Quick and Accurate False Data Detection in Mobile Crowd Sensing

Xiaocan Li, Kun Xie, Xin Wang, *Member, IEEE, ACM*, Gaogang Xie, Dongliang Xie, Zhenyu Li, *Member, IEEE*, Jigang Wen, Zulong Diao, and Tian Wang

***Abstract*— The attacks, faults, and severe communication/system conditions in Mobile Crowd Sensing (MCS) make false data detection a critical problem. Observing the intrinsic low dimensionality of general monitoring data and the sparsity of false data, false data detection can be performed based on the separation of normal data and anomalies. Although the existing separation algorithm based on Direct Robust Matrix Factorization (DRMF) is proven to be effective, requiring iteratively performing Singular Value Decomposition (SVD) for low-rank matrix approximation would result in a prohibitively high accumulated computation cost when the data matrix is large. In this work, we observe the quick false data location feature from our empirical study of DRMF, based on which we propose an intelligent Light weight Low Rank and False Matrix Separation algorithm (LightLRFMS) that can reuse the previous result of the matrix decomposition to deduce the one for the current iteration step. Depending on the type of data corruption, random or successive/mass, we design two versions of LightLRFMS. From a theoretical perspective, we validate that LightLRFMS only requires one round of SVD computation and thus has very low computation cost. We have done extensive experiments using a PM 2.5 air condition trace and a road traffic trace. Our results demonstrate that LightLRFMS can achieve very good false data detection performance with the same highest detection accuracy as DRMF but with up to 20 times faster speed thanks to its lower computation cost.**

***Index Terms*— Matrix separation, false data detection, mobile crowd sensing.**

Manuscript received August 25, 2019; revised February 19, 2020; accepted March 9, 2020; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor P. P. C. Lee. Date of publication April 16, 2020; date of current version June 18, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61972144, Grant 61572184, Grant 61725206, and Grant 61976087, in part by the Hunan Provincial Natural Science Foundation of China under Grant 2017JJ1010, in part by the U.S. NSF under Grant ECCS 78929 and Grant CNS 1526843, in part by the Open Project Funding of State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, under Grant CARCH201809, in part by the CERNET Innovation Project under Grant NGII20190118, in part by the Open Foundation of State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) under Grant SKLNST-2018-1-20, and in part by the Peng Cheng Laboratory Project of Guangdong Province under Grant PCL2018KP004. *(Corresponding author: Kun Xie.)*

Xiaocan Li is with the College of Computer Science and Electronics Engineering, Hunan University, Changsha 410082, China.

Kun Xie is with the College of Computer Science and Electronics Engineering, Hunan University, Changsha 410082, China, also with the Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen 518000, China, and also with the Purple Mountain Laboratory, Nanjing 211111, China (e-mail: xiekun@hnu.edu.cn).

Xin Wang is with the Department of Electrical and Computer Engineering, The State University of New York at Stony Brook, Stony Brook, NY 11794 USA.

Gaogang Xie is with the Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100190, China.

Dongliang Xie is with the Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China.

Zhenyu Li is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with Taobao.com, Beijing 100102, China.

Jigang Wen is with the Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China.

Zulong Diao is with the Purple Mountain Laboratory, Nanjing 211111, China, and also with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China.

Tian Wang is with the College of Computer Science and Technology, Huaqiao University, Quanzhou 362021, China.

Digital Object Identifier 10.1109/TNET.2020.2982685

I. INTRODUCTION

THE proliferation of smartphones contributes to the prosperity of a novel sensing paradigm called Mobile Crowd Sensing (MCS) [1]–[7]. Different from traditional wireless sensor networks (WSNs) which usually leverage dedicated sensors to acquire real-world conditions, MCS utilizes off-the-shelf smartphones carried by citizens to capture dynamic changes of social and urban information. The participatory and mobile nature of MCS provides a novel way for the monitoring of environment, transportation, and city infrastructure [8]–[11]. Although promising, there is a big challenge to ensure the data quality in MCS, as it is susceptible to different kinds of attacks, faults, severe communication and system conditions:

- *Attack during transmission*. With the transmissions over wireless networks, MCS data are subject to attacks such as eavesdropping, information tampering and malicious programs.
- *Attack by intelligent participation*. MCS involves human participation. Intelligent attackers can introduce bad data into certain state variables, exploiting the knowledge of the MCS application configurations to bypass existing techniques for detecting bad measurements. For example, to earn rewards, participants may submit fake data without performing the actual sensing task [12], or compromise the mobile devices to provide faulty sensor readings [13].
- *Failure of algorithms or sensors.* Mobile devices that do not work in normal states may also provide false measurements.
- *Severe communication/system conditions.* False measurements may be resulted from network congestion, node misbehavior, monitor failure, and unreliable wireless transmission channels.

The attacks, faults, and severe communication/system conditions in MCS cause a serious false data problem which affects the proper operation of the MCS systems. As many MCS applications are vulnerable to the false data, detecting false data as fast and precise as possible has a significant impact on the reliable operations in MCS.

Facilitated by the popularity of smart-phones and other GPS-enabled mobile devices (e.g., in-vehicle sensing devices), most of MCS applications are location-aware. Sensory data are usually reported along with the monitoring locations. The data collected from MCS can generally be represented by an $N \times T$ matrix, which records data from N locations over T time slots. As almost all physical conditions monitored are continuous without sudden changes, sensory data from environment monitoring generally exhibit strong spatio-temporal correlations [14] thus the data matrix has a low-rank feature. Since it is very costly for an attacker to compromise a large number of measurements and transmission units for a long period of time, and moreover, faults in sensors, communications, and systems usually rarely happen, false data over time form a sparse matrix [15]–[17]. The data matrix observed is usually the sum of a low-rank real measurement matrix and a sparse malicious data matrix.

Based on the observation, one effective way to detect false data is through the matrix separation. That is, we can recover and separate real measurement data in MCS from the corrupted ones by exploiting the low-rank feature of the real measurement matrix and the sparseness feature of the false data matrix. After obtaining the sparse false data matrix, we can further locate the attacks or faults in the MCS systems.

Principal Component Analysis (PCA) [18] and Robust PCA (RPCA) [19] are two typical matrix separation techniques. However, they can not achieve good performance in false data detection, as they either fail under the large corruption or resort to some relaxation techniques to solve the problem. Recent work in [20] proposes a Direct Robust Matrix Factorization (DRMF) to minimize the error of the low-rank matrix approximation subject to that the number of outliers is small without using the relaxation techniques. DRMF is proven to be very effective in video activity detection and USPS anomaly detection [20]. Although promising, to accurately separate the low-rank matrix and the false data components, DRMF requires iteratively solving a low-rank matrix approximation sub-problem and an anomaly detection sub-problem. The execution of low-rank approximation further requires performing SVD (Singular Value Decomposition), that has $O\left(\min\left\{mn^2, nm^2\right\}\right)$ time complexity to handle a matrix of $\mathbb{R}^{m\times n}$ in each iteration step, which results in a prohibitively high accumulated computation cost and makes it impractical for dealing with large data set in MCS.

For the security and robustness of MCS, it is critical to separate/detect the false data and recover the original measurement data. In light of the importance of DRMF and the above challenge in high computation complexity, in this work, we propose a computationally efficient algorithm to enable fast and accurate false data detection based on DRMF. Our contributions are summarized as follows:

- We find the feature of ***quick false data location***. DRMF operates iteratively to separate the false data from the observed noisy data. From the empirical study of applying DRMF to trace data sets for false data detection, we find that candidate false data locations can be detected in the first few iteration steps, and after which the locations will not change even though the false data values will change until the algorithm converges.
- We propose an intelligent Light weight Low Rank and False Matrix Separation algorithm (LightLRFMS), which takes advantage of the feature of ***quick false data location*** to largely speed up the whole iteration process by reusing the previous result of the matrix decomposition to deduce the one for the current iteration step. As the false data may be caused by random corruptions or success/mass corruptions in MCS, we propose two versions of LightLRFMS to detect the random false data and structured false data respectively.
- We validate from a theoretical perspective that LightLRFMS only requires one round of SVD computation and thus has very low computation cost.
- We compare LightLRFMS with the state of art false data separation algorithms using two real data sets. Our experiment results demonstrate that LightLRFMS can achieve the same highest false data detection accuracy as DRMF at significantly faster speed (up to 20 times faster than DRMF) thanks to its lower computational cost.

The rest of the paper is organized as follows. Section II presents the related work. We present the problem formulation and empirical studies with two real traces in Section III and Section IV, respectively. We present our LightLRFMS algorithm and its extension to detect structured false data in Section V and Section VI, respectively. Finally, we implement the proposed LightLRFMS and evaluate the performance in Section VII, and conclude the work in Section VIII.

II. Prior Art and Limitations

In this paper, we would like to solve the false data detection problem through low-rank and sparse matrix separation and propose a general solution for false data detection.

Current research interests in MCS mainly focus on issues such as sensing task allocation, sparse sensing, privacy, and data integrity [1]–[7] investigate the false data detection problem in MCS. The accuracy of the detection of false data highly depends on the accuracy of the recovery of low rank normal data. However, the work in [21] solves the low-rank normal data recovery problem and false data detection problem individually. As a result, [21] suffers from false data with large deviation. With the need of context information of the particular MCS application, [22] is not a general false data detection algorithm.

To detect false data, Principal Component Analysis (PCA) [18] is perhaps the best-known statistical analysis technique. The principal components (PCs) in the lower dimensional subspace capture the dynamical properties of the system and the PCs in the higher dimensional subspace represent noisy information. Using this property, it is hypothesized that outliers due to bad data will result in larger activity in

the high dimensional subspace. Although effective when the corruption is caused by small additive noise, recent studies show that traditional PCA-based approaches fail under the large corruption, even if the corruption affects only very few of the observations [23].

Although not targetting for MCS, to decompose a given observation (noisy) matrix X into a low-rank component X' and a sparse outlier component E, Candès et. al. [19] propose Robust PCA (RPCA). To make the problem solvable, the work in [24] replaces the matrix rank and the cardinality ($\|\|_0$) functions with their convex surrogates, the nuclear norm $\|\|_*$ (i.e., the sum of its singular values) and the L_1 norm $\|\|_1$, and solves the following convex optimization problem

$$\min_{X',E} \{\|X'\|_* + \lambda\|E\|_1\}$$
$$st.\ X' + E = X \tag{1}$$

where λ is a positive weighting parameter. To decompose the data into low-rank component and sparse component, these methods resort to some relaxation techniques which may largely impact the accuracy of false data detection.

To conquer the challenge in RPCA, work in [20] proposes DRMF which directly formulates the problem in its original way using the matrix rank to represent the low rank feature and the L_0-norm to represent the sparse feature of the false data. However, the solution involves the iterative execution of the SVD decomposition, which will bring very high computation cost and is not scalable to large monitoring data.

To the best of knowledge, we are the first that proposes a highly accurate and quick false data detection algorithm based on DRMF. Specially, from experiment studies, we find an interesting feature (**quick false data location**) hidden in the DRMF. That is, DRMF can quickly identify the candidate false data location in the first a few iteration steps although DRMF doesn't converge as the data values still change over iteration steps.

III. Problem and Challenge

In this section, we first show that the detection and identification of false data in MCS can be formulated as a low rank and sparse matrix separation problem. We then give an overview of our solution, and analyze the computation complexity of the problem.

A. Problem Formulation

The monitoring data captured by MCS at a time slot k is denoted as $l_k \in \mathbb{R}^N$ where N is the number of measurement locations. In the presence of attacks and faults, the measurement is contaminated by the false data vector $s_k \in \mathbb{R}^N$. Let $L = [l_1, l_2, \cdots, l_T] \in \mathbb{R}^{N\times T}$ be the matrix of monitoring data taken from N locations for a time period of T, and $S = [s_1, s_2, \cdots, s_T] \in \mathbb{R}^{N\times T}$ be the false data matrix. The obtained observation can be expressed as:

$$X = L + S \tag{2}$$

In the monitoring data matrix, the row and the column correspond respectively to a measurement location and a time slot. In a typical MCS platform, mobile users that are willing to provide sensing data can register as participants. The inherent mobility of mobile users provides unprecedented spatiotemporal coverage compared to static sensor networks. However, some locations may not have measurements due to the mobility and change of participants, so the corresponding entries in the matrix are empty. These entries can be inferred through matrix completion, which takes advantage of spatial and temple correlations to fill the missing measurement data [25]–[30].

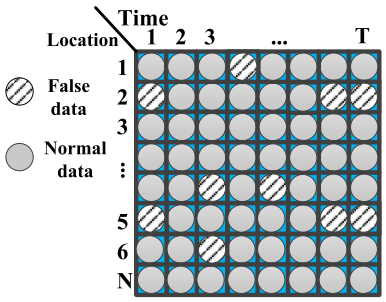


Fig. 1. Random distributed data corrupted monitoring data.

As discussed in introduction, data for environment monitoring generally have low-rank feature due to their strong spatial and temporal correlations, and false data are sparse because attacks, faults, and failures in MCS are rare. Therefore, the detection and identification of false data problem can be converted to a matrix separation problem as:

$$\min_{L,S} \|(X - S) - L\|_F^2$$
$$\text{s. t. } \operatorname{rank}(L) \le k$$
$$\|S\|_0 \le e \tag{3}$$

where S is the false data matrix, L is the low-rank approximation of matrix $X - S$, k is the truncation rank, and e is the maximal number of non-zeros entries in S that can not be ignored as outliers. In (3), we use $rank(L)$ to stand for the rank of the matrix L, and $\|S\|_0$ represents the number of nonzero entries of matrix S.

The outlier number constraint e does not need to match the actual number of outliers, but is only used to prevent that too many data items from being classified as outliers. Therefore, we do not need the actual number of outliers, but only use e to provide an upper limit. The formulation in (3) aims at minimizing the error of the low-rank approximation subject to that the number of outliers is small, without any further assumptions.

B. Solution Overview and Challenge

Usually, optimization problems (i.e., problem in (3)) involving the matrix rank or the L_0-norm i.e. set cardinality are difficult to solve. Some relaxation techniques are proposed to solve the low-rank matrix approximation, such as using the nuclear norm of matrix to replace the rank and L_1-norm to replace L_0-norm. However, these relaxations may largely impact the estimation accuracy of the low-rank matrix approximation and further impact the accuracy of false data detection.

Rather than resorting to relaxation techniques, following DRMF in [20], we adopt the block coordinate descent strategy to directly solve the problem (3). The block coordinate descent strategy solves the problem (3) iteratively, and in each iteration step the following two sub-problems need to be solved:

- Low-rank matrix approximation sub-problem

$$\begin{aligned} L = \arg\min_{L} \|C - L\|_F^2 \\ \text{s.t. } C = X - S \\ rank(L) \leq k \end{aligned} \tag{4}$$

- False data detection sub-problem

$$\begin{aligned} S = \arg\min_{S} \|E - S\|_F^2 \\ \text{s.t. } E = X - L \\ \|S\|_0 \leq e \end{aligned} \tag{5}$$

In each iteration, we first fix the current estimate of the set of outliers S and exclude them from the measurement X to get the "clean" data matrix C, and then fit L based on C. Next, we update the false data matrix S based on $E = X - L$.

Specially, a theorem proven by Eckart and Young [31] shows that the error in approximating a matrix A by A_k can be written as: $\|A - A_k\|_F^2 \leq \|A - B\|_F^2$ where B is any matrix with the rank k, A_k is the rank-k truncated SVD of the matrix A. According to Eckart and Young's theorem, the solution to L in problem (4) is the truncated SVD approximation to the "cleaned" matrix C.

Moreover, following the theorem in the work of [32] to solve the general problem of L_0-norm constrained minimization of the decomposable objective, the false data detection problem in (5) can also be solved as

$$s_{i,j} = \begin{cases} e_{i,j} & \beta_{i,j} > \beta(e) \\ 0 & otherwise \end{cases} \tag{6}$$

where $\beta_{i,j} = (e_{i,j})^2$, $\beta(e)$ is the e-th largest value in $\beta_{i,j}$ (e is outlier number constraint in (5)), and $e_{i,j}$ is the (i,j)-th entry in E where $E = X - L$.

C. Computation Complexity and Challenge

The block coordinate descent strategy requires a truncated SVD approximation to solve the problem in (4) in each iterative step. Such an operation, however, introduces a high computational cost and is not scalable to deal with large monitoring data.

Given a matrix $X \in \mathbb{R}^{m \times n}$, SVD decomposes the matrix into three factors:

$$X = U\Sigma \mathrm{V}^T, \tag{7}$$

where U is an $m \times m$ orthogonal matrix. Σ is a diagonal $m \times n$ matrix with non-negative real numbers on the diagonal. V is an $n \times n$ orthogonal matrix, and V^T is the conjugate transpose of V. The diagonal entries σ_i of Σ are known as the singular values of X. Generally, the rank of a matrix X, denoted by r, is equal to the number of its non-zero singular values. We call this rank definition as "precise rank". Based on the matrix rank, an alternate version of the SVD (in (7)) can be expressed as

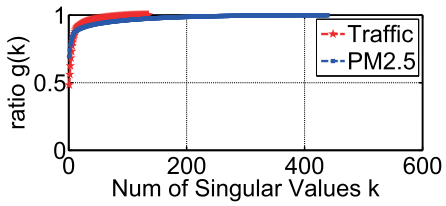


Fig. 2. Fraction captured by top k singular values.

$$\begin{aligned} X &= U\Sigma \mathrm{V}^T \\ &= [u_1, u_2, \ldots, u_r] \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \ldots & 0 \\ 0 & 0 & \sigma_r \end{bmatrix} [v_1, v_2, \ldots, v_r]^T \\ &= u_1 \sigma_1 {v_1}^T + u_2 \sigma_2 {v_2}^T + \cdots + u_r \sigma_r {v_r}^T \end{aligned} \tag{8}$$

where $U = [u_1, \cdots, u_r] \in \mathbb{R}^{m \times r}$ such that $U^T U = I$, Σ is $r \times r$ diagonal matrix with positive entries in the decreasing order on the diagonal, and $V = [v_1, \cdots, v_r] \in \mathbb{R}^{n \times r}$ such that $V^T V = I$. Given the truncation rank k, by setting all but the first k largest singular values equal to zero and using only the first k columns of U and V, we can obtain the truncated matrix X_k.

Although promising, the exact SVD has $O(\min\{mn^2, nm^2\})$ time complexity. This is highly unscalable, rendering the straightforward way of obtaining the truncated matrix through the exact SVD impractical for a large data matrix. Additionally, as the low-rank matrix approximation is executed iteratively in DRMF framework, the overall computation cost will be very high. Therefore, ***how to reduce the accumulated computation cost of the whole iteration process becomes the key challenge***.

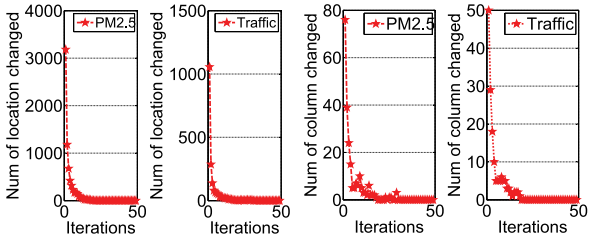
IV. Empirical Study With Real Trace Data

The low-rank feature is the prerequisite for matrix separation. In this section, we first validate that real PM 2.5 air condition data and road traffic data have a good low-rank feature. Then we present our empirical study of applying DRMF for matrix separation, which discovers an interesting feature hidden in DRMF and provides us the opportunity to speed up the whole iteration process.

A. Low-Rank Feature Validation

An $N \times T$ matrix is low-rank if its matrix rank $r \ll \min\{N, T\}$. Although the definition of the precise rank is of high theoretical interest, it is not realistic to use this definition for the practical data. The calculation of the precise rank of the matrix is an ill-posed problem in a practical environment because arbitrary small perturbations of matrix elements may change the rank [33]. Instead of calculating of precise rank, according to PCA, if a matrix is low-rank, its top k singular values occupy the total variance, that is, $\sum_{i=1}^{k} \sigma_i^2 \approx \sum_{i=1}^{r} \sigma_i^2$.

The PM 2.5 air condition data and road traffic data are respectively denoted as PM 2.5 [34] and Traffic [35] are utilized in the experiment. Fig.2 plots the fraction of the total



(a) false location: $N_l(t+1)$ (b) false column: $N_c(t+1)$

Fig. 3. False data candidate position change during iteration process.

variance captured by the top k singular values (i.e., $g(k) = \frac{\sum_{i=1}^{k}\sigma_i^2}{\sum_{i=1}^{r}\sigma_i^2}$) for these two traces. We find that the top 20 singular values capture more than 90% variance in the data, which indicates that real PM 2.5 air condition data and road traffic data have a good low-rank feature.

B. Quick False Location Identification With DRMF

In DRMF, two sub-problems (Low-rank matrix approximation sub-problem (4) and False data detection sub-problem(5)) need to solve iteratively until the two results converge. Among these two sub-problems, the false data detection sub-problem(5)) in each step will identify up to e locations as the candidate false data positions. When the whole iteration process converges, among all e positions, the positions with the large data values will be detected as the outliers. In each iteration step, after update the false data matrix, the clear data matrix C will also be updated and an SVD is required to be executed on C to solve the low-rank matrix approximation sub-problem.

As we will show in Section V, this paper exploits the matrix decomposition SVD in the previous step to deduce the SVD in the current step to speed up the whole iteration process. As a basis, we will investigate how the candidate false data positions change and thus the false data matrix S and the clear data matrix C change in the iteration steps when DRMF is carried out. We define two metrics to track the change of candidate false data position detected during the iteration process, which can be expressed as follows:

- $N_l(t+1)$, which counts the total number of locations that change from a candidate false data location to a normal location or from a normal location to a false location from step t to $t+1$.
- $N_c(t+1)$, which counts the total number of columns that change from a false column (a column that has false candidate locations) to a normal column or from a normal column to a false column from step t to $t+1$.

In the experiment, using the real PM 2.5 air condition data and road traffic data, we randomly select 0.1% locations to inject the false data, then run DRMF. Fig.3 tracks the changes of the false data locations during the iteration process. Obviously, at the initial iteration step, $N_l(t+1)$ (Fig.3(a)) is a large value. Although it takes a longer period of time for the whole DRMF to converge with the false data values change in a new iteration, $N_l(t+1)$ converges quickly. After $t=15$, $N_l(t+1)=0$, which means the candidate false data locations

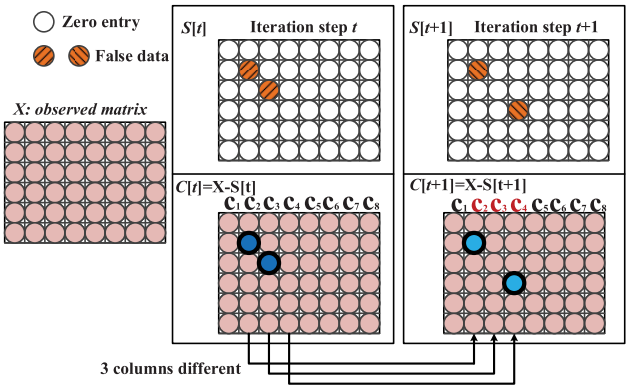


Fig. 4. The relationship between two sequential steps.

will not change any more. $N_c(t+1)$ has a similar trend as that of $N_l(t+1)$ while $N_c(t+1)$ is significantly smaller than $N_l(t+1)$ as one column may contain several false data entries.

This experiment results real a very interesting feature, that is, DRMF can quickly identify the candidate false data locations. We call this feature **quick false data location**. In section V, we will take advantage of this good feature to largely speed up the DRMF execution.

V. Lightweight Matrix Separation Algorithm

As the false data matrix S is updated in each iteration step, $C = X - S$ also changes. To search for the rank-k matrix that can approximate C in each iteration step, the straight-forward way is to perform SVD on the newly updated matrix C. However, as discussed in Section III-C, such a straight-forward solution involves a large accumulated computation cost. In this section, we will exploit the relationship between the sequential matrices $C[t]$ and $C[t+1]$ to significantly speed up the whole iteration process.

A. Opportunity for Speedup the Whole Iteration Process

The false matrix S is usually sparse with only up to e non-zero entries, and all others zeros. If we compare two false data matrices obtained in two sequential steps, the large number of columns in the matrix should remain unchanged except only a few columns (at most $2e$ columns). This relationship provides us an opportunity to reuse the previous result of SVD on $C[t]$ to deduce the SVD for the matrix $C[t+1]$ in current step so that the whole process can be sped up.

Fig.4 further illustrates the above relationship. There are two candidate false data positions with $e=2$. $S[t]$ and $S[t+1]$ denote the false data matrices at the iteration steps t and $t+1$, respectively. $C[t]$ and $C[t+1]$ are the clear matrix to perform SVD for the low-rank matrix approximation. Obviously, when $S[t]$ changes to $S[t+1]$, $C[t]$ changes to $C[t+1]$ with only 3 columns (c_2, c_3, c_4) different, and $N_l(t+1)=2$ and $N_c(t+1)=2$. From the empirical study in Section IV-B, we know that DRMF has the feature of **quick false data location**. Therefore, we can expect that after a small number of iterations, we will have $N_l(t+1)=0$ and $N_c(t+1)=0$, with no more false location changes. After that, the entry values at these false locations may change until they converge.

Therefore, instead of $2e$, there will be only totally e entries to change their values from $C[t]$ to $C[t+1]$, and thus at most e columns will change.

As S is a sparse matrix with e being a small value, only a small number of columns will be changed from $C[t]$ to $C[t+1]$, and the SVD of these two matrices must have strong relationship. In the next section, we will present how we exploit this relationship to design a lightweight matrix separation algorithm.

B. Algorithm Design

Before presenting our matrix separation algorithm, the following two theorems will illustrate how to deduce the SVD of a matrix A from the SVD of a matrix B, if there is only one column difference between the two matrices.

Theorem 1: Given a matrix $B = \left[M, \hat{d}\right] \in \mathbb{R}^{m\times n}$ and $A = [M, d] \in R^{m\times n}$, where $M \in \mathbb{R}^{m\times n-1}$ and d, $\hat{d} \in \mathbb{R}^m$. If $svd(B) = svd\left(\left[M, \hat{d}\right]\right) = U_0\Sigma_0{V_0}^T$ where $U_0 = [u_1, \cdots, u_r] \in \mathbb{R}^{m\times r}$ and $V_0 = [v_1, \cdots, v_r] \in \mathbb{R}^{n\times r}$, we have

$$svd(A) = svd([M, d]) = U_1\Sigma_1{V_1}^T \tag{9}$$

where $U_1 = U_0\tilde{U}(1:r, 1:r) + p\tilde{U}(r+1, 1:r)$, $\Sigma_1 = \tilde{\Sigma}(1:r, 1:r)$ and $V_1 = V_0\tilde{V}(1:r, 1:r) + q\tilde{V}(r+1, 1:r)$ where $\tilde{U}$, $\tilde{\Sigma}$, and $\tilde{V}$ are calculated from Eq.(15).

Proof: As only the last columns of the two matrices A and B are different, we can easily obtain

$$A = [M, d] = \left[M, \hat{d}\right] + ce^T \tag{10}$$

where $c = d - \hat{d}$ and $e^T = \left(\underbrace{0, 0, \cdots, 1}_{n}\right)$. With only the last item of e^T equal to 1, such a design can guarantee that ce^T in (10) is an $m \times n$ matrix with all entries zero except the last column which is c. Further we can rewrite (10) as follows:

$$A = [M, d] = [U_0, c]\begin{bmatrix}\Sigma_0 & 0\\ 0^T & 1\end{bmatrix}\begin{bmatrix}V_0^T\\ e^T\end{bmatrix} \tag{11}$$

To make (11) in the SVD style, it requires that columns in $[U_0, c]$ and $[V_0, e]$ be orthogonal unit vectors. As $svd(B) = svd\left(\left[M, \hat{d}\right]\right) = U_0\Sigma_0{V_0}^T$, we have $U_0^TU_0 = I$ and $V_0^TV_0 = I$, that is, the columns of U_0 and V_0 form orthonormal bases for $\mathbb{R}^r$.

To make columns of $[U_0, c]$ and $[V_0, e]$ become orthogonal unit vectors, we need to consider the Gram-Schmidt orthogonalization of e and c. For example, the orthogonal vector of e under Gram-Schmidt orthogonalization can be expressed as

$$t_a = e - (e, v_1)v_1 - (e, v_2)v_2 - \cdots - (e, v_r)v_r, \tag{12}$$

and we further have

$$t_a = e - V_0a \tag{13}$$

with $a = (V_0^T)_r$ where $(V_0^T)_r$ is the last column of (V_0^T). Further, $q = \frac{t_a}{\|t_a\|_2}$ is the orthogonal unit vector of e.

Similarly, the orthogonal unit vector of c can be calculated as $p = \frac{t_b}{\|t_b\|_2}$ where $t_b = c - U_0b$ and $b = U_0^Tc$.

After the Gram-Schmidt orthogonalization of c and e, we can rewrite the Eq.(11) as following:

$$[M, d] = [U_0, p]\begin{bmatrix}\Sigma_0 + ab^T & \|t_a\|_2a\\ \|t_b\|_2b^T & \|t_a\|_2\|t_b\|_2\end{bmatrix}\begin{bmatrix}V_0^T\\ q^T\end{bmatrix} \tag{14}$$

The SVD of the middle part of the Eq.(14) can be written as follows.

$$svd\left(\begin{bmatrix}\Sigma_0 + ab^T & \|t_a\|_2a\\ \|t_b\|_2b^T & \|t_a\|_2\|t_b\|_2\end{bmatrix}\right) = \tilde{U}\tilde{\Sigma}\tilde{V}^T \tag{15}$$

Based on (15), we can obtain that:

$$[M, d] = [U_0, p]\,\tilde{U}\tilde{\Sigma}\tilde{V}^T\begin{bmatrix}V_0^T\\ q^T\end{bmatrix} \tag{16}$$

Let

$$U_1 = U_0\tilde{U}(1:r, 1:r) + p\tilde{U}(r+1, 1:r) \tag{17}$$

$$\Sigma_1 = \tilde{\Sigma}(1:r, 1:r) \tag{18}$$

$$V_1 = V_0\tilde{V}(1:r, 1:r) + q\tilde{V}(r+1, 1:r) \tag{19}$$

we have $svd(A) = svd([M, d]) = U_1\Sigma_1{V_1}^T$.

Theorem 1 presents the way to quickly deduce the SVD of matrix A from the SVD of matrix B if these two matrices have the same size with only the last column different. In the following theorem, we will show the relationship between SVDs of A and B if only two of their columns are exchanged but all the remaining ones are the same.

Theorem 2: Given a matrix $B = [b_1, b_2, \cdots, b_n] \in R^{m\times n}$ with its $SVD(B) = U\Sigma V^T$, exchange the J-th column with the last column in B, the resulted new matrix is denoted as $A = [b_1, \cdots, b_{J-1}, b_n, b_{J+1}, \cdots, b_J] \in R^{m\times n}$, then we have $SVD(A) = \tilde{U}\tilde{\Sigma}\tilde{V}$ where $\tilde{U} = U$, $\tilde{\Sigma} = \Sigma$, $\tilde{V} = V$ except $\tilde{V}(n,:) = V(J,:)$ and $\tilde{V}(J,:) = V(n,:)$ where $V(n,:)$ and $V(J,:)$ denote the n-th row and the J-th row of the matrix V.

Proof: From the definition of SVD (in Eq.(8)), we have

$$\begin{aligned}B &= [b_1, b_2, \cdots, b_n]\\ &= [u_1, u_2, \ldots, u_r]\begin{bmatrix}\sigma_1 & 0 & 0\\ 0 & \ldots & 0\\ 0 & 0 & \sigma_r\end{bmatrix}\begin{bmatrix}v_{1,1} & \cdots & v_{1,r}\\ \vdots & \cdots & \vdots\\ \mathbf{v_{J,1}} & \cdots & \mathbf{v_{J,r}}\\ \vdots & \cdots & \vdots\\ \mathbf{v_{n,1}} & \cdots & \mathbf{v_{n,r}}\end{bmatrix}^T\end{aligned} \tag{20}$$

As A is obtained from exchanging two columns in B, A can be written as

$$\begin{aligned}A &= [b_1, \cdots, b_{J-1}, b_n, b_{J+1}, \cdots, b_J]\\ &= [u_1, u_2, \ldots, u_r]\begin{bmatrix}\sigma_1 & 0 & 0\\ 0 & \ldots & 0\\ 0 & 0 & \sigma_r\end{bmatrix}\begin{bmatrix}v_{1,1} & \cdots & v_{1,r}\\ \vdots & \cdots & \vdots\\ \mathbf{v_{n,1}} & \cdots & \mathbf{v_{n,r}}\\ \vdots & \cdots & \vdots\\ \mathbf{v_{J,1}} & \cdots & \mathbf{v_{J,r}}\end{bmatrix}^T\end{aligned} \tag{21}$$

Obviously, the proof completes.

Based on Theorem 1 and Theorem 2, given the SVD of matrix A, we can easily design Algorithm 1 to quickly deduce

Algorithm 1 Deduce the SVD When One Column Different

Input: $SVD(B) = U_0\Sigma_0{V_0}^T$, column index l, column difference vector v

Output: SVD of A where A and B have $n-1$ columns the same except the l-th column $a_l = b_l + v$

1: **if** $l \neq n$ **then**
2: Exchange the last row of V_0(i.e., $V_0(n,:)$) with the l-th row of V_0 (i.e., $V_0(l,:)$) according to Theorem 2
3: Update U_1, Σ_1, V_1 according to Theorem 1 using the column difference vector v
4: Exchange the last row of V_1 (i.e., $V_1(n,:)$) with the l-th row of V_1 (i.e., $V_1(l,:)$) according to Theorem 2
5: return $SVD(A) = U_1\Sigma_1{V_1}^T$
6: **else**
7: Update U_1, Σ_1, V_1 according to Theorem 1 using the column difference vector v
8: return $SVD(A) = U_1\Sigma_1{V_1}^T$
9: **end if**

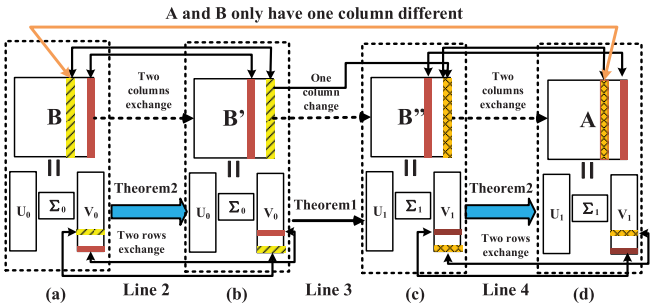


Fig. 5. Deduce SVD of A from SVD of B.

the SVD of matrix B if A and B only have one different column. Fig.5 shows the main operations of line 2-4 in Algorithm 1. In Fig.5(a), $SVD(B) = U_0\Sigma_0{V_0}^T$. In Fig.5(b), the updated U_0, Σ_0, and V_0 form the SVD of B' which is built by exchanging the l column with the last column of B. On line 3, by applying theorem 1, we further have the SVD (U_1, Σ_1, V_1) of $B'' = B'$ with the last column $(B'')_n = (B')_n + v$, as shown in Fig.5(c). On line 4, according to theorem 1, the SVD of A is deduced (in Fig.5(d)).

Based on Algorithm 1, we further design Algorithm 2 to quickly separate the false data matrix from the observed data matrix. On lines 4-5, we scan the false data matrices $S[t]$ to $S[t+1]$ obtained in two sequential steps to identify all possible columns changed from t to $t+1$, with the column index set denoted by R_t. On lines 6-9, update the corresponding SVD sequently until all columns are changed and we obtain the final SVD of $C[t+1]$. On line 10, set $L[t+1]$ as the rank-k truncated SVD of matrix $C[t+1]$, that is, $L[t+1] = C[t+1]_k$.

C. Algorithm Analysis

For the original DRMF, each iteration step requires SVD, which brings $O\left(\min\left\{mn^2, nm^2\right\}\right)$ time complexity.

Different from DRMF, our LightLRFMS only requires one time execution of SVD at the first iteration step. After that, LightLRFMS deduces each next SVD from the previous

Algorithm 2 Light Weight Low Rank and Sparse Matrix Separation

Input: X: the observation matrix
k: the maximal rank of the matrix factorizaiton
e: the maximal number of false data injected
S: the inial false data matrix
t: iterative step

1: $t = 1$, $S[t] = S$, $C[t] = X - S[t]$, $SVD(C[t]) = U_0\Sigma_0{V_0}^T$, $L[t] = C[t]_k$
2: **while** not converged **do**
3: Apply (6) to solve the false data detection problem:

$$S[t+1] = \arg\min_S \|E - S\|_F^2 \quad \text{s.t. } E = X - L[t] \quad \|S\|_0 \leq e \tag{22}$$

4: Scan $S[t]$ and $S[t+1]$, use the sets $R[t]$ and $R[t+1]$ to record the column index in $S[t]$ and $S[t+1]$ that have entry values not zero, respectively.
5: $R_t = R[t] \cup R[t+1]$
6: **for** each $r \in R_t$ **do**
7: $v = S[t]_r - S[t+1]_r$
8: update $SVD(C[t+1])$ by applying Algorithm 1 with $SVD(C[t+1]) = U_0\Sigma_0{V_0}^T$, r, and v as the input
9: **end for**
10: Set $L[t+1]$ as the rank-k truncated SVD of matrix $C[t+1]$, that is, $L[t+1] = C[t+1]_k$
11: $t = t+1$
12: **end while**

SVD by exploiting the relationship of matrices operated in sequential iteration steps. As shown in Algorithm 2, if there are $|R_t|$ columns of difference between $C[t]$ and $C[t+1]$, we will perform $|R_t|$ times SVD deductions through Algorithm 1.

The computation cost of Algorithm 1 is mainly caused by the step on line 3 or line 7, which applies the Theorem 1 to deduce the SVD when one column changes. According to Theorem 1, the main operations to deduce SVD include a Gram-Schmidt operation (in Eq.(13)) with the time complexity $O(mr)$, an SVD on a small size matrix (in Eq(15)) with the time complexity $O\left((r+1)^3\right)$, and the update operations on Eq.(17), Eq.(18), and Eq.(19), which requires $O\left(mr^2\right)$, $O(1)$, and $O\left(mr^2\right)$ time complexity, respectively. Therefore, applying Theorem 1 one time to deduce the SVD involves the complexity of $O\left(mr^2\right) + O\left((r+1)^3\right) + O(1)$. One iteration step in our Algorithm 2 requires $|R_t|$ times of SVD deduction, thus the complexity is $|R_t| \times (O\left(mr^2\right) + (O(mr) + O\left((r+1)^3\right) + O(1))$.

According to Fig.2, as PM 2.5 air condition data and road traffic data have strong correlations, we have the rank $r \ll \min\{m,n\}$. Moreover, according to the analysis in Section V-A, in the inial iteration steps, the number of updated columns $|R_t|$ is at most $2e$. After a few steps, the **quick false data location** makes the updated column number $|R_t|$ even less and at most e. As outlier seldom happens, e is

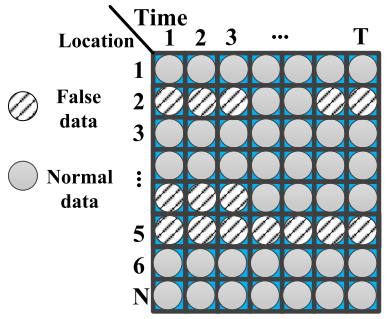


Fig. 6. Structured false data.

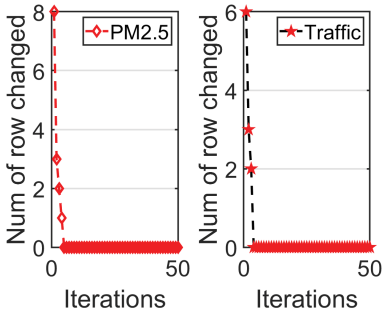


Fig. 7. False data candidate rows change during iteration process.

obviously a small value. Therefore, we can easily conclude $|R_t| \times \left(O\left(mr^2\right) + \left(O\left(mr\right) + O\left((r+1)^3\right) + O\left(1\right)\right) \ll O\left(\min\left\{mn^2, nm^2\right\}\right)$, that is, our LightLRFMS has much smaller computation cost than DRMF. In our performance studies, we will show the significant speed gain achieved by our LightLRFMS.

VI. Extensions to Structured False Data Detection

Problem in (3) uses l_0 norm penalty on the false data matrix S to introduce element-wise sparsity. However, recent studies show that successive/mass data corruption may be resulted from the matrix rows (Fig. 6) due to the failure of sensor nodes, continuous data tampering on nodes by attackers, and the severe communication conditions. In this paper, we call such successive or mass data corruption as structured false data.

In this section, we first introduce the structured false data detection problem and its iterative solution, then present our lightweight solution for the problem.

A. Structured False Data Detection Problem and Its Iterative Solution

For the structured false data, we would like to look for the outlier rows so that the evidences of anomalies can be aggregated to enhance the false data detection performance. Instead of counting the number of outlier entries in problem (3), we use the structure norm $\|S\|_{2,0}$ to represent the number of rows with nonzero entries in matrix S. The false row detection problem can be expressed as

$$\begin{aligned} &\min_{L,S} \|(X-S)-L\|_F^2 \\ &\text{s.t. } \text{rank}\,(L) \le k \\ &\quad \|S\|_{2,0} \le \tau \end{aligned} \tag{23}$$

where τ is the maximal number of false rows.

Similar to Section III-B, we adopt the block coordinate descent strategy to solve the problem (23), where two sub-problems need to be solved iteratively.

- Low-rank matrix approximation sub-problem

$$\begin{aligned} L = &\arg\min_{L} \|C-L\|_F^2 \\ &\text{s.t. } C = X - S \\ &\quad rank(L) \le k \end{aligned} \tag{24}$$

- Structured false data detection sub-problem

$$\begin{aligned} S = &\arg\min_{S} \|E-S\|_F^2 \\ &\text{s.t. } E = X - L \\ &\quad \|S\|_{2,0} \le \tau \end{aligned} \tag{25}$$

By treating each row of S as an element, based on (6), we obtain the solution for the sub-problem in (25) as follows:

$$S_{i,:} = \begin{cases} E_{i,:} & \beta_i > \beta\left(\tau\right) \\ 0 & otherwise \end{cases} \tag{26}$$

where $S_{i,:}$ and $E_{i,:}$ denote the i-th row of matrices S and E respectively. $\beta_i = \|E_{i,:}\|_F$ and $\beta\left(\tau\right)$ is the τ-th largest value in $\{\beta_i\}_{i=1,\ldots N}$.

B. Lightweight Algorithm

In Section IV, we find the feature of ***quick false data location*** when we follow DRMF to solve problem in (3) by adopting the block coordinate descent strategy. We expect that there exists a similar feature in the iterative solution in Section VI-A. Similar to Section IV, we conduct experiments to investigate whether there exists the similar feature by using the metric $N_r(t+1)$ to track the change of candidate false data rows detected during the iterative process.

- $N_r(t+1)$, which counts the total number of rows that change from a false row (a row that has false candidate locations) to a normal row or from a normal row to a false row from step t to $t+1$.

In the experiment, using the real PM 2.5 air condition data and road traffic data, we randomly select 5% rows to inject the false data, then run the iterative solution in Section VI-A. Fig.7 shows the experiment results. As expected, only after very few iterative steps, $N_r(t+1)$ converges quickly to zero. These experiment results show that DRMF can quickly identify the candidate false data rows. The feature of ***quick false data location*** also exists when we adopt block coordinate descent strategy to detect the structured false data.

Taking advantage of the feature, we propose another version of LightLRFMS in Algorithm 3 to detect the structured false data. To well utilize the properties hidden in Theorem 1 and Theorem 2, a simple matrix transpose is executed at matrices S and X to transform the rows of the matrices to the columns, as shown in line 1 in Algorithm 3.

Similar to the analysis in Section V-C, one iteration step in Algorithm 3 requires $|R_t|$ times of SVD deduction, thus the

Algorithm 3 Detection of Structured False Data With Light Weight Low Rank and Sparse Matrix Separation

Input: X: the observation matrix
k: the maximal rank of the matrix factorizaiton
τ: the maximal number of false data rows injected
S: the inial false data matrix
t: iterative step

1: $t = 1$, $S[t] = S^T$, $C[t] = X^T - S[t]$, $SVD(C[t]) = U_0\Sigma_0{V_0}^T$, $L[t] = C[t]_k$
2: **while** not converged **do**
3: Apply (26) to solve the structured false data detection sub-problem:

$$S[t+1] = \arg\min_{S} \|E - S\|_F^2 \quad \text{s.t. } E = X - L[t] \quad \|S\|_{2,0} \le \tau \tag{27}$$

4: Scan $S[t]$ and $S[t+1]$, use the sets $R[t]$ and $R[t+1]$ to record the column index in $S[t]$ and $S[t+1]$ that have structured data values not zero, respectively.
5: $R_t = R[t] \cup R[t+1]$
6: **for** each $r \in R_t$ **do**
7: $v = S[t]_r - S[t+1]_r$
8: update $SVD(C[t+1])$ by applying Algorithm 1 with $SVD(C[t+1]) = U_0\Sigma_0{V_0}^T$, r, and v as the input
9: **end for**
10: Set $L[t+1]$ as the rank-k truncated SVD of matrix $C[t+1]$, that is, $L[t+1] = C[t+1]_k$
11: $t = t+1$
12: **end while**

Output: $S = S[t]^T$ $L = X^T - S$

complexity is $|R_t| \times (O(mr^2) + (O(mr) + O((r+1)^3) + O(1))$. As indicated in Fig.7, after a few steps, $N_r(t+1)$ is equal to zero which makes the updated column number $|R_t|$ even less and at most τ. Therefore, after a few steps, the complexity of one iteration step is at most $|\tau| \times (O(mr^2) + (O(mr) + O((r+1)^3) + O(1))$, which is obviously much less than $O(\min\{mn^2, nm^2\})$, the complexity of SVD. Therefore, in the case of structured false data detection, our LightLRFMS has much smaller computation cost than DRMF.

VII. Performance Evaluations

We use one PM 2.5 air condition data (denoted as PM 2.5) and one road traffic data (denoted as Traffic) to evaluate the performance of our proposed LightLRFMS.

- PM 2.5 [34] includes PM 2.5 air condition data every one hour in the time span of 2014-05-01 to 2015-04-30 from 437 monitoring locations in 43 cities in China including Beijing, Tianjin, Guangzhou, Shenzhen, and other 39 adjacent cities.
- Traffic [35] includes traffic speed data collected from 142 road segments in Manhattan (New York City) every five minutes from 04:00 AM to 23:55 PM in every day during the time span from 2017-11-29 to 2018-01-11.

As trace PM 2.5 and trace traffic record the real measurements of PM 2.5 air condition and traffic speed, to evaluate the proposed false data detection algorithm, we inject the outliers to these raw data to generate the synthesized corrupted data with following steps.

1) ***Data normalization***. We denote the raw trace data as $L \in R^{N\times T}$. For more efficient data processing, given $l_{i,j}$, we adopt $l_{i,j} = \frac{l_{i,j} - \min_{u,v}\{l_{u,v}\}}{\max_{u,v}\{l_{u,v}\} - \min_{u,v}\{l_{u,v}\}}$ to normalize the data within the range [0,1], where $\max_{u,v}\{l_{u,v}\}$ and $\min_{u,v}\{l_{u,v}\}$ are the maximum value and minimum value of all the data, respectively.

2) ***False data generation.*** A false data matrix S is generated as

$$s_{i,j} = \begin{cases} s_{i,j} & (i,j) \in \Omega \\ 0 & otherwise \end{cases} \tag{28}$$

where s_{ij} is the generated outlier, and Ω is the set of false data locations.

The synthesized data X is: $x_{i,j} = l_{i,j} + s_{i,j}$ for all (i,j), and we adopt two data distributions to generate the false data value injected:

1) ***Gaussian distribution:*** $s_{ij} \in \Omega$ is generated following the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ with the defaults values of mean μ and variance σ^2 set to 0 and 0.1, respectively.

2) ***Exponential distribution:*** $s_{ij} \in \Omega$ is generated following the Exponential distribution $E\left(\frac{1}{\mu}\right)$ with the defaults values of mean $\mu = 0.1$.

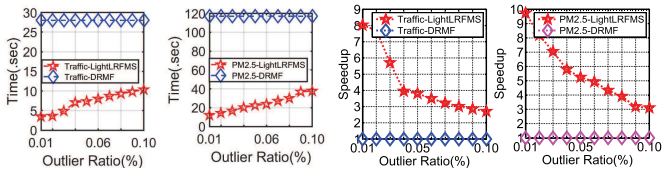
The following four metrics are utilized to evaluate our proposed LightLRFMS. **False Positive Rate (FPR):** It measures the proportion of non-outliers that are wrongly identified as outliers. **True Positive Rate (TPR):** It measures the proportion of outliers that are correctly identified. **Error(false):** is the average value of the differences between false values estimated and the false values injected. **Speedup:** Given the computation time under two different algorithms (alg_1 and alg_2), denoted as T_1 and T_2, the speedup in the computation time of the alg_2 with respect to the alg_1: $S_{1-2} = T_1/T_2$.

All experiments are run by a workstation equipped with two Intel(R) Core (R) i5-8300H CPUs(2.30 GHz) (totally 4 Cores) and 32.00GB RAM. To measure the processing time, we insert a timer into all the schemes implemented.

In Section V and Section VI, we propose two versions of our LightLRFMS, one is to detect the random false data and the other is to detect the structured false data. Actually, we can further easily extend our LightLRFMS to detect mixture false data that consists of both random and structured corruptions. To evaluate the detection performance, three groups of experiments are performed. The first group is to detect the false data with random locations, the second group is to detect the structured false data, and the third is to detect the mixture false data. All performance results are obtained with average of results from 10 random runs.

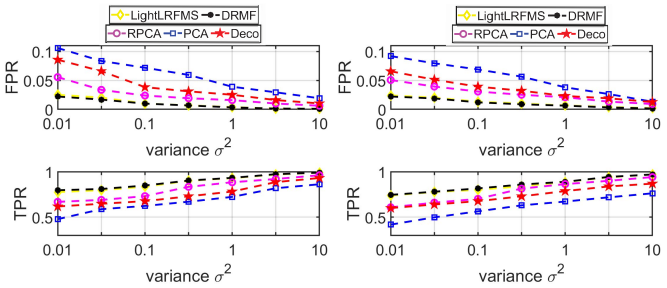
A. Group 1: Detection of Random False Data

In this group of experiments, to simulate false data that do not have fixed locations, we randomly select $\gamma \times (N \times T)$



(a) Time (b) SpeedUp

Fig. 8. Speed comparison - inject random false data.



(a) Traffic (b) PM2.5

Fig. 9. Detection accuracy - inject random Gaussian distribution outlier with different variance.

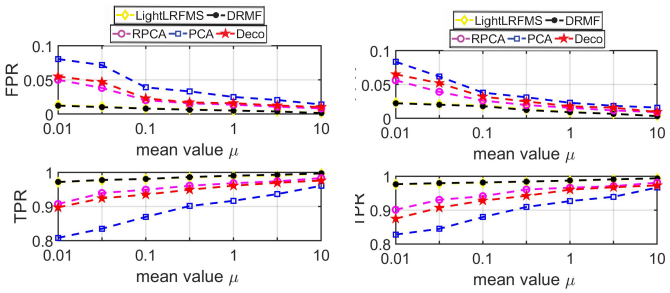
locations to form Ω, where γ is the false data ratio. We set γ=0.1% as the default setting.

Besides our LightLRFMS, we implement other four schemes including Deco [21], DRMF [20], RPCA [19] and PCA [36]. To fairly compare these algorithms, we adopt the same false detection principle: among all the candidate false locations, return the $\gamma \times (N \times T)$ data points with the largest $\gamma \times (N \times T)$ absolute values where $\gamma \times (N \times T)$ is the number of false data points injected.

1) Speed Comparison: In Fig.8, we compare the computation speeds between DRMF and our LightLRFMS under random attack. Fig.8(a) shows the computation time. Moreover, by using DRMF as the baseline algorithm and set $alg_1 = DRMF$, we also calculate the speedup metric and show in Fig.8(b). With the increase of γ, the Speedup of our LightLRFMS decreases as higher computation complexity is involved in the SVD deduction. When the false data ratio $\gamma = 0.01\%$, it takes LightLRFMS 12 seconds (PM 2.5) and 3.5 seconds (Traffic) to detect the false data, while it takes DRMF 120 seconds (PM 2.5) and 28 seconds (Traffic). Our LightLRFMS is up to 10 (PM 2.5) and 8 (Traffic) times faster than DRMF.

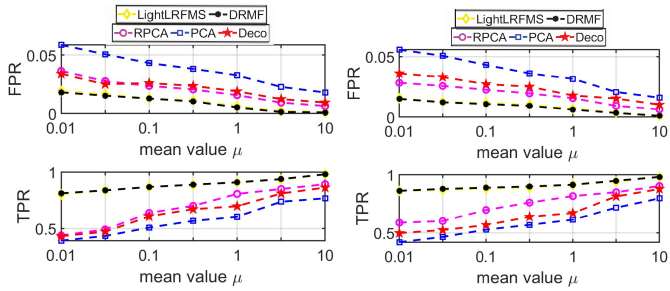
In following experiments, we will show although our LightLRFMS achieves significant speed gain, it can achieve the same and best false data detection performance as DRMF, which is much better than other peer algorithms.

2) Detection Accuracy (FPR and TPR): With other parameters fixed, we vary the variance σ^2, the average value μ, and the outlier ratio γ of the outliers injected. In Fig.9 - Fig.13, our LightLRFMS and DRMF achieve the lowest False Positive Rate and Highest True Positive Rate, under all the experiment scenarios using different data traces (PM 2.5 and Traffic) with different anomaly injected strategies, which demonstrates that our LightLRFMS in Algorithm 1 can highly accurately deduce the SVD by well utilizing the relationship between matrices in two sequential iteration steps.



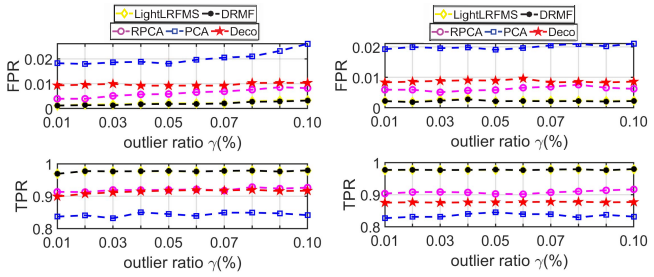
(a) Traffic (b) PM2.5

Fig. 10. Detection accuracy - inject random Gaussian distribution outlier with different mean value.



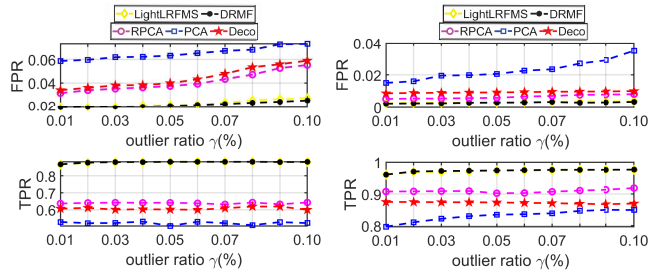
(a) Traffic (b) PM2.5

Fig. 11. Detection accuracy - inject random Exponential distribution outlier with different mean value.



(a) Traffic (b) PM 2.5

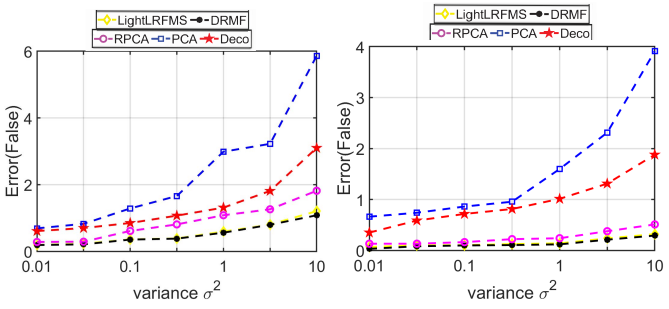
Fig. 12. Detection accuracy - inject random Gaussian distribution outlier with different outlier ratio.



(a) Traffic (b) PM 2.5

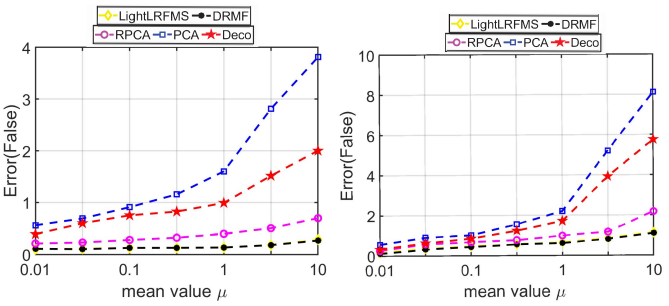
Fig. 13. Detection accuracy - inject random Exponential distribution outlier with different outlier ratio.

The false positive rate under RPCA is up to 2 times larger than that under DRMF and our LightLRFMS. RPCA uses the trace norm to relax the low-rank feature of monitoring matrix, which largely impacts the detection performance. Higher false positive rate would result in false anomaly alarms, which may largely increase the MCS system maintenance cost.



(a) Traffic (b) PM2.5

Fig. 14. Error on false data estimated - inject random Gaussian distribution outlier with different variance.



(a) Traffic (b) PM2.5

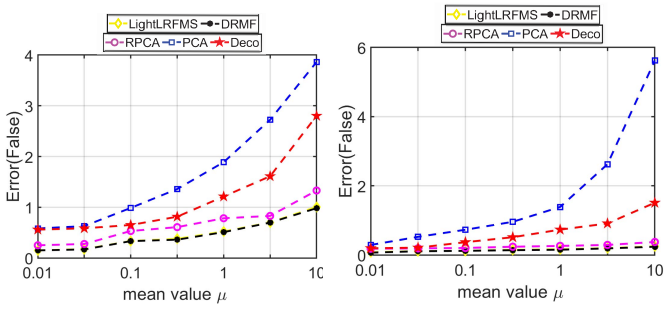
Fig. 15. Error on false data estimated - inject random Gaussian distribution outlier with different mean value.

For the Gaussian distribution, as shown in Fig.9 and Fig.10, 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. Moreover, with the increase of variance σ^2 and mean μ, the outlier value is generated in a large range, PCA is not robust to these outliers and results in the largest False Positive Rate and lowest True Positive Rate, which confirms the observation of [20]. For the Exponential distribution (in Fig.11), similar to the simulation results in Gaussian distribution, with the increase of mean μ of the outliers the True Positive Rate increases while the False Positive Rate decreases for all algorithms implemented.

Fig.12 and Fig.13 draw the experiment results under different outlier ratios when the outlier locations are randomly generated. As DRMF and our LightLRFMS formulate the problem directly using the matrix rank to represent the low rank feature and the L_0-norm to represent the sparse feature of the false data, the detection performance of these two algorithms are much more stable compared to peer detection algorithms.

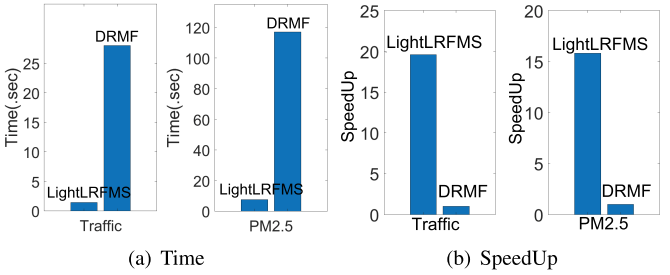
3) Error on False Data Estimated: From the experiment results in Fig.3, we find DRMF can quickly detect the false data location although DRMF doesn't converge as the false data values estimated still change over iteration steps. Fig.14- Fig.16 show the error on the false value estimated.

Obviously, our LightLRFMS and DRMF achieve the similar best performance with the lowest error. With the increase of



(a) Traffic (b) PM2.5

Fig. 16. Error on false data estimated - inject random Exponential distribution outlier with different mean value.



(a) Time (b) SpeedUp

Fig. 17. Speed comparison - inject structured false data.

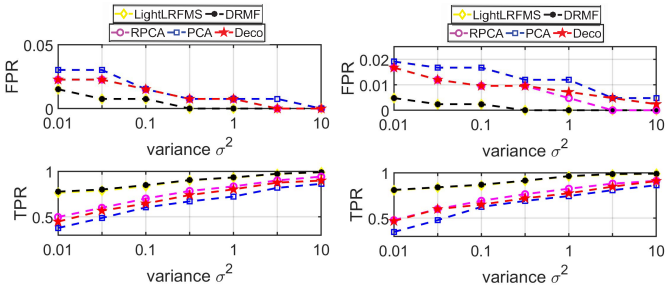
variance σ^2 and mean μ of the outliers, synthesized outlier data have larger and different values, which impacts the normal data recovery thus the estimation of the false data values. As a result, the errors under all schemes implemented increases. PCA only considers the constraint of low rank measurement data but ignores the sparsity feature of false data injected, which results in high error on the estimation of false data value.

B. Group 2: Detection of Structured False Data

In MCS, the failure of sensor nodes, constant data tampering by attackers, and the severe communication conditions may cause structured data corruption in matrix rows [27]. To simulate such data corruptions, we randomly select κN (κ=5%) locations in PM 2.5 trace and road segments in traffic trace be the ones under such an attack.

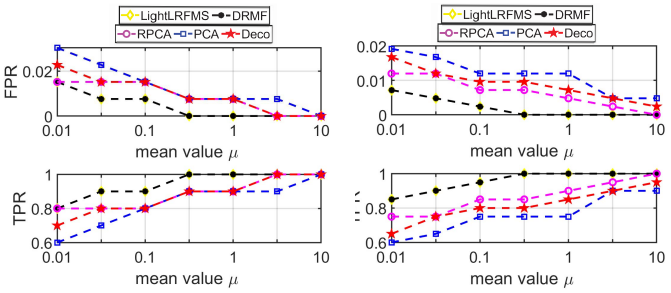
Similar to our LightLRFMS, by using the structure norm $\|S\|_{2,0}$ to represent the number of rows with nonzero entries in matrix S, DRMF can detect the structured false data. Besides LightLRFMS and DRMF, Deco [21], RPCA [19] and PCA [36] do not directly support the detection of structured false data. For fair comparison, we calculate the Frobenius norm of the rows in the false data matrix (S) in these algorithms and return the κN-largest rows as the structured false data rows.

Fig.17, Fig.18- Fig.20, and Fig.21- Fig.23 show the speed, the detection accuracy performance, and the error on the false data estimated under the experiment scenario of injecting structured corruptions. As expected, our LightLRFMS and DRMF achieve the best accuracy performance to detect the structured false data, while our LightLRFMS is almost 20 times (Traffic) and 15 times (PM2.5) faster than DRMF



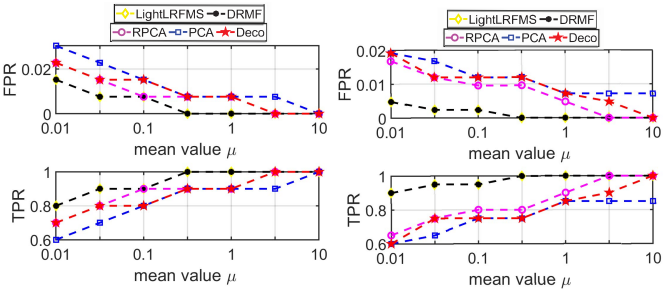
(a) Traffic (b) PM2.5

Fig. 18. Detection accuracy - inject structured Gaussian distribution outlier with different variance.



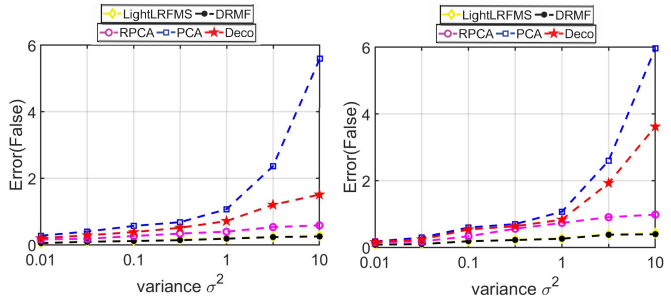
(a) Traffic (b) PM2.5

Fig. 19. Detection accuracy - inject structured Gaussian distribution outlier with different mean value.



(a) Traffic (b) PM2.5

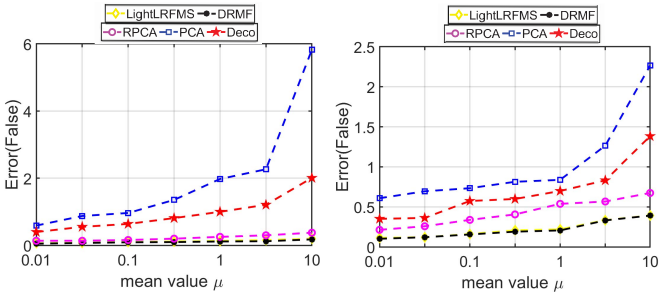
Fig. 20. Detection accuracy - inject structured Exponential distribution outlier with different mean value.



(a) Traffic (b) PM2.5

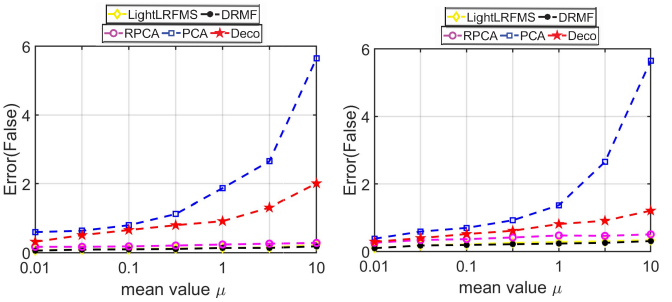
Fig. 21. Error on false data estimated - inject structured Gaussian distribution outlier with different variance.

for both data sets, which demonstrates that SVD deduction in our LightLRFMS is very effective in reducing the computation cost.



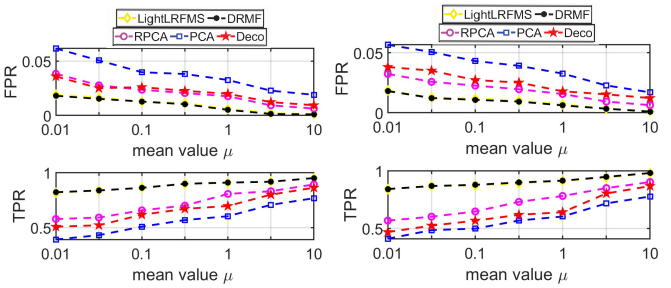
(a) Traffic (b) PM2.5

Fig. 22. Error on false data estimated - inject structured Gaussian distribution outlier with different mean value.



(a) Traffic (b) PM2.5

Fig. 23. Error on false data estimated - inject structured Exponential distribution outlier with different mean value.



(a) Traffic (b) PM2.5

Fig. 24. Detection accuracy - Inject random and structured Exponential distribution outlier with different mean value.

C. Group 3: Detection of Mixture False Data Consisting of Both Random and Structured Corruptions

To simulate the mixture false data scenario, we randomly select κN (κ=5%) rows to inject the structured corruptions. Besides these rows, we also randomly select $\gamma \times ((N - \kappa N) \times T)$ locations to inject the random corruptions.

To detect the mixture false data, we can apply both the L_0-norm and $L_{2,0}$-norm to constrain the number of random false data and the number of structured false rows in DRMF and our LightLRFMS. For other peer algorithms, we calculate the Frobenius norm of the rows in the false data matrix (S) in these algorithms and return the κN-largest rows as the structured false data rows. Besides these rows, we also return the largest $\gamma \times ((N - \kappa N) \times T)$ data points in matrix S with the largest absolute values. For space saving, we only

list the detection results with the injected outlier data values following the Exponential distribution in Fig.24. Obviously, both our LightRFMS and DRMF achieve the highest false data detection performance with the highest False Positive Rate and the smallest True Positive Rate.

VIII. Conclusion

To accurately and quickly detect the false data in MCS, we propose a novel LightLRFMS based on DRMF. From the empirical study, we find DRMF has an interesting feature called ***quick false data location***, that is, DRMF can quickly detect the candidate false data locations in the first few iteration steps. To conquer the computation complexity problem faced in DRMF, taking advantage of the sparsity feature of the false data and DRMF's quick false data location feature, we propose to reuse the previous result of the matrix SVD decomposition to deduce the one for the current iteration step, which largely speeds up the whole iteration process. Furthermore, as the false data may be caused by random corruptions or success/mass corruptions, our LightLRFMS is designed with two versions to detect the random false data and the structured false data respectively. Extensive experiment results demonstrate that LightLRFMS can achieve the same highest false data detection accuracy as DRMF at significantly faster speed (up to 20 times that of DRMF) thanks to its lower computation cost.

We don't expect that our scheme works for all MCS applications. This paper focuses on false data detection in the environment monitoring, one of the most popular MCS applications and with the sensory data often formed as a low-rank matrix. Some previous studies [25], [37]–[40] also show that sensory matrices of temperature, humidity, light, and sound level are low-rank and have high spatiotemporal correlations. Although this paper utilizes two data sets PM 2.5 air condition data and road traffic data as an example to evaluate our proposed LightLRFMS, we expect that our scheme can also work well to detect the false data in other sensory matrices. In our future work, we will evaluate the performance of our LightLRFMS in these cases.

References

[1] B. Guo *et al.*, "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm," *ACM Comput. Surv.*, vol. 48, no. 1, p. 7, 2015.

[2] R. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.

[3] Z. Wang *et al.*, "When mobile crowdsensing meets privacy," *IEEE Commun. Mag.*, vol. 57, no. 9, pp. 72–78, Sep. 2019.

[4] Z. Wang *et al.*, "Personalized privacy-preserving task allocation for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1330–1341, Jun. 2019.

[5] Y. Qu *et al.*, "Posted pricing for chance constrained robust crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 1, pp. 188–199, Jan. 2020.

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

[7] Q. Liu, P. Hou, G. Wang, T. Peng, and S. Zhang, "Intelligent route planning on large road networks with efficiency and privacy," *J. Parallel Distrib. Comput.*, vol. 133, pp. 93–106, Nov. 2019.

[8] M. Mun *et al.*, "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research," in *Proc. 7th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2009, pp. 55–68.

[9] B. Predić, Z. Yan, J. Eberle, D. Stojanovic, and K. Aberer, "ExposureSense: Integrating daily activities with air quality using mobile participatory sensing," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PERCOM Workshops)*, Mar. 2013, pp. 303–305.

[10] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-Air: When urban air quality inference meets big data," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2013, pp. 1436–1444.

[11] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "BikeNet: A mobile sensing system for cyclist experience mapping," *ACM Trans. Sensor Netw.*, vol. 6, no. 1, 2009, Art. no. 6.

[12] M. Talasila, R. Curtmola, and C. Borcea, "Improving location reliability in crowd sensed data with minimal efforts," in *Proc. 6th Joint IFIP Wireless Mobile Netw. Conf. (WMNC)*, Apr. 2013, pp. 1–8.

[13] S. Saroiu and A. Wolman, "I am a sensor, and I approve this message," in *Proc. 11th Workshop Mobile Comput. Syst. Appl. (HotMobile)*, 2010, pp. 37–42.

[14] M. C. Vuran, Ö. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: Theory and applications for wireless sensor networks," *Comput. Netw.*, vol. 45, no. 3, pp. 245–259, Jun. 2004.

[15] F. Xiao, C. Sha, L. Chen, L. Sun, and R. Wang, "Noise-tolerant localization from incomplete range measurements for wireless sensor networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 2794–2802.

[16] Y.-C. Chen, L. Qiu, Y. Zhang, G. Xue, and Z. Hu, "Robust network compressive sensing," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2014, pp. 545–556.

[17] S. Lazarova-Molnar, H. P. Logason, P. G. Andersen, and M. B. Kjaergaard, "Mobile crowdsourcing of data for fault detection and diagnosis in smart buildings," in *Proc. Int. Conf. Res. Adapt. Convergent Syst.*, 2016, pp. 12–17.

[18] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psychol.*, vol. 24, no. 6, pp. 417–441, 1933.

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

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

[21] L. Cheng *et al.*, "Deco: False data detection and correction framework for participatory sensing," in *Proc. IEEE 23rd Int. Symp. Qual. Service (IWQoS)*, Jun. 2015, pp. 213–218.

[22] B. Wang, L. Kong, L. He, F. Wu, J. Yu, and G. Chen, "I (TS, CS): Detecting faulty location data in mobile crowdsensing," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 808–817.

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

[24] 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.

[25] L. Kong, M. Xia, X.-Y. Liu, M.-Y. Wu, and X. Liu, "Data loss and reconstruction in sensor networks," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 1654–1662.

[26] 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.*, vol. 17, no. 7, pp. 1595–1608, Jul. 2018.

[27] 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.

[28] K. Xie, X. Li, X. Wang, G. Xie, J. Wen, and D. Zhang, "Active sparse mobile crowd sensing based on matrix completion," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, 2019, pp. 195–210.

[29] K. Xie *et al.*, "Accurate recovery of missing network measurement data with localized tensor completion," *IEEE/ACM Trans. Netw.*, vol. 27, no. 6, pp. 2222–2235, Dec. 2019.

[30] K. Xie *et al.*, "Accurate recovery of Internet traffic data under variable rate measurements," *IEEE/ACM Trans. Netw.*, vol. 26, no. 3, pp. 1137–1150, Jun. 2018.

[31] 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.

[32] Z. Lu and Y. Zhang, "Penalty decomposition methods for $L0$-norm minimization," 2010, *arXiv:1008.5372*. [Online]. Available: https://arxiv.org/abs/1008.5372

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

[34] Y. Zheng *et al.*, "Forecasting fine-grained air quality based on big data," in *Proc. SIGKDD KDD*, New York, NY, USA, 2015, pp. 2267–2276. [Online]. Available: http://doi.acm.org/10.1145/2783258.2788573
[35] *Flowmap*. Accessed: Jan. 16, 2018. [Online]. Available: http://flowmap.nyctmc.org/weborb4/flowmap/#
[36] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 219–230, Oct. 2004.
[37] L. Cheng *et al.*, "Compressive sensing based data quality improvement for crowd-sensing applications," *J. Netw. Comput. Appl.*, vol. 77, pp. 123–134, 2017.
[38] G. Chen *et al.*, "Multiple attributes-based data recovery in wireless sensor networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 103–108.
[39] J. Cheng, Q. Ye, H. Jiang, D. Wang, and C. Wang, "STCDG: An efficient data gathering algorithm based on matrix completion for wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 2, pp. 850–861, Feb. 2013.
[40] Q. Yuan, Z. Liu, J. Li, S. Yang, and F. Yang, "An adaptive and compressive data gathering scheme in vehicular sensor networks," in *Proc. IEEE 21st Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2015, pp. 207–215.

Xiaocan Li is currently pursuing the Ph.D. degree with Hunan University. His research interests include matrix/tensor factorization and anomaly detection.

Kun Xie received the Ph.D. degree in computer application from Hunan University, Changsha, China, in 2007. She is currently a Professor with Hunan University, the Peng Cheng Laboratory, and the Purple Mountain Laboratory. She has published over 60 articles in major journals and conference proceedings, including journals such as the IEEE/ACM TRANSACTIONS ON NETWORKING, the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and the IEEE TRANSACTIONS ON SERVICES COMPUTING, and conferences, including SIGMOD, INFOCOM, ICDCS, SECON, DSN, and IWQoS. Her research interests include network measurement, network security, big data, and AI.

Xin Wang (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Columbia University, USA. She is currently an Associate Professor with the Department of Electrical and Computer Engineering, The State University of New York at Stony Brook, USA. Her research interests include algorithm and protocol design in wireless networks and communications, mobile and distributed computing, and networked sensing and detection. She was a member of the ACM in 2004. She received the NSF Career Award in 2005 and the ONR Challenge Award in 2010.

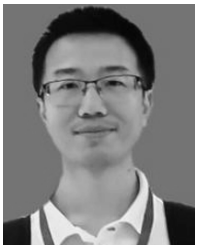
Gaogang Xie received the B.S. degree in physics and the M.S. and Ph.D. degrees in computer science from Hunan University in 1996, 1999, and 2002, respectively. He is currently a Professor with the Computer Network Information Center (CNIC), Chinese Academy of Sciences (CAS), and the University of Chinese Academy of Sciences. He is also the Vice President of CNIC. His research interests include Internet architecture, packet processing and forwarding, and Internet measurement.

Dongliang Xie received the Ph.D. degree from the Beijing Institute of Technology, China, in 2002. He is currently a Full Professor and the Director of the State Key Laboratory of Networking and Switching Technology, Broadband Network Research Center, Beijing University of Posts and Telecommunications (BUPT), China. His research interests have expanded from wireless sensor networks to the future information centric networks, mobile social networks, and mobile cloud computing.

Zhenyu Li (Member, IEEE) received the Ph.D. degree in 2009. He is currently a Full Professor with the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), and Taobao.com. His research interests include Internet architecture 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 with the Computer Network Information Center (CNIC), Chinese Academy of Sciences. His research interests include wireless networks and mobile computing, and high-speed network measurement and management.

Zulong Diao received the Ph.D. degree in software engineering from Hunan University, Changsha, China, in 2019. He is currently working as a Research Associate with the Institute of Computing Technology, Chinese Academy of Sciences, and the Purple Mountain Laboratory. His research interests include machine learning, edge computing, and abnormal detection.



Tian Wang received the B.Sc. and M.Sc. degrees in computer science from Central South University in 2004 and 2007, respectively, and the Ph.D. degree from the City University of Hong Kong in 2011. He is currently a Professor with the College of Computer Science and Technology, Huaqiao University, China. His research interests include the Internet of Things and edge computing.