2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)

Exploiting Outlier Value Effects in Sparse Urban CrowdSensing

1st En Wang
Computer Science and Technology
Jilin University
Changchun, China
wangen@jlu.edu.cn

2nd Mijia Zhang
Computer Science and Technology
Jilin University
Changchun, China
zhangmj19@mails.jlu.edu.cn

3rd Yongjian Yang
Computer Science and Technology
Jilin University
Changchun, China
yyj@jlu.edu.cn

4th Yuanbo Xu*
Computer Science and Technology
Jilin University
Changchun, China
yuanbox@jlu.edu.cn

5th Jie Wu
Computer and Information Sciences
Temple University
Philadelphia, USA
jiewu@temple.edu

2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS) | 978-1-6654-1494-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/IWQOS52092.2021.9521275

***Abstract*—Sparse spatiotemporal data completion is crucial in Mobile CrowdSensing for urban application scenarios. In fact, accurate urban data completion can enhance data expression, improve urban analysis, and ultimately guide city planning. However, it is a non-trivial task to consider outlier values caused by the special events (e.g., parking peak, traffic congestion, or festival parade) in spatiotemporal data completion because of the following challenges: 1) the rarity and unpredictability, 2) the inconsistency compared to normal values, and 3) the complex spatiotemporal relations. In spite of the considerable improvements, recent deep learning-based methods overlook the existence of outlier values, which results in misidentifying these values. To this end, focusing on spatiotemporal data, we propose a matrix completion method that takes outlier value effects into consideration. Specifically, an outlier value model is proposed by adding a memory network and modifying the loss function to traditional matrix completion. Along this line, we extract the features of outlier values and further efficiently complete and predict the unsensed data. Finally, we conduct both qualitative and quantitative experiments on three different datasets, and the results demonstrate that the performance of our method outperforms the state-of-the-art baselines.**

***Index Terms*—Sparse Mobile CrowdSensing, matrix completion, outlier value, memory network**

I. Introduction

With the rapid development of the Quality of Service (QoS) in Internet of Things (IoT), Mobile CrowdSensing (MCS) [1], which recruits mobile users carrying IoT devices to collect various urban sensing data [2]–[4], has become an increasingly powerful sensing paradigm. In real-world scenarios, MCS can not only help the government monitor the current urban status, but also predict the future situation by utilizing historical data. To reduce sensing cost of MCS, some researchers introduce data inference techniques into MCS for enhancing QoS [5]. Thus, Sparse MCS [6], [7] appears, which senses some limited areas and explores the correlations to infer the rest data for the unsensed areas.

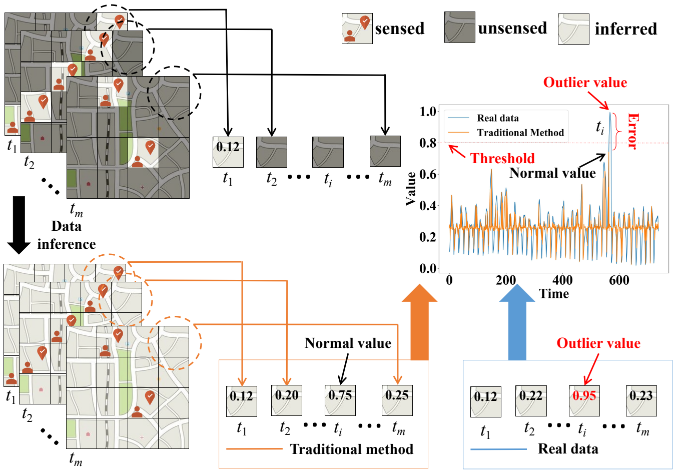


Fig. 1: An example to describe the outlier value effect in data inference problem.

The problem description for exploiting outlier value effects in sparse urban crowdsensing is shown in Fig. 1. The whole city map is equally divided into 4×4 grids, and we have collected sensing data of m time segments. Due to the sensing budget constraint and uncertain user mobilities, we can hardly sense all 16 subareas' data at each time segment [8], [9]. In this case, some vital information may be lost, which results in missing emergency data. For example, a temperature monitoring system sustains to monitor the temperature of each

This work was supported in part by the National Natural Science Foundations of China under Grant 61772230 and Grant 61972450, in part by the National Natural Science Foundations of China for Young Scholars under Grant 61702215, in part by the Natural Science Foundations of Jilin Province under Grant 20190201022JC, in part by the National Science Key Lab Fund Project under Grant 61421010418, in part by the Innovation Capacity Building Project of Jilin Province Development and Reform Commission under Grant 2020C017-2, and in the part by the Changchun Science and Technology Development Project under Grant 18DY005.

*Yuanbo Xu is the corresponding author.

978-0-7381-3207-5/21/$31.00 © 2021 IEEE

978-1-6654-1494-4/21/$31.00 ©2021 IEEE

urban subarea. If there is a fire disaster (which leads to a high temperature) in a forest park and the system does not sense the abnormal temperature increasing in this subarea, it will cause a serious loss of lives and properties. Generally, we call this abnormal sensing data an "outlier value," which is significantly higher (or lower) than the normal value (non-outlier value) distribution. Actually, the occurrence of outlier values may always satisfy some potential rules. For example, the occurrence probability of a forest fire in winter is higher than that in summer. Hence, if we focus on outlier values in sensed data, the unsensed outlier value data may be efficiently inferred and even predicted in time.

Actually, how to infer outlier values in sparse urban crowdsensing is still an open issue. As shown in Fig. 1, some existing data inference algorithms (e.g., DMF [10], IGMC [11], etc.) can help us complete the unsensed data. However, focusing on the circled grid where outlier values should appear, we find that traditional inference has a large inference error (0.95 minus 0.75) compared with the real data. This situation may lead the inference outlier value to be lower than the threshold (0.8). Thus, the vital outlier value is regarded as a normal value, which is unacceptable. The main reason for this phenomenon is that there are rare samples related to outlier values for training the existing models. In common situations, outlier values only account for less than 5% of the total data. The imbalance between normal value data and outlier value data may lead to the following two problems: 1) *Underfitting*: The model regards all sensed value data as normal value data, so unsensed outlier value data cannot be recovered entirely. 2) *Overfitting*: The model fits both the normal and outlier value data perfectly, so some unsensed normal value data are recovered with large error. Hence, traditional Deep Neural Network-based matrix completion methods cannot effectively recover these data when we do not sense enough outlier value data. Therefore, how to *recover outlier values from such rare outlier value data* is the first challenge.

From the perspective of occurrence probability distribution, the normal value distribution is relatively concentrated and the occurrence probability is always high. While, the outlier value distribution is relatively deconcentrated and the occurrence probability is always low. In other words, the probability distributions of normal values and outlier values are completely different. If we deal with outlier values in the same way as normal values for designing and training models, the data inference models may be underfitting and ineffective. Therefore, how to *deal with inconsistent data distribution between normal and outlier values* is the second challenge.

After recovering historical data completely and learning the characteristics of historical outlier value data, we could predict the future state through neural network-based methods such as RNN, LSTM, GRU, etc. To explore the complex spatiotemporal relationships, we hope to recover complete data from sparse data at both the current (data inference) and future (data prediction) time. In both cases, they need the help of spatiotemporal relationship among the sensed data. Therefore, how to *extract the complex spatiotemporal relationship of outlier value data* is the third challenge.

To deal with the challenges above, we must make full use of sparse spatiotemporal data. If an outlier value is unsensed at the current time segment, we can try to determine whether other time segments have similar spatiotemporal data distribution to the current time segment. Therefore, the key to solving the problem is to compare the similarity of time segments and explore the space relations among different areas. To this end, this paper focuses on outlier values in sparse urban crowdsensing problems via Sparse MCS, aiming to recover the full map data, especially for outlier value data. Along with this line, we propose Deep Matrix Factorization with exploiting Outlier Value (DMF-OV), which focuses on predicting outlier value data. Specifically, considering the complex spatiotemporal correlations of sensed data, we first apply Deep Matrix Factorization (DMF) 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 which indicates whether the empty value is an outlier data. With the help of the outlier value model, the proposed sparse matrix completion method could efficiently complete and predict the unsensed data for the city-scale map.

Our work has the following contributions:

- We formalize the sparse urban crowdsensing problem, with the goal of recovering the unsensed normal and outlier value data from the sparse sensed data.
- We propose an urban crowdsensing method named DMF-OV, which aims to solve the problem of inferring outlier value data from sparse sensed data based on DMF. Compared with the traditional methods, our method can effectively extract the complex spatiotemporal relationship between the outlier value data and the normal value data.
- We evaluate the proposed method on three real-world datasets with three typical urban sensing tasks. The results verify the effectiveness in improving the recovery and prediction accuracy with the sparse sensed data when considering outlier values.

The reminder of this paper is organized as follows. Section II reviews related works. Section III presents the system model and the problem formulation. In section IV, we describe the details of our DMF-OV. We evaluate the performance of our approach through extensive simulations in Section V, followed by the conclusion in Section VI.

II. Related Work

A. Mobile CrowdSensing

Mobile CrowdSensing (MCS) technology utilized mobile devices carried by users to perform series of urban crowdsensing tasks [1], [12]. For example, in the application of urban environmental MCS, measurements (e.g., noise levels [13], traffic speeds [14], etc.) enabled the mapping of various large-scale urban environmental phenomena by involving the common person. Compared to traditional sensor networks, MCS had a number of unique characteristics that brought opportunities to researchers and users. To provide QoS, some existing efforts [15], [16] recruited as many users as possible to collect data. Such a type of approaches could indeed provide better service, however, collecting a large number of users meant a huge cost. In our application scenario of the outlier value data inference, it did help to be aware of such large amounts of data, but it was unrealistic in most cases. Therefore, traditional MCS could not be applied directly. In fact, by utilizing spatiotemporal correlations among subareas, we needed only a small amount of the data to infer the data of other unsensed areas [6]. In this way, if the data inference algorithm performs well, the cost of data sensing will be greatly reduced.

B. Sparse MCS

In recent years, researchers have developed many urban crowdsensing systems based on Sparse MCS, which first sensed limited subareas and then infered the entire map. An example of the application of noise monitoring would reveal how effective the idea was. Rana *et al.* [13] built an urban noise monitoring system that used compressive sensing method to realize inferring a fine-grained urban noise map from the randomly and incompletely sensed data. What's more, He *et al.* [17] and Liu *et al.* [18] proposed a Sparse MCS-based urban air pollution and signal mapping systems. Furthermore, an incentive design had been added to guide users to sense more subareas of data. Wei *et al.* [19] also 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.* [20] focused on the critical problem that which data instances should be collected to maximize the performance of the trained model under the budget limit. They proposed the multi-round crowdsensing framework and came up with a novel data utility model to bridge the gap between the performance of the trained model and the collected series of instances effectively. Although existing Sparse MCS technology had not solve our problem directly, it was an inspiration for our work.

C. Matrix Completion for Spatiotemporal Data

For spatiotemporal data, sparse MCS was essentially the completion of the sparse spatiotemporal matrix. Matrix Factorization (MF) was a classical matrix completion algorithm, which took 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 [10] 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.* [21] 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 a 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.* [11] proposed an Inductive Graph-based Matrix Completion (IGMC) model without using any side information for recommender systems and applicable to the completion of sparse spatiotemporal data matrix because of its inductive model. In the matrix completion field, existing work usually ignored the impact of outlier values.

TABLE I: Main notations

Symbol	**Type**	**Meaning**
m, m_0	$\mathbb{N}_+$	number of time segments, the length of the time segments contained in each sampling period
n, $\tilde{n}$	$\mathbb{N}_+$	number of all subareas and sensed subareas
i	$\mathbb{N}_+$	index of a time segment
j	$\mathbb{N}_+$	index of a subarea
r	$\mathbb{N}_+$	rank we assume of the complete matrix
$\mathbf{Y}$, $\mathbf{Y}'$, $\hat{\mathbf{Y}}$	$\mathbb{R}^{m\times n}$	ground truth of the complete data, the sensed sparse data, the estimated data
$\mathbf{C}$	$\{0,1\}^{m\times n}$	label of which grids are sensed
$\mathbf{V}$, $\mathbf{V}'$	$\{0,\pm1\}^{m\times n}$	ground truth of complete label matrix, the label matrix of sensed sparse data
$\mathbf{Z}$	$\mathbb{R}^{r\times n}$	input of DMF
K, L	$\mathbb{N}_+$	number of target and reference matrices
k, l	$\mathbb{N}_+$	index of a target matrix and a reference matrix
$\mathbf{Y}'^{(k)}$	$\mathbb{R}^{m_0\times n}$	k-th target matrix
$\hat{\mathbf{Y}}^{(k)}$	$\mathbb{R}^{m_0\times n}$	completion matrix of $\mathbf{Y}'^{(k)}$ by DMF and OVL
$\mathbf{Z}^{(k)}$	$\mathbb{R}^{r\times n}$	embedding matrix of $\mathbf{Y}'^{(k)}$ by DMF and OVL
$\mathbf{W}^{(l)}$	$\mathbb{R}^{m_0\times n}$	l-th reference matrix
$\hat{\mathbf{Y}}_{\mathrm{W}}^{(l)}$	$\mathbb{R}^{m_0\times n}$	completion matrix of $\mathbf{W}^{(l)}$ by DMF and OVL
$\mathbf{Z}_{\mathrm{W}}^{(l)}$	$\mathbb{R}^{r\times n}$	embedding matrix of $\mathbf{W}^{(l)}$ by DMF and OVL
$\mathbf{V}_{\mathrm{W}}^{(l)}$	$\{0,\pm1\}^{m_0\times n}$	sparse label matrix of $\mathbf{W}^{(l)}$
α_{kl}	$\mathbb{R}$	inner product of vec($\mathbf{Z}^{(k)}$) and vec($\mathbf{Z}_{\mathrm{W}}^{(l)}$)
ω_{kl}	$[0,1]$	similarity of vec($\mathbf{Z}^{(k)}$) and vec($\mathbf{Z}_{\mathrm{W}}^{(l)}$)
$\mathbf{U}^{(k)}$	$[-1,1]^{m_0\times n}$	probability that there will be an outlier value
$\hat{\mathbf{O}}^{(k)}$	$\mathbb{R}^{m_0\times n}$	complete matrix of $\mathbf{Y}'^{(k)}$ by DMF-OV
$\hat{\mathbf{O}}$	$\mathbb{R}^{m\times n}$	matrix composed of all matrix $\hat{\mathbf{O}}^{(k)}$
$\hat{\mathbf{O}}_{\text{fix}}$	$\mathbb{R}^{m\times n}$	corrected output of DMF-OV

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.

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 ($\tilde{n} << n$) for each time segment. In order to represent which grids are sensed in the i-th time segment, we introduce the logical vector $\mathbf{c}_{1\times n}^{(i)} = [c_{i1}, c_{i2}, \ldots, c_{in}]$ to denote which grids are sensed. If the subarea j has been sensed, $c_{ij} = 1$, otherwise, $c_{ij} = 0$. $\mathbf{y}'^{(i)}_{1\times n} = [y'_{i1}, y'_{i2}, \ldots, y'_{in}]$ denotes the sparse sensed data. The unsensed data are recorded as some meaningless values (e.g., ∞). $\mathbf{y}_{1\times n}^{(i)} = [y_{i1}, y_{i2}, \ldots, y_{in}]$ denotes the complete data vector which includes both the sensed value data and the unsensed value data.

After m time segments, we can combine vectors to the following matrices: $\mathbf{C}_{m\times n} = [\mathbf{c}^{(1)\mathrm{T}}, \mathbf{c}^{(2)\mathrm{T}}, \dots, \mathbf{c}^{(m)\mathrm{T}}]^{\mathrm{T}}$ denotes which grids are sensed for each time segment; sparse matrix $\mathbf{Y'}_{m\times n} = [\mathbf{y'}^{(1)\mathrm{T}}, \mathbf{y'}^{(2)\mathrm{T}}, \dots, \mathbf{y'}^{(m)\mathrm{T}}]^{\mathrm{T}}$ denotes the whole sensed data of m time segments; complete matrix $\mathbf{Y}_{m\times n} = [\mathbf{y}^{(1)\mathrm{T}}, \mathbf{y}^{(2)\mathrm{T}}, \dots, \mathbf{y}^{(m)\mathrm{T}}]^{\mathrm{T}}$ denotes the ground truth of complete data. Then the relationship of the three matrices we introduce is as follows:

$$\mathbf{Y}' = \mathbf{Y} \circ \mathbf{C}, \tag{1}$$

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 data matrix $\hat{\mathbf{Y}}$:

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

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 outlier values (-1), right outlier values (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 segment. The calculation method of v_{ij} is as follows:

$$v_{ij} = v(y_{ij}) = \begin{cases} 1, & y_{ij} > \epsilon_1 \\ -1, & y_{ij} < \epsilon_2 \\ 0, & \text{else} \end{cases}, \tag{3}$$

where the large constant ϵ_1 and small constant ϵ_2 are thresholds. By solving each element of matrix $\mathbf{Y}_{m\times n}$ with (3), we can get the label matrix $\mathbf{V}_{m\times n} = v(\mathbf{Y}_{m\times n})$. Similarly, we can also get the sparse label matrix $\mathbf{V'}_{m\times n} = v(\mathbf{Y'}_{m\times n})$ from the sparse sensed matrix $\mathbf{Y'}_{m\times n}$.

B. Problem Formulation

Problem [Sparse Spatiotemporal Matrix Completion via Outlier Values Model]: Given a sparse matrix $\mathbf{Y}'_{m\times n}$, we need to accomplish the following two tasks:

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

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

$$obj = ||\hat{\mathbf{Y}} - \mathbf{Y}|| + ||v(g(\mathbf{Y}')) - \mathbf{V}||. \tag{4}$$

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. It will be a two-step process.

IV. Spatiotemporal Matrix Completion with Outlier Values Model

Inspired by Deep Matrix Factorization (DMF) [10], we design a novel method of matrix completion based on DMF. Given a sparse matrix $\mathbf{Y}'_{m\times n}$, DMF uses a Deep Neural Network (DNN) to generate a complete matrix, as shown in Fig. 2. We hope that the output vector of the neural network is as close as possible to each column of the sparse matrix $\mathbf{Y}'_{m\times n}$. Unlike traditional DNNs, we need to treat the input vector $\mathbf{z}_{r\times 1}$ 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 elements.

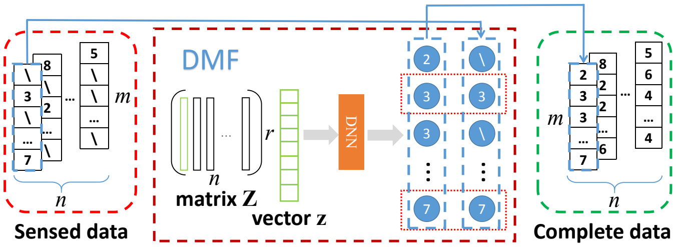


Fig. 2: Structure of traditional DMF.

We use the function $f(\cdot)$ to represent the structure of the DMF neural network. The loss function of traditional DMF is a MSE loss function with two regularization penalty terms and as shown in (5):

$$L_{\mathrm{MSE}} = \frac{1}{2mn}||(\mathbf{Y} - f(\mathbf{Z})) \circ \mathbf{C}||^2 + \lambda\pi(f) + \mu||\mathbf{Z}||^2, \tag{5}$$

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 [22], we design our Outlier Value Loss (OVL) function as follows:

$$L_{\mathrm{OVL}} = -\sum_{(i,j)\in\mathcal{A}} (1 - \beta^{(v_{ij})})(1 - \frac{p_{ij}^{(v_{ij})}}{\gamma})^{\gamma} \log(p_{ij}^{(v_{ij})}), \tag{6}$$

where $\mathcal{A} = \{(i,j)|c_{ij} = 1, 1 \le i \le m, 1 \le j \le n, (i,j) \in \mathbb{N}_+^2\}$. $\beta^{(0)}$, $\beta^{(-1)}$ 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 $[p_{1j}^{(0)}, p_{1j}^{(-1)}, p_{1j}^{(1)}, \dots, p_{mj}^{(0)}, p_{mj}^{(-1)}, p_{mj}^{(1)}]^{\mathrm{T}} \in \mathbb{R}^{3m}$, where $\mathbb{R}$ means the set of real numbers.

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

$$L_{\mathrm{mix}} = \xi L_{\mathrm{MSE}} + (1 - \xi) L_{\mathrm{OVL}}, \tag{7}$$

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.

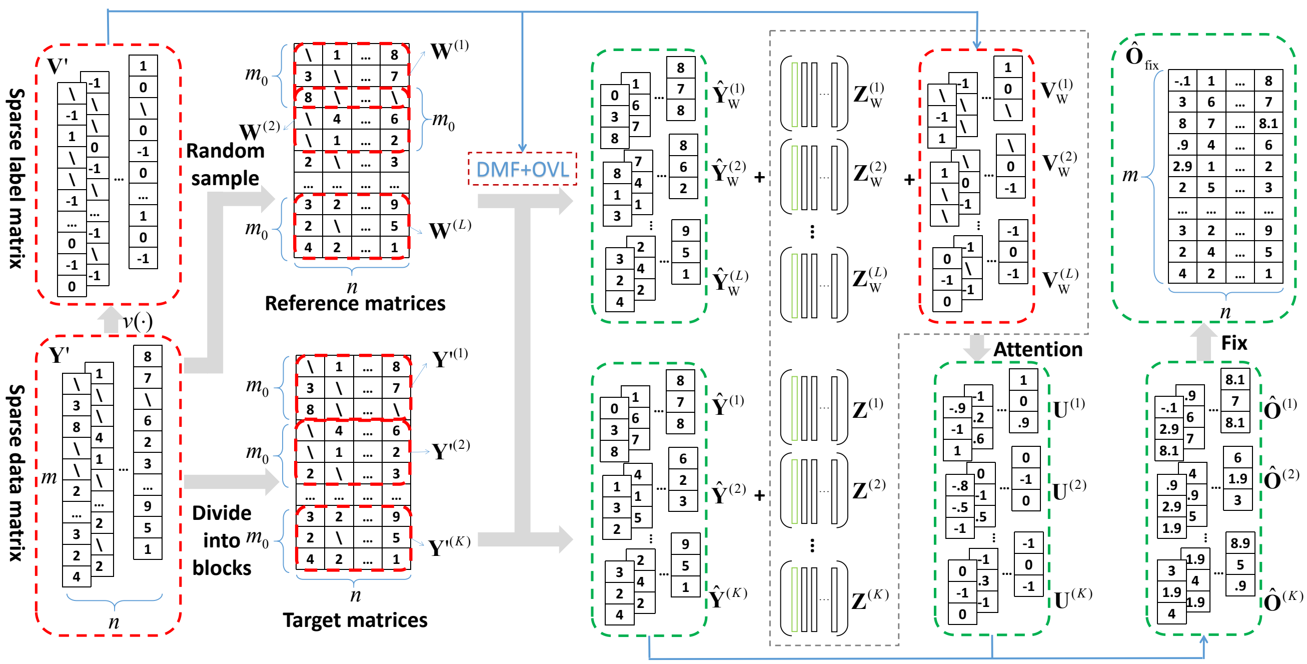


Fig. 3: Framework overview.

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

The detailed DMF-OV algorithm is introduced in the following several paragraphs. In order to make it more easy for readers to understand, we give a pseudo code flow table of DMF-OV in Alg. 1, and an example framework figure is also given in Fig. 3.

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. The meaning of the window is the size of this certain shape. As shown in Fig. 3 (left part), we randomly sample a sequence of windows by $\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, ..., \mathbf{W}^{(L)}$, where L is the size of the memory network and the number of rows in these matrices is m_0. We hope to obtain the characteristics of the outlier value data in sensed data from $\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, ..., \mathbf{W}^{(L)}$, so we call these L matrices reference matrices. Meanwhile, we divide the matrix $\mathbf{Y}'$ into $\mathbf{Y}'^{(1)}, \mathbf{Y}'^{(2)}, ..., \mathbf{Y}'^{(K)}$ of the same size in the way shown in Fig. 3. It is obvious that $K = [\frac{m}{m_0}]$. In particular, when $\frac{m}{m_0}$ is not an integer value which means that there will be remaining rows of the whole sparse data matrix, we will set $K = [\frac{m}{m_0}] + 1$, and $\mathbf{Y}'^{([\frac{m}{m_0}]+1)}$ will consist of the last m_0 rows of the whole sparse data matrix. In the following, for the sake of discussion, we assume that $\frac{m}{m_0}$ is an integer value. In other words, we assume that the whole sparse data matrix can just be divided into $K = \frac{m}{m_0}$ matrices. We want to recover outlier values in these K matrices one by one, so we call these K matrices target matrices. These K target matrices are our goals to recover.

We find that these target matrices and reference matrices are all sparse matrices. Most matrix operations are based on a complete matrix rather than a sparse matrix. Sparse matrix is not convenient for our subsequent processing of target matrices and reference matrices. Therefore, it is necessary to get the complete matrices first, even if they have not yet recovered outlier values.

Then, we propose applying the DMF module with the mixed loss function to complete each target matrix and each reference matrix to build a memory network to remember characteristics of outlier value data. 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 mainly consists of the following two modules:

- **Embedding Module** $\mathbf{Z}_{\mathrm{W}} = \{\mathbf{Z}_{\mathrm{W}}^{(1)}, \mathbf{Z}_{\mathrm{W}}^{(2)}, \cdots, \mathbf{Z}_{\mathrm{W}}^{(L)}\}$: $\mathbf{Z}_{\mathrm{W}}^{(l)} \in \mathbb{R}^{r\times n}$ is the latent representation of window l.

Algorithm 1 Deep Matrix Factorization with Exploiting Outlier Value

```
Input: $\mathbf{Y'}_{m\times n} = [\mathbf{y'}^{(1)\mathrm{T}}, \mathbf{y'}^{(2)\mathrm{T}}, \dots, \mathbf{y'}^{(m)\mathrm{T}}]^{\mathrm{T}}$ and logical
    matrix $\mathbf{C}_{m\times n} = [\mathbf{c}^{(1)\mathrm{T}}, \mathbf{c}^{(2)\mathrm{T}}, \dots, \mathbf{c}^{(m)\mathrm{T}}]^{\mathrm{T}}$
Output: $\hat{\mathbf{O}}_{\mathrm{fix}}$
Calculate sparse label matrix $\mathbf{V'}_{m\times n} = v(\mathbf{Y'}_{m\times n})$;
Divide $\mathbf{Y'}_{m\times n}$ into $K$ blocks: $\mathbf{Y'}^{(1)}$, $\mathbf{Y'}^{(2)}$,..., $\mathbf{Y'}^{(K)}$;
Random Sample $\mathbf{Y'}_{m\times n}$, get reference matrices: $\mathbf{W}^{(1)}$,
    $\mathbf{W}^{(2)}$,..., $\mathbf{W}^{(L)}$;
Random Init $\mathbf{Z}^{(1)}$, $\mathbf{Z}^{(2)}$,..., $\mathbf{Z}^{(K)}$, $\mathbf{Z}_{\mathrm{W}}^{(1)}$, $\mathbf{Z}_{\mathrm{W}}^{(2)}$,..., $\mathbf{Z}_{\mathrm{W}}^{(L)}$;
Build the Neural Networks by using $f(\cdot)$,
    $count = 0$;
while not convergent and $count <$ MAX_ITER do
    for $k$ is from $1$ to $K$ do
        Fix $f(\cdot)$, $\mathbf{Z}^{(k)}$,
        calculate $\hat{\mathbf{Y}}^{(k)} = f(\mathbf{Z}^{(k)})$,
        and then calculate and reduce $L_{\mathrm{mix}}$ by (7);
    end for
    for $l$ is from $1$ to $L$ do
        Fix $f(\cdot)$, $\mathbf{Z}_{\mathrm{W}}^{(l)}$,
        calculate $\hat{\mathbf{Y}}_{\mathrm{W}}^{(l)} = f(\mathbf{Z}_{\mathrm{W}}^{(l)})$,
        and then calculate and reduce $L_{\mathrm{mix}}$ by (7);
    end for
    $count := count + 1$;
end while
for $l$ is from $1$ to $L$ do
    Calculate $\mathbf{V}_{\mathrm{W}}^{(l)} = v(\hat{\mathbf{Y}}_{\mathrm{W}}^{(l)})$;
end for
for $k$ is from $1$ to $K$ do
    for $l$ is from $1$ to $L$ do
        Calculate $\alpha_{kl} = \mathrm{vec}(\mathbf{Z}^{(k)})^{\mathrm{T}}\mathrm{vec}(\mathbf{Z}_{\mathrm{W}}^{(l)})$;
    end for
    for $l$ is from $1$ to $L$ do
        Calculate $\omega_{kl} = \frac{\exp(\alpha_{kl})}{\sum_{l=1}^{L}\exp(\alpha_{kl})}$;
    end for
    Calculate $\mathbf{U}^{(k)} = \sum_{l=1}^{L}\omega_{kl}\mathbf{V}_{\mathrm{W}}^{(l)}$;
    Calculate $\mathbf{O}^{(k)} = \hat{\mathbf{Y}}^{(k)} + b\mathbf{U}^{(k)}$;
end for
return $\hat{\mathbf{O}}_{\mathrm{fix}} = \mathbf{Y'} \circ \mathbf{C} + \hat{\mathbf{O}} \circ \bar{\mathbf{C}}$.
```

- **Label Matrix Module**
  $\mathbf{V}_{\mathrm{W}} = \{\mathbf{V}_{\mathrm{W}}^{(1)}, \mathbf{V}_{\mathrm{W}}^{(2)}, \cdots, \mathbf{V}_{\mathrm{W}}^{(L)}\}$: $\mathbf{V}_{\mathrm{W}}^{(l)} \in \{-1, 0, 1\}^{m_0\times n}$ is the outlier value label matrix of window l.

At each submatrix k, we use DMF to complete the matrix with $\hat{\mathbf{Y}}^{(k)} = f(\mathbf{Z}^{(k)})$. Thus, $\mathbf{Z}$ could be regarded as a potential representation of $\hat{\mathbf{Y}}$. As we have discussed, $\hat{\mathbf{Y}}^{(k)}$ 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. In order to measure the similarity between the target matrix and the reference matrix by cosine similarity, we utilize the matrix vec operator $\mathrm{vec}(\cdot)$ to convert a matrix into a vector. We abbreviate $\mathrm{vec}(\mathbf{Z}^{(k)})$ as vector $\overrightarrow{\mathbf{z}}^{(k)}$ and abbreviate $\mathrm{vec}(\mathbf{Z}_{\mathrm{W}}^{(l)})$ as vector $\overrightarrow{\mathbf{z}}_{\mathrm{W}}^{(l)}$. Then we can easily use cosine similarity to measure the similarity between the k-th target matrix and the l-th reference matrix. The mathematical expression of the cosine similarity α_{kl} between $\overrightarrow{\mathbf{z}}^{(k)}$ and $\overrightarrow{\mathbf{z}}_{\mathrm{W}}^{(l)}$ can be expressed as (8) shown:

$$\alpha_{kl} = \frac{\overrightarrow{\mathbf{z}}^{(k)\mathrm{T}}\overrightarrow{\mathbf{z}}_{\mathrm{W}}^{(l)}}{||\overrightarrow{\mathbf{z}}^{(k)}|| \cdot ||\overrightarrow{\mathbf{z}}_{\mathrm{W}}^{(l)}||}, \tag{8}$$

where $||\cdot||$ denotes the L_2 vector norm. α_{kl} only represents the similarity between the k-th target matrix and the l-th reference matrix. In fact, we hope that the k-th target matrix can be compared with every reference matrix, and the comparison results can be scored in the range of $[0, 1]$. Therefore, we use a softmax function to realize the weighted average. So ω_{kl}, the new value of the similarity between the k-th target matrix and the l-th reference matrix, can be calculated as:

$$\omega_{kl} = \frac{\exp(\alpha_{kl})}{\sum_{l=1}^{L}\exp(\alpha_{kl})}. \tag{9}$$

According to (9), we can clearly find that $\omega_{k1}, \omega_{k2}, \cdots, \omega_{kL}$ constrained by (10). Our subsequent processing will depend on this important equation.

$$\omega_{k1} + \omega_{k2} + \cdots + \omega_{kL} = 1. \tag{10}$$

A large value of ω_{kl} means that the k-th target matrix is more similar to the l-th reference matrix. Therefore, we can use the linear combination of label matrices $\mathbf{V}_{\mathrm{W}}^{(1)}$, $\mathbf{V}_{\mathrm{W}}^{(2)}$, $\cdots$, $\mathbf{V}_{\mathrm{W}}^{(L)}$ as the k-th score matrix $\mathbf{U}^{(k)}$ corresponding to the k-th target matrix $\mathbf{Y'}^{(k)}$ and ω_{k1}, ω_{k2}, $\cdots$, ω_{kL} can be set as weight factors, i.e.,

$$\mathbf{U}^{(k)} = \sum_{l=1}^{L}\omega_{kl}\mathbf{V}_{\mathrm{W}}^{(l)}. \tag{11}$$

Unlike label matrix $\mathbf{V}_{\mathrm{W}}^{(1)}$, $\mathbf{V}_{\mathrm{W}}^{(2)}$, $\cdots$, $\mathbf{V}_{\mathrm{W}}^{(L)}$, the elements of the score matrix $\mathbf{U}^{(k)}$ with the size of $m_0 \times n$ are in the range of $[0, 1]$, i.e., $\mathbf{U}^{(k)} \in [-1, 1]^{m_0\times n}$. $\mathbf{U}^{(k)}$ 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, it means that 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 values.

The traditional 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 traditional 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 $\hat{\mathbf{Y}}^{(k)}$, we can get a better matrix completion result. The compensation result of the k-th target matrix is as shown in (12):

$$\mathbf{O}^{(k)} = \hat{\mathbf{Y}}^{(k)} + b\mathbf{U}^{(k)}, \tag{12}$$

where $b \in \mathbb{R}_+$ is the scale parameter. The value b is determined by that sensed outlier value data.

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 combine all the K matrices and get the complete matrix of optimized recovery outlier value data as follows:

$$\hat{\mathbf{O}} = [\mathbf{O}^{(1)\mathrm{T}}, \mathbf{O}^{(2)\mathrm{T}}, \ldots, \mathbf{O}^{(K)\mathrm{T}}]^{\mathrm{T}}. \tag{13}$$

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

$$\hat{\mathbf{O}}_{\text{fix}} = \mathbf{Y}' \circ \mathbf{C} + \hat{\mathbf{O}} \circ \bar{\mathbf{C}}, \tag{14}$$

where $\bar{\mathbf{C}}$ denotes the logical operation of the negation of bool matrix $\mathbf{C}$.

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 outlier value data.

V. 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 main research questions are:

- **RQ1**: Does our method really work for outlier value data effectively?
- **RQ2**: Does our method improve the accuracy of matrix completion and prediction?
- **RQ3**: What are the influences of hyper-parameters in the model?

A. Datasets

For evaluating our proposed outlier value data inference problem, we applied three famous and popular urban crowdsensing datasets, including *Sensor-Scope* [23], *U-Air* [24], and *Parking in Birmingham* [25]. *Sensor-Scope* contained various typical sensing data of urban environmental, such as humidity, temperature, etc. *U-Air* collected the sensing readings from a urban air quality monitoring systems, and *Parking in Birmingham* includes the parking capacity data and the number of parking cars. We provide a detailed description of these three datasets in Table II.

Note that although all of these three datasets were collected by using static sensors, we could also obtain the same sensing data from mobile devices (or even human self). Moreover, these three selected sensing tasks, including temperature, PM2.5, and parking occupancy rate 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.

B. Baselines

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

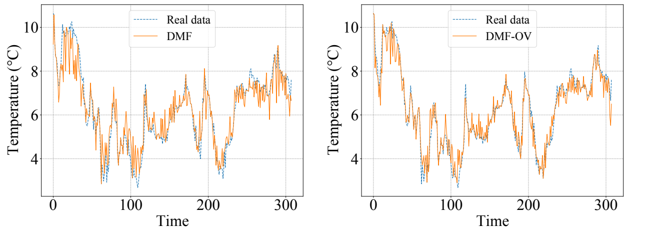
- KNN [26], which selects the top-K nearest sensed time segments and calculates the average value. KNN algorithm is a linear data inference algorithm.
- GP (Gaussian Process) [27], [28], which assumes that the spatial distribution of data in the same time segment obeys the Gaussian distribution. Unlike the KNN algorithm we have introduced, GP algorithm is a non-linear data inference algorithm.
- DMF [10], which is a deep neural network based on matrix full rank factorization, is a popular matrix completion algorithm. Data from different subareas are still inferred one by one. However, unlike the KNN and the GP algorithms we have introduced, when we infer a subarea's data, we also use the characteristics of the data from other subareas.
- IGMC [11], which is an inductive matrix completion method based on GNN (Graph Neural Networks) without using side information for recommender systems and applicable to the completion of sparse spatiotemporal data matrix because of its inductive model.

Then, we will introduce how these baselines are applied for the three sensing tasks. For KNN and GP, we focus on one subarea j. If there is no sensed data in the i-th time segment, KNN will collect sensed data for k time segments closest to time segment i. Then we calculate the average value of these k time segments as the estimated value of KNN for the time segment i. If there are $\tilde{m}$ time segments of sensed data, GP will build a Gaussian distribution. 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 $\mathbf{Y}'_{m\times n}$, the DMF method divides the matrix $\mathbf{Y}'_{m\times n}$ into n sparse column vectors $\mathbf{y}'^{(1)}_{m\times 1}$, $\mathbf{y}'^{(2)}_{m\times 1}$, $\cdots$, $\mathbf{y}'^{(n)}_{m\times 1}$. That 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.

In this paper, we use three famous urban sensing datasets with three representative types of urban crowdsensing tasks (Sparse spatiotemporal matrix completion for Temperature, PM2.5 and Parking occupancy rate.) For qualitative analysis, we compare our method (DMF-OV) with traditional DMF method to verify whether our method improves the completion of outlier value data. For quantitative analysis, comparing with KNN, GP, IGMC, and DMF, we verify our method can

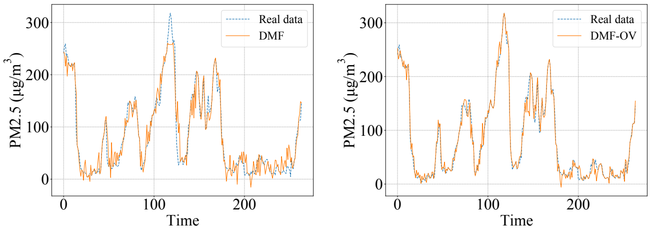
TABLE II: Statistics of three evaluation datasets.

	Datasets		
	Sensor-Scope	*U-Air*	*Parking in Birmingham*
City	Lausanne (Switzerland)	Beijing (China)	Birmingham (UK)
Data	Temperature	PM2.5	Parking occupancy rate
Subarea	57 subareas each with 50 × $30m^2$	36 subareas each with 1000 × $1000m^2$	30 parking lots
Period & Duration	$0.5h$ & $7d$	$1h$ & $11d$	$0.5h$ & $77d$
Mean ± Std. (Unit)	6.04 ± 1.87 (°C)	79.11 ± 81.21 (μg/m^3)	53.6 ± 26.3 (%)



(a) DMF (b) DMF-OV

Fig. 4: Complementary effects of outlier values over *Sensor-Scope*.



(a) DMF (b) DMF-OV

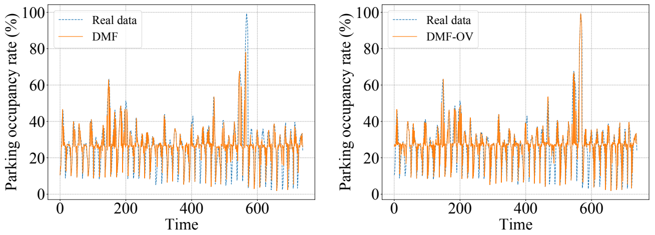
Fig. 5: Complementary effects of outlier values over *U-Air*.

improve the completion accuracy of the sparse spatiotemporal matrix. What's more, we give the effect of two hyper-parameters on the model. In this section, unless otherwise specified, the error we mention means Rooted Mean Square Error (RMSE).

In addition, our experiment platform is equipped with Intel(R) Core(TM) CPU i5-3317U CPU @ 1.70GHz and 4.00 GB RAM, and we implement the urban crowdsensing scheme using Python language with a famous open source machine learning framework Pytorch.

C. Complementary Effects of Outlier Values (*RQ1*)

We first qualitatively verify that our method can effectively deal with outlier values. We randomly extract 50% of the data from the three tasks, and then use DMF and DMF-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 give the ground truth of the original data simultaneously. As shown in Figs. 4, 5, and 6, we find that DMF-OV algorithm can recover data effectively, especially the outlier value data compared with the traditional DMF algorithm. Among all the three experimental results, the most obvious effect is the task over *U-Air* and the experimental result is shown in Fig. 5. As shown in Fig. 5, it is easy to find that the traditional DMF algorithm can not recover the values greater than 300 between the 100th time segment and the 150th time segment. However, when we use DMF-OV algorithm to complete the data again, the values greater than 300 can be recovered at a lower error. We also find a similar phenomenon in the experimental results of the *Parking in Birmingham* dataset, as shown in Fig. 6.



(a) DMF (b) DMF-OV

Fig. 6: Complementary effects of outlier values over *Parking in Birmingham*.

TABLE III: RMSE of outlier values over all three tasks.

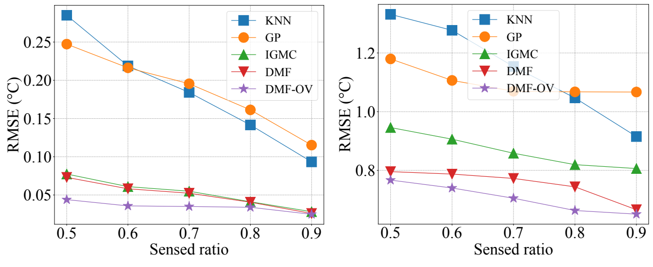
	DMF	DMF-OV
Temperature (°C)	0.67	0.45
PM2.5 (μg/m^3)	20.56	11.74
Parking occupancy rate (%)	10.4	8.9

Through the data inference results of DMF and DMF-OV algorithm, we notice a phenomenon that DMF-OV can improve the data inference effect of not only normal value data but also outlier value data. This is because our DMF-OV method has such a characteristic that DMF-OV also judges a normal value as a outlier value with a small probability. In the other words, DMF-OV fixes all data instead of only outlier value data. Therefore, we also calculate the completion errors of that column by DMF and DMF-OV, the results of which are shown in Table III. It is obvious that the error of DMF-OV is less than that of DMF, especially when considering the effect of outlier value data.

The above experiments show that our proposed DMF-OV algorithm could deal with outlier value problems and increase the accuracy of matrix completion.

D. Matrix Completion and Short-term Prediction (*RQ2*)

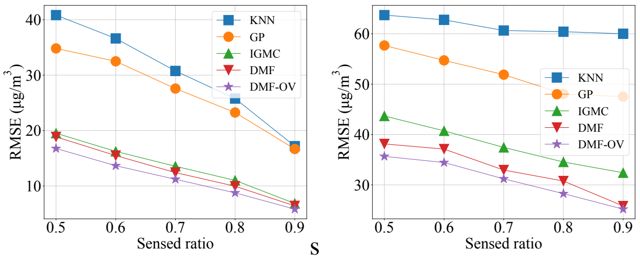
We start to test the sensed ratio's effect on the sparse matrix completion for the DMF-OV algorithm. In each urban sensing task, we randomly select 50%-90% of the data in the spatiotemporal data matrix as unsensed data, and the remaining



(a) Inference task

(b) Prediction task

Fig. 7: Inference and prediction accuracy under different sensed ratios over *Sensor-Scope*.



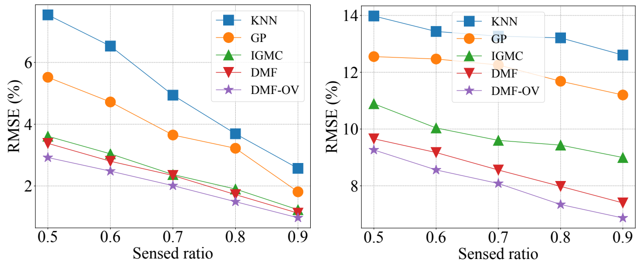
(a) Inference task

(b) Prediction task

Fig. 8: Inference and prediction accuracy under different sensed ratios over *U-Air*.

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 Figs. 7(a), 8(a), and 9(a). We can find out that no matter which matrix completion algorithm is used for unsensed data inference, the matrix completion error will decrease with the sensed ratio increase. This is because the effect of data inference is related to the amount of sensed data. Simultaneously, it can also be found that the matrix completion error of DMF-OV method is lower than other comparison methods, especially when the data is sparse. DMF-OV inherits the advantages of DMF in matrix completion, and focuses on handling outlier values, so the accuracy of matrix completion will be improved by DMF-OV.

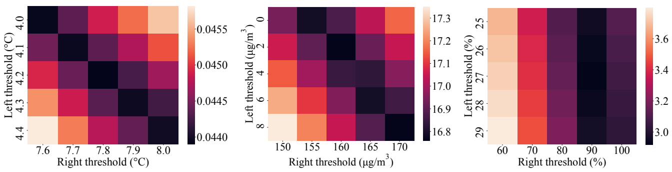
The matrix completion error shows that the result of data inference is very close to the real data. However, the smaller matrix completion error does not necessarily mean that the model can better fit the data's spatiotemporal characteristics. Considering that the accuracy of time series prediction is closely related to the spatiotemporal characteristics of historical data, we use the matrix completion results of DMF-OV and other comparison methods to predict the next time segment by Gated Recurrent Unit (GRU) [29]. The experimental results of next time segment prediction are shown in Figs. 7(b), 8(b), and 9(b). We can find that the prediction error of DMF-OV is also lower than other comparison methods, which indicates that DMF-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. It means that the more mobile users



(a) Inference task

(b) Prediction task

Fig. 9: Inference and prediction accuracy under different sensed ratios over *Parking in Birmingham*.



(a) Temperature

(b) PM2.5

(c) Occupancy Rate

Fig. 10: Inference accuracy under different hyper-parameters over *Sensor-Scope*, *U-Air*, and *Parking in Birmingham*.

we recruit, the better urban crowdsensing services we can provide. Obviously, the experimental result is consistent with our intuitive inference.

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 and GP algorithms have low accuracy of matrix completion and prediction. KNN and GP 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 and DMF is relatively high, and their experimental results are similar. But the prediction accuracy of IGMC is not as good as DMF. This is because IGMC is mostly used in recommendation systems, and the effect of capturing time series correlation is poorer than DMF. Therefore, when dealing with urban crowdsensing tasks, we prefer to combine the OV model with DMF instead of IGMC. Because these two methods do not consider the influence of outlier value data, the data inference error in some locations may be large, which leads to the final matrix completion effect not being as good as DMF-OV. On the other hand, with the increase in the sensed ratio, the outlier value data in unsensed data becomes less and less, and the advantage of DMF-OV algorithm may be weakened in many tasks.

*E. Impact of Hyper-parameters **(RQ3)***

Finally we test the influence of hyper-parameters in our DMF-OV model, where the results are shown in Fig. 10. In this paper, the right outlier value threshold ϵ_1 ("Right threshold" in Fig. 10) and the left outlier value threshold ϵ_2 ("Left threshold" in Fig. 10) are the most important 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 DMF-OV model. Empirically, we usually choose the value of $Mean \pm Std.$ to set the initial value of the thresholds.

As shown in Fig. 10, we keep the sensed ratio at a constant 50% and use our proposed DMF-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 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 DMF-OV algorithm is applied to other scenarios, we should first set reasonable outlier value threshold according to the data distribution, so as to make the model play the best effect.

VI. Conclusion

In this paper, we solve the problem of sparse spatiotemporal matrix completion and unsensed outlier value data recovery. In detail, we use sensed data of sparse spatiotemporal matrix to infer unsensed normal and outlier values. In order to solve this problem, we propose an urban sensing method named DMF-OV, which is based on DMF with an outlier value model. Specifically, an outlier value model is proposed by adding a memory network and modifying the loss function from traditional matrix completion. Experiments on three popular urban sensing datasets show that the proposed DMF-OV method can complete the sparse matrix with a high accuracy and recover outlier value data effectively. Moreover, the actual application areas of this paper are not limited to these three tasks. Our proposed method can also be applied for other urban sensing scenarios, such as traffic speed monitor, congestion detection, etc.

References

[1] 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, 2011.

[2] C. Xiang, P. Yang, C. Tian, L. Zhang, H. Lin, F. Xiao, M. Zhang, and Y. Liu, "CARM: crowd-sensing accurate outdoor RSS maps with error-prone smartphone measurements," *IEEE Trans. Mob. Comput.*, vol. 15, no. 11, pp. 2669–2681, 2016.

[3] C. Xiang, P. Yang, C. Tian, H. Cai, and Y. Liu, "Calibrate without calibrating: An iterative approach in participatory sensing network," *IEEE Trans. Parallel Distributed Syst.*, vol. 26, no. 2, pp. 351–361, 2015.

[4] X. Fan, C. Xiang, C. Chen, P. Yang, L. Gong, X. Song, P. Nanda, and X. He, "Buildsensys: Reusing building sensing data for traffic prediction with cross-domain learning," *IEEE Trans. Mob. Comput.*, vol. 20, no. 6, pp. 2154–2171, 2021.

[5] L. Wang, D. Zhang, D. Yang, A. Pathak, C. Chen, X. Han, H. Xiong, and Y. Wang, "SPACE-TA: cost-effective task allocation exploiting intradata and interdata correlations in sparse crowdsensing," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 2, pp. 20:1–20:28, 2018.

[6] K. Xie, X. Li, X. Wang, G. Xie, J. Wen, and D. Zhang, "Active sparse mobile crowd sensing based on matrix completion," in *SIGMOD 2019*. ACM, 2019, pp. 195–210.

[7] A. Abedin, S. H. Rezatofighi, Q. Shi, and D. C. Ranasinghe, "Sparsesense: Human activity recognition from highly sparse sensor datastreams using set-based neural networks," in *IJCAI 2019*. ijcai.org, 2019, pp. 5780–5786.

[8] 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 *IJCAI 2020*. ijcai.org, 2020, pp. 1310–1316.

[9] J. Zhou and Z. Huang, "Recover missing sensor data with iterative imputing network," in *AAAI 2018*, ser. AAAI Workshops, vol. WS-18. AAAI Press, 2018, pp. 209–216.

[10] J. Fan and J. Cheng, "Matrix completion by deep matrix factorization," *Neural Networks*, vol. 98, pp. 34–41, 2018.

[11] M. Zhang and Y. Chen, "Inductive matrix completion based on graph neural networks," in *ICLR 2020*. OpenReview.net, 2020.

[12] D. Zhang, L. Wang, H. Xiong, and B. Guo, "4w1h in mobile crowd sensing," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 42–48, 2014.

[13] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: an end-to-end participatory urban noise mapping system," in *IPSN 2010*. ACM, 2010, pp. 105–116.

[14] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *IEEE Trans. Mob. Comput.*, vol. 12, no. 11, pp. 2289–2302, 2013.

[15] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang, "An efficient prediction-based user recruitment for mobile crowdsensing," *IEEE Trans. Mob. Comput.*, vol. 17, no. 1, pp. 16–28, 2018.

[16] C. Zhang, Y. Guo, H. Du, and X. Jia, "Pfcrowd: Privacy-preserving and federated crowdsourcing framework by using blockchain," in *IWQoS 2020*. IEEE, 2020, pp. 1–10.

[17] S. He and K. G. Shin, "Steering crowdsourced signal map construction via bayesian compressive sensing," in *INFOCOM 2018*. IEEE, 2018, pp. 1016–1024.

[18] T. Liu, Y. Zhu, Y. Yang, and F. Ye, "Alc^2: When active learning meets compressive crowdsensing for urban air pollution monitoring," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9427–9438, 2019.

[19] X. Wei, Y. Wang, S. Gao, and Y. Tang, "Data quality aware task allocation under a feasible budget in mobile crowdsensing," in *IWQoS 2018*. IEEE, 2018, pp. 1–2.

[20] J. Li, J. Wu, and Y. Zhu, "Data utility maximization when leveraging crowdsensing in machine learning," in *IWQoS 2018*. IEEE, 2018, pp. 1–6.

[21] E. Wang, M. Zhang, X. Cheng, Y. Yang, W. Liu, H. Yu, L. Wang, and J. Zhang, "Deep learning-enabled sparse industrial crowdsensing and prediction," *IEEE Trans. Ind. Informatics*, pp. 1–1, 2020.

[22] D. Ding, M. Zhang, X. Pan, M. Yang, and X. He, "Modeling extreme events in time series prediction," in *KDD 2019*. ACM, 2019, pp. 1114–1122.

[23] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "Sensorscope: Application-specific sensor network for environmental monitoring," *ACM Trans. Sens. Networks*, vol. 6, no. 2, pp. 17:1–17:32, 2010.

[24] Y. Zheng, F. Liu, and H. Hsieh, "U-air: when urban air quality inference meets big data," in *KDD 2013*. ACM, 2013, pp. 1436–1444.

[25] D. H. Stolfi, E. Alba, and X. Yao, "Predicting car park occupancy rates in smart cities," in *Smart-CT 2017*, ser. Lecture Notes in Computer Science, vol. 10268. Springer, 2017, pp. 107–117.

[26] L. Wang, D. Zhang, A. Pathak, C. Chen, H. Xiong, D. Yang, and Y. Wang, "CCS-TA: quality-guaranteed online task allocation in compressive crowdsensing," in *UbiComp 2015*. ACM, 2015, pp. 683–694.

[27] F. Yin and F. Gunnarsson, "Distributed recursive gaussian processes for RSS map applied to target tracking," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 3, pp. 492–503, 2017.

[28] H. Yen and C. Wang, "Cross-device wi-fi map fusion with gaussian processes," *IEEE Trans. Mob. Comput.*, vol. 16, no. 1, pp. 44–57, 2017.

[29] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP 2014*. ACL, 2014, pp. 1724–1734.