IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 17, 2018, accepted September 4, 2018, date of publication September 17, 2018, date of current version August 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2870298

EdgeSense: Edge-Mediated Spatial-Temporal Crowdsensing

SIJIA YANG[1,2], JIANG BIAN[3], LICHENG WANG[1], HAOJIN ZHU[4], YANJIE FU[5], AND HAOYI XIONG[5]

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Télécom Ecole de Management, 91000 Évry, France
[3]Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816, USA
[4]Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200240, China
[5]Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409, USA

Corresponding author: Haoyi Xiong (xhyccc@gmail.com)

This work was supported in part by the National Key R&D Program of China under Grant 2016YFB0800602, in part by the National Natural Science Foundation of China (NSFC) under Grant 61502048, in part by the Shandong Provincial Key R&D Program of China under Grant 2018CXGC0701, and in part by the Missouri S&T Smart CPS Startup Grant.

ABSTRACT Edge computing recently is increasingly popular due to the growth of data size and the need of sensing with the reduced center. Based on Edge computing architecture, we propose a novel crowdsensing framework called *Edge-Mediated Spatial-Temporal Crowdsensing*. This algorithm targets on receiving the environment information such as air pollution, temperature, and traffic flow in some parts of the goal area, and does not aggregate sensor data with its location information. Specifically, *EdgeSense* works on top of a secured peer-to-peer network consisted of participants and propose a novel *Decentralized Spatial-Temporal Crowdsensing* framework based on *Parallelized Stochastic Gradient Descent*. To approximate the sensing data in each part of the target area in each sensing cycle, *EdgeSense* uses the local sensor data in participants' mobile devices to learn the low-rank characteristic and then recovers the sensing data from it. We evaluate the *EdgeSense* on the real-world data sets (temperature [1] and PM2.5 [2] data sets), where our algorithm can achieve low error in approximation and also can compete with the baseline algorithm which is designed using centralized and aggregated mechanism.

INDEX TERMS Edge computing, crowdsourcing, and distributed sensing.

I. INTRODUCTION

Mobile Crowdsensing (MCS) leverages consumer-centric mobile devices, such as smartphones and wearables, as a scalable data collection platform that measures phenomena of common interest [3], [4]. Typically, MCS applications focus on community sensing tasks for large-scale phenomena that cannot easily be measured by a single individual. Instead, these phenomena can only be measured accurately when data are aggregated spatio-temporally from many individuals.

Extensive works have been done to study MCS-empowered applications, including environment monitoring (e.g. noise and air pollutions [5]–[9]), mobility (e.g. trajectories [10] and place characterization [11]), transportation (e.g. traffic dynamics [12]), road conditions (e.g. potholes [13], [14]), public health (e.g. mood and behavioral wellbeing [15], [16]), and wireless network monitoring [17], [18].

To the best of our knowledge, all existing MCS works assume a ***cloud-centric approach*** where participant selection algorithms make global decisions based on various coverage and incentive models. This creates ***two significant challenges*** for any large-scale MCS task deployment. First, the existing cloud-centric MCS system aggregates sensor data from users to a centralized cloud server, which bottlenecks scalability of the system to monitor a large target area by an extremely large number of mobile users participating in the MCS tasks. Further, the existing solution localizes the collected sensor data by tracking the real-time location of each participant, which raises serious location privacy concerns. The real-time locations of a participant might be identified by other parties (e.g., the cloud server collecting the data).

On the other hand, as aforementioned, to enable the efficient environment monitoring, MCS applications usually use *spatial-temporal coverage* of the collected sensor data as the objective for data collection. Given a target region split

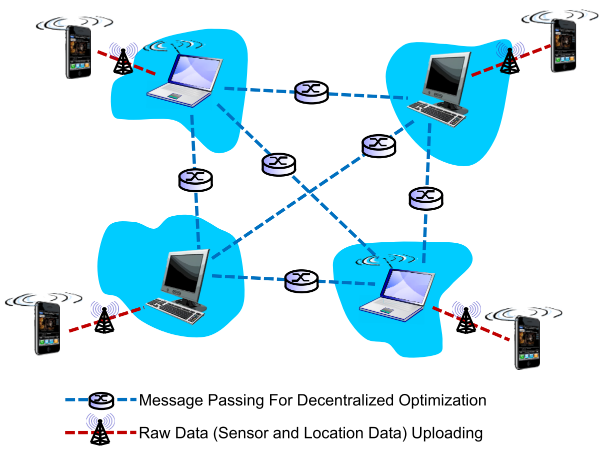


FIGURE 1. ***EdgeSense* : The edge-mediated secure P2P network for crowdsensing.**

into multiple *subareas* and a sensing task divided into a sequence of equal-length *sensing cycles*, the spatial-temporal coverage of a MCS task coverage refers to a type of metrics that characterize the proportion of subareas covered by at least one sensor data in each sensing cycle. For example, [19], [20] proposed to use the *full spatial-temporal coverage* as the criterion of the participants selection for crowdsensing, while [21], [22] studied the *partial spatial-temporal coverage* as the objective of the optimization for budget-constrained participant selection. With the sensor data that partially cover the target area, our previous work [23]–[25] proposed *compressive crowdsensing*, which is capable of recovering the missing sensor data of the uncovered subareas from the collected data. Through compressive crowdsensing, it is possible to accurately monitor the target area with even lower spatial-temporal coverage, thus resulting in reduced cost on participant incentives as fewer participants are selected. Please provide the in-text citation for Fig. 1.

To monitor the environment of a target region, the traditional MCS systems are designed to collect sensor data from mobile users while tracking their real-time locations (e.g., GPS or WiFi). Such sensor/location data aggregation design raises the serious location privacy concerns, as the real-time location or past trajectories of each participant might be identified by other participants or organizations.

To protect the location privacy of participants, we propose to study a novel *Edge-Mediated* Mobile Crowd Sensing system, namely *EdgeSense*, which is designed to obtain the real-time spatial-temporal environmental information of the target area, through message-passing between participants' trusted edge services (for distributed optimization and learning), without aggregating or collecting the sensor/location data from participants.

As shown in Fig. 2, in the proposed system, each participant carries a sensor-equipped mobile device, such as a smartphone or a tablet, to collect spatial-temporal sensor data, e.g., temperature, PM 2.5 (for air quality) and road traffic (GPS with speed information). Each participating mobile device is paired with a trusted edge server/device via a wireless network such as a Cellular or WiFi networks. Participants are free to select their home/office machines or cloudlets that they trust as their trusted edge server. The trusted edge servers will be exposed to public and other participants.

These edge services/devices then organize and operate a *Virtual Secure Peer-to-Peer Network* connecting the MCS participants, as shown in Figure 2. With the virtual secure P2P network, *EdgeSense* recovers the spatial-temporal information of sensor data using Gossip algorithms [26], while each trusted edge server does not share the participant's location and sensor data to others. The proposed system provides two levels of participant-location de-identification:

- **System Level**: The location of each participant can be (re)-identified by addressing his/her mobile devices in wireless communication. Thus, in our system, we separate each participant with their trusted edge server. Only the trusted edge server can address the paired participant through wireless communication. All communications between participants are proxied through the paired trusted edge servers, so as to avoid location information leakages (to other participants) caused by wireless communication (e.g., IP address tracking). In this way, we can de-identify each participant with his/her real-time location (e.g., IP address) in the network of wireless communications; and
- **Data Level**: The traditional MCS mechanism aggregates sensor and location data from each participant, which causes serious location privacy concerns. Instead of aggregating the sensor/location data from each participant (or edge server), *EdgeSense* leverages a decentralized stochastic low-rank approximation procedure to recover the spatial-temporal matrix of sensor data, over Gossip-based optimization. In each round of gossiping, each trusted edge server updates the latent space of the low-rank approximation, with explicit noise, according to their locally stored location/sensor data. This design de-identifies each participant's location from the spatial-temporal data aggregation.

With above two levels of de-identification, *EdgeSense* offers the following features: 1) each participant's locations are not shared to any other participant or any central component; 2) thanks to the low-rank approximation, *EdgeSense* can accurately recover the sensor data of each subarea in each sensing cycle (i.e., full spatial-temporal coverage), while the full coverage of participants' mobility is not required; 3) in each sensing cycle, the number of subareas where are covered by the participants are not known.

The rest of the paper is organized as follows. In the Preliminaries and Problem formulation section, we review the compressive crowdsensing and the matrix factorization approach. Then we introduce the parallelized stochastic gradient descent and present the problem formulation. In the Frameworks and Algorithms section, we propose framework of *EdgeSense* and present the algorithms in details. In the

Experiments section, we evaluate *EdgeSense* on real-world datasets and compare it with baseline algorithms. Finally, in the Conclusion section, we summarize the work in this paper.

II. PRELIMINARIES AND PROBLEM FORMULATION

In this section, we first briefly introduce the previous work on the compressive crowdsensing. Then, we formulate the problem of our research.

A. COMPRESSIVE CROWDSENSING

To derive the target full sensing matrix from partially collected sensing readings, the compressive crowdsensing [23], [24] is commonly considered to be an effective approach, which consists of two parts:

1) SENSING DATA AGGREGATION

Given the target region splitting into a set of subareas (denoted as S) and a set of m participants, in order to obtain the full picture of the target region for each sensing cycle (e.g., the t^{th} cycle), the Compressive Crowdsensing system first collects the sensing data from all participants. Specifically, the subareas covered by the j^{th} participant in the t^{th} sensing cycle ($t \in T$) is denoted as $S_j^t \subset S$. Thus, the overall coverage in the sensing cycle t can be denoted as $S^t = S_1^t \cup S_2^t \cup \ldots \cup S_m^t$. Due to the limited mobility of each participant and limited number of participants involved, the overall coverage can usually include a subset of subareas, i.e., $S^t \subseteq S$. Given the collected sensing data, the compressive crowdsensing system aggregates the data and assigns each covered subarea an unique sensor data value. For example, if multiple sensor data values are collected (from multiple participants) that cover the same subarea in a sensing cycle, the *averaged* value would be used as the value of such subarea in the sensing cycle. In this way, each subarea $s \in S^t$ has been covered with one sensor data value, through data aggregation, and the compressive crowdsensing system needs to infer the missing sensor data of the subareas in $S \backslash S^t$ to obtain the sensor data of the whole target area.

2) MISSING DATA INFERENCE

Given the aggregated sensor data of the covered subareas (S^t), there exists a wide-range of inferring techniques to infer the missing data of the uncovered subareas, such as expectation maximization [27] and singular spectrum analysis [28]. One of the powerful approach is the spatial-temporal compressive sensing [29], [30]. The essential idea of this approach is based on the nonnegative matrix factorization (NMF) [31], [32]. Given the aggregated sensor data of recent sensing cycles (the number of recent sensing cycles used for NMF is denoted as w), this approach first sorts the subareas using their indices from $1 \ldots$ to $|S|$, then maps the data into a $|S| \times w$ matrix denote as R, where the element $R_{a,t}$ ($1 \leq a \leq |S|$ and $1 \leq t \leq w$) refers to the aggregated sensing value of the a^{th} subarea and t^{th} sensing cycle (in the window). To recover the missing values in R, this approach obtains two non-negative ***matrix factors*** $P \in \mathbb{R}^{|S| \times l}$ and $Q \in \mathbb{R}^{l \times w}$ such that $R \approx PQ$, through NMF, where l stands for the *Size of Latent Space* of NMF.

Typically, there are four key factors affecting the performance of the compressive community sensing: (1) *The Number of Subareas* that each participant covers in each sensing cycle; (2) *The Number of Participants (m)* which, together with the number of subareas per participant, can determine the coverage of collected sensor data; (3) *The Size of Windows (w)* that refers to the number of past sensing cycles used for matrix recovery (i.e., the width of the matrix for matrix completion); (4) *The Size of Latent Space (l)* that determines the rank of matrices for low-rank matrix recovery/completion.

B. PROBLEM FORMULATION

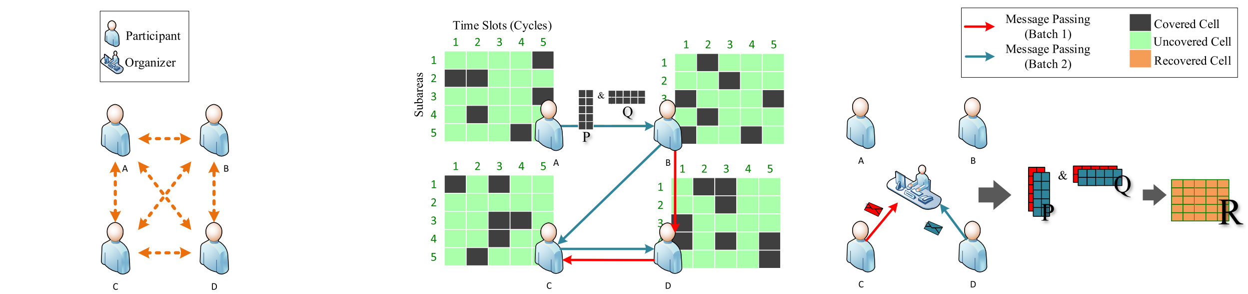
Given a set of participants, where each participant's mobile device stores the raw sensor data locally (without raw data sharing), our proposed work intends to recover the sensing data of the target area while assuming that the organizer is not allowed to aggregate the sensor data from any participants. Specifically, we make following assumptions:

- For all the sensing cycles in T and subareas in S, there exists an unknown spatial-temporal sensor data matrix R^* ($R^* \in \mathbb{R}^{|S| \times T}$), where each element $R^*_{a',t'}$ ($1 \leq a' \leq |S|$ and $1 \leq t' \leq |T|$) refers to the real value of sensor data in the corresponding subarea a' and sensing cycle t'.
- In each sensing cycle (e.g., the t^{th} cycle), each participant (e.g., the j^{th} participant) covers a subset of subareas (i.e., $S_j^t \subseteq S$) in the target area. Thus, all the collected sensor data from the 1^{st} to the t^{th} sensing cycle of the j^{th} participant can be represented as a matrix $R^j \in \mathbb{R}^{|S| \times t}$, where each element refers to the value of the sensor data collected in the corresponding subarea and cycle. Note that, to protect the location privacy, R^j is not known by the organizer.
- We denote the value of the sensor data collected by the j^{th} participant in sensing cycle t at subarea a as $R^j_{a,t}$. Each sensor datum obtained is assumed to be the true value with (unknown) random noise, i.e., $R^j_{a,t} = R^*_{a,t} + \varepsilon^j_{a,t}$. For any two participants (i.e., the j^{th} and k^{th} participants), they might cover the same subarea (say, $a_j^t \cap S_k^t \neq \emptyset$ is possible), but are with *different* sensor data value obtained, due to the noise.

Our problem is that, in each sensing cycle t, with R^j ($1 \leq j \leq N$) locally stored on each participant's device, there needs to estimate $\widehat{R}_{a,t}$ to

$$minimize \sum_{a=1}^{|S|} (\widehat{R}_{a,t} - R^*_{a,t})^2 \; for \; 1 \leq t \leq T,$$

while ensuring that the organizer is prohibited to aggregate R^j from any participant and the raw sensor/location data sharing is not allowed between the participants.



(a) **Phase I:** Secured P2P Network Establishment and Initialization

(b) **Phase II:** Distributed Compressive Crowdsensing over Secured P2P

(c) **Phase III:** Spatial-Temporal Data Recovery

FIGURE 2. Overall Framework of *EdgeSense*. (a) Phase I: Secured P2P Network Establishment and Initialization. (b) Phase II: Distributed Compressive Crowdsensing over Secured P2P. (c) Phase III: Spatial-Temporal Data Recovery.

III. FRAMEWORK AND ALGORITHMS

In this section, we present the proposed framework of *EdgeSense* and the underlying algorithms. Specifically, we introduce a novel *Decentralized Spatial-Temporal Compressive Sensing* framework based on *Parallelized Stochastic Gradient Descent*.

A. FRAMEWORK DESIGN

Before elaborating the proposed framework and algorithms, we make the following settings: (1) In order to simulate a secure peer-to-peer network over the participants, we define a set of participants, where these participants can receive or send messages (factor matrices) to each other trustfully and randomly; (2) When passing the message between two participants, the *receiver* can not send the updated matrix factors back to the *sender*, while the *sender* can easily recover the *receiver's* local sensing data by recalculating the return messages; (3) The organizer can only receive or access the related message when the updates (message passing) are finished. In this way, the private information such as real-time locations of the participants in each sensing cycle can be protected from the organizer.

The overall framework of *EdgeSense* consists of the following three phases (as illustrated in Figure 2):

1) PHASE I: SECURE P2P NETWORK ESTABLISHMENT AND INITIALIZATION

Prior to initializing the *batch* on the organizer, we first establish a secure peer-to-peer (P2P) network among m participants, while all the collected sensor data on the j^{th} participant are mapped to a local data matrix R^j. Then, as shown in Algorithm 1, *EdgeSense* randomly picks a set of participants which is the *batch* (denoted as the set L with size N) from the secure network of m participants. Next, given the target data matrix $R \in \mathbb{R}^{|S|\times w}$, *EdgeSense* extracts the row and column number of R to construct the initial matrix factors $\hat{P}$ and $\hat{Q}$ on the organizer. Specifically, $\hat{P}_j$ is generated by a $|S| \times l$ Gaussian Random Matrix on the j^{th} participant. Similarly, $\hat{Q}_j$ is generated by a $l \times w$ Gaussian Random Matrix on the same j^{th} participant. To avoid the aforementioned message transferring back between two participants, we initialize a counter i to record passing times (iterations) among participants and set j_p to mark the last participant's index, where the (i, j_p) will be transferred along with the updated matrix factors so that the participant who receives the message can randomly select the next one excluding participant j_p. When the initialization ends, each participant (I_j) in the predefined set L (*batch*) will be assigned a pair of starting matrix factors $\hat{P}_j$ and $\hat{Q}_j$.

Algorithm 1 Initializing Batch and Matrix Factors ($\hat{P}$, $\hat{Q}$) on Organizer

---

Data:
$R_{|S|\times w}$ — the target data matrix
Parameter:
/* Subareas covered by per participant */
$|S|$ — the maximum numbers of subareas
w — the size of windows
l — the size of latent space
begin
  /* Predefine a set of participants */
  Randomly Draw N *Participants* into **Set** L
  /* $L = \{I_1, I_2, \ldots, I_N\}$ */
  for *each* $I_j \in L$ **do**
    /* Initialize matrix factors P, Q on I_j */
    $\hat{P}_j \leftarrow |S| \times l$ Gaussian Random Matrix
    $\hat{Q}_j \leftarrow l \times w$ Gaussian Random Matrix
    /* Initialize the counter and the previous participant index */
    SEND ($\hat{P}_j$, $\hat{Q}_j$, **0**, **null**) to **L**;
  end
end

---

2) PHASE II: DISTRIBUTED COMPRESSIVE COMMUNITY SENSING VIA PARALLELIZED LOW-RANK APPROXIMATION

Given the mapped local data matrix R^j on j^{th} participant, *EdgeSense* intends to approximate the optimal estimation of matrix factors $\hat{P}_j$ and $\hat{Q}_j$ via parallelized stochastic gradient descent on top of non-negative matrix factorization algorithm. Specifically, the initialized ($\hat{P}_j$, $\hat{Q}_j$, **0**, **null**) has been allocated on the j^{th} participant, where **0** refers to the fact that no update has been executed and "**null**" refers to there is no previous participant (coming from the organizer) which has updated the matrix factors (the index of previous participant is empty). Then the algorithm processes the updating task on each participant from the predefined *batch* (L) in parallel.

Suppose two dense matrix factors are $P \in \mathbb{R}^{|S|\times l}$ and $Q \in \mathbb{R}^{l\times w}$, the target minimization loss function over m participants through parallelized stochastic gradient descent is as follow:

$$\hat{P}, \hat{Q} \leftarrow \underset{P\in\mathbb{R}^{|S|\times l}, Q\in\mathbb{R}^{l\times w}}{argmin} \left\{ \frac{1}{m} \sum_{j=1}^{m} \left\| F_j \circ (R^j - PQ) \right\|_F^2 + \lambda_P \|P\|_F^2 + \lambda_Q \|Q\|_F^2 \right\}, \quad (1)$$

where l is the size of latent space, "$\circ$" means element-wise matrix multiplication, $\|\cdot\|_F$ is the Frobenius norm, λ_P and λ_Q are regularization parameters. Particularly, parallelly starting on each participant I_j, Algorithm 2 first receives the input $(\hat{P}_j, \hat{Q}_j)$ from the last involved participant in the secure network (or initialized from the organizer in the first run). Next it updates the $(\hat{P}_j, \hat{Q}_j)$ using the mapped local data matrix R^j with the missing-value filter matrix F_j, and randomly picks up the next participant except the previous one (j_p) from the secure participants network and sends the updated $(\hat{P}_j, \hat{Q}_j)$ to this chosen participant. The matrix F_j is a matrix filling with 0 (missing) and 1 (collected) which can set the missing elements in matrix R^j to zero by the element-wise multiplication. We mainly use it to prevent the missing value in the local data matrix R^j from affecting the gradient updating in $(\hat{P}_j, \hat{Q}_j)$. In addition, we leverage the *Truncate()* function, where the negative values in matrix factors $(\hat{P}_j, \hat{Q}_j)$ will be set to zero, then ensuring the non-negativeness of $(\hat{P}_j, \hat{Q}_j)$ when finishing each update.

Algorithm 2 keeps picking up the next participant for updating, until the times of updates i exceeds the maximal number of updates, or the updating process converges (i.e., $max \left\{ |g_p|_\infty, |g_q|_\infty \right\} \leq \Delta_{max}$). Similar procedures are starting on each participant I_j and the related matrix factors keep updating independently. Once the updating process completes on each participant, Algorithm 2 sends $(\hat{P}_j, \hat{Q}_j)$ where $j = 1, 2, \ldots, N$ to the organizer. When all the parallel processes are finished, the organizer has received N pairs of the estimated $(\hat{P}, \hat{Q})$ for recovery of the target data matrix.

3) PHASE III: SPATIAL-TEMPORAL DATA RECOVERY

As we have introduced in the Preliminaries, the organizer can recover the target data matrix $\hat{R}$ based on the optimal estimated matrix factors $(\hat{P}, \hat{Q})$.

Given the received matrix factors $(\hat{P}_j, \hat{Q}_j)$ which are from the *batch*, Algorithm 3 first separately average the $\hat{P}$ and $\hat{Q}$ from $j = 1$ to N. Then, to recover the target data matrix, the algorithm multiplies the averaged matrix factors $(\bar{P}, \bar{Q})$ and obtains the well-estimated target data matrix $\hat{R}$.

Algorithm 2 Parallelized Optimization on the j^{th} Participant

Data:
R^j — the local data matrix on the j^{th} participant
F_j — the filter matrix on the j^{th} participant
Parameter:
i — the number iterations
j_p, j — the index of previous and current participant
η — step size
Δ_{min} — the minimum allowed perturbation
t_{max} — the maximum number of allowed updates
λ_P, λ_Q — regularization parameter on P and Q matrices

```
begin
    /* On receiving the message from the previous
       participant                                        */
    RECEIVE (P̂_j, Q̂_j, t, j_p)
    /* Noting that "A ∘ B" means element-wise matrix
       multiplication                                     */
    g_p ← (F_j ∘ (R^j − P̂_j Q̂_j)) Q̂_j^T − λ_P · P̂_j
    g_q ← P̂_j^T (F_j ∘ (R^j − P̂_j Q̂_j)) − λ_Q · Q̂_j
    P̂_j ← P̂_j − η · g_p
    Q̂_j ← Q̂_j − η · g_q
    /* Set the negative elements to zero                 */
    P̂_j, Q̂_j ← Truncate(P̂_j, Q̂_j)
    i ← i + 1
    /* Checking convergence conditions                   */
    Δ = max { |g_p|_∞ , |g_q|_∞ }
    if Δ ≥ Δ_max ANDi ≤ t_max then
        /* Not converged, continuing the algorithm       */
        j_next ← Draw a random number from 1 to m
        except j_p;
        SEND (P̂_j, Q̂_j, i, j) to the j_next^th Participant;
    else
        /* Converged, find out the optimal estimates
           */
        SEND (P̂_j, Q̂_j) to the Organizer;
    end
end
```

Algorithm 3 Mobile Sensing Recovery on the Organizer

Data:
$\hat{P}_j, \hat{Q}_j$ — the received matrix factors from the *batch*

```
begin
    /* Average all P̂_j, Q̂_j on organizer                */
    P̄ ← (1/N) Σ_{j=1}^N P̂_j
    Q̄ ← (1/N) Σ_{j=1}^N Q̂_j
    /* Recover the target overall data matrix           */
    R̂ ← P̄ Q̄
end
```

4) ALGORITHM ANALYSIS

In this section, we brief the analytical results of the proposed algorithms.

Given the overall set of subareas (S), the size of the latent space (l), the size of the windows (w), in each iteration, N participants in the system would send out messages, while each participant sends a $|S| \times l$ matrix and a $l \times w$ matrix (i.e., P and Q matrices). In this way, the system-wide communication complexity in the worst-case (after the completion of t_{max} iterations of message-passing) should be $\mathcal{O}\left((|S| \cdot l + l \cdot w) \cdot t_{max} \cdot N\right)$.

Suppose P^* and Q^* are the optimal solutions of the problem listed in Eq. 1, while $\bar{P}$ and $\bar{Q}$ (appeared in Algorithm 3) are two approximation results obtained by our algorithm. According to [33], the approximation error of $||P^* - \bar{P}||_F \to 0$ and $||Q^* - \bar{Q}||_F \to 0$, when $t_{max \to +\infty}$ and N is sufficiently large. Note that with a larger N, the proposed algorithm can achieve a faster rate of convergence of the approximation error with increasing t_{max}. For more theoretical analysis, please refer to [33].

IV. EXPERIMENTS

In order to evaluate the *EdgeSense* algorithm, we use the *Temperature (TEMP)* and *PM 2.5 air quality (PM25)* dataset, where the Experimental Setup section will cover all the settings and assumptions. Based on the above dataset, we first introduce the baseline algorithms which are commonly used in sensor data recovery. Specifically, the baseline algorithms adopt the ***matrix completion*** method and leverage the ***centralized computing*** patterns to recover the target sensing data. Then, we compare the performance of *EdgeSense* with baseline algorithms on two real-world datasets.

A. EXPERIMENTAL SETUP

For *TEMP* [1] and *PM25* [2] datasets, the sensing value of temperature (°C)/PM2.5 (air quality index) are located on each participant's mobile sensor in varying time slots (sensing cycle) and at different subareas. In details, the *TEMP* dataset contains the temperature readings in 57 cells (Subareas) and each sensing cycle lasts for 30 minutes. The *PM25* dataset includes the PM2.5 air quality values on 36 stations (Subareas) with the same sensing cycle.

In order to simulate the settings of the centralized computing patterns, we aggregate the collected sensing data from each participant. In details, we follow the aforementioned three phases to set the appropriate value of four key factors: *the Number of Participants (m), the Number of Subareas that each participant covers in each sensing cycle, the Size of Windows (w)* and *the Size of Latent Space (l)*. Note that each participant can sense the temperature/PM2.5 at a subset of subarea. Specifically, we use the maximum number of subareas s ($1 \leq s \leq |S|$) in the experiments, assuming the participant can cover no more than s subareas. To simulate the scenario that each participant can cover various number of subareas, the actual number of subareas covered by the participant will follow the discrete uniform distribution U$\{1, s\}$.

B. BASELINE ALGORITHMS

In this section, we briefly introduce three baseline algorithms, where their advantages and drawbacks are listed as compared to *EdgeSense* algorithm.

- ***Spatio-Temporal Compressive Sensing (STCS) –*** *STCS* [23], [30] leverages the sparsity regularized matrix factorization to fill in the missing values in a certain matrix accounting for spatial-temporal properties. Based on the low-rank nature of real-world data matrices, STCS first exploits global and subarea structures in the data metrics. Then, it recovers the original matrices through matrix factorization under spatial-temporal constraints. Indeed, STCS advances ideas from compressive sensing and provides a highly effective (high accuracy and robustness) approach to solve the problem of missing data interpolation.
- ***Robust Principle Component Analysis (RPCA)*** and ***Truncated Singular Value Decomposition (TSVD) –*** *RPCA* [34] is derived from a widely used statistical procedure of principal component analysis (PCA), where RPCA performs well on solving the problem of matrices recovering. With respect to a mass of missing observations, RPCA aims to recover a low-rank matrix through random sampling techniques [35]. *TSVD* [36] is also commonly used to approximate a low-rank matrix. Different from the traditional singular value decomposition, TSVD sets all but the first k largest singular values equal to zero and use only the first k columns of the corresponding unitary matrices. With the optimality property, this method provides an efficient way to recover the target sensing matrix.

C. EXPERIMENTAL RESULTS

In this section, we report the performance of *EdgeSense* and other three baselines on *TEMP* and *PM25* datasets. Specifically, we use the *Absolute Error*, which is the averaged element-wise difference $\left(\sum_{a=1}^{|S|}\sum_{t=1}^{|T|}\left|\widehat{R}_{a,t} - R^*_{a,t}\right|/(|S| \cdot |T|)\right)$ between the recovered matrix ($\widehat{R}$) and the original data matrix (R^*), as the indicator of the performance.

1) TEMP DATASETS

First, we present a comparison of algorithms with the settings of the maximum number of subareas (covered by each participant) ranging from 1 to 5 in Fig. 3. Due to the overall better performances of *EdgeSense* and STCS, we present the entire comparison in (a) and only compare *EdgeSense* with STCS in other three settings (the same in Figures 4, 5 and 6 as well). Specifically, in Fig. 3(a), 10 participants are involved. Then we vary the number of participants from 10 to 40 in the increment of 10 in Figs. 3(b), (c) and (d). We observe that the error is around 0.2 to 0.45 with varying maximum number of subareas from 1 to 5. It is noteworthy that *EdgeSense* can compete to STCS under these settings.

Second, we also compare *EdgeSense* with baseline algorithms by varying the number of participants in the secure P2P network. In Fig. 4(a), the maximum number of subareas is 1. Then we increase it from 1 to 4 in the increment of 1 in Figs. 4(b), (c) and (d). In each comparison between

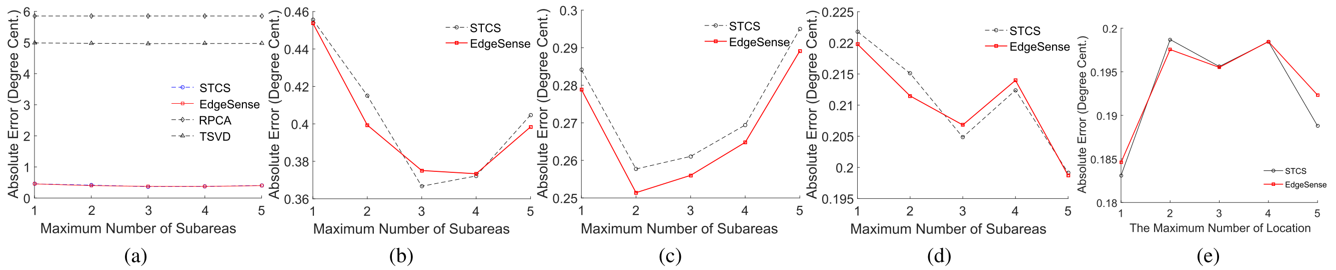


FIGURE 3. On *TEMP* Datasets, we compare the performance of increasing *Max Number of Subareas* (*s*) from 1 to 5 with different number of participants in each cycle. (We remove the significantly poor algorithms in (b), (c), (d) and (e), the same as following experiments.)

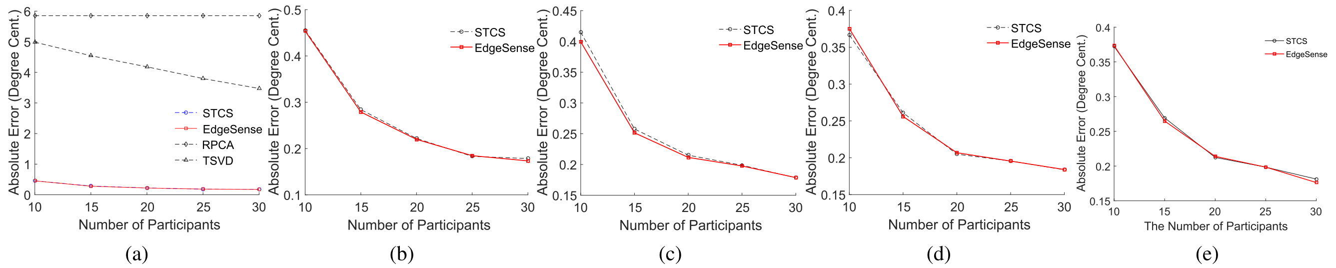


FIGURE 4. On *TEMP* Datasets, we compare the performance of increasing *Number of Participants* (*m*) from 10 to 30 with different max number of subareas in each cycle. (a) $s = 1$ (Maximum Number of Subareas). (b) $s = 1$ (Maximum Number of Subareas). (c) $s = 2$ (Maximum Number of Subareas). (d) $s = 3$ (Maximum Number of Subareas). (e) $s = 4$ (Maximum Number of Subareas).

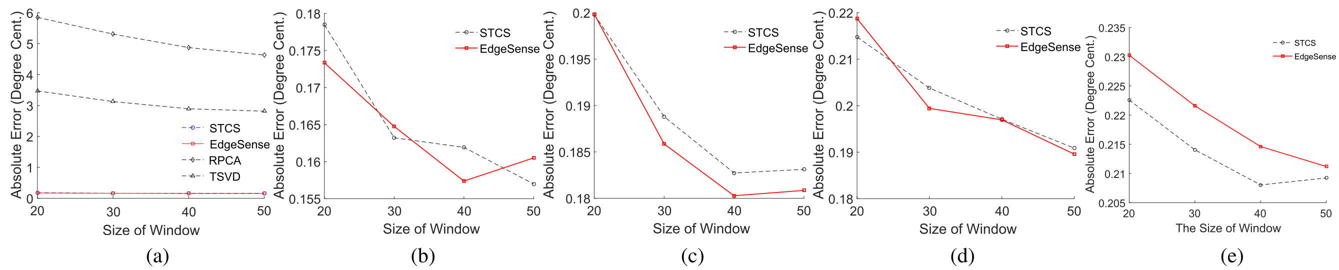


FIGURE 5. On *TEMP* Datasets, we compare the performance with the increasing *Size of Window* (*w*) from 20 to 50 with different size of latent space in each cycle. (a) $l = 2$ (Size of Latent Space). (b) $l = 2$ (Size of Latent Space). (c) $l = 4$ (Size of Latent Space). (d) $l = 6$ (Size of Latent Space). (e) $l = 8$ (Size of Latent Space).

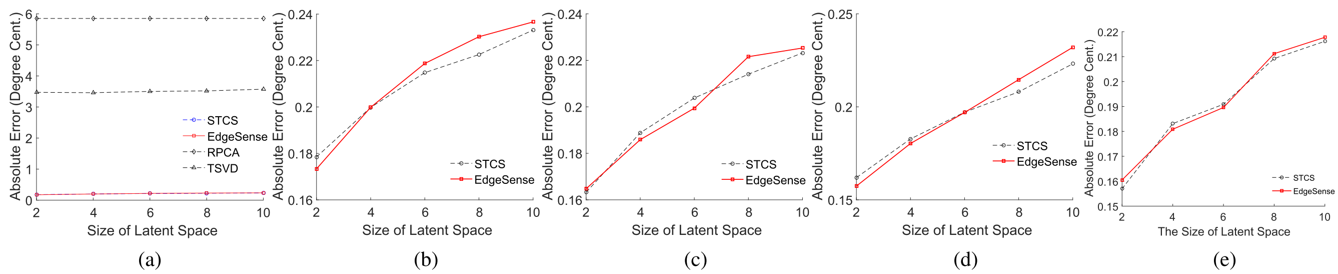


FIGURE 6. On *TEMP* Datasets, we compare the performance with the increasing *Size of Latent Space* (*l*) from 2 to 10 with different size of windows in each cycle. (a) $w = 20$ (Size of Windows). (b) $w = 20$ (Size of Windows). (c) $w = 30$ (Size of Windows). (d) $w = 40$ (Size of Windows). (e) $w = 50$ (Size of Windows).

EdgeSense and STCS, the error decreases when the number of participants increases for both of these two algorithms. This demonstrates that the larger group of participants can improve the performance of the matrix recovery, where intuitively the participants can cover more subareas and sensing cycles. Similar to the previous setting, *EdgeSense* can approximate the performance of STCS as well.

Further, we alter the values of two aforementioned key factors, such as *Size of Windows* and *Size of Latent Space*, to observe the variation of the error. Fig. 5 shows that the error decreases when the window size increases from 20 to 50. Note that for each size of latent space in Figs. 5(b), (c) and (d), the decreasing trends of the error are almost the same and the performance of *EdgeSense* still can compete with STCS. Fig. 6 exhibits that the error increases when the size of the latent space increases from 2 to 10. Thus, for *TEMP* datasets, the small size of latent space can better approximate the original data matrix when it is low-rank. Thus the performance of *EdgeSense* is still competitive to STCS, as shown in Figs. 6(b), (c) and (d).

TABLE 1. On PM25 Datasets, we compare the (*Absolute Error*) with Varying *Number of Participants* (*m*) and *Size of Windows* (*w*).

	Number of Participants (m)			Size of Windows (w)		
	10	20	30	20	30	40
EdgeSense	15.563	11.686	9.561	8.844	10.232	12.028
STCS	15.185	11.864	9.353	8.517	10.090	11.955

2) PM25 DATASETS

We conduct experiments with similar settings as *TEMP* datasets. Since the performances of RPCA and TSVD are still not as good as the other two algorithms, we only present the comparison between the proposed *EdgeSense* and STCS here. Specifically, in Table 1, we list the *Absolute Error* of these two algorithms with varying number of participants (m) and the window size (w). When the number of participants increases, the error is decreasing intuitively. On the contrary, the error increases with increased size of the window. However, *EdgeSense* performs comparably to STCS, sometimes even better (e.g., for $m = 20$). In Table 2, we show the performance with varying size of latent space and the number of subareas covered by each participant. The results reveal that the number of subareas does not affect the error significantly, while with the larger latent space the error is smaller with *PM25* datasets. Under these two settings, the performance of *EdgeSense* can still compete with STCS. Note that for each setting, we present the performance on the varying factor while keeping the other factor at optimal value. Also it is worth noting that the overall error is small on the average (10 with PM2.5 index ranging from 1 to 500) in both of *EdgeSense* and STCS.

TABLE 2. on PM25 Datasets, we compare the *Absolute Error* with Varying *Size of Latent Space* (*l*) and *Maximum Number of Subareas* (*s*).

	Size of Latent Space (l)			Maximum Number of Subareas (s)		
	2	4	6	1	2	3
EdgeSense	11.777	9.561	8.945	8.166	8.945	8.844
STCS	11.518	9.353	8.719	8.220	8.719	8.516

3) SUMMARY

With two real-world datasets, we compared the proposed *EdgeSense* with the baseline algorithms STCS, RPCA and TSVD. For both of the datasets, *EdgeSense* significantly outperforms RPCA and TSVD in most cases. Moreover, compared to the centralized algorithm STCS, *EdgeSense* also presents its competitiveness, with a low approximation error (0.2° in city-wide temperature and 10 units of PM2.5 index in urban air quality). Even in some settings, the *EdgeSense* has a lower approximation error than STCS, which demonstrates the superiority of *EdgeSense*.

V. CONCLUSION

In this paper, we proposed *EdgeSense* – a novel crowdsensing paradigm. *EdgeSense* is designed to extract the environmental information in each subarea, without aggregating sensor and location data from the participants who partially cover the monitored area. On top of a secure peer-to-peer network over the participants, *EdgeSense* proposes a novel Decentralized Spatial-temporal Compressive Sensing framework based on Parallelized Stochastic Gradient Descent. Specifically, through learning the low-rank matrix structure via distributed optimization, *EdgeSense* approximates the value of sensor data in each subarea (both covered and uncovered) for each sensing cycle using the sensor data that locally stored in each participant's mobile device. According to the theoretical analysis on the parallelized stochastic gradient decent [33], *EdgeSense* is capable of recovering the Spatial-Temporal information with bounded approximation error using the P2P communications of controllable complexity. The experiment results based on real-world datasets demonstrates that *EdgeSense* has low approximation error and performs comparably to (sometimes even better than) state-of-the-art algorithms based on the data aggregation and centralized computation.

REFERENCES

[1] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "Sensorscope: Application-specific sensor network for environmental monitoring," *ACM Trans. Sensor Netw.*, vol. 6, no. 2, 2010, Art. no. 17.

[2] 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*, 2013, pp. 1436–1444.

[3] R. K. 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.

[4] 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, Sep. 2015, Art. no. 7, doi: 10.1145/2794400.

[5] N. Maisonneuve, M. Stevens, and B. Ochab, "Participatory noise pollution monitoring using mobile phones," *Inf. Polity*, vol. 15, no. 1, pp. 51–71, Apr. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1858974.1858981

[6] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: An end-to-end participatory urban noise mapping system," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, New York, NY, USA, 2010, pp. 105–116, doi: 10.1145/1791212.1791226.

[7] L. Ruge, B. Altakrouri, and A. Schrader, "SoundOfTheCity—Continuous noise monitoring for a healthy city," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PERCOM)*, Mar. 2013, pp. 670–675.

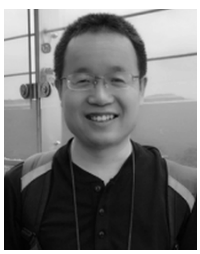
[8] P. Dutta *et al.*, "Common sense: Participatory urban sensing using a network of handheld air quality monitors," in *Proc. 7th ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, New York, NY, USA, 2009, pp. 349–350, doi: 10.1145/1644038.1644095.

[9] 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)*, New York, NY, USA, 2013, pp. 1436–1444, doi: 10.1145/2487575.2488188.

[10] Y. Chon, S. Kim, S. Lee, D. Kim, Y. Kim, and H. Cha, "Sensing wifi packets in the air: Practicality and implications in urban mobility monitoring," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. (UbiComp)*, New York, NY, USA, 2014, pp. 189–200, doi: 10.1145/2632048.2636066.
[11] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao, "Automatically characterizing places with opportunistic crowdsensing using smartphones," in *Proc. ACM Conf. Ubiquitous Comput. (UbiComp)*, New York, NY, USA, 2012, pp. 481–490, doi: 10.1145/2370216.2370288.
[12] F. Calabrese, M. Colonna, P. Lovisolo, D. Parata, and C. Ratti, "Real-time urban monitoring using cell phones: A case study in Rome," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 141–151, Mar. 2011.
[13] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," in *Proc. 6th ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, New York, NY, USA, 2008, pp. 323–336, doi: 10.1145/1460412.1460444.
[14] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: Using a mobile sensor network for road surface monitoring," in *Proc. 6th Int. Conf. Mobile Syst., Appl. Services (MobiSys)*, New York, NY, USA, 2008, pp. 29–39, doi: 10.1145/1378600.1378605.
[15] N. Lathia, V. Pejovic, K. K. Rachuri, C. Mascolo, M. Musolesi, and P. J. Rentfrow, "Smartphones for large-scale behavior change interventions," *IEEE Pervasive Comput.*, vol. 12, no. 3, pp. 66–73, Jul. 2013.
[16] R. Wang *et al.*, "Studentlife: Assessing mental health, academic performance and behavioral trends of college students using smartphones," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. (UbiComp)*, New York, NY, USA, 2014, pp. 3–14, doi: 10.1145/2632048.2632054.
[17] S. Sen, J. Yoon, J. Hare, J. Ormont, and S. Banerjee, "Can they hear me now?: A case for a client-assisted approach to monitoring wide-area wireless networks," in *Proc. ACM SIGCOMM Conf. Internet Meas. Conf. (IMC)*, New York, NY, USA, 2011, pp. 99–116, doi: 10.1145/2068816.2068827.
[18] A. Nikravesh, D. R. Choffnes, E. Katz-Bassett, Z. M. Mao, and M. Welsh, *Mobile Network Performance From User Devices: A Longitudinal, Multidimensional Analysis*. Cham, Switzerland: Springer, 2014, pp. 12–22, doi: 10.1007/978-3-319-04918-2_2.
[19] H. Xiong, D. Zhang, L. Wang, and H. Chaouchi, "EMC^3: Energy-efficient data transfer in mobile crowdsensing under full coverage constraint," *IEEE Trans. Mobile Comput.*, vol. 14, no. 7, pp. 1355–1368, Jul. 2015.
[20] X. Sheng, J. Tang, and W. Zhang, "Energy-efficient collaborative sensing with mobile phones," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1916–1924.
[21] D. Zhang, H. Xiong, L. Wang, and G. Chen, "Crowdrecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2014, pp. 703–714.
[22] S. Hachem, A. Pathak, and V. Issarny, "Probabilistic registration for large-scale mobile participatory sensing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2013, pp. 132–140.
[23] L. Wang *et al.*, "CCS-TA: Quality-guaranteed online task allocation in compressive crowdsensing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2015, pp. 683–694.
[24] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, and A. M'hamed, "Sparse mobile crowdsensing: Challenges and opportunities," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 161–167, Jul. 2016.
[25] L. Wang *et al.*, "SPACE-TA: Cost-effective task allocation exploiting intradata and interdata correlations in sparse crowdsensing," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 2, 2018, Art. no. 20.
[26] D. Shah, *Gossip Algorithms*. Boston, MA, USA: Now, 2009.
[27] T. Schneider, "Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values," *J. Climate*, vol. 14, no. 5, pp. 853–871, 2001.
[28] D. Kondrashov and M. Ghil, "Spatio-temporal filling of missing points in geophysical data sets," *Nonlinear Processes Geophys.*, vol. 13, no. 2, pp. 151–159, 2006.
[29] 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.
[30] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and Internet traffic matrices," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 267–278, 2009.
[31] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.
[32] Y. Mao and L. K. Saul, "Modeling distances in large-scale networks by matrix factorization," in *Proc. 4th ACM SIGCOMM Conf. Internet Meas.*, 2004, pp. 278–287.
[33] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola, "Parallelized stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 2595–2603.
[34] H. Gao, J.-F. Cai, Z. Shen, and H. Zhao, "Robust principal component analysis-based four-dimensional computed tomography," *Phys. Med. Biol.*, vol. 56, no. 11, p. 3181, 2011.
[35] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[36] S. Isam, I. Kanaras, and I. Darwazeh, "A Truncated SVD approach for fixed complexity spectrally efficient FDM receivers," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2011, pp. 1584–1589.

SIJIA YANG received the B.Eng. degree from Zhejiang Gongshang University, Hangzhou, China, in 2011, and the M.Sc. degree in information and communications technology business management from the Télécom Ecole de Management, Paris, France, in 2015. She is currently pursuing the Ph.D. degree in cyberspace security with the Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include cyber security, data analytics, and machine learning.

JIANG BIAN received the B.Eng. degree in logistics systems engineering from the Huazhong University of Science and Technology, and the M.Sc. degree in industrial systems engineering from the University of Florida. He is currently pursuing the Ph.D. degree with the University of Central Florida, under the supervision of Prof. Z. Guo (co-supervision by Prof. H. Xiong). His current research interests include human-subject data learning, ubiquitous computing, and intelligent systems.



LICHENG WANG received the B.S. degree in engineering from Northwest Normal University, Lanzhou, China, in 1995, the M.S. degree in mathematics from Nanjing University, Nanjing, China, in 2001, and the Ph.D. degree in engineering from Shanghai Jiao Tong University, Shanghai, China, in 2007. He is currently an Associate Professor with the Beijing University of Posts and Telecommunications, Beijing, China. His current research interests include cryptography, blockchain, and future Internet architecture.

HAOJIN ZHU received the B.Sc. degree from Wuhan University, in 2002, the M.Sc. degree from Shanghai Jiao Tong University, China, in 2005, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, in 2009.

He is currently a Full Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. He is leading the Network Security and Privacy Protection (NSEC) Laboratory. He has published more than 50 journals, including: JSAC, TDSC, TPDS, TMC, TIFS, TWC, and TVT, Top 100 most cited Chinese Papers published in international journals of 2014, a Supervisor of Shanghai Excellent Master Thesis, and more than 60 international conference papers, including the IEEE SECURITY AND PRIVACY, ACM Conference on Computer and Communications Security, ACM MOBICOM, Network and Distributed System Security, ACM MOBIHOC, IEEE Conference on Computer Communications (IEEE INFOCOM), and IEEE International Conference on Distributed Computing Systems (IEEE ICDCS). He is a member of ACM. He received the IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award in 2014 for the contribution to wireless network security and privacy, best paper awards of the IEEE ICC 2007 and Chinacom 2008, and the Best Paper Award runner up for Globecom 2014 and WASA 2017.

YANJIE FU received the B.E. degree from the University of Science and Technology of China, in 2008, the M.E. degree from the Chinese Academy of Sciences, in 2011, and the Ph.D. degree from the Rutgers, The State University of New Jersey, in 2016.

He has been an Assistant Professor with the Department of Computer Science, Missouri University of Science and Technology (University of Missouri-Rolla), since 2016. He has research experience in industry research labs, such as Microsoft Research Asia, Huawei Research Labs, and the IBM Thomas J. Watson Research Center. He has published prolifically in refereed journals and conference proceedings, such as the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (IEEE TKDE), *ACM Transactions on Knowledge Discovery From Data* (TKDD), the IEEE TRANSACTIONS ON MOBILE COMPUTING (IEEE TMC), *ACM Transactions on Intelligent Systems and Technology* (ACM TIST), ACM SIGKDD, AAAI, IJCAI, and IEEE ICDM. His research interests include data mining and big data analytics.

HAOYI XIONG received the B.Eng. degree in electrical engineering from the Huazhong University of Science and Technology, in 2009, the M.Sc. degree in information technology from The Hong Kong University of Science and Technology, in 2010, and the Ph.D. degree in computer science from Telecom SudParis jointly with Pierre and Marie Curie University, Paris, France, in 2015.