pair. In Abilene [46], traffic measurements are taken every 5 minutes, one day have 288 time intervals. Therefore, the gap between each pair of measurements in adjacent days captured in two time slots (k, and $k+288$) is equal to

$$day(i,j,k) = |z(i,j,k) - z(i,j,k+288)| \tag{4}$$

where $1 \leqslant i,j \leqslant N$ and $1 \leqslant k \leqslant \Gamma - 288$ and Γ is time intervals present. Obviously, the smaller the $day(i,j,k)$, the more similar the traffic data for OD pair (i,j) around the same time slot of adjacent days.

By computing the normalized difference values between adjacent days, we measure the traffic similarity at OD pair (i,j) and time slot k according to

$$\Delta day(i,j,k) = \frac{|z(i,j,k) - z(i,j,k+288)|}{\max\limits_{1\leqslant k\leqslant\Gamma-288} |z(i,j,k) - z(i,j,k+288)|} \tag{5}$$

where $\max\limits_{1\leqslant k\leqslant\Gamma-288} |z(i,j,k) - z(i,j,k+288)|$ is the maximal gap between any two adjacent days in the traffic data from origin i to destination j.

We plot the CDF of $\Delta day(i,j,k)$ in Fig.3(c). The X-axis represents the normalized difference values between two adjacent days, i.e., $\Delta day(i,j,k)$. The Y-axis represents the cumulative probability. We observe that more than 90% $\Delta day(i,j,k)$ are very small (< 0.05). These results indicate that traffic periodic pattern exists in real Internet traffic trace.

V. System Model and Problem Formulation

In this section, we first present our traffic tensor model, and then formulate the traffic data recovery problem.

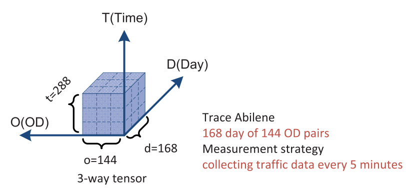


Fig. 4. Traffic tensor model.

A. Traffic Tensor Model

Current traffic interpolation approaches usually model the traffic data with a traffic matrix $\mathbf{X} \in \mathbb{R}^{o\times\Gamma}$ $(o = N \times N)$, where a column of $\mathbf{X}$ represents the traffic data of all OD pairs at one time slot, while a row of $\mathbf{X}$ represents the time evolution of a single OD pair. As discussed in the introduction, modeling the traffic data in the matrix format cannot sufficiently capture spatial and temporal characteristics of the traffic data. Therefore, although matrix-based approaches work well when the ratio of the missing data is low, their performances degrade significantly when the data missing ratio becomes large.

To address the issues of the matrix-based methods mentioned above, we propose to apply the tensor to model traffic data. As a straightforward way of modeling [35], traffic tensor may be formed with a 3-way tensor $\mathcal{Z} \in \mathbb{R}^{N\times N\times\Gamma}$, corresponding respectively to the origin, destination and the total number of time intervals to consider. However, such a 3-way tensor model can not fully exploit the similarity structures hidden in the traffic data.

To fully exploit the traffic features of temporal stability, spatial correlation, as well as the periodicity pattern, we model the traffic data as a 3-way tensor $\mathcal{X} \in \mathbb{R}^{o\times t\times d}$ (as shown in Fig.4), where o corresponds to $N \times N$ OD pairs, and there are d days to consider with each day having t time intervals. Obviously, we have $\Gamma = t \times d$. Fig.4 uses Abilene trace data [46] 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 $\mathcal{X} \in \mathbb{R}^{o\times t\times d}$ with $o = 144$, $t = 288$, and $d = 168$. According to [47], the missing data recovery performance becomes better when the dimensions of the tensor are more balanced. If we use the traffic tenor in [35] with the tensor formed with source, destination and time slot, for the data taken from 12 sources and 12 destinations (totally 144 OD pairs) over 288×168 time slots, the tensor would be in a oblong shape with the dimension of the time slot much larger than the other two dimensions. Instead, with our tensor setup, the sizes of three dimensions are much more balanced with 144(OD), 168(Day), and 288(time slot). Therefore, the tensor model in this paper is better than the one in [35] for missing data recovery.

B. Problem Formulation

Before we present our problem formulation, we first utilize following table to summarize some most used variables.

Variables	
$\mathcal{X} \in \mathbb{R}^{o\times t\times d}$	Raw traffic tensor
$\mathcal{M} \in \mathbb{R}^{o\times t\times d}$	Measurement traffic tensor
Ω	Set of indices of the observed entries in $\mathcal{M}$
$\mathbf{X}_{(i)}$, $\mathbf{M}_{(i)}$	ith-mode unfolding matrix of $\mathcal{X}$ and $\mathcal{M}$
$l_{(i)}$, $n_{(i)}$	No. rows and No. columns of $\mathbf{X}_{(i)}$ and $\mathbf{M}_{(i)}$
$\mathbb{U}_{ij}$	Vector space of the matrices $\in \mathbb{R}^{i\times j}$

$\mathcal{M}$ is generally an incomplete tensor due to sample-based traffic monitoring and the unavoidable data loss resulted from severe communication conditions. We define the operation $\mathcal{M}_\Omega = \mathcal{X}_\Omega$ as

$$m_{ijk} = \begin{cases} x_{ijk} & if\ (i,j,k) \in \Omega \\ 0 & otherwise \end{cases} \tag{6}$$

If there are no traffic data between a particular pair of nodes in a given time interval, of course, it leaves the corresponding entry in $\mathcal{M}$ to be empty. In our study, we use zero as a placeholder to replace the empty entry.

To recover the missing traffic data, the traffic data recovery problem can be formulated as a tensor completion problem with the goal of finding its missing entries through the minimization of the tensor rank as

$$\begin{aligned} &\min_{\mathcal{X}}\ rank\,(\mathcal{X}) \\ &s.t.\ \mathcal{X}_\Omega = \mathcal{M}_\Omega \end{aligned} \tag{7}$$

According to the definition of n-rank, the n-rank of a given n-way tensor can be analyzed by means of matrix techniques. Therefore, the tensor completion problem defined in (7) can be further transformed to

$$\begin{aligned} &\min_{\mathcal{X}} \sum_{i}^{3} rank\left(\mathbf{X}_{(i)}\right) \\ &s.t.\ \mathcal{X}_\Omega = \mathcal{M}_\Omega \end{aligned} \tag{8}$$

VI. Traffic Data Recovery

In order to reduce the computation cost, we propose two sequential tensor completion algorithms to quickly recover the traffic data. In this section, we first analyze the challenges, then present the proposed algorithms.

A. Challenges

The problem in (8) considers the tensor as multiple matrices and forces the matrix unfolded along each mode of the tensor to be low rank. Therefore, the tensor completion problem is transformed to the low-rank matrix completion problem along each mode, and the final tensor data can be obtained by folding the recovered data of each mode.

Rather than straight-forwardly recovering the matrix data, to speed up the tensor completion process, we consider an approach with two steps: 1) Finding a recovering method that allows for the reuse of the previous recovered data to reduce the computation complexity, and 2) Developing an algorithm that can efficiently recover new data during the continuous monitoring process. Although promising, there are a few challenges to address:

- ***What algorithm to use to recover each unfolded matrix of the tensor?*** Existing solutions generally recover a matrix via convex relaxation with the nuclear norm minimization. This method can work efficiently if the matrix satisfies certain incoherence conditions [49], [50] and sufficiently number of entries are observed. However, it may bring long computation time and even not converge when the sample data are not sufficient.

- ***How to reduce the computation to reuse previous tensor factorization ?*** To reuse the previous results of the tensor factorization, the derived tensor factorization should be able to capture the main features of both historical data and the current data based only on past data and partially observed new data, and the solution needs to be simple for implementation for online monitoring.

B. Completion of Unfolded Matrix With the Reuse of Previous Results

Given the limitation of conventional methods based on nuclear norm minimization, to support efficient online monitoring with sequential sampling scheme while ensuring low-overhead data recovery, we consider a completion method that searches for the orthogonal column space on the Grassmann manifold to match the partial measurement data.

To find a rank-$r_{(i)}$ matrix $\mathbf{X}'_{(i)}$ that is consistent with the observations $(\mathbf{M}_{(i)})_\Omega$, the column space searching problem for the matrix completion can be expressed as

$$\min_{\mathbf{U}_{(i)} \in \mathbb{U}_{l_{(i)} r_{(i)}}} \left\| \left(\mathbf{M}_{(i)} - \mathbf{U}_{(i)} \mathbf{W}_{(i)}^{tr} \right)_\Omega \right\|_F^2 \tag{9}$$

where $\mathbf{U}_{(i)}$ is the column orthogonal matrix of matrix $\mathbf{M}_{(i)}$. $\|\cdot\|_F$ denotes the Frobenius norm, and $\mathbf{W}_{(i)}^{tr}$ denotes the transpose of $\mathbf{W}_{(i)}$. Given a $\mathbf{U}_{(i)}$, the $\mathbf{W}_{(i)}$ in Eq.(9) can be calculated through the following function:

$$\mathbf{W}_{(i)} = \arg\min_{\mathbf{W}_{(i)} \in \mathbb{R}^{n_{(i)} \times r_{(i)}}} \left\| \left(\mathbf{M}_{(i)} - \mathbf{U}_{(i)} \mathbf{W}_{(i)}^{tr} \right)_\Omega \right\|_F^2 \tag{10}$$

The low-rank matrix completion is transformed to the column space searching problem with the aim of finding a column space consistent with the observed entries. As we don't know $\mathbf{W}_{(i)}$ in advance, to find the optimal column orthogonal matrix $\mathbf{U}_{(i)}$, problems in (9) and (10) should be iteratively solved until it converges. After we obtain the column orthogonal matrix $\mathbf{U}_{(i)}$ and matrix $\mathbf{W}_{(i)}$, through $\mathbf{U}_{(i)}\mathbf{W}_{(i)}^{tr}$, the incomplete ith-mode matrix $\mathbf{M}_{(i)}$ can be recovered and the $\mathbf{X}'_{(i)} = \mathbf{U}_{(i)}\mathbf{W}_{(i)}^{tr}$ is the resulted recovery matrix.

In the following contents, we will further utilize the good feature of the column orthogonal matrix to propose a sequential tensor completion approach to significantly speed up traffic recovery process.

Traffic measurement data generally come in sequence. To obtain the complete traffic data for the advanced network management, the tensor completion task will be invoked periodically or upon the request of the network operators. It would involve a large computation cost if we directly solve the column space searching problem by iteratively executing (9) and (10) to find the column space of each unfolding matrix when the tensor completion task is invoked.

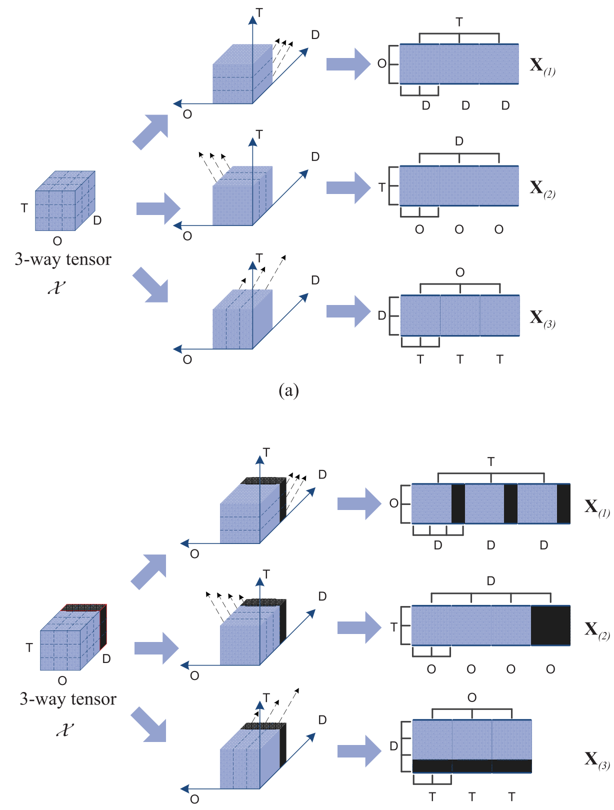


Fig. 5. Sequence tensor completion tasks at time t and $t+1$. (a) Traffic tensor and its unfolding matrices obtained at time t. (b) Traffic tensor and its unfolding matrices obtained at time $t+1$.

Fig. 5 shows the sequence of the tensor completion tasks at time t and time t+1. Comparing the Fig.5 (a) with the Fig.5(b), the major parts of the traffic data (undertone color data) are the same in both figures, where the traffic data are recovered from the previous measurements. The only difference is that the tensor in Fig.5(b) has more traffic data than the tensor in Fig. 5(a), and consequently more columns and rows in the unfolding matrices. The additional data are obtained from the new measurements. This relationship provides us an opportunity to reuse the previous result of tensor decomposition to deduce the tensor decomposition for the current data so the data can be quickly recovered.

For a rank-r matrix $\mathbf{X} = [\mathbf{X}_{\{1,\ldots,n-1\}}, x_{\{n\}}] \in \mathbb{R}^{l\times n}$, where $\mathbf{X}_{\{1,\ldots,n-1\}}$ is the submatrix of $\mathbf{X}$ by removing the last column from $\mathbf{X}$. Let $\mathbf{M}$ be the observation matrix of $\mathbf{X}$, that is, $\mathbf{M}_\Omega = \mathbf{X}_\Omega$ and $\left(m_{\{n\}}\right)_{\Omega_n} = \left(x_{\{n\}}\right)_{\Omega_n}$ where $m_{\{n\}}$ and $x_{\{n\}}$ are the last column of $\mathbf{M}$ and $\mathbf{X}$, respectively, with its observed entry set Ω_n. To recover matrix $\mathbf{X}$ from $\mathbf{M}$, before we present our sequential tensor completion algorithm in Algorithm 1, the following theorem will illustrate how to calculate the column orthogonal matrix of $\mathbf{M} = [\mathbf{M}_{\{1,\ldots,n-1\}}, m_{\{n\}}] \in \mathbb{R}^{l\times n}$ based on the obtained column orthogonal matrix of $\mathbf{M}_{\{1,\ldots,n-1\}}$.

Theorem 1: Let $\mathbf{U}_1 = \arg\min_{\mathbf{U}\in\mathbb{U}_{lr}} \|(\mathbf{M} - \mathbf{U}\mathbf{W}^{tr})_\Omega\|_F^2$, *and define*

$$\mathbf{U}' = \mathbf{U}_0 + \left((cos\,(\sigma\eta) - 1)\,\frac{p}{\|p\|} + sin\,(\sigma\eta)\,\frac{\ell}{\|\ell\|} \right) \frac{\omega^{tr}}{\|\omega\|} \tag{11}$$

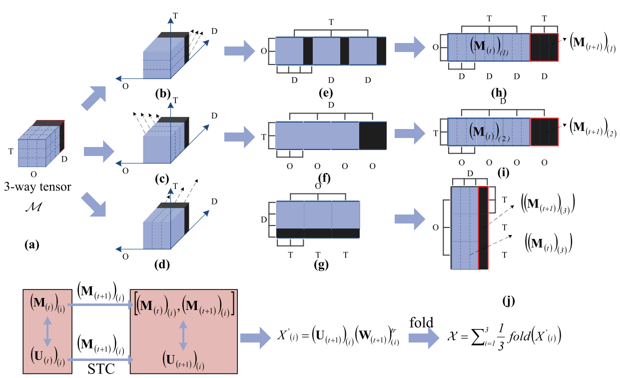


Fig. 6. Sequential tensor completion for traffic data.

where $\mathbf{U}_0$ *is an* $l \times r$ *matrix whose orthogonal columns span* $\mathbf{M}_{\{1,\ldots,n-1\}}$*, and* $\eta > 0$ *is a small stepsize,* $\omega = argmin_{\omega} \left\| (\mathbf{U}_0)_{\Omega_n} \omega - \left(m_{\{n\}}\right)_{\Omega_n} \right\|_2^2$ *is the least-squares weight,* $p = \mathbf{U}_0 \omega$*,* $\ell = \left(m_{\{n\}}\right)_{\Omega_\eta} - p$ *is the residual vector, and* $\sigma = \|\ell\| \, \|p\|$*. Then* $\mathbf{U}_1$ *and* $\mathbf{U}'$ *are identical with a specific choice for step size* η*.*

According to the orthonormal columns of $\mathbf{U}_0$ which spans $\mathbf{M}_{\{1,\ldots,n-1\}}$, we can get

$$\mathbf{U}_0 = \underset{\mathbf{U} \in \mathbb{U}_{lr}}{argmin} \left\| \left(\mathbf{M}_{\{1,\ldots,n-1\}} - \mathbf{U}\mathbf{W}^{tr}\right)_{\Omega} \right\|_F^2$$

From $m_{\{n\}} = \mathbf{U}_0 w + \ell$, we have

$$m_{\{n\}} = \begin{bmatrix} \mathbf{U}_0 & \frac{\ell}{\|\ell\|} \end{bmatrix} \begin{bmatrix} w \\ \|\ell\| \end{bmatrix}.$$

And then, we can get

$$[\mathbf{M}_{\{1,\ldots,n-1\}}, m_{\{n\}}] = \begin{bmatrix} \mathbf{U}_0 \frac{\ell}{\|\ell\|} \end{bmatrix} \begin{bmatrix} \mathbf{I} & w \\ 0 & \|\ell\| \end{bmatrix} \begin{bmatrix} \mathbf{W} & 0 \\ 0 & 1 \end{bmatrix}^{tr}.$$

Furthermore, we have

$$\begin{bmatrix} \mathbf{U}_0 \frac{\ell}{\|\ell\|} \end{bmatrix} = \underset{\mathbf{U} \in \mathbb{U}_{l(r+1)}}{argmin} \|(\mathbf{M} - \mathbf{U}\mathbf{W}^{tr})_{\Omega}\|_F^2$$

Taking the SVD of the center matrix to be

$$\begin{bmatrix} \mathbf{I} & w \\ 0 & \|\ell\| \end{bmatrix} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^{tr}; \tilde{\mathbf{\Sigma}} = \begin{bmatrix} \delta_1 & & & \\ & \ddots & & \\ & & \delta_r & \\ & & & \delta_{r+1} \end{bmatrix}.$$

To find a matrix $\in \mathbb{U}_{lr}$ by catching most energy of the first r singular values of matrix $\mathbf{M}$, set

$$\mathbf{U}_{t+1} = \left(\begin{bmatrix} \mathbf{U}_0 & \frac{\ell}{\|\ell\|} \end{bmatrix} \tilde{\mathbf{U}} \right)_{\{1,\ldots,r\}},$$

$$\mathbf{W}_{t+1} = \left(\begin{bmatrix} \mathbf{W} & 0 \\ 0 & 1 \end{bmatrix} \tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}} \right)_{\{1,\ldots,r\}},$$

only the top r singular vectors are needed.

According the ISVD algorithm in [51], we can get $\mathbf{U}_{t+1} = \underset{\mathbf{U} \in \mathbb{U}_{lr}}{argmin} \left\| \left([\mathbf{M}_{\{1,\ldots,n-1\}}, m_{\{n\}}] - \mathbf{U}\mathbf{W}^{tr}\right)_{\Omega} \right\|_F^2$

Proof:

It was shown in [52] that updating $\mathbf{U}_0$ to $\mathbf{U}_{t+1}$ is equivalent to GROUSE for a specific step size η, which performs the gradient descent directly on the Grassmann manifold, that is, $\mathbf{U}_{t+1} = \mathbf{U}' = \mathbf{U}_0 + \left((cos(\sigma\eta) - 1) \frac{p}{\|p\|} + sin(\sigma\eta) \frac{\ell}{\|\ell\|} \right) \frac{\omega^{tr}}{\|\omega\|}$, which completes the proof. ■

According to Theorem 1, when a new column vector v is appended to the matrix $\mathbf{M}$, we do not need a new column space searching procedure to calculate the orthogonal column matrix $\mathbf{U}'$ for the matrix $[\mathbf{M}, v]$. Instead, $\mathbf{U}'$ can be derived from $\mathbf{U}$ and v only, where $\mathbf{U}$ is the orthogonal column matrix of $\mathbf{M}$ whose orthogonal columns span $\mathbf{M}$. Therefore, Theorem 1 provides a good approach to reuse the column space found for the previous traffic data to quickly recover the current traffic data.

C. Sequential Tensor Completion

To design our sequential tensor completion algorithm, we first provide some notations. As shown in Fig.6, the undertone color data are processed in the previous tensor completion procedure, while the dark color data are newly obtained. The three unfolding matrices of the traffic tensor (in Fig. 6 (a)) are shown in Fig. 6 (e), Fig. 6(f), and Fig. 6(g), which can be further transformed into Fig. 6(h), Fig. 6(i), and Fig.6(j), respectively. Note that, the traffic matrices in Fig. 6(h) and (i) are the elementary transformation of matrices in Fig. 6(e) and (f), the traffic matrix in Fig. 6(j) is the transpose of the matrix in Fig. 6(g).

As shown in Fig. 6(h), Fig. 6(i), and Fig. 6(j), we denote undertone color data as $(\mathbf{M}_{(t)})_{(1)}$, $(\mathbf{M}_{(t)})_{(2)}$, and $(\mathbf{M}_{(t)})_{(3)}$, and the remainder dark color data as $(\mathbf{M}_{(t+1)})_{(1)}$, $(\mathbf{M}_{(t+1)})_{(2)}$, and $(\mathbf{M}_{(t+1)})_{(3)}$, respectively. According to Theorem 1, by utilizing the column space of $(\mathbf{M}_{(t)})_{(1)}$, $(\mathbf{M}_{(t)})_{(2)}$, and $(\mathbf{M}_{(t)})_{(3)}$ to calculate the column space of the whole $\left[\left(\mathbf{M}_{(t)}\right)_{(1)}, \left(\mathbf{M}_{(t+1)}\right)_{(1)}\right]$, $\left[\left(\mathbf{M}_{(t)}\right)_{(2)}, \left(\mathbf{M}_{(t+1)}\right)_{(2)}\right]$, and $\left[\left(\mathbf{M}_{(t)}\right)_{(3)}, \left(\mathbf{M}_{(t+1)}\right)_{(3)}\right]$, we design our Sequential Tensor Completion algorithm (STC), as shown in Algorithm 1.

As shown on lines 4-6 in Algorithm 1, for the newly coming traffic data in $(\mathbf{M}_{(t+1)})_{(i)}$ $(1 \leq i \leq 3)$, we add each column in $(\mathbf{M}_{(t+1)})_{(i)}$ sequentially to existing data, and update the corresponding column space by utilizing the previous $\left(\mathbf{U}_{(t)}\right)_{(i)}$ and the new column to add.

To train the column orthogonal matrix $\left(\mathbf{U}_{(t+1)}\right)_{(i)}$ to more accurately capture the column information of the newly traffic data in $(\mathbf{M}_{(t+1)})_{(i)}$, we repeat the training procedure (lines 4-6) $CycleNum$ rounds.

Then according to Eq.(10), calculate the optimal $\left(\mathbf{W}_{(t+1)}\right)_{(i)}$ and set $\mathbf{X}'_{(i)} = \left(\mathbf{U}_{(t+1)}\right)_{(i)} \left(\mathbf{W}_{(t+1)}\right)_{(i)}^{tr}$ as the recovery matrix for this unfolding matrix. After folding each recovered unfolding matrix $\mathbf{X}'_{(1)}$, $\mathbf{X}'_{(2)}$, and $\mathbf{X}'_{(3)}$, the recovered traffic tensor can obtained as shown on line 9.

D. Dynamic Sequential Tensor Completion

In Algorithm 1, the number of columns of the orthogonal matrix from $\left(\mathbf{U}_{(t)}\right)_{(i)}$ to $\left(\mathbf{U}_{(t+1)}\right)_{(i)}$ is kept the same. However, our recent studies [17], [53] reveal that the rank of the matrix varies over time, although the network monitoring data can be represented as a low rank matrix. When new traffic data comes, the rank of whole traffic data may increase as more data is added. Therefore, fixing the number of columns (thus the

Algorithm 1 Sequential Tensor Completion (STC)

Input: The orthogonal matrices $\left(\mathbf{U}_{(t)}\right)_{(1)}$, $\left(\mathbf{U}_{(t)}\right)_{(2)}$, $\left(\mathbf{U}_{(t)}\right)_{(3)}$ for $(\mathbf{M}_{(t)})_{(1)}$, $(\mathbf{M}_{(t)})_{(2)}$, and $(\mathbf{M}_{(t)})_{(3)}$

Output: The recovered traffic tensor $\mathcal{X}$

1: **for** $i \leftarrow 1, \ldots, 3$ **do**

2: $\left(\mathbf{U}_{(t+1)}\right)_{(i)} = \left(\mathbf{U}_{(t)}\right)_{(i)}$

3: **for** $k \leftarrow 1, \cdots, CycleNum$ **do**

4: **for** each column vector v in $(\mathbf{M}_{(t+1)})_{(i)}$ with its observed entry set Ω_v **do**

5: Apply Theorem 1 to update the column orthogonal matrix

$$\left(\mathbf{U}_{(t+1)}\right)_{(i)} = \left(\mathbf{U}_{(t+1)}\right)_{(i)} + \left((cos\,(\sigma\eta) - 1)\,\frac{p}{\|p\|} + sin\,(\sigma\eta)\,\frac{\ell}{\|\ell\|}\right)\frac{\omega^{tr}}{\|\omega\|} \quad (12)$$

where

$$\omega = argmin_{\omega}\left\|\left(\mathbf{U}_{(t+1)}\right)_{(i)\Omega_v}\omega - (v)_{\Omega_v}\right\|_2^2 \quad (13)$$

$p = \left(\mathbf{U}_{(t+1)}\right)_{(i)}\omega$, residual $\ell = (v)_{\Omega_v} - p$, and $\sigma = \|\ell\|\,\|p\|$.

6: **end for**

7: **end for**

8: According to Eq.(10), $\left(\mathbf{W}_{(t+1)}\right)_{(i)}$ can be calculated from

$$\left(\mathbf{W}_{(t+1)}\right)_{(i)} = \underset{\mathbf{W}\in\mathbb{R}^{n_{(i)}\times r_{(i)}}}{\arg\min}\left\|\left(\begin{matrix}\left[\left(\mathbf{M}_{(t)}\right)_{(i)}, \left(\mathbf{M}_{(t+1)}\right)_{(i)}\right]\\ -\left(\mathbf{U}_{(t+1)}\right)_{(i)}\mathbf{W}^{tr}\end{matrix}\right)_{\Omega}\right\|_F^2 \quad (14)$$

where $n_{(i)} = (n_{(t)})_{(i)} + (n_{(t+1)})_{(i)}$ with $(n_{(t)})_{(i)}$ and $(n_{(t+1)})_{(i)}$ being the numbers of columns of matrices $(\mathbf{M}_{(t)})_{(i)}$ and $(\mathbf{M}_{(t+1)})_{(i)}$, respectively.

9: $\mathbf{X}'_{(i)} = \left(\mathbf{U}_{(t+1)}\right)_{(i)}\left(\mathbf{W}_{(t+1)}\right)_{(i)}^{tr}$

10: **end for**

11: $\mathcal{X} = \sum_{i=1}^{3}\frac{1}{3}fold\left(\mathbf{X}'_{(i)}\right)$

12: Return traffic tensor $\mathcal{X}$.

rank r) of the orthogonal matrix can not well represent practical network monitoring data and the recovery performance has room to improve. Therefore, based on Algorithm 1, we further design DSTC (Dynamic Sequential Tensor Completion) in Algorithm 2 to dynamically change the rank of the orthogonal matrix when needed to more accurately capture the feature of the data to more accurately recover the data.

In Algorithm 2, the orthogonal matrix is trained multiple times on lines 10-14 similar to lines 3-7 in Algorithm 1. However, when scanning each column of the newly coming traffic data the first time, the rank will change and new column will be added when needed in Algorithm 2.

Specially, on line 4, as v_j is a sparse vector with only small portion of its entries (in Ω_{v_j}) having values, we set $v_j = (v_j)_{\Omega_{v_j}} + (e)_{\bar{\Omega}_{v_j}}$, where e is a small vector with the values of its entries very small (i.e., 10^{-6}) and $\bar{\Omega}_{v_j}$ is the set of locations of unobserved entries in v_j. As the column vectors in the orthogonal matrix act as the basis vectors to span the traffic data space, adding vector (e) is to avoid the newly added column vector in the orthogonal matrix to be a sparse vector. On line 5, $\ell_j = (v_j) - p$ is the residual vector of column v_j represented by the current orthogonal matrix. Our column expanding principle is that: when the ratio of $\|\ell_j\|$ is larger than $\alpha\|\ell_{j-1}\|$, a new column is added to the orthogonal matrix and the orthogonal matrix is updated to $\left(\mathbf{U}_{(t+1)}\right)_{(i)} = [\left(\mathbf{U}_{(t+1)}\right)_{(i)}, \frac{\ell_j}{\|\ell_j\|}]$. Obviously, as ℓ_j is the residual vector of column v_j, it is orthogonal to the columns of the matrix updated before. On line 6, α is constant with $\alpha > 1$. In this paper, we set $\alpha = 2$ in the simulation part.

In the simulation part, we will show that such a dynamic design can effectively improve the recovery performance of the static algorithm.

E. Algorithm analysis

In this section, we compare the computation complexity of our two sequential tensor completion algorithms (STC and DSTC) with that of the direct matrix completion solution.

According to the definition of n-rank, to complete a tensor, we first solve the low-rank matrix completion problem along each mode, and then obtain the final tensor data by folding the recovered data of each mode. Moreover, for accurate missing data recovery, we further translate the matrix completion problem to a column space searching problem and propose two sequential algorithms (STC and DSTC) to solve the problem. As the final recovered tensor is folded by the recovered matrices of all modes, we analyze the computation complexity of the matrix completion algorithm for each mode.

As shown in Algorithm 1, our STC mainly includes two parts. The first is to deduce the column orthogonal matrix $\left(\mathbf{U}_{(t+1)}\right)_{(i)}$ based on the matrix decomposition result of previous data. The second part calculates the matrix $\left(\mathbf{W}_{(t+1)}\right)_{(i)}$ by solving the matrix least squares minimization problem in (14).

According to Theorem 1, the matrix $\left(\mathbf{U}_{(t+1)}\right)_{(i)}$ is updated from $\left(\mathbf{U}_{(t)}\right)_{(i)}$ by using each column of the new measurement data matrix $\left(\mathbf{M}_{(t+1)}\right)_{(i)}$ to train $\left(\mathbf{U}_{(t+1)}\right)_{(i)}$ through operation in (12), which requires solving a least squares problem in (13). This least squares problem further requires obtaining $r_{(i)}$ unknowns in $\left|\Omega_{v_j}\right|$ equations, so the complexity is $O\left(\left|\Omega_{v_j}\right| r_{(i)}^2\right)$. As each column at most has $l_{(i)}$ entries and the measurement data matrix $\left(\mathbf{M}_{(t+1)}\right)_{(i)}$ has $(n_{(t+1)})_{(i)}$ columns, one round of such a training process requires at most $O\left(l_{(i)} r_{(i)}^2 (n_{(t+1)})_{(i)}\right)$. For more accurate data recovery, the training process is executed in STC CycleNum rounds. Thus the complexity of the first part is $O\left(CycleNum l_{(i)} r_{(i)}^2 (n_{(t+1)})_{(i)}\right)$.

The least squares problem in (14) is solved in the second part of STC. As the measurement data matrix $\left[\left(\mathbf{M}_{(t)}\right)_{(i)}, \left(\mathbf{M}_{(t+1)}\right)_{(i)}\right]$ including $(n_{(t)})_{(i)} + (n_{(t+1)})_{(i)}$

Algorithm 2 Dynamic Sequential Tensor Completion (DSTC)

Input: The orthogonal matrices $\left(\mathbf{U}_{(t)}\right)_{(1)}$, $\left(\mathbf{U}_{(t)}\right)_{(2)}$, $\left(\mathbf{U}_{(t)}\right)_{(3)}$ for $(\mathbf{M}_{(t)})_{(1)}$, $(\mathbf{M}_{(t)})_{(2)}$, and $(\mathbf{M}_{(t)})_{(3)}$

Output: The recovered traffic tensor $\mathcal{X}$

1: **for** $i \leftarrow 1, \ldots, 3$ **do**

2: $\left(\mathbf{U}_{(t+1)}\right)_{(i)} = \left(\mathbf{U}_{(t)}\right)_{(i)}$, $\|\ell_0\| = 0$

3: **for** each column vector v_j in $(\mathbf{M}_{(t+1)})_{(i)}$ with its observed entry set Ω_{v_j} **do**

4: $v_j = (v_j)_{\Omega_{v_j}} + (e)_{\bar{\Omega}_{v_j}}$

5:

$$\left(\mathbf{U}_{(t+1)}\right)_{(i)} = \left(\mathbf{U}_{(t+1)}\right)_{(i)} + \left((cos\,(\sigma\eta) - 1)\,\frac{p}{\|p\|} + sin\,(\sigma\eta)\,\frac{\ell_j}{\|\ell_j\|}\right)\frac{\omega^{tr}}{\|\omega\|} \quad (15)$$

where

$$\omega = argmin_{\omega}\left\|\left(\mathbf{U}_{(t+1)}\right)_{(i)}\omega - (v_j)\right\|_2^2 \quad (16)$$

$p = \left(\mathbf{U}_{(t+1)}\right)_{(i)}\omega$, residual $\ell_j = (v_j) - p$, and $\sigma = \|\ell_j\|\,\|p\|$.

6: **if** $(\|\ell_j\| > \alpha\|\ell_{j-1}\|$ **then**

7: $\left(\mathbf{U}_{(t+1)}\right)_{(i)} = [\left(\mathbf{U}_{(t+1)}\right)_{(i)}, \frac{\ell_j}{\|\ell_j\|}]$

8: **end if**

9: **end for**

10: **for** $k \leftarrow 1, \cdots, CycleNum$ **do**

11: **for** each column vector v in $(\mathbf{M}_{(t+1)})_{(i)}$ with its observed entry set Ω_v **do**

12: Apply Theorem 1 to update the column orthogonal matrix

$$\left(\mathbf{U}_{(t+1)}\right)_{(i)} = \left(\mathbf{U}_{(t+1)}\right)_{(i)} + \left((cos\,(\sigma\eta) - 1)\,\frac{p}{\|p\|} + sin\,(\sigma\eta)\,\frac{\ell}{\|\ell\|}\right)\frac{\omega^{tr}}{\|\omega\|} \quad (17)$$

where $\omega = argmin_{\omega}\left\|\left(\mathbf{U}_{(t+1)}\right)_{(i)_{\Omega_v}}\omega - (v)_{\Omega_v}\right\|_2^2$, $p = \left(\mathbf{U}_{(t+1)}\right)_{(i)}\omega$, residual $\ell = (v)_{\Omega_v} - p$, and $\sigma = \|\ell\|\,\|p\|$.

13: **end for**

14: **end for**

15: According to Eq.(10), $\left(\mathbf{W}_{(t+1)}\right)_{(i)}$ can be calculated from

$$\left(\mathbf{W}_{(t+1)}\right)_{(i)} = \underset{\mathbf{W}\in\mathbb{R}^{n_{(i)}\times r_{(i)}}}{\arg\min}\left\|\left(\begin{matrix}\left[\left(\mathbf{M}_{(t)}\right)_{(i)}, \left(\mathbf{M}_{(t+1)}\right)_{(i)}\right] \\ -\left(\mathbf{U}_{(t+1)}\right)_{(i)}\mathbf{W}^{tr}\end{matrix}\right)_{\Omega}\right\|_F^2 \quad (18)$$

where $n_{(i)} = (n_{(t)})_{(i)} + (n_{(t+1)})_{(i)}$ with $(n_{(t)})_{(i)}$ and $(n_{(t+1)})_{(i)}$ being the numbers of columns of matrices $(\mathbf{M}_{(t)})_{(i)}$ and $(\mathbf{M}_{(t+1)})_{(i)}$, respectively.

16: $\mathbf{X}'_{(i)} = \left(\mathbf{U}_{(t+1)}\right)_{(i)}\left(\mathbf{W}_{(t+1)}\right)_{(i)}^{tr}$

17: **end for**

18: $\mathcal{X} = \sum_{i=1}^{3}\frac{1}{3}fold\left(\mathbf{X}'_{(i)}\right)$

19: Return traffic tensor $\mathcal{X}$.

columns, according to the complexity of problem in (13), the complexity of solving problem in (14) is at most $O\left(l_{(i)}r_{(i)}^2((n_{(t)})_{(i)} + (n_{(t+1)})_{(i)})\right)$.

Therefore, the main operations to train $\left(\mathbf{U}_{(t+1)}\right)_{(i)}$ and the matrix $\left(\mathbf{W}_{(t+1)}\right)_{(i)}$ in STC requires $O\left(CycleNuml_{(i)}r_{(i)}^2(n_{(t+1)})_{(i)}\right) + O\left(l_{(i)}r_{(i)}^2((n_{(t)})_{(i)} + (n_{(t+1)})_{(i)})\right)$.

Different from STC, in Algorithm 2, DSTC need one more round to train $\left(\mathbf{U}_{(t+1)}\right)_{(i)}$ by testing whether new measurement data will change the rank of $\left(\mathbf{U}_{(t+1)}\right)_{(i)}$ in line 1-9 in Algorithm 2. The main operation in this round requires solving a least squares problem in (16) which is slightly different from (13) in STC. The problem in (16) requires obtaining $r_{(i)}$ unknowns in $l_{(i)}$ equations, which results in $O\left(l_{(i)}r_{(i)}^2\right)$ complexity. As the measurement matrix $\left(\mathbf{M}_{(t+1)}\right)_{(i)}$ has $(n_{(t+1)})_{(i)}$ columns, one round of such training process requires $O\left(l_{(i)}r_{(i)}^2(n_{(t+1)})_{(i)}\right)$.

Different from STC and DSTC, directly solving the column space searching problem for matrix completion requires solving the least squares minimizations (9) and (10) iteratively with all measurement data. Solving the two problems involves the complexity $O(((n_{(t)})_{(i)} + (n_{(t+1)})_{(i)})r_{(i)}^2 l_{(i)})$ and $O(l_{(i)}(r_{(i)})^2((n_{(t)})_{(i)} + (n_{(t+1)})_{(i)}))$, respectively. Obviously, $O(((n_{(t)})_{(i)} + (n_{(t+1)})_{(i)})r_{(i)}^2 l_{(i)}) = O(l_{(i)}(r_{(i)})^2((n_{(t)})_{(i)} + (n_{(t+1)})_{(i)}))$. Assuming the total number of iteration rounds is INum, the total computation complexity under the direct solution is $O\left(2INum \times l_{(i)}\left(r_{(i)}\right)^2\left(\left(n_{(t)}\right)_{(i)} + \left(n_{(t+1)}\right)_{(i)}\right)\right)$.

In our STC and DSTC, only the matrix $\left(\mathbf{U}_{(t+1)}\right)_{(i)}$ is trained multiple rounds, where only new measurement data are involved in the calculations in each round. While in the direct solution, both $\left(\mathbf{U}_{(t+1)}\right)_{(i)}$ and $\left(\mathbf{W}_{(t+1)}\right)_{(i)}$ are trained multiple rounds using all the measurement data. Therefore, compared with the direct solution, our STC and DSTC have much smaller computation complexity. Moreover, although DSTC needs one more round of training process for the matrix $\left(\mathbf{U}_{(t+1)}\right)_{(i)}$, compared with STC, our simulation results show that DSTC can achieve better recovery performance.

VII. Performance Evaluations

We evaluate the performance of our proposed algorithm using the public traffic trace data Abilene [46]. The metrics we consider include: Error Ratio, Successful Recovery Ratio, Recovery Loss, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Recovery Computation Time.

As mentioned in Section VI, traffic measurement data generally come in sequence. In the simulation, in each sequential step, we add one more day measurement data. Then we apply the tensor completion to the measurement data to recover the full data. Finally, we calculate the error ratio by comparing the recovered data with the raw data trace. In this paper, one sequence recovery step in the simulations includes the above three operations.

Definition 5 (Error Ratio): a metric for measuring the recovery error of entries in the tensor after the interpolation, which can be calculated as $\frac{\sqrt{\sum_{i,j,k=d}(x_{ijk}-\hat{x}_{ijk})^2}}{\sqrt{\sum_{i,j,k=d}x_{ijk}^2}}$*. where* $1 \le i \le o$*,* $1 \le j \le t$ *and* $k = d$*.* x_{ijk} *and* $\hat{x}_{ijk}$ *denote the*

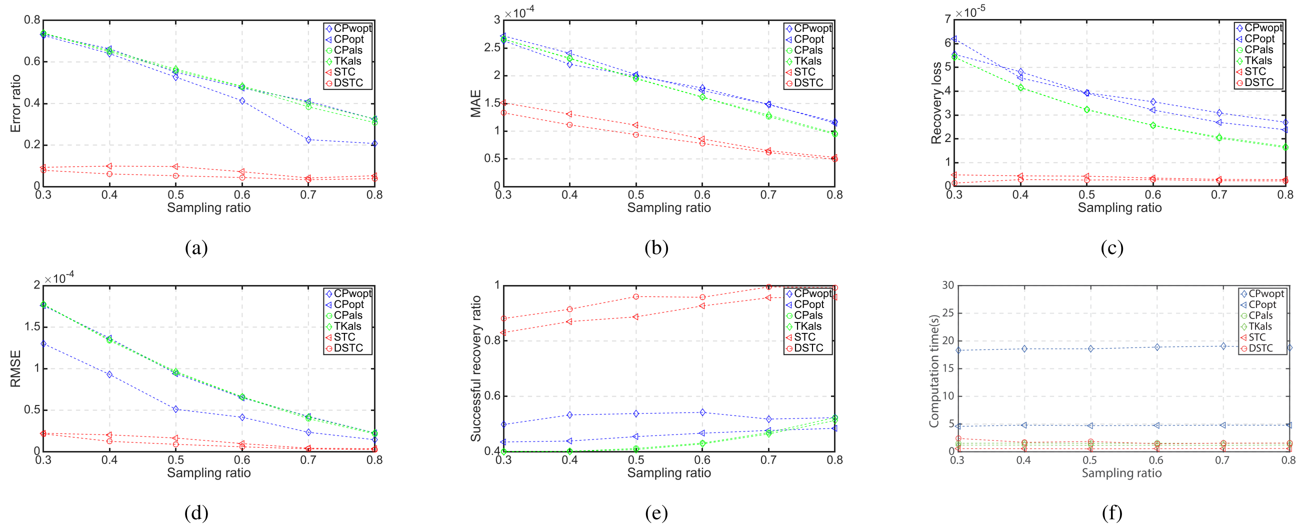


Fig. 7. One sequence step with different sampling ratio. (a) Error ratio. (b) MAE. (c) Recovery loss. (d) RMSE. (e) Successful recovery ratio. (f) Recovery computation time.

raw data and the recovered data at (i,j,k)-th element of $\mathcal{X}$, respectively.

Definition 6 (Successful Recovery Ratio): a metric for measuring the successful recovery of entries in the tensor after the interpolation, which can be calculated as:

$$\frac{\sum_{i,j,k=d} \rho_{ijk}}{o \times t} \quad where\ \rho_{ijk} = \begin{cases} 1 & \left|\frac{\hat{x}_{ijk} - x_{ijk}}{x_{ijk}}\right| \leq \lambda \\ 0 & otherwise. \end{cases} \tag{19}$$

where $1 \leq i \leq o$, $1 \leq j \leq t$ and $k = d$. x_{ijk} and $\hat{x}_{ijk}$ in (19) denote the raw data and the recovered data at (i,j,k)-th element of $\mathcal{X}$, respectively.

In this paper, we set $\lambda = 0.1$. Note that $k = d$ in definition 5 and 6, that is, only the last measurement data in the last day is counted into the performance metric calculation.

Definition 7 (Recovery Loss): is defined as $Loss = \sqrt{\sum_{(i,j,k)\in\Omega,k=d} (x_{i,j,k} - \hat{x}_{i,j,k})^2}$ where $1 \leq i \leq o$, $1 \leq j \leq t$ and $k = d$. Only entry observed (i.e., $(i,j,k) \in \Omega$) is counted in Recovery Loss.

Definition 8 (MAE): is an average of the absolute errors after the interpolation, which can be calculated as $MAE = \frac{1}{T} \sum_{i,j,k} |x_{ijk} - \hat{x}_{ijk}|$, where $1 \leq i \leq o$, $1 \leq j \leq t$ and $k = d$. T is the total number of data entries in the day corresponding to $k = d$, that is, $T = o \times t$.

Definition 9 (RMSE): represents the sample standard deviation of the differences between recovered values and raw values, which can be calculated as $RMSE = \sqrt{\frac{1}{T} \sum_{i,j,k} (x_{ijk} - \hat{x}_{ijk})^2}$ where $1 \leq i \leq o$, $1 \leq j \leq t$, $k = d$, and $T = o \times t$.

Definition 10 (Recovery Computation Time): a metric for measuring the average number of seconds of one sequence recovery step.

All simulations are run on a Microstar workstation, which is equipped with two Intel (R) Xeon (R) E5-2620 CPUs (2.00GHz) (totaliy 24 Cores) and 32.00GB RAM. To measure the recovery computation time, we insert a timer to all the implemented approaches.

A. Comparison With Other Tensor Completion Algorithms

Besides our STC, we implement other four tensor completion algorithms.

2) CP_{opt} [25]: CP_{opt} addresses the problem of fitting the CP model to incomplete data sets by solving a least-squares (ALS) optimization problem ([25, eq. 2]) with a gradient-based optimization approach.
1) CP_{wopt} [24]: different from CP_{opt}, CP_{wopt} (CP Weighted Optimization) addresses the problem of fitting the CP model to incomplete data sets by solving a weighted least squares problem ([24, eq. 2]).
3) CP_{als}: CP_{als} addresses the problem of fitting the CP model to incomplete data sets by solving an alternating least-squares problem. It is implemented using the Tensor Toolbox [54].
4) TK_{als}: TK_{als} addresses the problem of fitting the Tucker model to incomplete data sets by solving an alternating least-squares problem. It is implemented using the Tensor Toolbox [54].

Among the four peer tensor completion algorithms, the first three CP_{wopt}, CP_{opt}, and CP_{als} are designed based on CP model, the last TK_{als} is designed based on the Tucker model. As all the tensor completion approaches are executed iteratively to train the parameters needed, for a fair comparison, we adopt the same two stop conditions: 1) The difference in the recovery loss between two consecutive iterations is smaller than a given threshold value, set to 10^{-4} in this paper; 2) The maximum number of iterations is reached, set to 50 in this paper. The iteration process will continue until either of the two stop conditions is satisfied.

Fig.7 shows the performance results under different sampling ratio for one sequence step. Obviously, Fig.8(a), Fig.8(b), Fig.8(c), Fig.8(d), and Fig.8(e) show that our algorithms STC, DSTC can achieve the highest recovery performance with the least error ratio, MAE, recovery loss, RMSE, and the highest successful recovery ratio. The recovery ratio is very high even with a small sampling rate 0.3, which demonstrates the

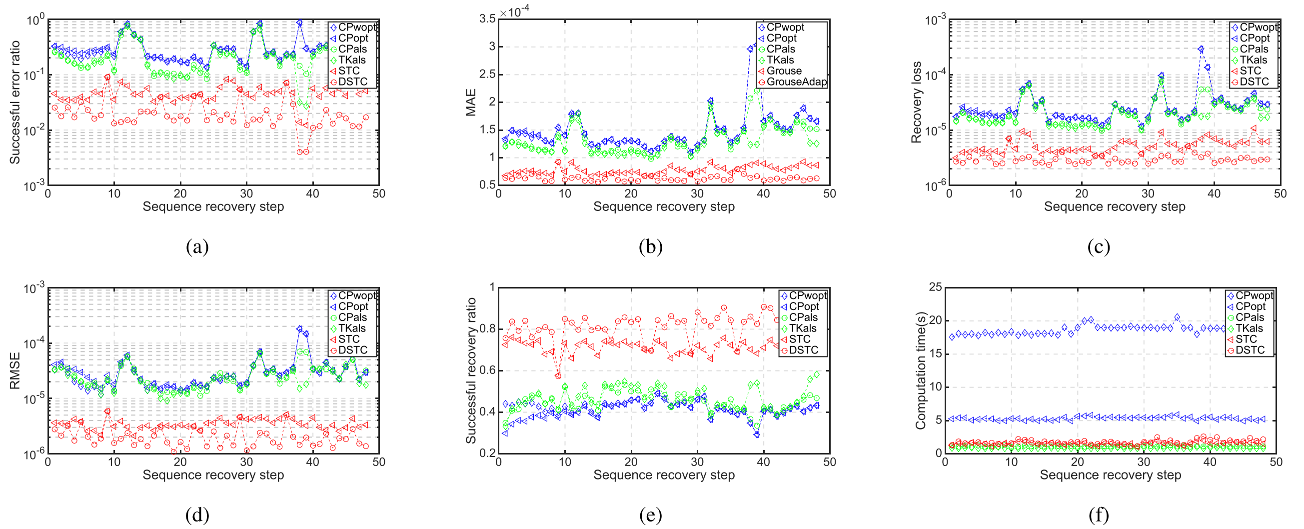


Fig. 8. Multiple sequence steps. (a) Error ratio. (b) MAE. (c) Recovery loss. (d) RMSE. (e) Successful recovery ratio. (f) Recovery computation time.

effectiveness of our algorithm in supporting high accuracy and low cost traffic monitoring. This good performance demonstrates that our STC, DSTC algorithms have the good ability of capturing the global information in the traffic data to recover the missing data with a high accuracy. Moreover, the performance of DSTC is slightly better than that of STC, as it can dynamically change the rank of orthogonal matrix and add new column to the matrix when needed to more accurately capture the feature of the traffic data dynamics to recover the data.

Fig.8(f) compares the computation time under different algorithms. All the tensor completion approaches execute iteratively to train the parameters needed for data recovery until either of the two stop conditions is satisfied. We observe that CP_{als} and TK_{als} usually stop their iterations quickly, much before reaching the iteration limit, thus their speeds are fast and close to STC and DSTC. However, they have high error ratios in data recovery as shown in Fig.8(a). CP_{wopt} adopts an approach similar to CP_{opt} in training the parameters but involves additional weight calculation, thus its computation time is larger than CP_{opt}. Compared with STC, as DSTC requires one more round to train $\left(\mathbf{U}_{(t+1)}\right)_{(i)}$ by testing whether new measurement data will change the rank of $\left(\mathbf{U}_{(t+1)}\right)_{(i)}$, its time consumption for missing data recovery is slightly larger. This is consistent with the computation complexity analyzed in Section VI-E.

Fig. 8 shows the recovery performance under multiple sequential steps by fixing the sampling ratio to 50%. The results are consistent with those in Fig.7. Among all the algorithms implemented, our DSTC achieves the best recovery performance with the least error ratio, MAE, recovery loss, RMSE, and the highest successful recovery ratio. Although different sequential steps involve different newly coming traffic data, DSTC achieves more accurate data recovery than the static algorithm STC in all the steps as DSTC has the ability to change the rank upon needed according the dynamic data changes. Besides having the same training phase as STC, DSTC will add new column to the orthogonal matrix when needed, thus the speed of STC is slightly faster than DSTC.

B. Comparison With Matrix Completion Algorithms

Among all the current traffic inferring studies, the matrix-completion-based recovery algorithm is proven to achieve the best performance. In this part, we further implement other five matrix completion algorithms for the performance comparison.

1) NMF [55]: NMF performs non-negative matrix factorization, where the non-negative matrix factorization is a recently developed technique for finding part-based, linear representations of non-negative data. Given a non-negative matrix $\mathbf{V}$, the goal of NMF is to find the non-negative matrix factors $\mathbf{W}$ and $\mathbf{H}$ such that $\mathbf{V} = \mathbf{W}\mathbf{H}^{tr}$.
2) $SRMF$ [12]: $SRMF$ is a matrix interpolation technique which uses an alternating least squares procedure to find the global sparse, low-rank approximation of the traffic matrix that accounts for the spatial and temporal properties.
3) $SRSVD$ [12]: $SRSVD$ is a matrix interpolation technique which uses an alternating least squares procedure to find the sparse, low-rank approximation of the traffic matrix.
4) SET [56]: SET is proposed for solving the consistent matrix completion problem. The SET algorithm consists of two parts, subspace evolution and subspace transferring.
5) $LMaFit$ [57]: LMaFit is based on a nonlinear successive over-relaxation (SOR) method that only requires solving a linear least squares problem per iteration. Following the idea of the nonlinear SOR technique, LMaFit uses a weighted average between the current updated data and data from the previous iteration to achieve a faster convergence.

All the seven matrix completion algorithms are applied to the traffic matrix which is defined in $SRMF$ [12].

Fig.9(a), Fig.9(b), Fig.9(c), Fig.9(d), Fig.9(e) shows the error ratio, MAE, recovery loss, RMSE, and successful recovery ratio under sample $ratio = 50\%$ for a sequence of recovery step. Obviously, our DSTC achieves the best recovery performance among all the algorithms studied.

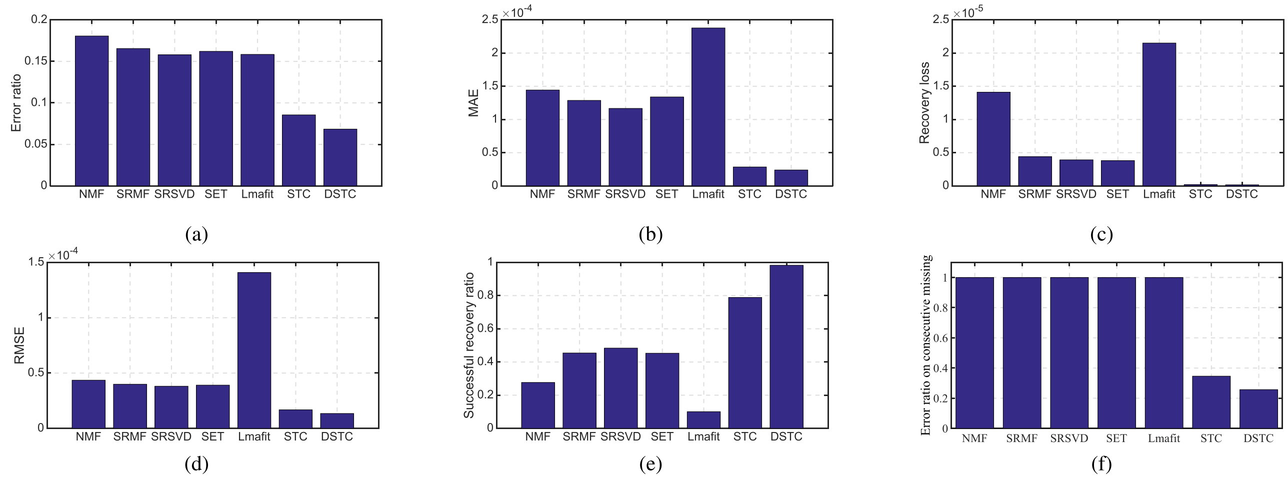


Fig. 9. Performance comparison with matrix completion algorithms. (a) Error ratio (sampling ratio 50%). (b) MAE (sampling ratio 50%). (c) Recovery loss (sampling ratio 50%). (d) RMSE (sampling ratio 50%). (e) Successful recovery ratio (sampling ratio 50%). (f) Error ratio on consecutive data missing.

Moreover, among the one day measurement data in the sequential step, we let consecutive measurements over 50 minutes all lost, and then calculate the error ratio on the 50 minutes data, as shown in Fig.9(f). The consecutive data missing, obviously, results in the consecutive column missing in the traffic matrix. From the literature work, we know that the conventional matrix completion algorithms can only recover data if there is no row or column to be completely empty. If a row or a column is missing, matrix completion algorithms do not have effect on these missing entries. Because we use zero as a placeholder to replace the empty entry, the error ratio on this kind of consecutive missing is 1 under all the matrix completion algorithms, while the error ratios on the consecutive missing data are only 0.37 and 0.33 under STC and DSTC, respectively. STC and DSTC exploit the information along three dimensions, while the matrix completion only considers the constraints along two particular dimensions. This is the key reason why STC and DSTC outperform the matrix completion-based algorithms.

VIII. Conclusion

In this paper, we apply the emerging concept of tensor completion to the recovery of the missing Internet traffic data. To well capture the spatial-temporal features inherent in the traffic data, we first analyze a large trace of real traffic data, and our studies reveal that the traffic data have the features of the temporal stability, the spatial correlation, and the periodicity. To fully exploit theses hidden structures for the data recovery, we model the traffic data as a traffic tensor which can combine and utilize the multi-mode correlations. To reduce the computation cost for tensor completion, we propose two novel Sequential Tensor Completion algorithms STC and DSTC to quickly recover the missing traffic data. We have done extensive simulations to evaluate the performance of our proposed algorithms, and the simulation results demonstrate that our algorithms can achieve significantly better performance compared with current of state tensor and matrix completion algorithms.

Although we apply the tensor to capture the traffic volume in this paper, our tensor modeling and the proposed sequential tensor completion approach are useful for the representation of other factors of the network, for instance, delay, jitter, loss, bottleneck-bandwidth, and distance (RTT). In our future work, we will evaluate the performance of our proposed algorithms on other network monitoring data. Moreover, as data missing is more severe in mobile and wireless sensor networks due to interference, user movement, and link unreliability [58] thus network dis-connectivity [59]–[62], in our future work, we will extend our proposed algorithms to infer the missing data when the information is gathered by mobile and wireless sensor networks [63]–[66] and other systems of IOT (Internet of things) [67]–[70].

References

[1] K. Xie *et al.*, "Accurate recovery of Internet traffic data: A tensor completion approach," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.

[2] M. Roughan, M. Thorup, and Y. Zhang, "Traffic engineering with estimated traffic matrices," in *Proc. ACM IMC*, 2003, pp. 248–258.

[3] L. Fu, Y. Qin, X. Wang, and X. Liu, "Throughput and delay analysis for convergecast with mimo in wireless networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 4, pp. 768–775, Apr. 2012.

[4] K. Xie, X. Wang, J. Wen, and J. Cao, "Cooperative routing with relay assignment in multiradio multihop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 859–872, Apr. 2016.

[5] K. Xie, X. Wang, X. Liu, J. Wen, and J. Cao, "Interference-aware cooperative communication in multi-radio multi-channel wireless networks," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1528–1542, May 2016.

[6] X. Deng, Q. Peng, L. He, and T. He, "Interference-aware QoS routing for neighbourhood area network in smart grid," *IET Commun.*, vol. 11, no. 5, pp. 756–764, Dec. 2016.

[7] A. Lakhina *et al.*, "Structural analysis of network traffic flows," *ACM SIGMETRICS*, vol. 32, no. 1, pp. 61–72, 2004.

[8] Y. Zhang, M. Roughan, C. Lund, and D. L. Donoho, "Estimating point-to-point and point-to-multipoint traffic matrices: An information-theoretic approach," *IEEE/ACM Trans. Netw.*, vol. 13, no. 5, pp. 947–960, Oct. 2005.

[9] 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.

[10] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," *J. Amer. Stat. Assoc.*, vol. 91, no. 433, pp. 365–377, 1996.

[11] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proc. ACM IMW*, 2002, pp. 71–82.

[12] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and Internet traffic matrices (extended version)," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 662–676, Jun. 2012.

[13] M. Mardani and G. Giannakis, "Robust network traffic estimation via sparsity and low rank," in *Proc. IEEE ICASSP*, May 2013, pp. 4529–4533.

[14] R. Du, C. Chen, B. Yang, and X. Guan, "Vanet based traffic estimation: A matrix completion approach," in *Proc. IEEE GLOBECOM*, Dec. 2013, pp. 30–35.

[15] G. Gürsun and M. Crovella, "On traffic matrix completion in the Internet," in *Proc. ACM IMC*, 2012, pp. 399–412.

[16] Y.-C. Chen, L. Qiu, Y. Zhang, G. Xue, and Z. Hu, "Robust network compressive sensing," in *Proc. ACM MobiCom*, 2014, pp. 545–556.

[17] K. Xie *et al.*, "Sequential and adaptive sampling for matrix completion in network monitoring systems," in *Proc. IEEE INFOCOM*, Apr. 2015, pp. 2443–2451.

[18] B. Liu, D. Niu, Z. Li, and H. V. Zhao, "Network latency prediction for personal devices: Distance-feature decomposition from 3D sampling," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 307–315.

[19] R. Zhu, B. Liu, D. Niu, Z. Li, and H. V. Zhao, "Network latency estimation for personal devices: A matrix completion approach," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 724–737, Apr. 2017.

[20] H. Zhou, D. Zhang, and K. Xie, "Accurate traffic matrix completion based on multi-Gaussian models," *Comput. Commun.*, vol. 102, pp. 165–176, Apr. 2017.

[21] K. Xie *et al.*, "Fast tensor factorization for accurate Internet anomaly detection," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3794–3807, Dec. 2017.

[22] K. Xie *et al.*, "Graph based tensor recovery for accurate Internet anomaly detection," in *Proc. 37th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, to be published.

[23] 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.

[24] 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.

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

[26] 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.

[27] M. Filipović and A. Jukić, "Tucker factorization with missing data with application to low-n-rank tensor completion," *Multidimensional Syst. Signal Process.*, vol. 26, no. 3, pp. 677–692, 2015.

[28] H. Li, K. Li, A. Jiyao, and K. Li, "MSGD: A novel matrix factorization approach for large-scale collaborative filtering recommender systems on GPUs," *IEEE Trans. Parallel Distrib. Syst.*, to be published, doi: 10.1109/TPDS.2017.2718515.

[29] K. Xie, L. Wang, X. Wang, G. Xie, and J. Wen, "Low cost and high accuracy data gathering in WSNs with matrix completion," *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2017.2775230.

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

[31] A. Cichocki *et al.* (Mar. 2014). "Tensor decompositions for signal processing applications from two-way to multiway component analysis." [Online]. Available: https://arxiv.org/abs/1403.4462

[32] B. D. Haeffele and R. Vidal. (Jun. 2015). "Global optimality in tensor factorization, deep learning, and beyond." [Online]. Available: https://arxiv.org/abs/1506.07540

[33] H. Tan *et al.*, "A tensor-based method for missing traffic data completion," *Transp. Res. C, Emerg. Technol.*, vol. 28, pp. 15–27, Mar. 2013.

[34] Q. Zhao, C. Chen, R. Du, S. Bi, and B. Yang, "HaTTC: An urban traffic sensing method based on tensor completion technique," in *Proc. IEEE GLOBECOM*, Dec. 2014, pp. 175–180.

[35] 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.

[36] C. Da Silva and F. J. Herrmann, "Optimization on the hierarchical Tucker manifold–applications to tensor completion," *Linear Algebra Appl.*, vol. 481, pp. 131–173, Sep. 2015.

[37] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of 'Eckart-Young' decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[38] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an 'explanatory' multi-modal factor analysis," *UCLA Work. Papers Phonetics*, vol. 16, no. 1, pp. 1–84, 1970.

[39] L. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

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

[41] X.-Y. Liu, S. Aeron, V. Aggarwal, and X. Wang. (Oct. 2016). "Low-tubal-rank tensor completion using alternating minimization." [Online]. Available: https://arxiv.org/abs/1610.01690

[42] X.-Y. Liu, S. Aeron, V. Aggarwal, X. Wang, and M.-Y. Wu, "Adaptive sampling of RF fingerprints for fine-grained indoor localization," *IEEE Trans. Mobile Comput.*, vol. 15, no. 10, pp. 2411–2423, Oct. 2016.

[43] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, Mar. 2015.

[44] E. Kernfeld, M. Kilmer, and S. Aeron, "Tensor–tensor products with invertible linear transforms," *Linear Algebra Appl.*, vol. 485, pp. 545–570, Nov. 2015.

[45] J. Håstad, "Tensor rank is NP-complete," *J. Algorithms*, vol. 11, no. 4, pp. 644–654, 1990.

[46] *The Abilene Observatory Data Collections*. Accessed: May 2004. [Online]. Available: http://abilene.internet2.edu/observatory/data-collections.html

[47] C. Mu, B. Huang, S. J. Wright, and D. Goldfarb, "Square deal: Lower bounds and improved relaxations for tensor recovery," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 73–81.

[48] I. Markovsky, *Low Rank Approximation: Algorithms, Implementation, Applications*. London, U.K.: Springer, 2011.

[49] D. Gross, "Recovering low-rank matrices from few coefficients in any basis," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1548–1566, Mar. 2011.

[50] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2053–2080, May 2010.

[51] M. Brand, "Incremental singular value decomposition of uncertain data with missing values," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2002, pp. 707–720.

[52] L. Balzano and S. J. Wright, "On grouse and incremental SVD," in *Proc. IEEE CAMSAP*, Dec. 2013, pp. 1–4.

[53] K. Xie, L. Wang, X. Wang, J. Wen, and G. Xie, "Learning from the past: Intelligent on-line weather monitoring based on matrix completion," in *Proc. IEEE 34th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2014, pp. 176–185.

[54] B. W. Bader *et al.* (Jan. 2012). *MATLAB Tensor Toolbox Version 2.5*. [Online]. Available: http://www.sandia.gov/~tgkolda/TensorToolbox/

[55] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.

[56] W. Dai, E. Milenkovic, and E. Kerman, "Subspace evolution and transfer (SET) for low-rank matrix completion," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3120–3132, Jul. 2011.

[57] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Math. Program. Comput.*, vol. 4, no. 4, pp. 333–361, 2012.

[58] K. Xie *et al.*, "Decentralized context sharing in vehicular delay tolerant networks with compressive sensing," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2016, pp. 169–178.

[59] L. Fu *et al.*, "Determining source–destination connectivity in uncertain networks: Modeling and solutions," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3237–3252, Dec. 2017.

[60] L. Fu, X. Wang, and P. Kumar, "Are we connected? Optimal determination of source–destination connectivity in random networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 751–764, Apr. 2017.

[61] X. Fu *et al.*, "ConMap: A novel framework for optimizing multicast energy in delay-constrained mobile wireless networks," in *Proc. 18th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2017, Art. no. 2.

[62] J. An, X. Liu, and G. Wen, "Stability analysis of delayed Takagi-Sugeno fuzzy systems: A new integral inequality approach," *J. Nonlinear Sci. Appl.*, vol. 10, no. 4, pp. 1941–1959, 2017.

[63] K. Xie, J. Cao, X. Wang, and J. Wen, "Optimal resource allocation for reliable and energy efficient cooperative communications," *IEEE Trans. Wireless Commun.*, vol. 12, no. 10, pp. 4994–5007, Oct. 2013.

[64] 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.

[65] K. Xie *et al.*, "An efficient privacy-preserving compressive data gathering scheme in WSNs," *Inf. Sci.*, vol. 390, pp. 82–94, Jun. 2017.

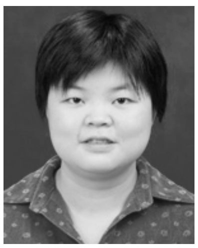
[66] X. Deng, T. He, L. He, J. Gui, and Q. Peng, "Performance analysis for IEEE 802.11 s wireless mesh network in smart grid," *Wireless Pers. Commun.*, vol. 96, no. 1, pp. 1537–1555, 2017.

[67] X. Liu, K. Li, G. Min, Y. Shen, A. X. Liu, and W. Qu, "Completely pinpointing the missing RFID tags in a time-efficient way," *IEEE Trans. Comput.*, vol. 64, no. 1, pp. 87–96, Jan. 2015.

[68] X. Liu *et al.*, "RFID estimation with blocker tags," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 224–237, Feb. 2017.

[69] X. Liu, S. Zhao, A. Liu, N. Xiong, and A. V. Vasilakos, "Knowledge-aware proactive nodes selection approach for energy management in Internet of Things," *Future Generat. Comput. Syst.*, to be published, doi: 10.1016/j.future.2017.07.022.

[70] Y. Liu *et al.*, "Context-aware collect data with energy efficient in cyber–physical cloud systems," *Future Generat. Comput. Syst.*, to be published, doi: 10.1016/j.future.2017.05.029.



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, State University of New York at Stony Brook, from 2012 to 2013. Her research interests include network management and control, cloud computing and mobile cloud, and big data.

Lele Wang is currently pursuing the Ph.D. degree with Hunan University, Changsha, China.

Xin Wang (M'01) received the B.S. and M.S. degrees in telecommunications engineering and wireless communications engineering, respectively, from Beijing University of Posts and Telecommunications, Beijing, China, and 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, State University of New York at Stony Brook, Stony Brook, NY, USA. Her research interests include algorithm and protocol design in wireless networks and communications, mobile and distributed computing, and networked sensing and detection.

Gaogang Xie received the B.S. degree in physics, the M.S. degree, and the Ph.D. degree in computer science from Hunan University in 1996, 1999, and 2002, respectively. He is currently a Professor and the Director of the Network Technology Research Center with the 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, and high-speed network measurement and management.

Guangxing Zhang received the B.S. degree and the Ph.D. degree in computer science from Hunan University in 2002 and 2011, respectively. He is currently an Associate Professor with the Network Technology Research Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His research interests include wireless network and mobile computing, Internet architecture, packet processing and forwarding, and Internet measurement.

Jiannong Cao (M'93–SM'05–F'14) is currently a Chair Professor and the head of the Department of Computing with The Hong Kong Polytechnic University, Hong Kong. He is also the Director of the Internet and Mobile Computing Laboratory with the Department of Computing. His research interests include parallel and distributed computing, computer networks, mobile and pervasive computing, fault tolerance, and middleware. He received numerous prizes and awards, and is very active in professional activities locally and internationally. He is a senior member of China Computer Federation and a member of ACM.

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.