

Outlier-Concerned Data Completion Exploiting Intra- and Inter-Data Correlations in Sparse CrowdSensing

En Wang^{ID}, Member, IEEE, Mijia Zhang^{ID}, Student Member, IEEE, Wenbin Liu^{ID}, Member, IEEE,
Haoyi Xiong, Senior Member, IEEE, Bo Yang^{ID}, Yongjian Yang^{ID}, and Jie Wu^{ID}, Fellow, IEEE

Abstract—Mobile CrowdSensing (MCS) is a popular data collection paradigm which usually faces the problem of sparse sensed data because of the limited sensing cost. In order to address the situation of sparse data, sparse MCS recruits users to sense important areas and infers completed data by data completion, which is crucial in sparse MCS for urban sensing applications (e.g. enhancing data expression, improving urban analysis, guiding city planning, etc.) To achieve accurate completion results, previous methods usually utilize the universal similarity and conventional tendency while incorporating only a single dataset to infer the full map. However, in real-world scenarios, there may exist many kinds of data (inter-data), that could help to complement each other. Moreover, for each kind of data (intra-data), there usually exist a few but important outliers caused by the special events (e.g., parking peak, traffic congestion, or festival parade), which may behave in a different way as the statistical patterns. These outliers cannot be ignored, while it is difficult to detect and recover them in data completion because of the following challenges: 1) the infrequency and unpredictability of outliers' occurrence, 2) the large deviations against the means compared to normal values, and 3) the complex spatiotemporal relations among inter-data. To this end, focusing on spatiotemporal data with both intra- and inter-data correlations, we propose a matrix completion method that takes outliers' effects into consideration and exploits both intra- and inter-data correlations for enhancing performance. Specifically, we first conduct the Deep Matrix Factorization (DMF) with multiple auxiliary Neural Networks, which named Stacked Deep Matrix Factorization (SDMF). Note that the loss function of

Manuscript received 19 December 2021; revised 31 May 2022; accepted 30 July 2022; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor N. Bartolini. Date of publication 1 September 2022; date of current version 18 April 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2021ZD0112501 and Grant 2021ZD0112502; in part by the National Natural Science Foundations of China under Grant 61772230, Grant 61972450, Grant 62102161, and Grant 62072209; in part by the China Computer Federation (CCF)-Baidu Open Fund under Grant 2021PP15002000; and in part by the National Science Foundation (NSF) under Grant CNS 1824440, Grant CNS 1828363, Grant CNS 1757533, Grant CNS 1629746, Grant CNS 1651947, and Grant CNS 1564128. An earlier version of this paper was presented at the 29th IEEE/ACM International Symposium on Quality of Service 2021 (IWQoS) [DOI: 10.1109/IWQOS52092.2021.9521275]. (Corresponding author: Wenbin Liu.)

En Wang, Mijia Zhang, Wenbin Liu, Bo Yang, and Yongjian Yang are with the College of Computer Science and Technology, Jilin University, Changchun 130012, China (e-mail: wangen@jlu.edu.cn; zhangmj19@mails.jlu.edu.cn; liwenbin@jlu.edu.cn; ybo@jlu.edu.cn; yyj@jlu.edu.cn).

Haoyi Xiong is with the Big Data Laboratory, Baidu Research, Beijing 100193, China (e-mail: xionghaoyi@baidu.com).

Jie Wu is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: jiewu@temple.edu).
Digital Object Identifier 10.1109/TNET.2022.3201545

SDMF is no longer the previous MSE loss function, but replaced with an Outlier Value Loss (OVL) function to effectively detect and recover the outliers. Moreover, a spatiotemporal outlier value memory network is added for further enhancing the outlier inference. Finally, we take extensive qualitative and quantitative experiments on two popular datasets each with two types of sensing data, and the experimental results indicate the advantages of our method that outperforms the state-of-the-art methods.

Index Terms—Sparse mobile crowdsensing, matrix completion, intra- and inter-data correlation, outlier.

I. INTRODUCTION

WITH the rapid development of the mobile computing in Internet of Things (IoT), Mobile CrowdSensing (MCS) [2], [3], [4], which recruits mobile users carrying IoT devices to collect various urban sensing data [5], [6], [7], [8], [9], has become an increasingly powerful sensing paradigm. In real-world scenarios, traditional MCS usually recruits a huge amount of users to collect all of the required data, which obviously costs a lot. To reduce sensing cost, some researchers introduce data inference techniques, called Sparse MCS [10], [11], which can sense a part of data, explore the correlations, and infer the remaining ones.

Thus, sparse MCS will be a multi-step process: 1) selecting important subareas, 2) recruiting user to sense parts of data, 3) inferring the complete data. The past several years has witnessed that data completion (the 3rd step) receives more attention from the field of network [12], [13] and plays an important role in sparse MCS [14], [15]. In practical sparse MCS application, there usually exist a few but important spatiotemporal outliers that effect the urban sensing applications. As illustrated in Fig. 1, the whole city map is equally divided into 4×4 grids, and we aim to obtain the full sensing map of m time slots. Due to the sensing budget constraint and uncertain user mobilities, it is difficult to collect data from all 16 subareas at each time slot [16], [17]. In this case, some vital information may be lost, which results in missing emergency data. For example, a pedestrian flow monitoring system [18] sustained to monitor the pedestrian flow of each urban subarea. Supposing that there suddenly gathers a lot of people (which leads to a high pedestrian flow) in a street or other place and the system does not sense the abnormal pedestrian flow increasing in this subarea, it would cause personnel trample and even a serious loss of lives and properties. Generally,

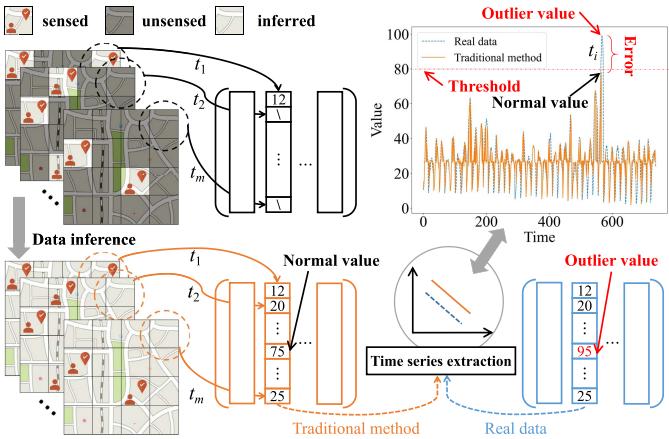


Fig. 1. An example to show the outlier value effect in data inference problem.

we call this abnormal sensing data an “**outlier value**,” which is significantly out of the normal value (non-outlier value) distribution. Fortunately, the occurrence of outlier values may always satisfy some potential rules. For example: 1) The high incidence of fire in an area always occurs in a fixed period of time every year (temporal correlations); 2) The change of forest fire risk level in two different regions may be highly consistent (spatial correlations); 3) Low humidity areas are more prone to fire, and fire is usually accompanied by deterioration of air quality (inter-data correlations); etc. The spatiotemporal correlations like example 1) and example 2) are originated by intra-data and uniformly called intra-data correlations. When we focus on the correlations of both intra- and inter-data, the unsensed outliers may be efficiently detected, inferred, or predicted in time.

Nowadays, the way to infer outliers in sparse urban crowdsensing is still an open issue. The existing data inference methods (e.g., Compressive Sensing [10], [19], [20], [21], Matrix Completion [22], [23], etc.) usually utilize the universal similarity and conventional tendency to infer the full map. However, as shown in Fig. 1, giving a glance at the circled grid where outliers appear, we find that the previous inference method that relies on usual spatiotemporal correlations or distributions can infer the outlier value as 75, which has a large inference error compared with the real data. At this time, the inferred value is lower than the alarm threshold. Thus, the vital outlier is regarded as a normal value, which is dangerous and unacceptable for real applications (e.g. temperature monitoring).

The main reason that previous methods misidentify the outliers is that there are lack of outlier samples to train the existing models. Ding *et al.* [24] have proved that empirical distribution of real-world data always appears to be heavy-tailed. By combining PauTa Criterion theory of Statistical Mathematics, we can easily draw the conclusion that outliers only account for less than 5% of the total data in common situations. The imbalance between normal value data and outlier data may lead to the following two problems: 1) The model regards all sensed data as normal values, so the unsensed outlier cannot be recovered entirely. 2) The model fits both the normal value and the outlier perfectly, so some

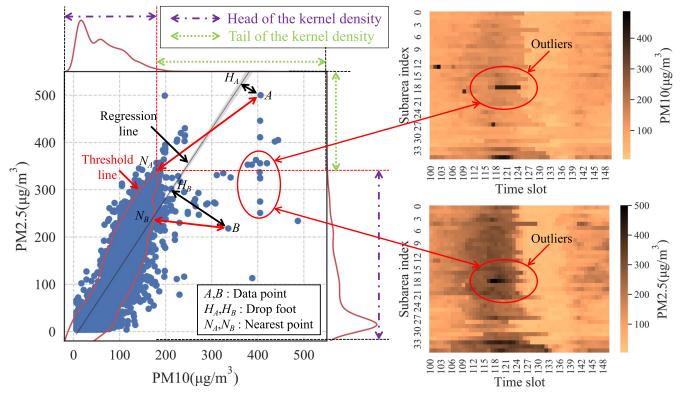


Fig. 2. An example to show the data distribution of two datasets with correlation.

unsensed normal values are recovered to outliers. Without enough sensed outliers, traditional machine learning-based data inference methods cannot effectively recover these data. While the use of other type of sensed data would help, it is difficult to establish effective associations between different datasets because of the sparse sensed data, especially for outlier data points. Therefore, how to *detect and recover outliers from such rare sensed outlier data* is the first challenge.

Note that the outliers usually go against the usual correlations in normal data, which makes data completion more challenging. From the perspective of intra-data probability distribution, the normal value distribution is relatively concentrated around the means and the occurrence probability is always strictly higher than outliers. This suggests us to exploit more inter-data correlations when inferring outliers. However, inter-data only plays an auxiliary role, and depending on inter-data solely leads to the degradation of overall inference accuracy. In addition, different auxiliary datasets make different contributions to data inference and outlier detection. When facing multiple auxiliary datasets, we must correctly measure the different contributions of each auxiliary dataset. Hence, how to *deal with inconsistent data distribution between normal and outlier values from multiple datasets* is the second challenge.

While inter-data correlations provide tremendous information gain for modeling the outlier values in crowdsensing, it is necessary to implement algorithms to leverage the real spatiotemporal data. We provide an example of two types of air quality data, as shown in Fig. 2, and we find that the outliers’ inter-data correlations are more complex than intra-data. The left part of Fig. 2 shows the value distribution of these two types of data and the right part shows the spatiotemporal distribution. It can be seen that there is strong correlation between these two types of air quality data. Hence, we also provide the linear regression line (the black line H_AH_B) to show this correlation trend. Thus, we can use this correlation to recover those unsensed outliers. We select two points A and B from all data points. Since the data point A is closer to the linear regression line H_AH_B than the data point B ($AH_A < BH_B$), the data point B is more easily to be inferred than the data point A , although the data point A is closer to normal value ($BN_B < AN_A$). Therefore, how to *extract*

such complex inter-data correlations of outliers is the third challenge.

To deal with the three key challenges above, we must make full use of sparse spatiotemporal data from every dataset. If an outlier is unsensed at the current time slot, we try to determine whether there is similar spatiotemporal data distribution from other time slots, other subareas and even other datasets. Along this line, we propose the Stacked Deep Matrix Factorization with exploiting Outlier Value (SDMF-OV), which focuses on detecting and inferring the outliers. Specifically, considering the complex spatiotemporal correlations of sensed data from different datasets, we apply Stacked Deep Matrix Factorization (SDMF) algorithm with Outlier Value Loss (OVL) function to initially recover the sparse sensing matrix. Unlike existing data inference models, we construct an outlier value memory network to predict the label that indicates whether the unsensed value is an outlier value. With the help of the outlier value memory network, the proposed sparse matrix completion method can efficiently complete and predict the unsensed data for the city-scale map.

Our work has the following contributions:

- We formulate the sparse urban crowdsensing problem, where we intend to recover the unsensed MCS data including both normal value data and outlier value data by leveraging the sparsely sensed data with both intra- and inter-data correlations.
- We propose an urban crowdsensing method named SDMF-OV, which aims to solve the problem of inferring outlier from sparse sensed data based on SDMF (an intra- and inter-data-based sparse data inference method we proposed). Compared with the previous methods, SDMF-OV can effectively extract the complex spatiotemporal relationship between the outlier values and the normal values, and the performance of SDMF-OV is improved by using both intra- and inter-data correlations.
- We evaluate the proposed method using two real-world datasets with two typical urban sensing tasks. The results based on both datasets verify the effectiveness in improving the recovery and prediction accuracy and the ability of outliers' classification with the sparse sensed data when considering outliers and exploiting both intra- and inter-data correlations.

II. RELATED WORK

A. Sparse MCS

MCS technology utilizes mobile devices carried by users to perform series of urban crowdsensing tasks [2], [25]. For example, in urban environmental MCS applications, measurements (e.g., traffic speeds [26], noise levels [27], etc.) enabled the mapping of various large-scale urban environmental phenomena by involving the common person. Compared with traditional sensor networks, MCS has a number of unique characteristics that bring opportunities to researchers and users. To provide QoS, some existing efforts [28], [29] recruited as many users as possible to collect data. These types of approaches could indeed provide better service, however, collecting a large number of users means a huge cost. In our

application scenario of the outlier value data inference, it does help to be aware of such large amounts of data, but it was unrealistic in most cases. Therefore, traditional MCS can not be applied directly. In recent years, researchers have developed many urban crowdsensing systems based on Sparse MCS, which first sensed limited subareas and then inferred the entire map. Wei *et al.* [30] made efforts in the field of task allocation. They achieved a highly diverse and spatial quality coverage level within a limited budget for different application scenarios. Li *et al.* [31] focused on the critical problem that which data instances should be collected to maximize the performance of the trained model under the budget limits. Bian *et al.* [32] proposed to use decentralized MF algorithm to enable sparse MCS without location/data aggregation to a central server.

B. Spatiotemporal Data Inference

For spatiotemporal data, sparse MCS is essential for the completion of the sparse spatiotemporal matrix. Matrix Factorization (MF) is a classical matrix completion algorithm, which takes the advantage of the low-rank properties of the complete matrix. With the wide application of deep neural networks in the past several years, Fan and Cheng [22] proposed the Deep Matrix Factorization (DMF) method by combining traditional linear matrix factorization with deep neural network. Using DMF to complete the sparse matrix could obtain the non-linear spatiotemporal characteristics effectively. Wang *et al.* [33] utilized the DMF method in the field of sparse urban sensing and prediction, and combined it with the time series prediction method to realize the use of sparse data to predict the future value by an end-to-end model. In recent years, with the rapid development of graph neural network (GNN), the algorithm of matrix completion using GNN arised at the historic moment. Zhang *et al.* [23] proposed an Inductive Graph-based Matrix Completion (IGMC) model without using any side information for recommender systems, and it was also applicable to the data completion scene of sparse spatiotemporal data matrix because of its inductive model.

C. Intra- and Inter-Data

With the wide application of big data in recent years, researchers are increasingly exploring the use of intra- and inter-data. Cui *et al.* [34] combined big data with software-defined networking (SDN) and proposed SDN-based intra- and inter-data-center networks with big data. In the next several years, intra- and inter-data was applied in more and more fields. Wang *et al.* [19] studied the inter-data correlations in multi-task scenarios and carried out a significance test of the Pearson correlation to find out which was the dominant correlation. Liu *et al.* [35] designed a network structure with multiple branches, where branches extract the intra-task correlations for each task, respectively, and then catenated the results from all branches to capture the inter-data correlations among the multiple tasks. For the air quality and weather prediction of practical applications, Han *et al.* [36] proposed the Multi-adversarial spatiotemporal recurrent Graph Neural Networks for jointing air quality and weather predictions.

TABLE I
MAIN NOTATIONS

Symbol	Meaning
m, m_0	number of time slots, the length of the time slots contained in each sampling period
n, n_0	number of subareas, the number of the subareas contained in each sampling grids
i, j	index of a time slot or a subarea
r	rank we assume of the complete matrix
ϵ_1, ϵ_2	right/left outlier threshold
Φ	number of the auxiliary datasets
$\rho^{(\phi)}$	correlation coefficient between objective dataset and ϕ -th auxiliary dataset
L	length of the memory network
$\theta^{(\phi)}$	weight coefficient of the auxiliary dataset
b	compensation factor of outliers
$\mathbf{Y}'_o, \mathbf{Y}'_{ok}$	sparse objective matrix/ k -th sub-matrix of \mathbf{Y}'_o
$\mathbf{Y}'^{(\phi)}, \mathbf{Y}'_{ak}^{(\phi)}$	sparse auxiliary matrix/ k -th sub-matrix of $\mathbf{Y}'^{(\phi)}$
$\mathbf{W}'_{ol}, \mathbf{W}'_{al}$	l -th window sub-matrix of matrix $\mathbf{Y}'_o/\mathbf{Y}'_a^{(\phi)}$
$\hat{\mathbf{Y}}_{ok}, \mathbf{Z}_{ok}$	completed/embedding matrix of sub-matrix \mathbf{Y}'_{ok}
$\hat{\mathbf{W}}_{ol}, \mathbf{Z}_{Wal}$	completed/embedding matrix of sub-matrix \mathbf{W}'_{ol}
$\mathbf{Z}_{ak}^{(\phi)}, \mathbf{Z}_{Wal}^{(\phi)}$	embedding matrix of k -th target sub-matrix/ l -th reference sub-matrix of the ϕ -th auxiliary dataset

In the intra- and inter-data correlation field, existing work usually ignored the impact of outlier values, especially outliers of intra- and inter-data.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

Our research is based on a common urban sensing scheme that recruits mobile users to collect data from some target areas for recovering outlier value data. In this subsection, we are going to show the mathematical representation of the system model we design. A large number of mathematical symbols will be involved in this section and the following sections, so we sort out some main notations in TABLE I.

Given a whole urban sensing area which is divided into n subareas (grids), we aim at achieving the whole n subareas with only \tilde{n} sensed grids ($\tilde{n} \ll n$) for each time slot. In order to represent which grids are sensed in the i -th time slot, we introduce the logical vector $\mathbf{c}^{(i)} = [c_{i1}, c_{i2}, \dots, c_{in}] \in \{0, 1\}^n$ to denote which grids are sensed. If the subarea j has been sensed, $c_{ij} = 1$, otherwise, $c_{ij} = 0$. The objective vector $\mathbf{y}'^{(i)} = [y'_{i1}, y'_{i2}, \dots, y'_{in}] \in \mathbb{R}^n$ denotes the sparse sensed data. The unsensed data are recorded as some meaningless values (e.g., ∞). $\mathbf{y}^{(i)} = [y_{i1}, y_{i2}, \dots, y_{in}] \in \mathbb{R}^n$ denotes the complete data vector which includes both the sensed value data and the unsensed value data.

After m time slots, we can combine vectors with the following matrices: $\mathbf{C} = [\mathbf{c}^{(1)\top}, \mathbf{c}^{(2)\top}, \dots, \mathbf{c}^{(m)\top}]^\top \in \{0, 1\}^{m \times n}$ denotes which grids are sensed for each time slot; sparse matrix $\mathbf{Y}' = [\mathbf{y}'^{(1)\top}, \mathbf{y}'^{(2)\top}, \dots, \mathbf{y}'^{(m)\top}]^\top \in \mathbb{R}^{m \times n}$ denotes the whole sensed data of m time slots; complete matrix $\mathbf{Y} = [\mathbf{y}^{(1)\top}, \mathbf{y}^{(2)\top}, \dots, \mathbf{y}^{(m)\top}]^\top \in \mathbb{R}^{m \times n}$ denotes the ground truth of complete data. Then we introduce the relationship of the three matrices as follows:

$$\mathbf{Y}' = \mathbf{Y} \circ \mathbf{C}, \quad (1)$$

where the symbol \circ denotes the Hadamard product. Then, we want to find a data inference function $g(\cdot)$ to infer the

unsensed data from \mathbf{Y}' and we can get the estimated complete data matrix $\hat{\mathbf{Y}}$:

$$\hat{\mathbf{Y}} = g(\mathbf{Y}'). \quad (2)$$

Considering the influence of outlier values on the results of matrix completion, it is necessary to define outlier values. We can classify the data into left outliers (-1), right outliers ($+1$) and normal values (0) according to the value of the data. $v_{ij} \in \{-1, 0, 1\}$ denotes the type of outlier values in the j -th subarea of the i -th time slot. An outlier label matrix \mathbf{V} can be formed by v_{ij} and the calculation method of the matrix \mathbf{V} is as follows:

$$\mathbf{V} = v(\mathbf{Y}) = \mathbb{I}(\mathbf{Y} > \epsilon_1) - \mathbb{I}(\mathbf{Y} < \epsilon_2), \quad (3)$$

where the large constant ϵ_1 and small constant ϵ_2 are thresholds. $\mathbb{I}(\cdot)$ denotes the indicator function. By solving each element of matrix \mathbf{Y} with Eq. (3), we can get the label matrix $\mathbf{V} = v(\mathbf{Y})$. It is noted that the unsensed elements in the matrix \mathbf{Y} should be regarded as normal values with a high probability.

In addition to the objective dataset that we are committed to complete, we also try to use several sparse auxiliary datasets. (For enhancing the accuracy of data sensing and prediction, it will be a better case if there are complete auxiliary datasets.) To facilitate differentiation, the mathematical symbols $\mathbf{C}_o, \mathbf{Y}_o, \mathbf{Y}'_o, \mathbf{V}_o$ and \mathbf{V}'_o denote the matrices about the objective dataset and the mathematical symbols $\mathbf{C}_a^{(\phi)}, \mathbf{Y}_a^{(\phi)}, \mathbf{Y}'_a^{(\phi)}, \mathbf{V}_a^{(\phi)}$ and $\mathbf{V}'_a^{(\phi)}$ denote the matrices about the ϕ -th auxiliary dataset. We can calculate the correlation coefficient $\rho^{(\phi)}$ between the objective matrix \mathbf{Y}'_o and any auxiliary matrix $\mathbf{Y}'_a^{(\phi)}$ by the following formulas:

$$\bar{y}'_o = \text{Mean}(\mathbf{Y}'_o | \mathcal{S}_o), \quad (4)$$

$$\bar{y}'_a^{(\phi)} = \text{Mean}(\mathbf{Y}'_a^{(\phi)} | \mathcal{S}_a^{(\phi)}), \quad (5)$$

$$\rho^{(\phi)} = \frac{\sum_{(i,j) \in \mathcal{S}_{oa}^{(\phi)}} (y'_{oj} - \bar{y}'_o)(y'_{aj}^{(\phi)} - \bar{y}'_a^{(\phi)})}{\sqrt{\sum_{(i,j) \in \mathcal{S}_{oa}^{(\phi)}} (y'_{oj} - \bar{y}'_o)^2 \sum_{(i,j) \in \mathcal{S}_{oa}^{(\phi)}} (y'_{aj}^{(\phi)} - \bar{y}'_a^{(\phi)})^2}}, \quad (6)$$

where $\mathcal{S}_o \triangleq \{(i,j) | c_{oj} = 1\}$, $\mathcal{S}_a^{(\phi)} \triangleq \{(i,j) | c_{aj}^{(\phi)} = 1\}$ and the set $\mathcal{S}_{oa}^{(\phi)} \triangleq \mathcal{S}_o \cap \mathcal{S}_a^{(\phi)}$. $\text{Mean}(\cdot | \mathcal{S})$ denotes the mean value of some elements (corresponding the set \mathcal{S}) of a matrix.

It is necessary to emphasize the following two notes. First, all of these matrices must be with the shape of $m \times n$ in order to enhance data sensing and prediction effectively. In other words, in any auxiliary matrix or objective matrix, the elements in any row i and any column j must correspond to the same time slot and subarea. Second, it does not mean that any dataset can be used as an auxiliary dataset. We require that there is strong positive or negative correlation between auxiliary dataset and objective dataset.

B. Problem Formulation

Problem [Sparse Spatiotemporal CrowdSensing via Intra-and Inter-Data with Outlier Model]: Given a sparse objective matrix $\mathbf{Y}'_o \in \mathbb{R}^{m \times n}$ and the corresponding label matrix $\mathbf{C}_o \in \{0, 1\}^{m \times n}$, a set of sparse auxiliary matrices $\mathcal{Y}'_a = \{\mathbf{Y}'_a^{(1)}, \mathbf{Y}'_a^{(2)}, \dots, \mathbf{Y}'_a^{(\Phi)}\}$ and a set of corresponding label

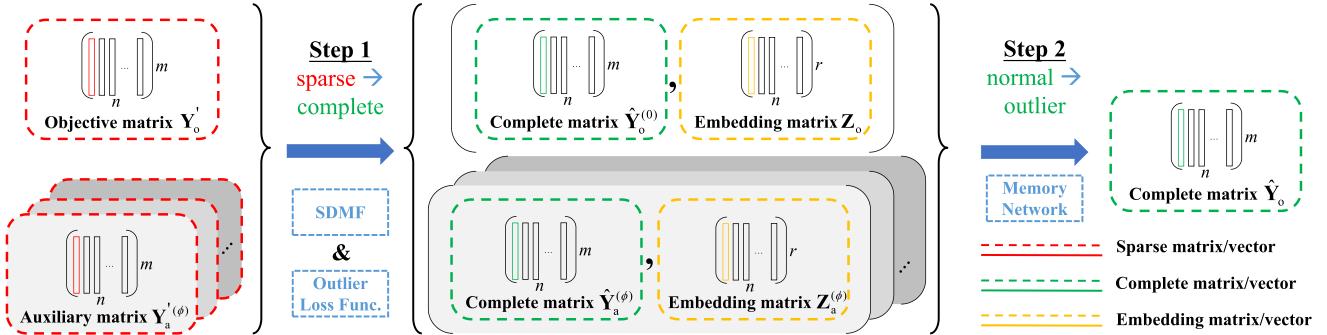


Fig. 3. Framework overview.

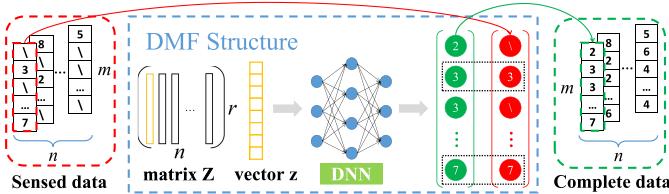


Fig. 4. Mapping relationship between low-dimensional embedding vector and high-dimensional complete vector.

matrices $\mathcal{C}_a = \{\mathbf{C}_a^{(1)}, \mathbf{C}_a^{(2)}, \dots, \mathbf{C}_a^{(\Phi)}\}$, we need to accomplish the following two tasks:

- Find a function $g(\cdot)$ to recover the complete matrix $\hat{\mathbf{Y}}_o$ and make $\hat{\mathbf{Y}}_o = g(\mathbf{Y}'_o, \mathbf{C}_o, \mathcal{Y}'_a, \mathcal{C}_a)$ be established.
- Ensure that the unsensed outlier value data can be detected and inferred correctly.

In this process, the following objective value obj should be kept as small as possible:

$$obj = \|\hat{\mathbf{Y}}_o - \mathbf{Y}_o\| + \|v(g(\mathbf{Y}'_o, \mathbf{C}_o, \mathcal{Y}'_a, \mathcal{C}_a)) - \mathbf{V}_o\|. \quad (7)$$

In order to minimize the objective value obj , we should first minimize the error of matrix completion and then focus on outlier value data to correct outlier errors. As shown in Fig. 3, it will be a two-step process.

IV. STACKED DEEP MATRIX FACTORIZATION WITH OUTLIER VALUE LOSS FUNCTION

Inspired by Deep Matrix Factorization (DMF) [22], we design a novel method of matrix completion based on DMF. Given a sparse matrix $\mathbf{Y} \in \mathbb{R}^{m \times n}$, DMF uses a Deep Neural Network (DNN) to generate a complete matrix, as shown in Fig. 4. We hope that the output vector of the neural network is as close as possible to each column of the sparse matrix $\mathbf{Y}' \in \mathbb{R}^{m \times n}$. Unlike traditional DNNs, we need to treat the input vector $\mathbf{z} \in \mathbb{R}^r$ of the DNN as the parameters of the DNN for training. In addition, for the output layer, we only focus on the values of sensed elements of the output vector instead of all output elements.

The structure of our Stacked Deep Matrix Factorization (SDMF) is shown in Fig. 5 and the algorithm pseudo code is shown in Alg. 1. All of the input datasets can be divided into two parts: the objective dataset and the auxiliary datasets. The objective dataset is the task dataset and our goal is to infer the missing data from the objective dataset. To represent a more general case, we assume that all auxiliary datasets

Algorithm 1 Stacked Deep Matrix Factorization With Outlier Value Loss Function

Input: Sparse objective matrix \mathbf{Y}'_o , logical label matrix \mathbf{C}_o , auxiliary matrix set $\mathcal{Y}'_a = \{\mathbf{Y}'_a^{(1)}, \mathbf{Y}'_a^{(2)}, \dots, \mathbf{Y}'_a^{(\Phi)}\}$, logical label matrix set $\mathcal{C}_a = \{\mathbf{C}_a^{(1)}, \mathbf{C}_a^{(2)}, \dots, \mathbf{C}_a^{(\Phi)}\}$

Output: Completed objective matrix $\hat{\mathbf{Y}}_o$

- 1: Calculate the correlation coefficient $\rho^{(1)}, \rho^{(2)}, \dots, \rho^{(\Phi)}$ between \mathbf{Y}'_o and each element in \mathcal{Y}'_a , respectively, by Eqs. (4), (5) and (6);
- 2: Randomly initialize $\mathbf{Z}_o, \mathbf{Z}_a^{(1)}, \mathbf{Z}_a^{(2)}, \dots, \mathbf{Z}_a^{(\Phi)}$, respectively;
- 3: Build the Neural Networks by using $f_o(\cdot), f_a^{(1)}(\cdot), f_a^{(2)}(\cdot), \dots, f_a^{(\Phi)}(\cdot)$;
- 4: Set $count := 0$;
- 5: **while** not convergent **and** $count < MAX_ITER_1$ **do**
- 6: **for** ϕ is from 1 to Φ **do**
- 7: Fix $f_a^{(\phi)}(\cdot), \mathbf{Z}_a^{(\phi)}$, calculate $\hat{\mathbf{Y}}_a^{(\phi)} := f_a^{(\phi)}(\mathbf{Z}_a^{(\phi)})$, and then calculate and reduce \mathcal{L}_{mix} by Eq. (12);
- 8: **end for**
- 9: $count := count + 1$;
- 10: **end while**
- 11: Reset $count := 0$;
- 12: **while** not convergent **and** $count < MAX_ITER_2$ **do**
- 13: Fix $f_o(\cdot), \mathbf{Z}_o$, calculate $\hat{\mathbf{Y}}_o^{(0)} := f_o(\mathbf{Z}_o)$, and then calculate and reduce \mathcal{L}_{mix} by Eq. (12);
- 14: **for** ϕ is from 1 to Φ **do**
- 15: Fix $f_o(\cdot)$, calculate $\hat{\mathbf{Y}}_o^{(\phi)} := f_o(\rho^{(\phi)} \mathbf{Z}_a^{(\phi)})$, and then calculate and reduce \mathcal{L}_{mix} by Eq. (12);
- 16: **end for**
- 17: $count := count + 1$;
- 18: **end while**
- 19: Set the weight coefficient $\theta^{(0)}$ for the objective dataset, and then calculate $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(\Phi)}$, respectively, by Eq. (8);
- 20: **return** $\hat{\mathbf{Y}}_o := \theta^{(0)} \hat{\mathbf{Y}}_o^{(0)} + \theta^{(1)} \hat{\mathbf{Y}}_o^{(1)} + \dots + \theta^{(\Phi)} \hat{\mathbf{Y}}_o^{(\Phi)}$.

also have missing data. Unlike the objective dataset, there is no requirement to infer missing data from auxiliary datasets. Both the objective dataset and the auxiliary datasets can be represented by the spatiotemporal matrix, which name is objective matrix or auxiliary matrix. We first try to use traditional DMF to complete each auxiliary matrix, but our

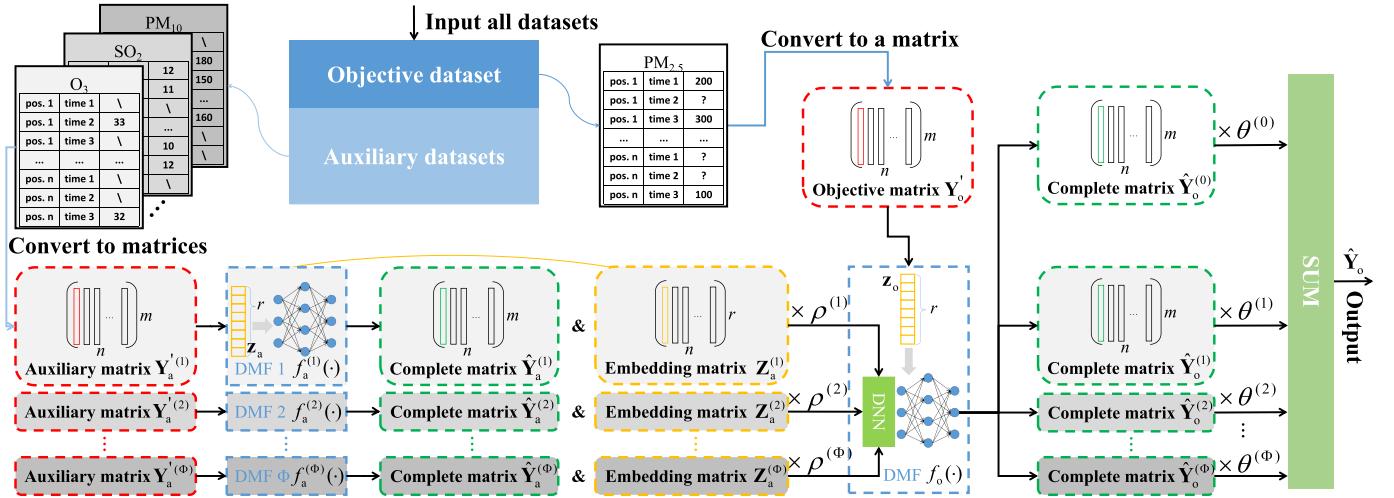


Fig. 5. Structure of Stacked Deep Matrix Factorization.

goal is to get the embedded representation of the auxiliary matrix rather than the matrix completion result. Based on the assumption that there is strong correlation between the objective dataset and each auxiliary datasets. The dimensions of these embedding vectors must all be set as r , which equals to the rank of objective matrix. For the sparse objective matrix, we only input it into a traditional neural network to complete it. It should be noted that while training this DMF, those embedding vectors from the auxiliary matrix are also input by multiplying the correlation coefficient. Whether we use the objective matrix or the embedding vector from the auxiliary matrix for training, we all hope that the output of this DMF converges towards the direction of objective sensed value.

Through the above steps, we will get $(\Phi + 1)$ complete matrices related to the objective dataset and auxiliary datasets. The weighted sum of these matrices will be the output of SDMF. The weight coefficient of the auxiliary dataset is calculated as follows:

$$\theta^{(\phi)} = \frac{|\rho^{(\phi)}|(1 - \theta^{(0)})}{\sum_{\phi=1}^{\Phi} |\rho^{(\phi)}|}, \quad (8)$$

where $\theta^{(0)} \in (0, 1)$ is the weight coefficient for the objective dataset and the value of $\theta^{(0)}$ represents the weight of intradata. Then, we will get the output matrix of SDMF which is completed and shown as follows:

$$\hat{\mathbf{Y}}_o = \sum_{\phi=0}^{\Phi} \theta^{(\phi)} \hat{\mathbf{Y}}_o^{(\phi)}. \quad (9)$$

Then, it is necessary to introduce how to set the loss function to generate outliers. We use the function $f(\cdot)$ to represent the structure of the SDMF neural network. The loss function of traditional DMF is a MSE loss function with two regularization penalty terms and is shown in Eq. (10):

$$\mathcal{L}_{\text{MSE}} = \frac{1}{2mn} \|(\mathbf{Y} - f(\mathbf{Z})) \circ \mathbf{C}\|^2 + \lambda \pi(f) + \mu \|\mathbf{Z}\|^2, \quad (10)$$

where $\pi(f)$ and $\|\mathbf{Z}\|^2$ represent the regularization penalty terms of the deep neural network and input vector, respectively. λ and μ are weight parameters.

However, considering the impact of outlier values, it is necessary to add a penalty term to identify outlier value data correctly. According to the probability distribution of outlier value data, inspired by Extreme Value Loss (EVL) function for extreme event prediction problem [24], we design our Outlier Value Loss (OVL) function as follows:

$$\mathcal{L}_{\text{OVL}} = - \sum_{(i,j) \in \mathcal{S}} (1 - \beta^{(v_{ij})})(1 - \frac{p_{ij}^{(v_{ij})}}{\gamma})^\gamma \log(p_{ij}^{(v_{ij})}), \quad (11)$$

where $\mathcal{S} \triangleq \{(i,j) | c_{ij} = 1\}$, $\beta^{(0)}$, $\beta^{(-1)}$ and $\beta^{(1)}$ are the proportion of normal values, left outlier values, and right outlier values in the dataset, respectively. γ is the hyper-parameter, which is the outlier value index in the approximation. $p_{ij}^{(0)}$, $p_{ij}^{(-1)}$, and $p_{ij}^{(1)}$ are calculated through \mathbf{Z} by a full connection layer neural network. The input of this neural network is vector $\mathbf{z}_j \in \mathbb{R}^r$ and the output vector is $[p_{1j}^{(0)}, p_{1j}^{(-1)}, p_{1j}^{(1)}, \dots, p_{mj}^{(0)}, p_{mj}^{(-1)}, p_{mj}^{(1)}]^\top \in \mathbb{R}^{3m}$.

From what has been discussed about loss functions, we may easily conclude that we can not only reduce the matrix completion error effectively but also identify the outlier values correctly by combining the MSE and OVL loss functions. So the mixed loss function we get is:

$$\mathcal{L}_{\text{mix}} = \xi \mathcal{L}_{\text{MSE}} + (1 - \xi) \mathcal{L}_{\text{OVL}}, \quad (12)$$

where ξ represents the weight parameter. The influence of two loss functions on the experimental results can be balanced by adjusting the weight parameter ξ .

When applying SDMF, we can select different loss functions according to different usage scenarios. For the convenience of the following description, SDMF algorithm is abbreviated as the following expression:

$$(\hat{\mathbf{Y}}_o, \mathbf{Z}_o, \mathcal{Z}_a) = \text{SDMF}(\mathbf{Y}'_o, \mathcal{Y}'_a, \text{loss}), \quad (13)$$

where loss denotes the loss function we choose (e.g. $\text{loss} = \mathcal{L}_{\text{mix}}$) and the embedding matrix set of auxiliary matrices is $\mathcal{Z}_a = \{\mathbf{Z}_a^{(1)}, \dots, \mathbf{Z}_a^{(\Phi)}\}$. $\hat{\mathbf{Y}}_o$ denotes the completed objective matrix and \mathbf{Z}_o denotes the corresponding embedding matrix.

The matrix $\mathbf{Z}_a^{(\phi)}$ denotes the embedding presentation of the ϕ -th auxiliary matrix $\mathbf{Y}_a^{(\phi)}$.

V. OUTLIER VALUE MEMORY NETWORK FOR ENHANCING OUTLIER INFERENCE

There is also diversity among outliers. The outliers that are just exceeding the threshold are different from those far exceeding the threshold. In order to memorize these outliers, we propose to use a memory network, which has been proved to be effective in recognizing inherent patterns contained in sensed information. So far, we can design a whole model named SDMF-OV model. The SDMF-OV model exploits the SDMF algorithm with the mixed loss function and an outlier value memory network which aims to remember and identify the distribution characteristics of the outlier value data. We combine the SDMF algorithm with the OVL function, which can not only use SDMF to achieve high-precision matrix completion, but also effectively reduce the influence of outlier inference results.

The detailed SDMF-OV algorithm is introduced in the following several paragraphs. In order to make it easier for readers to understand, we provide a pseudo code flow table of SDMF-OV in Alg. 2, and an example memory network figure is also given in Fig. 6.

First, we define the concept of a window in our context. In this paper, we will block or sample the data matrix and label matrix according to a certain shape $m_0 \times n_0$. We divide the objective matrix \mathbf{Y}'_o into objective sub-matrices $\mathbf{Y}'_{o1}, \mathbf{Y}'_{o2}, \dots, \mathbf{Y}'_{oK} \in \mathbb{R}^{m_0 \times n_0}$ and divide every auxiliary matrix $\mathbf{Y}'_a^{(\phi)}$ into auxiliary sub-matrices $\mathbf{Y}'_{a1}^{(\phi)}, \mathbf{Y}'_{a2}^{(\phi)}, \dots, \mathbf{Y}'_{aK}^{(\phi)} \in \mathbb{R}^{m_0 \times n_0}$, respectively. It is obvious that $K = \lceil \frac{m}{m_0} \rceil \times \lceil \frac{n}{n_0} \rceil$. Meanwhile, we also randomly sample a sequence of windows by $\mathbf{W}'_{o1}, \mathbf{W}'_{o2}, \dots, \mathbf{W}'_{oL} \in \mathbb{R}^{m_0 \times n_0}$ from the objective matrix \mathbf{Y}'_o and $\mathbf{W}'_{a1}^{(\phi)}, \mathbf{W}'_{a2}^{(\phi)}, \dots, \mathbf{W}'_{aL}^{(\phi)} \in \mathbb{R}^{m_0 \times n_0}$ from every auxiliary matrix $\mathbf{Y}'_a^{(\phi)}$ by a same sample form, where L is the length of the memory network we build.

Then, we propose applying the SDMF module with the mixed loss function to complete each target sub-matrix and each reference sub-matrix then build a memory network to remember characteristics of outliers. Suppose we train an SDMF neural network $SDMF(\cdot, \cdot, \mathcal{L}_{\text{mix}})$. We can easily get the completed sub-matrices and the corresponding embedding matrices: $(\hat{\mathbf{Y}}_{ok}, \mathbf{Z}_{ok}, \mathcal{Z}_{ak}) = SDMF(\mathbf{Y}'_{ok}, \mathcal{Y}'_{ak}, \mathcal{L}_{\text{mix}})$, where $\mathcal{Z}_{ak} = \{\mathbf{Z}_{ak}^{(1)}, \dots, \mathbf{Z}_{ak}^{(\Phi)}\}$ denotes the embedding matrix set of auxiliary matrices and $k = 1, 2, \dots, K$. Similarly, $(\hat{\mathbf{W}}_{ol}, \mathbf{Z}_{Wal}, \mathcal{Z}_{Wal}) = SDMF(\mathbf{W}'_{ol}, \mathcal{W}'_{al}, \mathcal{L}_{\text{mix}})$, where $\mathcal{Z}_{Wal} = \{\mathbf{Z}_{Wal}^{(1)}, \dots, \mathbf{Z}_{Wal}^{(\Phi)}\}$ denotes the embedding matrix set of auxiliary matrices and $l = 1, 2, \dots, L$.

After the processing of matrix completion, we will get some parameters that can characterize the memory network that we want to build. We require that these parameters represent the characteristics of sensed data and represent the characteristics of outlier values. In order to achieve these goals, we build the architecture of our memory network that mainly consists of the following two modules:

Algorithm 2 Outlier Value Memory Network

Input: Sparse objective matrix \mathbf{Y}'_o , logical label matrix \mathbf{C}_o , auxiliary matrix set $\mathcal{Y}'_a = \{\mathbf{Y}'_a^{(1)}, \mathbf{Y}'_a^{(2)}, \dots, \mathbf{Y}'_a^{(\Phi)}\}$, logical label matrix set $\mathcal{C}_a = \{\mathbf{C}_a^{(1)}, \mathbf{C}_a^{(2)}, \dots, \mathbf{C}_a^{(\Phi)}\}$, and the shape of windows: (m_0, n_0)

Output: $\hat{\mathbf{O}}$

- 1: Divide the objective matrix \mathbf{Y}'_o into K blocks each with the shape of $m_0 \times n_0$: $\mathbf{Y}'_{o1}, \mathbf{Y}'_{o2}, \dots, \mathbf{Y}'_{oK}$ and sample L blocks randomly each with the same shape as \mathbf{Y}'_{ok} : $\mathbf{W}'_{o1}, \mathbf{W}'_{o2}, \dots, \mathbf{W}'_{oL}$ from the objective matrix \mathbf{Y}'_o .
- 2: **for** ϕ is from 1 to Φ **do**
- 3: Divide the ϕ -th sparse auxiliary matrix $\mathbf{Y}'_a^{(\phi)}$ into blocks $\mathbf{Y}'_{a1}^{(\phi)}, \mathbf{Y}'_{a2}^{(\phi)}, \dots, \mathbf{Y}'_{aK}^{(\phi)} \in \mathbb{R}^{m_0 \times n_0}$ and sample blocks $\mathbf{W}'_{a1}^{(\phi)}, \mathbf{W}'_{a2}^{(\phi)}, \dots, \mathbf{W}'_{aL}^{(\phi)} \in \mathbb{R}^{m_0 \times n_0}$ by the same sample form as \mathbf{W}'_{ol} ;
- 4: **end for**
- 5: Build and train the neural network $SDMF(\cdot, \cdot, \mathcal{L}_{\text{mix}})$ by Alg. 1 and then calculate the output for objective sub-matrices $(\hat{\mathbf{Y}}_{ok}, \mathbf{Z}_{ok}, \mathcal{Z}_{ak}) = SDMF(\mathbf{Y}'_{ok}, \mathcal{Y}'_{ak}, \mathcal{L}_{\text{mix}})$ for each k in $\{1, 2, \dots, K\}$ and $(\hat{\mathbf{W}}_{ol}, \mathbf{Z}_{Wal}, \mathcal{Z}_{Wal}) = SDMF(\mathbf{W}'_{ol}, \mathcal{W}'_{al}, \mathcal{L}_{\text{mix}})$ for each l in $\{1, 2, \dots, L\}$;
- 6: Calculate the correlation coefficient $\rho^{(1)}, \rho^{(2)}, \dots, \rho^{(\Phi)}$ between \mathbf{Y}'_o and each element in \mathcal{Y}'_a , respectively, by Eqs. (4), (5) and (6);
- 7: **for** k is from 1 to K **do**
- 8: **for** l is from 1 to L **do**
- 9: Calculate α_{k-l0} by Eq. (14);
- 10: **for** ϕ is from 1 to Φ **do**
- 11: Calculate $\alpha_{k-l\phi}$ by Eq. (15);
- 12: **end for**
- 13: **end for**
- 14: **for** l is from 1 to L **do**
- 15: **for** ϕ is from 1 to Φ **do**
- 16: Calculate $\omega_{k-l\phi}$ by Eq. (16);
- 17: **end for**
- 18: **end for**
- 19: Calculate $\hat{\mathbf{O}}^{(k)} = \hat{\mathbf{Y}}_{ok} + b\mathbf{U}^{(k)}$;
- 20: **end for**
- 21: **return** $\hat{\mathbf{O}} = Splice(\hat{\mathbf{O}}^{(1)}, \hat{\mathbf{O}}^{(2)}, \dots, \hat{\mathbf{O}}^{(K)})$.

- **Embedding Module**

$\mathbf{Z}_{Wal} \in \mathbb{R}^{r \times n_0}$ and $\mathcal{Z}_{Wal} = \{\mathbf{Z}_{Wal}^{(1)}, \dots, \mathbf{Z}_{Wal}^{(\Phi)}\}$ are the latent representations of window l .

- **Label Matrix Module**

$\mathbf{V}_{Wal} = v(\mathbf{W}'_{ol}) \in \mathbb{R}^{m_0 \times n_0}$ and the outlier label matrix set $\mathcal{V}_{Wal} = \{\mathbf{V}_{Wal}^{(1)}, \dots, \mathbf{V}_{Wal}^{(\Phi)}\}$, which $\mathbf{V}_{Wal}^{(\phi)} = v(\mathbf{W}'_{al}^{(\phi)})$, are the outlier label matrices of window l from the ϕ -th auxiliary dataset.

At each sub-matrix k , we use SDMF to complete the matrix with $(\hat{\mathbf{Y}}_{ok}, \mathbf{Z}_{ok}, \mathcal{Z}_{ak}) = SDMF(\mathbf{Y}'_{ok}, \mathcal{Y}'_{ak})$. Thus, \mathbf{Z}_{ok} could be regarded as a potential representation of $\hat{\mathbf{Y}}_{ok}$. As we have discussed, $\hat{\mathbf{Y}}_{ok}$ may lack the ability to detect outlier values in the future. Therefore, we also require our model to retrospect its memory to check whether there is a similarity between the target value and outlier value in the sensed data. Hence, we propose to employ an attention mechanism in our model.

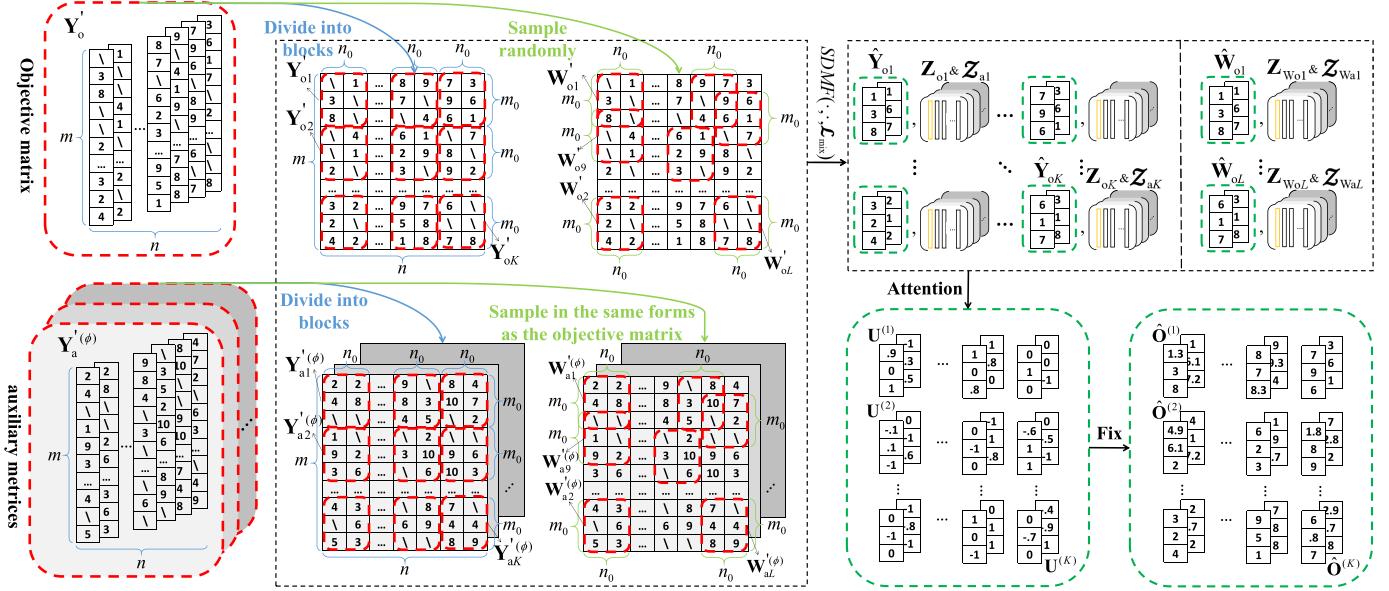


Fig. 6. Spatiotemporal memory network for outliers.

In order to measure the similarity between the target matrix and the reference matrix by cosine similarity, we utilize the matrix vector operator $\text{vec}(\cdot)$ to convert a matrix into a vector. We abbreviate $\text{vec}(\mathbf{Z}_{ok})$ as vector $\overrightarrow{\mathbf{z}}_{ok}$, abbreviate $\text{vec}(\mathbf{Z}_{Wal})$ as vector $\overrightarrow{\mathbf{z}}_{Wal}$ and abbreviate $\text{vec}(\mathbf{Z}_{Wal}^{(\phi)})$ as vector $\overrightarrow{\mathbf{z}}_{Wal}^{(\phi)}$. Then we can easily use cosine similarity to measure the similarity between the k -th target matrix and the l -th reference matrix from the ϕ -th auxiliary dataset or objective dataset self ($\phi = 0$). The mathematical expression of the cosine similarity $\alpha_{k-l\phi}$ between $\overrightarrow{\mathbf{z}}_{ok}$ and $\overrightarrow{\mathbf{z}}_{Wal}$ or $\overrightarrow{\mathbf{z}}_{Wal}^{(\phi)}$ can be expressed as the following Eqs. (14) and (15):

$$\alpha_{k-l0} = \frac{\overrightarrow{\mathbf{z}}_{ok}^\top \overrightarrow{\mathbf{z}}_{Wal}}{\|\overrightarrow{\mathbf{z}}_{ok}\| \cdot \|\overrightarrow{\mathbf{z}}_{Wal}\|}, \quad (14)$$

$$\alpha_{k-l\phi} = \frac{\overrightarrow{\mathbf{z}}_{ok}^\top \rho^{(\phi)} \overrightarrow{\mathbf{z}}_{Wal}^{(\phi)}}{\|\overrightarrow{\mathbf{z}}_{ok}\| \cdot \|\rho^{(\phi)} \overrightarrow{\mathbf{z}}_{Wal}^{(\phi)}\|}, \quad (15)$$

where $\|\cdot\|$ denotes the L_2 vector norm. $\alpha_{k-l\phi}$ only represents the similarity between the k -th target matrix and the l -th reference matrix from the ϕ -th auxiliary dataset or objective dataset self ($\phi = 0$). In fact, we hope that the k -th target matrix can be compared to every reference matrix, and the comparison results can be scored in the range of $[-1, 1]$. Therefore, we use a softmax function to realize the weighted average. So $\omega_{k-l\phi}$, the new similarity between the k -th target matrix and the l -th reference matrix from the ϕ -th auxiliary dataset or objective dataset self ($\phi = 0$), can be calculated as the following equation:

$$\omega_{k-l\phi} = \frac{\exp(\alpha_{k-l\phi})}{\sum_{l=1}^L \sum_{\phi=0}^{\Phi} \exp(\alpha_{k-l\phi})}. \quad (16)$$

A large value of $\omega_{k-l\phi}$ means that the k -th target matrix is more similar to the l -th reference matrix from the ϕ -th auxiliary dataset or objective dataset self ($\phi = 0$). Therefore, we can use the linear combination of label matrices \mathbf{V}_{Wal} , $\mathbf{V}_{Wal}^{(1)}, \dots, \mathbf{V}_{Wal}^{(\Phi)}$ as the k -th score matrix $\mathbf{U}^{(k)}$ corresponding to the k -th target matrix \mathbf{Y}'_{ok} and $\omega_{k-l\phi}$ ($l = 1, 2, \dots, L$ and

$\phi = 0, 1, \dots, \Phi$) can be set as weight factors of the summation operation, i.e.,

$$\mathbf{U}^{(k)} = \theta^{(0)} \sum_{l=1}^L \omega_{k-l0} \mathbf{V}_{Wal} + \sum_{\phi=1}^{\Phi} \theta^{(\phi)} \sum_{l=1}^L \omega_{k-l\phi} \mathbf{V}_{Wal}. \quad (17)$$

Unlike the label matrix \mathbf{V}_{Wal} , $\mathbf{V}_{Wal}^{(1)}, \dots, \mathbf{V}_{Wal}^{(\Phi)}$, the elements of the score matrix $\mathbf{U}^{(k)}$ with the size of $m_0 \times n_0$ are in the range of $[-1, 1]$, i.e., $\mathbf{U}^{(k)} \in [-1, 1]^{m_0 \times n_0}$. $\mathbf{U}^{(k)}$ impresses the probability that there will be an outlier value or a normal value. If the value of an element in the score matrix is negative, then the position may correspond to a left outlier value. On the contrary, a positive element value may correspond to a right outlier value. We use absolute values to measure the probability and the sign (positive or negative) to indicate the classification (right or left outlier value) of outlier points.

The previous matrix completion algorithm based on DNN is usually acceptable for normal value, but not for outlier value. Take the right outlier value as an example: the value recovered by previous methods is usually smaller than the ground truth value. If we can judge which unsensed data are outlier value data and compensate them in different degrees, the data inference error of outlier value will be reduced. In fact, we achieve this goal by calculating the matrix $\mathbf{U}^{(k)}$. By compensating the matrix $\hat{\mathbf{Y}}_{ok}$, we can get a better matrix completion result. The compensation result of the k -th target matrix is as shown in Eq. (18):

$$\hat{\mathbf{O}}^{(k)} = \hat{\mathbf{Y}}_{ok} + b \mathbf{U}^{(k)}, \quad (18)$$

where $b \in \mathbb{R}_+$ is the scale parameter and named compensation factor. The value b is determined by that sensed outliers.

Intuitively, the main advantage of our model is that it enables a flexible switch between yielding predictions of normal values and outlier values.

Similarly, by processing K target matrices, we can splice all the K matrices and get the complete matrix of optimized

TABLE II
STATISTICS OF TWO EVALUATION DATASETS

	Datasets	
	Sensor-Scope	U-Air
Country/Region - City	Switzerland - Lausanne	Chinese mainland - Beijing
Subarea	57 subareas each with $30m \times 50m$	36 subareas each with $1km \times 1km$
Time slot	336 time slots each with half an hour	264 time slots each with an hour
Objetive data (Unit)	Temperature ($^{\circ}C$)	PM2.5 ($\mu g/m^3$)
Auxiliary data (Unit)	Humidity (%)	PM10 ($\mu g/m^3$)
Correlation coefficient	-0.6829	0.8863
Mean \pm Std.	6.04 ± 1.87 (Tem.) / 84.52 ± 6.32 (Hum.)	79.11 ± 81.21 (PM2.5) / 63.12 ± 48.56 (PM10)

recovery outlier value data as shown in Fig. 6 and Eq. (19):

$$\hat{\mathbf{O}} = Splice(\hat{\mathbf{O}}^{(1)}, \hat{\mathbf{O}}^{(2)}, \dots, \hat{\mathbf{O}}^{(K)}). \quad (19)$$

Finally, since values of sensed data are known, it is necessary to fix these values to sensed value instead of inferred value. The corrected output is:

$$\hat{\mathbf{O}}_{\text{fix}} = \mathbf{Y}'_o \circ \mathbf{C} + \hat{\mathbf{O}} \circ (\mathbf{1} - \mathbf{C}), \quad (20)$$

where bool matrix $\mathbf{1}$ is with all *True* elements.

From what has been discussed above, we may safely claim that we can recover the unsensed data by our proposed method. With its seemingly magic power, it can not only recover normal value data, but also serve as an important role in recovering outliers.

VI. PERFORMANCE EVALUATION

In this section, we first introduce the datasets and the baselines. Then we present performance evaluation results for our proposed method. In particular, the most important research questions are:

- **RQ1:** How does we set or select the hyper-parameters for SDMF-OV?
- **RQ2:** Does SDMF-OV really work for outliers effectively, especially compared with traditional DMF?
- **RQ3:** Does SDMF-OV improve the accuracy of matrix completion and prediction?
- **RQ4:** Is each submodule of SDMF-OV necessary for enhancing performance?
- **RQ5:** What is the training cost of SDMF-OV?

A. Datasets and Settings

For evaluating our proposed outlier value data inference problem, we applied two famous and popular urban crowdsensing datasets, including *Sensor-Scope* [37] and *U-Air* [38]. *Sensor-Scope* contained various typical sensing data of urban environment, such as humidity, temperature, etc. *U-Air* collected the sensing readings from an urban air quality monitoring systems. We provide a detailed description of these two datasets in TABLE II and more detailed information of these two datasets is shown as follows:

- The *Sensor-Scope* [37] dataset contained various typical sensing data of urban environmental from the readings of static sensors deployed in the campus of École Polytechnique Fédérale de Lausanne (EPFL). We only selected the representative temperature data for testing our outlier value data inference problem although there were also

other type data such as humidity. In order to facilitate the statistical data of different subareas, the EPFL campus with the size of about $300m \times 500m$ was divided into 10×10 grids, each with the size of $30m \times 50m$. After a simple data preprocessing, we finally obtained 57 subareas with continual sensing data readings.

- The *U-Air* [38] dataset contained various typical significant data about air quality, such as PM10 and PM2.5, by monitor stations deployed in Beijing City, China. Since there was a similar tendency from the different type data about air quality, we selected one of the most important PM2.5 value to evaluate our outlier value data inference problem. Similar to [38], we finally obtained 36 subareas, each with the size of $1km \times 1km$ from the *U-Air* dataset, and thus we worked these urban sensing data for our evaluations.

Note that although these two datasets were collected by static sensors, we can also obtain the same sensing data from mobile devices or even a human self. Moreover, these two selected sensing tasks, including temperature and PM2.5 were typical urban crowdsensing tasks and also in need of outlier value data inference. Therefore, we use these datasets in our evaluations to show the effectiveness and the improvement of our proposed urban crowdsensing problem by different methods.

B. Baselines and Measures

In order to effectively utilize the sparse sensed data to infer outlier value data, we present the matrix completion algorithm referred as SDMF-OV. We mainly compare our method with the following matrix completion algorithms:

- KNN (K-Nearest Neighbor) [39], which selects the top- K nearest sensed time slots and calculates the average value. KNN is a linear data inference method.
- GP (Gaussian Process) [40], [41], which assumes that the spatial distribution of data in the same time slot obeys the Gaussian distribution. Unlike the KNN algorithm we have introduced, GP algorithm is a non-linear data inference algorithm.
- KDE (Kernel Density Estimation) [42], which is a non-parametric test method that is used to estimate the unknown probability density function. Compared with GP, the probability density constructed by KDE is more in line with the actual data distribution.
- DMF (Deep Matrix Factorization) [22], which is a deep neural network based on matrix full rank factorization,

TABLE III
COMPARISON OF DIFFERENT DATA INFERENCE METHODS

Method	Non-linear	DNN-based	Outlier	Interdata
KNN [39]	✗	✗	✗	✗
GP [40], [41]	✓	✗	✗	✗
KDE [42]	✓	✗	✗	✗
DMF [22]	✓	✓	✗	✗
IGMC [23]	✓	✓	✗	✗
DMF-OV [1]	✓	✓	✓	✗
SDMF ^a	✓	✓	✗	✓
SDMF-OV ^b	✓	✓	✓	✓

^a It is proposed in Section 4 of this paper.

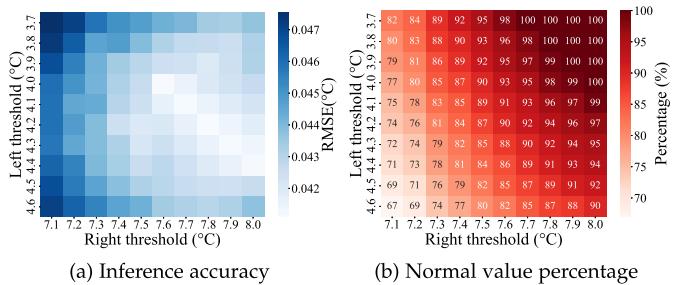
^b It is proposed in Section 4 and Section 5 of this paper.

is a popular matrix completion algorithm. Data from different subareas are still inferred one by one. However, unlike KNN, GP and KDE, when we infer a subarea's data, we also use the characteristics of the data from other subareas.

- IGMC (Inductive Graph-based Matrix Completion) [23], which is an inductive matrix completion method based on Graph Neural Networks without side information for recommender systems and is applicable to the completion of sparse spatiotemporal data matrix because of its inductive model.
- DMF-OV (Deep Matrix Factorization with exploiting Outlier Values) [1], which is a matrix completion method that takes outlier value effects into consideration. It should be noted that DMF-OV does not add the use of inter-data to the model framework.

Then, we will introduce how these baselines are applied for the two sensing tasks. For KNN, GP, and KDE, we focus on one subarea j . If there is no sensed data in the i -th time slot, KNN will collect sensed data for k time slots closest to time slot i . Then we calculate the average value of these k time slots as the estimated value of KNN for time slot i . If there are \tilde{m} time slots of sensed data, GP and KDE will build a Gaussian distribution and generate a kernel density, respectively. The mean value of the Gaussian distribution is set to be the statistical mean of this \tilde{m} sensed data, and the Gaussian distribution's variance value is set to be the statistical variance of this \tilde{m} sensed data. Then, GP generates a series of random values that obey the Gaussian distribution we built as the estimated value of the unsensed data of the subarea j . For DMF and IGMC, we transform the unsensed data inference problem into a sparse matrix completion problem. For a sparse matrix $\mathbf{Y}' \in \mathbb{R}^{m \times n}$, the DMF method divides the matrix \mathbf{Y}' into n sparse column vectors $\mathbf{y}'^{(1)}, \mathbf{y}'^{(2)}, \dots, \mathbf{y}'^{(n)} \in \mathbb{R}^m$. It means we have a dataset of a deep neural network whose batch size equals n . In addition, IGMC algorithm uses the currently popular GNN method to implement matrix completion instead of traditional DNN structure. It's easy to see that there are complex relationships and various characteristics among these baselines. Therefore, we summarize the relationships and characteristics between these methods in TABLE III.

In this paper, we use two famous urban sensing datasets with two representative types of urban crowdsensing tasks (Sparse spatiotemporal matrix completion for Temperature and



(a) Inference accuracy

(b) Normal value percentage

Fig. 7. Inference accuracy and normal value percentage under different thresholds over *Sensor-Scope*.

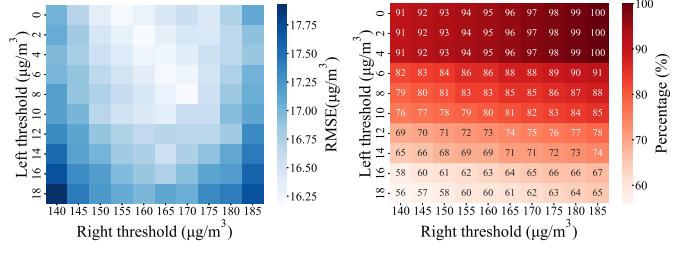


Fig. 8. Inference accuracy and normal value percentage under different thresholds over *U-Air*.

PM2.5.) For qualitative analysis, we compare our method (SDMF-OV) with traditional DMF method to verify whether our method improves the completion of outliers. We first test the effect of four hyper-parameters on the model and select the best values. For quantitative analysis, compared to KNN, GP, KDE, IGMC, etc., we verify SDMF-OV can not only improve the completion accuracy of the sparse spatiotemporal matrix, but also enhance performance of outlier classification. What's more, we also prove that each submodule of SDMF-OV is necessary by ablation Experiments. In this section, unless otherwise specified, the error we mention means Rooted Mean Square Error (RMSE).

C. Impact of Hyper-Parameters (*RQ1*)

1) *Outlier Value Threshold*: First, we test the influence of outlier value thresholds in our SDMF-OV model, where the results are shown in Figs. 7(a) and 8(a). In this paper, the right outlier value threshold ϵ_1 ("Right threshold" in Figs. 7(a) and 8(a)) and the left outlier value threshold ϵ_2 ("Left threshold" in Figs. 7(a) and 8(a)) are the most important two hyper-parameters. The setting of the right and left threshold has a great influence on the performance of the model. We infer that, a large right outlier value threshold or a small left outlier value threshold will cause more values to be classified as normal values, which in turn reduces the effectiveness of our method. On the contrary, a small right outlier value threshold or a too large left outlier value threshold will cause some normal values to be classified as outlier values, which will interfere with the model's extraction of outlier values. In other words, the frequency and range of outlier data affect the quality of the experimental results. Therefore, the selection of appropriate thresholds has a great impact on the effect of the SDMF-OV model. Empirically, we usually choose the values

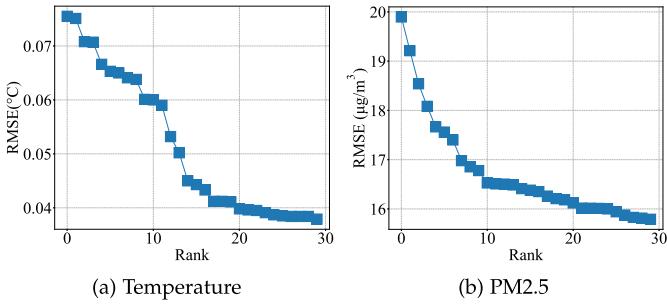


Fig. 9. Inference accuracy under different rank over *Sensor-Scope* and *U-Air*.

of $\text{Mean} \pm \text{Std}$ or $\text{Mean} \pm 2 \times \text{Std}$ to set the initial values of these two thresholds.

As shown in Figs. 7-8, we keep the sensed ratio at a constant 50% and use our proposed SDMF-OV method to recover the complete matrix. By setting different thresholds, we get different matrix completion errors. In this way, we not only test the influence of hyper-parameters ϵ_1 and ϵ_2 on the matrix completion error, but also find the best right outlier value threshold ϵ_1 and left outlier value threshold ϵ_2 that minimize the matrix completion error. Therefore, when SDMF-OV algorithm is applied to other scenarios, we should set a reasonable outlier value threshold according to the data distribution firstly, so as to make the best effect.

What's more, we also tested the percentage of normal value under different outlier thresholds, as shown in Figs. 7(b) and 8(b). We not only show the distributions of the normal value percentage, but more importantly, we find that when the outlier thresholds are set as the optimal solution, the corresponding normal value percentage is close to 95%. For example, the optimal setting of temperature task is $\epsilon_1 = 7.8$ and $\epsilon_2 = 4.2$, and the corresponding normal value percentage is just 95.19%. This is consistent with the results of our theoretical analysis.

2) *Rank of the Matrix*: Second, we test the effect of the rank's value on the results of matrix completion. We sample 50% of the data in each cycle of the two tasks for testing. In order to compare the error variation rules under different tasks more clearly, we represent six test results in the same coordinate system. As shown in Fig. 9, the errors of the two tasks are in an inverse ratio with the increasing rank. And in each task, the speed of error decline shows a trend of rapid decline first and then slow decline. A larger rank makes the structure of the neural network more complicated, and thus can obtain more information, but it also brings a longer running time. In order to balance the running time and the matrix completion error, we set $r = 10$ in subsequent experiments.

3) *Compensation Factor*: Third, we test the effect of the compensation factor b on the results of matrix completion. The compensation factor b determines the correction degree of outliers. If the value of the compensation factor b is too small, it will lead to insufficient correction, and if the value of the compensation factor b is too large, it will lead to excessive correction. Both cases will lead to the reduction of matrix completion accuracy. We tested the effect of the compensation factor b in both two urban sensing tasks. The result can be

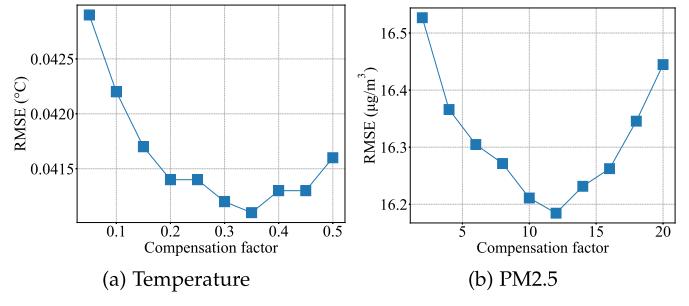


Fig. 10. Inference accuracy under different compensation factor over *Sensor-Scope* and *U-Air*.

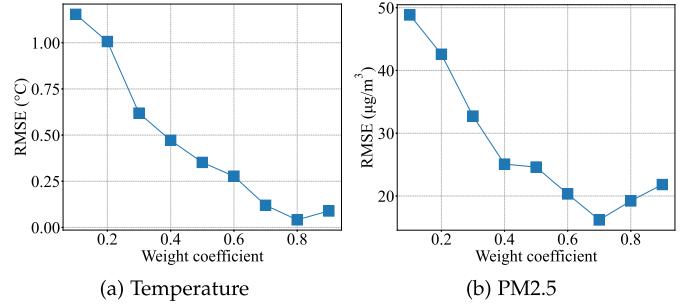


Fig. 11. Inference accuracy under different weight coefficient over *Sensor-Scope* and *U-Air*.

found in Fig. 10. Indeed, the curve image of the compensation factor b is a concave function in every task. We will select the value of b which minimizes the matrix completion error for the next series of tests ($b = 0.35$ for the dataset of *Sensor-Scope* and $b = 12$ for the dataset of *U-Air*).

4) *Weight Coefficient*: Compared with previous methods, one of the advantages of our SMDF-OV is the use of both intra- and inter- data for sparse data completion. However, the use of auxiliary datasets is not always “the more the better”. Excessive use of auxiliary datasets will result in the less use of objective dataset information, which is not conducive to high-precision matrix completion. Therefore, we try to find the most suitable weight coefficient $\theta^{(0)}$ in order to achieve the best auxiliary effect. We set the weight coefficient $\theta^{(0)}$ from 0.1 to 0.9 and then calculate the RMSE value. The experimental results are shown in Fig. 11. We found the best weight coefficient setting scheme from the experimental results: $\theta^{(0)} = 0.8$ for *Sensor-Scope* and $\theta^{(0)} = 0.7$ for *U-Air*. It indicates that the auxiliary function of auxiliary datasets is limited but promotive.

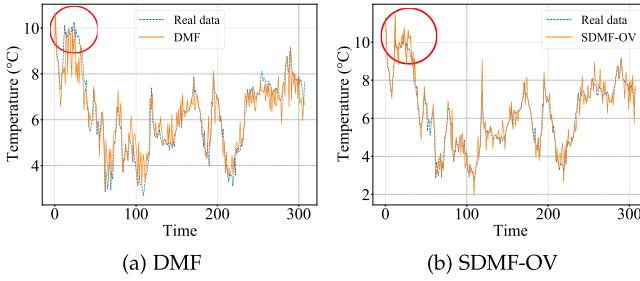
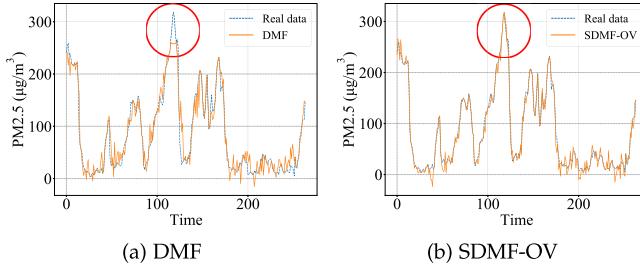
5) *Parameter Setting Summary*: From what has been tested above, we summarily provide the parameter setting of the experiment, as shown in TABLE IV. Although we did not provide the setting process of all parameters in the previous words, we provide a more complete list in TABLE IV. Readers can reproduce our experimental results for *Sensor-Scope* and *U-Air* by referring to our parameter settings. It should be noted that for more datasets, it may be necessary to reset different parameters to get better performance.

D. Complementary Effects of Outlier Values (RQ2)

1) *Time Series Analysis*: We first qualitatively verify that our method can effectively deal with outliers. We randomly

TABLE IV
PARAMETER SETTINGS

Task	Value								
	Φ	ϵ_1	ϵ_2	r	b	$\theta^{(0)}$	m_0	n_0	L
Temp.	1	7.8	4.2	10	0.35	0.8	16	4	32
PM2.5	1	160	2	10	12	0.7	24	9	32

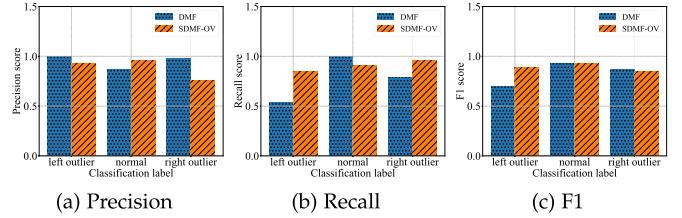
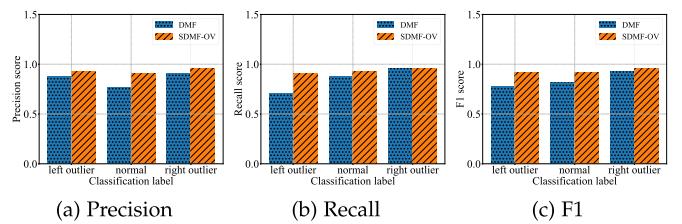
Fig. 12. Complementary effects of outlier values over *Sensor-Scope*.Fig. 13. Complementary effects of outlier values over *U-Air*.

extract 50% of the data from the two tasks, and then use DMF and SDMF-OV algorithms to complete the matrix, respectively. To show the experimental results, we randomly select a column from the matrix completion results of each dataset, and provide the ground truth of the original data simultaneously. As shown in Figs. 12 and 13, we find that SDMF-OV can recover data effectively, especially the outliers compared with the traditional DMF. Among all the two experimental results, the most obvious effect is the task over *U-Air* and the experimental result is shown in Fig. 13. As shown in Fig. 13, it is easy to find that the traditional DMF can not recover the values greater than 300 between the 100th time slot and the 150th time slot. However, when we use SDMF-OV to complete the data again, the values greater than 300 can be recovered at a lower error.

2) *Outlier Error Analysis*: Through the data inference results of DMF and SDMF-OV algorithm, we notice a phenomenon that SDMF-OV can improve the data inference effect of not only normal value data, but also outlier value data. This is because our SDMF-OV method has such a characteristic that SDMF-OV also judges a normal value as an outlier value with a small probability. In the other words, SDMF-OV fixes all data instead of only outlier value data. Therefore, we also calculate the completion errors of that column by DMF and SDMF-OV, the results of which are shown in TABLE V. It is obvious that the error of SDMF-OV is less than that of DMF, especially when considering the effect of outlier value data. The above experiments show that our proposed SDMF-OV

TABLE V
RMSE OF OUTLIER VALUES OVER BOTH TWO TASKS

Method	Temperature (°C)		PM2.5 ($\mu\text{g}/\text{m}^3$)	
	all data	only outliers	all data	only outliers
DMF	0.67	0.71	20.56	24.76
SDMF-OV	0.43	0.52	10.82	13.37
Improvement	35.8%	26.8%	47.4%	46.0%

Fig. 14. Several classical classification effect evaluation indexes over *Sensor-Scope*.Fig. 15. Several classical classification effect evaluation indexes over *U-air*.

algorithm can deal with outlier value problems and increase the accuracy of matrix completion.

3) *Outlier Classification Analysis*: Through the analysis of the theoretical parts, SDMF-OV has advantages in inferring outliers and the most important reason is that we effectively classify the outliers and normal values. So, for the inference of real data, does SDMF-OV really improve the classification level? In order to verify this view, we calculated the classification of inference results by using DMF and SDMF-OV, respectively. Due to the great difference between the number of outliers and normal value in our data set, we don't choose accuracy to measure our classification results. We calculated three classical evaluation indexes of classification results: precision score, recall score, and F1 score. As shown in Figs. 14 and 15, when the data inference method is converted from DMF to SDMF-OV, the precision score of normal value and the recall score of outlier values are improved at the same time. This means that the number of outliers incorrectly classified as normal value is decreasing, and the data classified as normal value does have more normal values. This is consistent with our original intention to solve the outlier problem.

E. Matrix Completion and Short-Term Prediction (**RQ3**)

We start to test the robustness of sensed ratio on the sparse matrix completion for the SDMF-OV algorithm. In each urban sensing task, we randomly select 50%-90% of the data in the spatiotemporal data matrix as unsensed data, and the remaining unextracted data is regarded as sensed data. We use different matrix completion methods to infer the unsensed data and calculate the matrix completion error. The experimental results are shown in Fig. 16 (upper part).

TABLE VI
INFERENCE ACCURACY UNDER DIFFERENT SENSED RATIOS OVER *Sensor-Scope*

Method	Sensed ratio								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
DMF	0.1324	0.1191	0.0934	0.0819	0.0731	0.0581	0.0523	0.0405	0.0258
SDMF	0.0949	0.0914	0.0723	0.0671	0.0621	0.0533	0.0469	0.0389	0.0257
DMF-OV	0.0708	0.0654	0.0571	0.0499	0.0439	0.0357	0.0349	0.0339	0.0247
SDMF-OV	0.0537	0.0505	0.0453	0.0422	0.0411	0.0343	0.0336	0.0332	0.0245
Improvement	0.0171	0.0149	0.0118	0.0077	0.0028	0.0014	0.0013	0.0007	0.0002

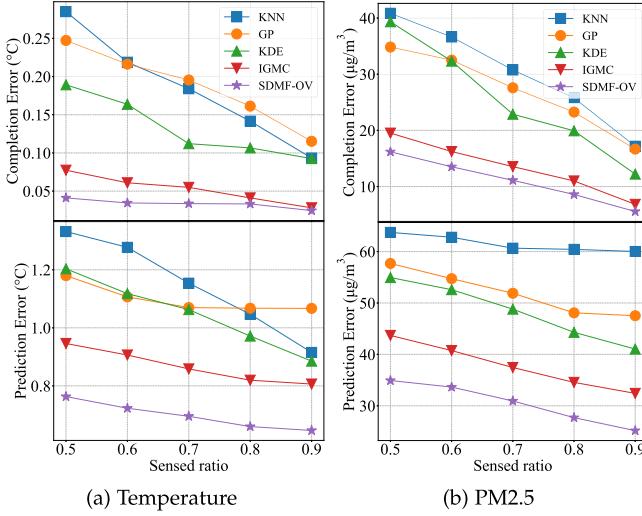


Fig. 16. Inference and prediction accuracy under different sensed ratios over *Sensor-Scope* and *U-Air*.

We can see that no matter which matrix completion algorithm is used for unsensed data inference, the matrix completion error will decrease with the sensed ratio increases. This is because the effect of data inference is related to the amount of sensed data. Simultaneously, it can also be seen that the matrix completion error of SDMF-OV method is lower than other comparison methods, especially when the data is sparse. SDMF-OV inherits the advantage of traditional deep learning methods in matrix completion, and focuses on handling outlier values, so the accuracy of matrix completion will be improved by SDMF-OV.

The matrix completion error shows that the result of data inference is very close to the real data. However, the smaller matrix completion error does not necessarily mean that the model can better fit the data's spatiotemporal characteristics. Considering that the accuracy of time series prediction is closely related to the spatiotemporal characteristics of historical data, we use the matrix completion results of SDMF-OV and other comparison methods to predict the next time slot by Gated Recurrent Unit (GRU) [43]. The experimental results of next time slot prediction are shown in Fig. 16 (lower part). We can find that the prediction error of SDMF-OV is also lower than other comparison methods, which indicates that SDMF-OV not only optimizes the outlier value inference, but also extracts spatiotemporal features effectively. Similarly, the increase in the sensed ratio is also helpful in improving the prediction accuracy. This means that the more mobile users we recruit, the better urban crowdsensing services we can provide. Obviously, the experimental result is consistent with both our intuitive inferences and theoretical analysis results.

At the end of this experiment, we will discuss the differences among different matrix completion methods. From the results of matrix completion and short-term prediction, we can find that KNN, GP, and KDE algorithms have low accuracy of matrix completion and prediction. KNN, GP, and KDE only use simple statistics instead of exploring particularly complex spatiotemporal relationships. It may get better results in some cases, but it is not a general method. The matrix completion accuracy of IGMC is higher than KNN, GP, and KDE and close to SDMF-OV. But the prediction accuracy of IGMC is obviously lower than SDMF-OV. This is because IGMC is mostly used in recommendation systems, and the effect of capturing time series correlation is poorer. Because KNN, GP, KDE, and IGMC do not consider the influence of outliers, the data inference error in some locations may be large, which leads to the final matrix completion effect not being as good as SDMF-OV. On the other hand, as the sensed ratio increases, the outlier value data in unsensed data decreases, and the advantage of SDMF-OV algorithm may be weakened in many tasks.

F. Ablation Experiments (*RQ4*)

Compared with the previous data inference methods, the structure of SDMF-OV is very complex. Without considering some subtle processing, the utilization of multiple datasets and the construction of outlier memory network bring a huge amount of computation. So, is each sub module necessary? If the addition and removal of a module have little impact on the results of data inference, it seems that this module is not necessary. In order to verify the necessity of each sub module, we conducted ablation experiments. We use DMF, DMF-OV and SDMF to verify the necessity of multiple datasets and outlier memory network. DMF is the most original matrix completion method, which uses a single dataset and does not focus on outliers. SDMF is added a multiple dataset module and DMF-OV is added a outlier memory network based on DMF, respectively. Except that we tested more sensed ratios (10%-90%), we conducted the same experiment as Subsection VI-E. The experimental results are shown in TABLE VI - TABLE IX, respectively. We find that the advantages of SDMF-OV become more obvious as the sensed ratio decreases.

Through the experimental results, we found that SDMF-OV is the smallest in both completion error and prediction error. This means that each sub module of SDMF-OV is indispensable. Although SDMF-OV is so complex, it is necessary to increase the amount of calculation in order to obtain higher completion and prediction accuracy. In addition, we also

TABLE VII
PREDICTION ACCURACY UNDER DIFFERENT SENSED RATIOS OVER *Sensor-Scope*

Method	Sensed ratio								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
DMF	0.9549	0.9233	0.8851	0.8489	0.7957	0.7874	0.7725	0.7439	0.6659
SDMF	0.9345	0.8982	0.8662	0.8345	0.7948	0.7617	0.7343	0.7202	0.6615
DMF-OV	0.9141	0.8783	0.8409	0.8089	0.7669	0.7399	0.7051	0.6633	0.6509
SDMF-OV	0.8834	0.8532	0.8217	0.7868	0.7628	0.7228	0.6949	0.6597	0.6462
Improvement	0.0307	0.0251	0.0192	0.0221	0.0041	0.0171	0.0102	0.0036	0.0047

TABLE VIII
INFERENCE ACCURACY UNDER DIFFERENT SENSED RATIOS OVER *U-Air*

Method	Sensed ratio								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
DMF	33.8589	29.4172	25.9169	22.5781	18.8601	15.4776	12.4089	9.9446	6.3982
SDMF	33.1532	29.2604	25.7761	21.5533	18.5921	15.0147	11.9862	9.4114	6.1139
DMF-OV	31.8491	28.3565	23.8294	20.1593	16.7592	13.6481	11.1998	8.7478	5.8056
SDMF-OV	28.1653	25.4225	22.4006	19.1944	16.1844	13.5212	11.1095	8.5937	5.5564
Improvement	3.6838	2.9340	1.4288	0.9649	0.5748	0.1269	0.0903	0.1541	0.2492

TABLE IX
PREDICTION ACCURACY UNDER DIFFERENT SENSED RATIOS OVER *U-Air*

Method	Sensed ratio								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
DMF	53.1945	49.9563	46.6291	42.7459	38.1412	37.1123	32.9519	30.7681	25.8105
SDMF	52.2998	48.8786	45.4098	41.6074	36.9077	36.1752	32.3743	29.3896	25.7031
DMF-OV	51.0813	47.7574	44.3578	40.0603	35.6669	34.4436	31.2008	28.2632	25.1839
SDMF-OV	47.8332	44.5238	41.7498	39.0716	34.9174	33.6389	30.9344	27.6941	25.1648
Improvement	3.2481	3.2336	2.6080	0.9887	0.7495	0.8047	0.2664	0.5691	0.0191

found that the outlier memory network seems to play a more important role than multi dataset module because the error of DMF-OV is less than SDMF.

G. Training Cost (*RQ5*)

1) *Data Scale Requirements*: In previous several paragraphs, we tested the performance of SDMF-OV in data inference accuracy. However, the data scale required to train an SDMF-OV model is still an important technical index. Therefore, in this part, we start with only one time slot and gradually increase the amount of data and then calculate the data inference error. It is very difficult to train the model due to very few time slots. Therefore, we test at least 10 time slots. According to the experimental results in Fig. 17, we found that the data inference error decreases with the increase of data scale in both two crowdsensing tasks. What's more, while the data scale increases to a certain extent, the fluctuation of data inference error will tend to be stable. For the temperature task of *Sensor-Scope* dataset, at least about 100 time slots are required to make the model perform best. When we focus on the PM2.5 task of *U-Air* dataset, This number will change to about 70. It means that our model does not need too many training samples to work. In other words, our SDMF-OV is easier to apply in actual MCS tasks.

2) *Training Time*: Finally, we test the methods proposed in this paper in terms of training time, as shown in TABLE X. Our experiment platform is equipped with Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz and 8 GB RAM, and we implement the sparse spatiotemporal crowdsensing scheme using Python 3 programming language with a popular toolkit

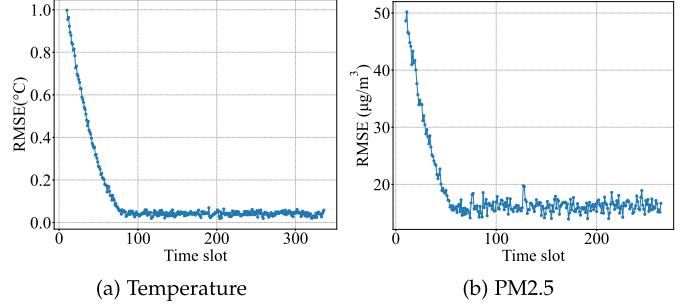


Fig. 17. Inference accuracy under different input time slots over *Sensor-Scope* and *U-Air*.

TABLE X
TRAINING TIME OVER *Sensor-Scope* AND *U-Air*

Dataset - Task	<i>Sensor-Scope</i> - Temp.	<i>U-Air</i> - PM2.5
Training Time (s)	56.372	48.263

PyTorch. For the temperature task of *Sensor-Scope* dataset, the SDMF-OV costs 56.372s to recover a complete matrix for all sensing time slots and the SDMF-OV costs 48.263s for the PM2.5 task of *U-Air* dataset. The size of the dataset used in our experiment is dozens of subareas and hundreds of time slots. Therefore, the training time of SMDF-OV on two classic tasks is less than 1 minute. But in most practical application cases, with the increase of data set size, it may take longer than the experiment in this paper.

VII. CONCLUSION

In this paper, we investigate the problem of MCS via spatiotemporal data matrix completion with unsensed outlier value data recovery. Specifically, we use sensed data of sparse

spatiotemporal matrix to infer unsensed normal and outlier values. In order to solve this problem, we propose a data completion method named SDMF-OV, which is based on DMF with an outlier value model. Specifically, we first conduct an intra- and inter-data-based matrix completion network named Stacked Deep Matrix Factorization (SDMF). The loss function of SDMF is no longer the previous MSE loss function, but replaced by an Outlier Value Loss (OVL) function that can effectively deal with outliers. Then, a spatiotemporal outlier value memory network is added for enhancing the outlier inference. Experiments on two popular urban sensing datasets show that the proposed SDMF-OV method can complete the sparse matrix with a high accuracy and recover outlier value data effectively.

REFERENCES

- [1] E. Wang, M. Zhang, Y. Yang, Y. Xu, and J. Wu, "Exploiting outlier value effects in sparse urban crowdsensing," in *Proc. IEEE/ACM 29th Int. Symp. Quality Service (IWQoS)*, Jun. 2021, pp. 1–10.
- [2] 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.
- [3] F. Restuccia, P. Ferraro, T. S. Sanders, S. Silvestri, S. K. Das, and G. L. Re, "FIRST: A framework for optimizing information quality in mobile crowdsensing systems," *ACM Trans. Sensor Netw.*, vol. 15, no. 1, pp. 1–35, Feb. 2019.
- [4] Y. Zhang and X. Zhang, "Price learning-based incentive mechanism for mobile crowd sensing," *ACM Trans. Sens. Netw.*, vol. 17, no. 2, pp. 1–17, Jun. 2021.
- [5] Y. Liu *et al.*, "Vernier: Accurate and fast acoustic motion tracking using mobile devices," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 754–764, Feb. 2021.
- [6] C. Xiang *et al.*, "CARM: Crowd-sensing accurate outdoor RSS maps with error-prone smartphone measurements," *IEEE Trans. Mobile Comput.*, vol. 15, no. 11, pp. 2669–2681, Nov. 2016.
- [7] C. Xiang, P. Yang, C. Tian, H. Cai, and Y. Liu, "Calibrate without calibrating: An iterative approach in participatory sensing network," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 2, pp. 351–361, Feb. 2015.
- [8] X. Fan *et al.*, "BuildSenSys: Reusing building sensing data for traffic prediction with cross-domain learning," *IEEE Trans. Mobile Comput.*, vol. 20, no. 6, pp. 2154–2171, Jun. 2021.
- [9] Y. Liu, J. Comden, Z. Liu, and Y. Yang, "Online resource provisioning for wireless data collection," *ACM Trans. Sensor Netw.*, vol. 18, no. 1, pp. 1–27, Feb. 2022.
- [10] 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.
- [11] K. Xie, X. Li, X. Wang, G. Xie, J. Wen, and D. Zhang, "Active sparse mobile crowd sensing based on matrix completion," in *Proc. SIGMOD*, 2019, pp. 195–210.
- [12] X. Li *et al.*, "Quick and accurate false data detection in mobile crowd sensing," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1339–1352, Jun. 2020.
- [13] 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.
- [14] Z. Yu, H. Ma, B. Guo, and Z. Yang, "Crowdsensing 2.0," *Commun. ACM*, vol. 64, no. 11, pp. 76–80, 2021.
- [15] J. Wang *et al.*, "Learning-assisted optimization in mobile crowd sensing: A survey," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 15–22, Jan. 2019.
- [16] Y. Gong, Z. Li, J. Zhang, W. Liu, B. Chen, and X. Dong, "A spatial missing value imputation method for multi-view urban statistical data," in *Proc. IJCAI*, 2020, pp. 1310–1316.
- [17] J. Zhou and Z. Huang, "Recover missing sensor data with iterative imputing network," in *Proc. Workshops 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 209–216.
- [18] B. Huang, G. Mao, Y. Qin, and Y. Wei, "Pedestrian flow estimation through passive WiFi sensing," *IEEE Trans. Mobile Comput.*, vol. 20, no. 4, pp. 1529–1542, Dec. 2021.
- [19] 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, pp. 1–20, Oct. 2018.
- [20] S. He and K. G. Shin, "Steering crowdsourced signal map construction via Bayesian compressive sensing," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 1016–1024.
- [21] T. Liu, Y. Zhu, Y. Yang, and F. Ye, "ALC²: When active learning meets compressive crowdsensing for urban air pollution monitoring," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9427–9438, Dec. 2019.
- [22] J. Fan and J. Cheng, "Matrix completion by deep matrix factorization," *Neural Netw.*, vol. 98, pp. 34–41, Feb. 2018.
- [23] M. Zhang and Y. Chen, "Inductive matrix completion based on graph neural networks," in *Proc. ICLR*, 2020, pp. 1–14.
- [24] D. Ding, M. Zhang, X. Pan, M. Yang, and X. He, "Modeling extreme events in time series prediction," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1114–1122.
- [25] D. Zhang, L. Wang, H. Xiong, and B. Guo, "4W1H in mobile crowd sensing," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 42–48, Aug. 2014.
- [26] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2289–2302, Nov. 2013.
- [27] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Earphone: An end-to-end participatory urban noise mapping system," in *Proc. IPSN*, 2010, pp. 105–116.
- [28] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang, "An efficient prediction-based user recruitment for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 17, no. 1, pp. 16–28, Jan. 2018.
- [29] C. Zhang, Y. Guo, H. Du, and X. Jia, "PFCrowd: Privacy-preserving and federated crowdsourcing framework by using blockchain," in *Proc. IEEE/ACM 28th Int. Symp. Quality Service (IWQoS)*, Jun. 2020, pp. 1–10.
- [30] X. Wei, Y. Wang, S. Gao, and Y. Tang, "Data quality aware task allocation under a feasible budget in mobile crowdsensing," in *Proc. IEEE/ACM 26th Int. Symp. Quality Service (IWQoS)*, Jun. 2018, pp. 1–2.
- [31] J. Li, J. Wu, and Y. Zhu, "Data utility maximization when leveraging crowdsensing in machine learning," in *Proc. IEEE/ACM 26th Int. Symp. Quality Service (IWQoS)*, Jun. 2018, pp. 1–6.
- [32] J. Bian, H. Xiong, Y. Fu, and S. K. Das, "CSWA: Aggregation-free spatial-temporal community sensing," in *Proc. AAAI Conf. Artif. Intell.*, S. A. McIlraith and K. Q. Weinberger, Eds., 2018, pp. 2087–2094.
- [33] E. Wang *et al.*, "Deep learning-enabled sparse industrial crowdsensing and prediction," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6170–6181, Sep. 2021.
- [34] L. Cui, F. R. Yu, and Q. Yan, "When big data meets software-defined networking: SDN for big data and big data for SDN," *IEEE Netw.*, vol. 30, no. 1, pp. 58–65, Jan./Feb. 2016.
- [35] W. Liu, Y. Yang, E. Wang, L. Wang, D. Zeghlache, and D. Zhang, "Multi-dimensional urban sensing in sparse mobile crowdsensing," *IEEE Access*, vol. 7, pp. 82066–82079, 2019.
- [36] J. Han, H. Liu, H. Zhu, H. Xiong, and D. Dou, "Joint air quality and weather prediction based on multi-adversarial spatiotemporal networks," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 4081–4089.
- [37] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "SensorScope: Application-specific sensor network for environmental monitoring," *ACM Trans. Sen. Netw.*, vol. 6, no. 2, pp. 1–17, Mar. 2010.
- [38] 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*, Aug. 2013, pp. 1436–1444.
- [39] L. Wang *et al.*, "CCS-TA: Quality-guaranteed online task allocation in compressive crowdsensing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. (UbiComp)*, 2015, pp. 683–694.
- [40] F. Yin and F. Gunnarsson, "Distributed recursive Gaussian processes for RSS map applied to target tracking," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 3, pp. 492–503, Apr. 2017.
- [41] H. C. Yen and C. C. Wang, "Cross-device Wi-Fi map fusion with Gaussian processes," *IEEE Trans. Mobile Comput.*, vol. 16, no. 1, pp. 44–57, Jan. 2017.
- [42] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065–1076, Sep. 1962.
- [43] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.



En Wang (Member, IEEE) received the B.E. degree in software engineering and the M.E. and Ph.D. degrees in computer science and technology from Jilin University, Changchun, in 2011, 2013, and 2016. He was a joint Ph.D. student at the Department of Computer and Information Science, Temple University. He is currently a Professor with the Department of Computer Science and Technology, Jilin University, Changchun. His current research interests include mobile computing, crowd intelligence, and data mining.



Bo Yang is currently a Professor with the College of Computer Science and Technology, Jilin University. He is the Director of the Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, China. His current research interests include data mining, complex network analysis, and self-organized and self-adaptive multi-agent systems, with applications to knowledge engineering and intelligent health informatics.



Mijia Zhang (Student Member, IEEE) received the B.E. and M.E. degrees in electronic information specialty from Harbin Engineering University, Harbin, China, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree in computer system architecture with the College of Computer Science and Technology, Jilin University, Changchun, China. His current research interests include sparse mobile crowdsensing, neural networks, spatiotemporal data inference, and matrix completion.



Yongjian Yang received the B.E. degree in automation from the Jilin University of Technology, Changchun, Jilin, China, in 1983, the M.E. degree in computer communication from the Beijing University of Post and Telecommunications, Beijing, China, in 1991, and the Ph.D. degree in software and theory of computer from Jilin University, Changchun, in 2005. He is currently a Professor and a Ph.D. Supervisor with Jilin University, the Director of Key Laboratory under the Ministry of Information Industry, and the Standing Director of the Communication Academy. His research interests include network intelligence management, wireless mobile communication and services, and wireless mobile communication.



Wenbin Liu (Member, IEEE) received the B.S. degree in physics and the Ph.D. degree in computer science and technology from Jilin University, Changchun, China, in 2012 and 2020, respectively. He was also a joint Ph.D. student at the Wireless Networks and Multimedia Services Department, Telecom SudParis, Institut Mines-Telecom, Evry, France. He is currently a Post-Doctoral Researcher with the Dingxin Scholar Program, College of Computer Science and Technology, Jilin University. His research interests include mobile crowdsensing, mobile computing, and ubiquitous computing.



Jie Wu (Fellow, IEEE) is the Director of the Center for Networked Computing and a Laura H. Carnell Professor with Temple University. He is the Director of International Affairs at College of Science and Technology. He was the Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and an Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. He was a Program Director at the National Science Foundation and was a Distinguished Professor at Florida Atlantic University. He regularly publishes in scholarly journals, conference proceedings, and books. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. He was an IEEE Computer Society Distinguished Visitor, an ACM Distinguished Speaker, and the Chair of the IEEE Technical Committee on Distributed Processing (TCDP). He is a CCF Distinguished Speaker. He was a recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award. He was the General Co-Chair of IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016, and IEEE CNS 2016, and the Program Co-Chair of IEEE INFOCOM 2011 and CCF CNCC 2013. He serves on several editorial boards, including IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON SERVICES COMPUTING, Journal of Parallel and Distributed Computing, and Journal of Computer Science and Technology.



Haoyi Xiong (Senior Member, IEEE) received the Ph.D. degree in computer science from Telecom SudParis, Université Pierre et Marie Curie, Paris, France, in 2015. From 2016 to 2018, he was a Tenure-Track Assistant Professor at the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO, USA (formerly known as the University of Missouri at Rolla). From 2015 to 2016, he was a Post-Doctoral Research Associate at the Department of Systems and Information Engineering, University of Virginia, Charlottesville, VA, USA. He is currently the Principal Research and Development Architect and a Researcher with the Big Data Laboratory, Baidu Research, Beijing, China.