Fine-grained Urban Prediction via Sparse Mobile CrowdSensing

Wenbin Liu[1], Yongjian Yang[1], En Wang[1,*], and Jie Wu[2]
[1]College of Computer Science and Technology, Jilin University, China
[2]Department of Computer and Information Sciences, Temple University, Philadelphia, USA

Abstract*—Mobile CrowdSensing (MCS) has recently emerged as a practical paradigm for large-scale and fine-grained urban sensing systems. To reduce sensing cost, Sparse MCS only senses data from a few subareas instead of sensing the full map, while the other unsensed subareas could be inferred by the intradata correlations among the sensed data. In certain applications, users are not only interested in inferring the data of other unsensed subareas in the current sensing cycle, but also interested in predicting the full map data of the near future sensing cycles. However, the intradata correlations exploited from the historical sparse sensed data cannot be effectively used for predicting full data in the temporal-spatial domain. To address this problem, in this paper, we propose an urban prediction scheme via Sparse MCS consisting of the matrix completion and the near-future prediction. To effectively utilize the sparse sensed data for prediction, we first present a bipartite-graph-based matrix completion algorithm with temporal-spatial constraints to accurately recover the unsensed data and preserve the temporal-spatial correlations. Then, for predicting the fine-grained future sensing map, with the historical full sensing data, we further propose a neural-network-based continuous conditional random field, including a Long Short-Term Memory component to learn the non-linear temporal relationships, and a Stacked Denoising Auto-Encoder component to learn the pairwise spatial correlations. Extensive experiments have been conducted on three real-world urban sensing data sets consisting of five typical sensing tasks, which verify the effectiveness of our proposed algorithms in improving the prediction accuracy with the sparse sensed data.

***Index Terms*—Mobile crowdsensing, matrix completion, continuous conditional random field.**

I. Introduction

With the rapid development of sensor-rich mobile devices and wireless communications, Mobile CrowdSensing (MCS) [1] has recently become a promising paradigm which recruits users carrying mobile devices to perform various sensing tasks, such as environment monitoring [2], infrastructure identification [3], and vehicle tracking [4], etc. To achieve high sensing quality, the traditional MCS applications would like to recruit a large number of users to collect data from all of the target sensing subareas. However, considering the sensing costs, we cannot afford to recruit too many users. More importantly, due to the uncertain mobilities of users, some target sensing subareas even have no available users at times [5], [6]. Hence, in most cases, we can only obtain the incomplete, random and even sparse sensed data, especially when we face the large-scale and fine-grained urban sensing tasks.

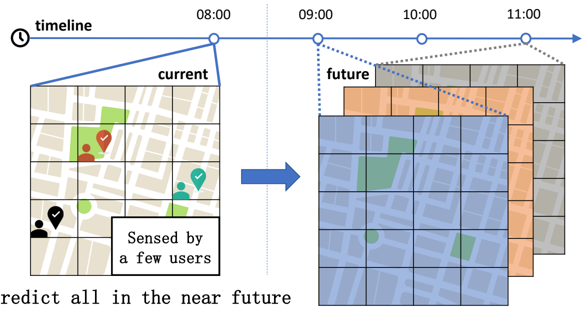


Fig. 1: Users sense data from a few subareas, which could be used to predict the data of all subareas in the near future.

In order to deal with this problem, some researchers propose a more practical paradigm, called Sparse MCS [7]–[11], which utilizes the sensing data of a few subareas to infer the rest of data from unsensed subareas. As shown in Fig. 1 (left part), suppose that there is a Sparse MCS scenario where we want to get a fine-grained urban sensing map at the current time, while only 3 out of 5×4 subareas have been sensed, and the data of the remaining subareas need to be inferred by the sparse sensed data. To solve this problem, most of the existing works exploit the intradata correlations among the sensed data to design various data inference algorithms. Liu *et al.* [12], Wang *et al.* [7], [8], and He *et al.* [13] separately use the Compressive Sensing (CS) and its variants (*e.g.*, Spatio-Temporal-CS and Bayesian-CS) as the data inference algorithms to recover the complete sensing matrix from the sparse sensed data. Another interesting problem is that if we could choose some effective subareas to sense data, and identify which subareas are the most useful for data inference. To deal with this problem, Liu *et al.* [9], [11] introduce Reinforcement Learning and Xie *et al.* [10] present a sensing scheduling scheme to actively determine the next subareas to be sensed, according to the already sensed data. In this way, the high data inference accuracy can still be achieved, while the requirement of the sensed data can be significantly reduced.

Although the existing works on Sparse MCS provide an effective way to infer the full sensing map from sparse sensed data, in some cases, as shown in Fig. 1, we are still more interested in predicting the full map data of the near future sensing cycles, rather than just inferring the data of other unsensed subareas in the current cycle. For example, in the traffic congestion or parking capacity monitoring tasks, the

[*] En Wang is the corresponding author.

current data are not very important, since users still need some time to drive there, and thus the data of the next several cycles are more instructive. Unfortunately, the existing works fail to do predict on the future data only based on the sparse sensed data. To achieve this goal, firstly, we have to infer a historical complete data matrix by sensing over a certain number of data[1]. However, the previous inference methods are not prediction-oriented, because the intradata correlations could not show the temporal-spatial correlations among all the subareas/cycles and hence are not enough to assist in predicting the future data. Secondly, the non-linear temporal relationships between the past and future sensing cycles and the complex spatial correlations across different subareas make the urban prediction problem more challenging. Therefore, how to effectively *utilize the sparse sensed data for prediction* and how to *learn the complex temporal-spatial correlations* for predicting the near future are two main challenges in the urban prediction problem via Sparse MCS.

In this paper, we turn attention from inferring the current unsensed data to predicting the near future full map from the sparse sensed data, and propose an urban prediction scheme via Sparse MCS. To effectively utilize the sparse sensed data for prediction, we first use a Bipartite-Graph-based Matrix Completion algorithm with Temporal-Spatial constraints (TS-BGMC) to conduct the unsensed data inference, which not only extracts intradata correlations from the sparse sensed data, but also preserves the temporal-spatial correlations among all the subareas and sensing cycles. In this way, we can accurately recover the historical unsensed data kept with effective temporal-spatial correlations, which provides more sufficient conditions for prediction. To further capture the temporal-spatial correlations for prediction, we propose a Neural-Network-based Continuous Conditional Random Field (NN-CCRF) model, including a Long Short-Term Memory (LSTM) component to learn the non-linear temporal relationships, and a Stacked Denoising Auto-Encoder (SDAE) component to learn the pairwise spatial correlations. To summarise, we aim to make full use of the sparse sensed data, and iteratively update and recover the historical complete data matrix at each sensing cycle, in order to predict the fine-grained near future sensing map as accurately as possible.

Our work has the following contributions:

- We formulate the urban prediction problem and propose an urban prediction scheme via Sparse Moblie CrowdSensing, with the goal of predicting the fine-grained near future sensing map from the historical sparse sensed data.
- To effectively utilize the sparse sensed data for prediction, we present a Bipartite-Graph-based Matrix Completion algorithm with Temporal-Spatial constraints (TS-BGMC) to recover the historical complete matrix kept with effective temporal-spatial relationships, which provides sufficient conditions for prediction.
- To further capture the temporal-spatial correlations for prediction, we propose a Neural-Network-based Continuous Conditional Random Field (NN-CCRF) model, including a Long Short-Term Memory (LSTM) component to learn the non-linear temporal relationships, and a Stacked Denoising Auto-Encoder (SDAE) component for the pairwise spatial correlations.
- We evaluate the proposed algorithms on three real-world data sets with five typical sensing tasks, and verify the effectiveness of our proposed algorithms in improving the prediction accuracy with the sparse sensed data.

II. Related Work

Mobile CrowdSensing is a promising paradigm which uses the mobile devices carried by users to perform various urban sensing tasks [1], [2]. In order to achieve high sensing quality, most of the existing works on MCS would like to recruit a large number of mobile users to sense data from as many target sensing subareas as possible [5], [6]. However, these works obviously cost a lot and still cannot sense data from the subareas with no users, especially when we face the large-scale and fine-grained urban sensing tasks. In order to reduce the costs and deal with the incomplete and random sensed data, some researchers introduce the data inference into MCS, called Sparse MCS [7]–[11], where we can sense data from only a few subareas and use some data inference algorithms to infer the unsensed data.

Recently, many fine-grained urban sensing systems have been developed via Sparse MCS. Rana $et\ al.$ [16] conducted an urban noise monitoring system which randomly senses data from some target subareas and uses compressive sensing to infer a fine-grained urban noise map. Zhu $et\ al.$ [17] also used a modified compressive sensing approach to estimate the urban traffic speeds, based on the data periodically collected by probe vehicles. To make more generalization, Wang $et\ al.$ [7], [8] formally presented the Sparse MCS paradigm that consists of three stages, including cell selection, data inference, and quality assessment. Briefly, Sparse MCS will first select some subareas to sense, and then uses the sensed data to infer the full map. If the inferred results are of poor quality, Sparse MCS continues to sense data and infer the full map. The authors also conducted various experiments over four urban sensing tasks, including temperature, humidity, air quality and traffic monitoring, in order to verify the effectiveness of Sparse MCS.

Similarly, He $et\ al.$ [13] and Liu $et\ al.$ [12] also proposed the urban air pollution and signal mapping systems based on Sparse MCS, while they further added the incentive design to steer users to sense data from some subareas. Furthermore, to effectively sense the useful subareas which hold sufficient information for data inference, Liu $et\ al.$ [9], [11] introduced the Reinforcement Learning technology and Xie $et\ al.$ [10] studied the sensing scheduling scheme to actively determine the next sensed subareas. Although Sparse MCS provides an effective way for the fine-grained urban sensing systems, the existing works mainly focus on inferring the current unsensed data but can hardly predict the near future full map.

[1] From the theoretical level, Candès and Recht [14] prove that we should use at least $Cn^{1.2}r\log n$ sensed data for inferring a $n \times n$ matrix of rank r with high accuracy, and several follow-on studies further slight the thresholds to about $O(nr\log n)$ [15].

III. System Model and Problem Formulation

A. System Model

We consider a general urban sensing scenario where the sensing system recruits some users to collect data from a large-scale target sensing area to provide the fine-grained urban sensing services. We divide the whole sensing campaign into many equal-length sensing cycles and the target area is split into m subareas, in order to provide the fine-grained results. Note that the lengths of cycles and the sizes of subareas are predetermined according to the tasks' requirements[2].

Under such a fine-grained urban sensing scenario, we recruit users to sense data from some subareas, and then use the data inference algorithms to infer the unsensed data for each sensing cycle. Specifically, for each sensing cycle, we obtain some data sensed from a few subareas, which are recorded in a vector $y'_{m\times 1}$, and the unsensed data are recorded as 0 (if 0 is not the reasonable sensing value, then we should use another value). Let the vector $y_{m\times 1}$ denote the ground truth and the binary sensed vector $c_{m\times 1}$ mark whether one subarea has been sensed: if subarea i has been sensed at the current cycle, $c[i] = 1$; otherwise, $c[i] = 0$, and thus

$$y' = y \bullet c, \tag{1}$$

where $\bullet$ represents an element-wise product, i.e., $y'[i] = y[i] \times c[i]$. Then, we use a data inference algorithm $f()$ to infer the unsensed data from the sensed y', with the inference error $\varepsilon(y, \hat{y})$, as follows:

$$f(y') = \hat{y} \approx y, \tag{2}$$

$$\varepsilon(y, \hat{y}) = \Sigma_{i=1}^{m} |y[i] - \hat{y}[i]|. \tag{3}$$

At the n-th sensing cycle, we have already sensed n vector and obtained the actually sensed matrix $Y'_n \triangleq \{y_1'^T, y_2'^T, ..., y_n'^T\}$. Similarly, let $Y_n \triangleq \{y_1^T, y_2^T, ..., y_n^T\}$ and $C_n \triangleq \{c_1^T, c_2^T, ..., c_n^T\}$, we have $Y'_n = Y_n \bullet C_n$. Then, with the goal to accurately predict the fine-grained near future full map from the sparse sensed data, we first use a matrix completion algorithm $m()$ to recover the historical sensed matrix $\hat{Y}_n$ from Y'_n, and then use an near-future prediction method $p()$ to predict the future sensing vector:

$$p(m(Y'_n)) = p(\hat{Y}_n) = \hat{y}_{n+1} \approx y_{n+1}, \tag{4}$$

$$\varepsilon(y_{n+1}, \hat{y}_{n+1}) = \Sigma_{i=1}^{m} |y_{n+1}[i] - \hat{y}_{n+1}[i]|. \tag{5}$$

Then, we can add the predicted $\hat{y}_{n+1}$ to $\hat{Y}_n$ and obtain $\hat{Y}_{n+1}$, in order to iteratively calculate the next k cycles, denoted as $p(m(Y'_n), k) = \hat{y}_{n+k}$.

B. Problem Formulation

Problem [Urban Prediction via Sparse MCS]: Given a MCS task with m subareas and n sensing cycles, for each cycle, we can sense data from quite a few subareas, and then use the historical sparse sensed data to infer the future full map of the k near future sensing cycles, with the objective of minimizing the prediction errors:

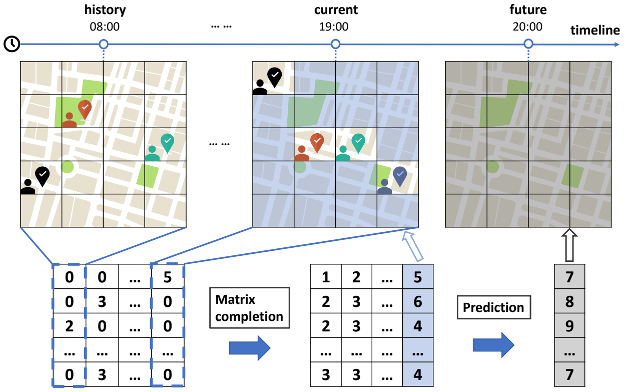


Fig. 2: Urban sensing and prediction via Sparse MCS.

$$\min \quad \sum_{j=1}^{n-k} \varepsilon(y_{j+k}, \hat{y}_{j+k}) \tag{6}$$

$$\text{s.t.} \quad p(m(Y'_j), k) = \hat{y}_{j+k}, \forall j \in \{1, 2, ..., n-k\}. \tag{7}$$

We now provide a running example to describe the urban prediction problem in more details, as shown in Fig. 2. Suppose that we have an urban sensing task which needs to sense data from a target area from 8:00 to 20:00. To provide the fine-grained results, we divide the target sensing area into 5×4 subareas and provide sensing data for these subareas every one hour. To reduce the costs and deal with the unavailable subareas, for each sensing cycle, only a few subareas will be sensed, e.g., at 8:00, we only obtain the data sensed from 3 subareas. After 12 hours, the current time is 19:00, and we obtain the data sensed from 4 subareas in this sensing cycle. We can use a data inference algorithm to infer the unsensed data of the current sensing cycle (marked as the blue grids). Furthermore, we would like to predict the near future full map (marked as the grey grids) from the historical sparse sensed data. In this paper, we first use a matrix completion algorithm to recover the historical complete matrix from the sparse sensed matrix, which provides accurate and sufficient data for prediction. Then, with the inferred matrix, we use a prediction method to learn the temporal-spatial correlations for predicting the near-future full map.

IV. Temporal-Spatial Matrix Completion with Bipartite Graph

A. Temporal-Spatial Matrix Factorization

The urban sensing is actually to collect the various readings from the urban regions. Note that most of the sensing data are continuous in the physical world, which generally exhibit strong temporal-spatial correlations, and thus the complete sensing matrix Y usually has the low-rank property. Given an incompletely and randomly (or even sparse) sensed data

[2] To reduce the complexity of the problem, we assume that the lengths of cycles and the sizes of subareas are set properly that the data sensed from the same subarea within the same cycle are the same. If we have irregular lengths and sizes, we should further use some algorithms, i.e., the numerical interpolation methods, to readjust them first.

matrix Y', we would like to recover the full sensing matrix $\hat{Y}$ based on the low-rank property:

$$\min \quad \mathit{rank}(\hat{Y}) \tag{8}$$

$$\text{s.t.,} \quad \hat{Y} \bullet C = Y'. \tag{9}$$

Note that the above optimization is nonconvex, so we can hardly solve it directly. Given the complete sensing matrix Y with rank k [3], i.e., $\mathit{rank}(Y) = \mathit{rank}(\hat{Y}) = k$, we can factor our inferred matrix $\hat{Y}$ into the product of a spatial factor matrix $L_{m\times k}$ and a temporal factor matrix $R_{n\times k}$, as shown in Fig. 3 (left part), in order to capture the low rank feature and change the above rank optimization problem with constraints to the error minimization problem for missing data recovery:

$$\min \quad \|(Y - \hat{Y}) \bullet C\|_F^2 = \|Y' - LR^T \bullet C\|_F^2, \tag{10}$$

$$\text{s.t.,} \quad \mathit{rank}(Y) = \mathit{rank}(\hat{Y}) = k, \hat{Y} = LR^T, \tag{11}$$

where $\|\|_F$ is the Frobenius norm and used to present the error between the inferred matrix and the actually sensed matrix.

To obtain the optimal $\hat{Y}$ from Y', many existing methods, such as the Alternating Least Squares [17], [18], can be used to train the two factors to solve the problem, i.e., $\hat{Y} = LR^T$ according to the Eq. 10. However, the Eq. 10 only focuses on the learning from the sensed data but ignores the temporal-spatial correlations existing in the unsensed data. Therefore, we further consider the important and naturally occurring correlations as the supplement and constraint for Eq. 10, and thus obtain the error minimization problem with temporal-spatial constraints as follows:

$$\min \ \|Y' - \hat{Y} \bullet C\|_F^2 + \lambda_t \|\hat{Y}\mathbb{T}^T\|_F^2 + \lambda_s \|\mathbb{S}\hat{Y}\|_F^2, \tag{12}$$

where $\mathbb{T}$ and $\mathbb{S}$ are the temporal and spatial constraint matrices:

- $\mathbb{T}$ presents the temporal constraint among the sensing data of the same subarea between different sensing cycles. Intuitively, two continuously sensed data from the same subarea are usually similar. Thus, we choose a temporal constraint matrix $\mathbb{T}_{n\times n} = \mathit{Toeplitz}(0, 1, -1)$ for Eq. 12 to constrain that two continuous data from one subarea are the same. Moreover, the prior domain knowledge and the sufficient historical data may provide more temporal constraints, such as the periodicity and statistical characteristics, which can be used to conduct a more sophisticated $\mathbb{T}$.
- $\mathbb{S}$ presents the spatial constraint among the sensing data of the same sensing cycle between different subareas. Similar with the temporal constraint, the data sensed from the closer subareas usually have the similar values. Thus, we use the Euclidean distance to characterize the spatial correlation, denoted as $\mathbb{S}_{m\times m}[i, j]$=exp$(-\text{distance}(i, j)/\sigma_s^2)$. Then, for each row i in $\mathbb{S}$, we normalize them as $\sum_{j=1, j\neq i}^{m} \mathbb{S}[i, j] = 1$ and set $\mathbb{S}[i, i] = -1, \forall i = \{1, ..., m\}$.

Note that the temporal and spatial constraint matrices $\mathbb{T}$ and $\mathbb{S}$ are used as a supplement to further constrain the unsensed data in the sensing matrix, which also preserve the temporal-spatial relationships among all the subareas and sensing cycles for the near-future prediction. Meanwhile, as shown in Eq. 12, we can use the weighted parameters λ_t and λ_s to balance the weights of different elements. Furthermore, many other correlations could be easily modified into our error minimization problem.

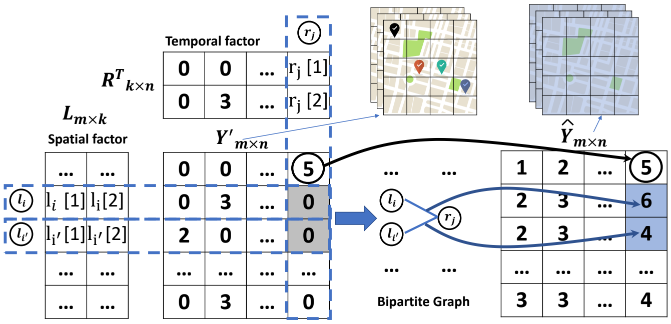


Fig. 3: Matrix factorization with bipartite graph.

B. Graph-based Matrix Completion

To conduct the matrix completion, we first consider the relationships between the temporal-spatial factors and the inferred results. As discussed above, we obtain that the inferred matrix $\hat{Y} = LR^T$. Specifically, as shown in Fig. 3 (right part), let $L \triangleq \{l_1^T, l_2^T, ..., l_m^T\}^T$ and $R \triangleq \{r_1, r_2, ..., r_m\}^T$, we have $\hat{Y}[i, j] = l_i r_j^T = \sum_{c=1}^{k} l_i[c] r_j[c]$. Actually, the elements in the inferred matrix $\hat{Y}$ can be seen as the linear combination of the elements in temporal-spatial factors L and R. As an example, suppose that we obtain the rank $k = 2$ and the well trained temporal-spatial factors L and R, as shown in Fig. 3, the unsensed data $Y'[i, j] = 0$ can be inferred as follows:

$$\begin{aligned} \hat{Y}[i, j] &= l_i r_j^T = l_i[1] r_j[1] + l_i[2] r_j[2], \\ \hat{Y}[i', j] &= l_{i'} r_j^T = l_{i'}[1] r_j[1] + l_{i'}[2] r_j[2]. \end{aligned} \tag{13}$$

Thus, we conduct the bipartite graph that consists of l_i and r_j as the vertexes and $\hat{Y}[i, j]$ as the edges, where $i \in [1, m]$ and $j \in [1, n]$. Then, the matrix completion can be seen as the linear calculations of l_i and r_j to obtain $\hat{Y}[i, j]$.

After understanding the relationships between the factors and the inferred matrix, we further consider how to train the temporal-spatial factors L and R. Based on the error minimization problem in Eqs. 11 and 12, we have $Y' = LR^T \bullet C$. Go back to the example in Fig. 3 (left part), we have

$$Y'[1, n] = l_1[1] r_n[1] + l_1[2] r_n[2] = 5. \tag{14}$$

Consider that in the first $(n - 1)$ sensing cycles, we already calculate the suitable $L_{m\times k}$ and $R_{(n-1)\times k}$. For the current n-th sensing cycle, we know the sensed data $Y'[1, n]$ and the spatial factor l_1, and then we can use the linear Eq. 14 to calculate the unknown temporal factor r_n. With the well trained $R = \{R_{(n-1)\times k}, r_n\}$ and L, we can recover the unsensed data, e.g., the $\hat{Y}[i, j]$ and $\hat{Y}[i', j]$ in our example, through the linear calculations.

The detailed graph-based matrix completion algorithm is summarized in Alg. 1 with an example shown in Fig. 4. For the

[3] k is a property of Y, but in practice, we cannot obtain the complete matrix Y and have to collect it for some sensing cycles to estimate the initial rank k and readjust it according to the recovery errors during the sensing campaign.

Algorithm 1 Graph-based Matrix Completion

```
Input: $Y'_{m\times n} = \{Y'_{m\times(n-1)}, y_n'^T\}$, $L_{m\times k} = \{l_1, l_2, ..., l_m\}$,
    $R_{(n-1)\times k} = \{r_1, r_2, ..., r_{n-1}\}$
Output: $\hat{Y}_{m\times n}$
1: Init $r_n$, $R_{n\times k} = \{R_{(n-1)\times k}, r_n\}$, $count = 0$;
2: Build the linear system by using $y_n'$, $L_{m\times k}$, and $R_{(n-1)\times k}$,
   and then calculate $r_n$;
3: while not convergent and $count <$ MAX_ITER do
4:     Fix $R_{n\times k}$ and treat $L_{m\times k}$ as unknown, build the linear
       system by using $Y'_{m\times n}$, $L_{m\times k}$ and $R_{n\times k}$, and then
       calculate and update $L_{m\times k}$;
5:     Fix $L_{m\times k}$ and treat $R_{n\times k}$ as unknown, build the linear
       system by using $Y'_{m\times n}$, $L_{m\times k}$ and $R_{n\times k}$, and then
       calculate and update $R_{n\times k}$, $\hat{Y} = LR^T$, and $count$++;
6: return $\hat{Y}$.
```

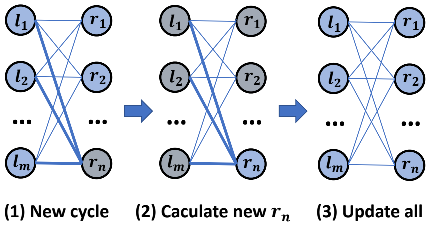


Fig. 4: Matrix completion based on bipartite graph.

current n-th, we have already held the well trained temporal and spatial factors $L_{m\times k}$ and $R_{(n-1)\times k}$ for the historical sensing matrix. We first build the linear system as Eq. 14 (line 2) to calculate the newly added temporal factor r_n. Note that the current factors $L_{m\times k}$ and $R_{(n-1)\times k}$ only hold the temporal and spatial information learned from the historical matrix $Y'_{m\times(n-1)}$, we then iteratively train and update the factor L or R by using an Alternating Least Squares [8], [10], [17], [18] while keeping the others fixed (lines 3-5), until the inferred $\hat{Y}$ is convergent or the maximum number of iterations is reached. Finally, the graph-based matrix completion algorithm outputs the complete inferred matrix $\hat{Y}$, which further provides sufficient data with effective temporal-spatial relationships for the near-future prediction in the next section.

V. Continuous Conditional Random Field for Urban Prediction

A. Neural Network-based CCRF

Continuous Conditional Random Field (CCRF) has been used for prediction in the recent years [19], [20], which aims to capture not only the relationships between the input and output data, but also the correlations among the output data. As shown in Fig. 5 (upper part), in the urban prediction, the input data is the time series of the data sensed from each subarea and the output data is the predicted results for the subareas. Thus, we model the relationships between the input and output data as the temporal relationships and the correlations among the output data as the spatial correlations in the urban prediction problem. Obviously, we should learn both of the temporal and spatial correlations to predict the near future sensing map, which can be naturally modeled as a CCRF model.

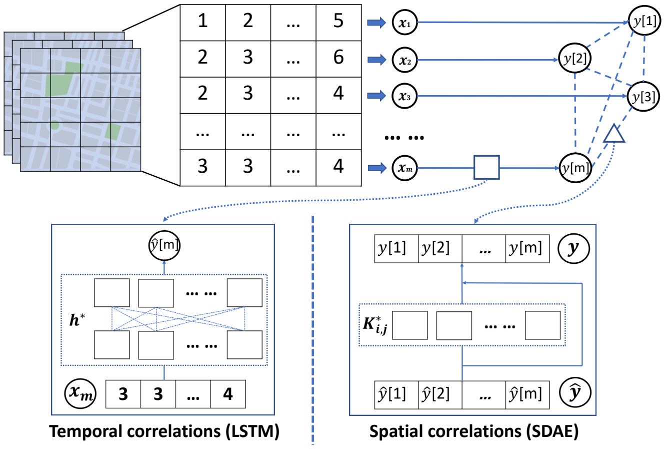


Fig. 5: The model of NN-CCRF for urban prediction.

Specifically, for the temporal relationships in the sequence of the data sensed from each subarea, we propose to use a Long Short-Term Memory (LSTM) [21] component to learn the non-linear relationships from the input data to the output results. For the spatial correlations among the sensing data of different subareas, we introduce a Stacked Denoising Auto-Encoder (SDAE) [22] component to learn the pairwise influences among the output results. Note that both LSTM and SDAE are the famous neural networks which have been proved to be effective on capturing the non-linear temporal and spatial correlations, which can be added into CCRF model to enhance the performances. As shown in Fig. 5 (lower part), we present an overview of the proposed NN-CCRF model for urban prediction, consisting of a LSTM component to learn the non-linear temporal relationships and a SDAE component to learn the pairwise spatial correlations, which can benefit from both CCRF model and NN algorithms. To reduce complexity, we simply consider the next-cycle prediction in the NN-CCRF (i.e., the LSTM and SDAE) and the algorithm is illustrated in Alg. 2. For the other near future cycles, we can add the predicted next-cycle results $\hat{y}_{n+1}$ to $\hat{Y}_n$ and obtain $\hat{Y}_{n+1}$, in order to iteratively calculate the next several cycles.

B. LSTM for Temporal Relationships

Long Short-Term Memory (LSTM) is a famous neural network architecture and well-suited to make prediction on time series data [21]. Obviously, in our urban prediction problem, there exist strong temporal relationships in the sequence of data sensed from the same subareas, i.e., the input data x_i. More importantly, the temporal relationships are usually non-linear and irregular, and dealing with these non-linear and irregular temporal relationships is actually the advantage of LSTM. Therefore, in this work, we conduct the LSTM component to capture the temporal relationships from the input sequence of sensed data and predict the output sensing data of the next sensing cycle.

Algorithm 2 NN-CCRF-based Next-Cycle Prediction

```
Input: $\hat{Y}_{m\times n} = \{x_1^T, x_2^T, ..., x_m^T\}^T$, $h^*$, $K^*$, $\omega_t$, $\omega_s$
Output: $\hat{y}_{n+1}$
1: Init $\hat{y}'_{n+1}$, $count = 0$;
2: for i=1,2,...,m do
3:     Use the LSTM with parameter $h^*$ to predict $\hat{y}'_{n+1}[i]$ from $x_i$, $\hat{y}'_{n+1}[i] = lstm(x_i|h^*)$;
4: $\hat{y}_{n+1} = \hat{y}'_{n+1}$
5: while $count <$ MAX_ITER do
6:     Use the SDAE with parameter $K^*$ to predict $\hat{y}_{n+1}$ from $\hat{y}'_{n+1}$, $\hat{y}_{n+1} = sdae(\hat{y}_{n+1}|K^*)$;
7: Conduct the linear combination of the results from SDAE and LSTM, $\hat{y}_{n+1} = \omega_s \cdot \hat{y}_{n+1} + \omega_t \cdot \hat{y}'_{n+1}$;
8: return $\hat{y}_{n+1}$.
```

Specifically, we conduct the LSTM component with parameter h^* to predict the output data $\hat{y}'_{n+1}$ from the input data $\hat{Y}_{m\times n} = \{x_1^T, x_2^T, ..., x_m^T\}^T$, as follows:

$$\hat{y}'_{n+1}[i] = lstm(x_i|h^*), \tag{15}$$

$$\hat{y}'_{n+1} = lstm(\hat{Y}, h^*), \tag{16}$$

$$lstm(\hat{Y}, h^*) = \sigma(W_h h_t^* + b_h|\hat{Y}, W_*, U_*, b_*), \tag{17}$$

where $\sigma()$ is the sigmod function and W_*, U_*, and b_* are the well-trained weighted matrices and bias vectors for different subareas within the LSTM component. More specifically, a common LSTM unit, composed of a memory cell c, an input gate i, an output gate o, and a forget gate f, specifies the W_*, U_*, and b_* by iteratively updating h_t^*:

$$f_t = \sigma(W_f \hat{Y}_t + U_f h_{t-1}^* + b_f), \tag{18}$$

$$i_t = \sigma(W_i \hat{Y}_t + U_i h_{t-1}^* + b_i), \tag{19}$$

$$o_t = \sigma(W_o \hat{Y}_t + U_o h_{t-1}^* + b_o), \tag{20}$$

$$c_t = f_t \bullet c_{t-1} + i_t \bullet \phi(W_c \hat{Y}_t + U_c h_{t-1}^* + b_c), \tag{21}$$

$$h_t^* = o_t \bullet \phi(c_t), \tag{22}$$

where $\bullet$ denotes the element-wise product and $\phi()$ is the hyperbolic tangent function.

The LSTM component is illustrated in Alg. 2 (lines 2-3). Note that the predicted results obtained from the LSTM component are only the preliminary estimations without considering the spatial correlations. Thus, we use $\hat{y}'_{n+1}$ to record these preliminary estimations (line 4), and further learn the spatial correlations to adjust the results in the next subsection.

C. SDAE for Spatial Correlations

Stacked Denoising Auto-Encoder (SDAE) [22] is a feedforward neural network consisting of multiple layers of denoising autoencoders, which inputs the corrupted data and trains to output the revised data. In our next-cycle prediction problem, the preliminary estimations, i.e., $\hat{y}'_{n+1}$, obtained from the LSTM component can be actually considered as the corrupted input data, since they only capture the temporal relationships but ignore the spatial correlations. Then, the output revised data can be modeled as the predictions on the ground truth, i.e., $\hat{y}_{n+1} \sim y_{n+1}$, which means that the SDAE matches the corrupted data $\hat{y}'_{n+1}$ to ground truth y_{n+1}. In this way, we adopt the SDAE component to capture the spatial correlations from the input preliminary estimations from the LSTM component, in order to further constrain and smooth the preliminary estimations to obtain the better next-cycle prediction results.

Specifically, the SDAE contains two processes, i.e., encoding and decoding. With the preliminary estimations $\hat{y}'_{n+1}$ and the ground truth y_{n+1}, the SDAE component first encodes $\hat{y}'_{n+1}$ into image z and then decodes the z to $\hat{y}_{n+1}$ as the prediction of y_{n+1}, as follows:

$$z = \sigma(W_z \hat{y}'_{n+1} + b_z), \tag{23}$$

$$\hat{y}_{n+1} = \sigma(W_y + b_y), \tag{24}$$

where W_z and W_y are the weighted matrices and b_z and b_y are the bias vectors. Then, the objective function is

$$min \quad \|y_{n+1} - \hat{y}_{n+1}\|_F^2 + \omega\|W\|_F^2, \tag{25}$$

where W denotes the weighted matrices which match the corrupted data $\hat{y}'_{n+1}$ to ground truth y_{n+1} in Eqs. 23 and 24.

To effectively capture the spatial correlations among the sensing data, we propose to use a same spatial correlation matrix $K_{m\times m}^*$ to replace all of the weighted matrices in the encoding and decoding processes, i.e., the W_z, W_y, and W in Eqs. 23, 24, and 25. In this way, for each encoding or decoding process, we actually use the pairwise correlations between i-th and j-th subareas to constrain and smooth the predicted results, and training the SDAE is actually to learn the spatial correlation matrix K^*. Therefore, our SDAE component can focus on the pairwise spatial correlations:

$$\hat{y}_{n+1} = sdae(\hat{y}'_{n+1}|K^*), \tag{26}$$

with the objective function converted as follows:

$$min \quad \|y_{n+1} - \sigma_n(K^* \ \hat{y}'_{n+1})\|_F^2 + \omega\|K^*\|_F^2, \tag{27}$$

where $\sigma_n()$ denotes the n times encoding-decoding processes.

The SDAE component is illustrated in Alg. 2 (lines 5-6). Note that the parameter MAX_ITER is the number of the SDAE layers, since the layers of SDAE are the same with the purpose to extract features layer-by-layer. Finally, we conduct a linear combination of the results from SDAE and LSTM, with the parameters ω_s and ω_t to balance the weights of temporal and spatial correlations.

Combining the TS-BGMC with NN-CCRF, we finally present the urban prediction scheme via Sparse MCS. For each sensing cycle, we first utilize the TS-BGMC to recover a complete sensing matrix and then employ the NN-CCRF to predict the near-future full map. For the next sensing cycle, we use the newly sensed data to update and recover the complete sensing matrix and employ the NN-CCRF to predict the future ones. To summarise, our proposed urban prediction scheme aims to make full use of the sparse data sensed from different subareas and different sensing cycles, which iteratively updates and recovers the complete sensing matrix, in order to predict the full map data of the near future as accurately as possible.

TABLE I: Statistics of three evaluation data sets

	Sensor-Scope	***U-Air***	***TaxiSpeed***
City	Lausanne (Switzerland)	Beijing (China)	Beijing (China)
Data	Temperature-Humidity	PM2.5-PM10	Traffic speed
Subarea	57 subareas each with 50*30m^2	36 subareas each with 1000*1000m^2	100 road segments as subareas
Cycle & Duration	$0.5h$ & $7d$	$1h$ & $11d$	$1h$ & $4d$
Mean ± Std.	$6.04 \pm 1.87°C$ (T)/$84.52 \pm 6.32\%$ (H)	79.11 ± 81.21 (PM2.5)/63.12 ± 48.56 (PM10)	$13.01 \pm 6.97m/s$

VI. Performance Evaluation

A. Data Sets

To evaluate our proposed urban prediction scheme, we adopt three well-known urban sensing data sets, including *Sensor-Scope* [23], *U-Air* [24], and *TaxiSpeed* [25]. The detailed statistics are shown in Table I with the descriptions as follows:

The ***Sensor-Scope*** [23] data set collected various types of environmental readings by using many static sensors deployed in the EPFL campus. We select two representative types of sensing data, $i.e.$, temperature and humidity, and use the Mean Absolute Error (MAE) to evaluate the data accuracy.

The ***U-Air*** [24] data set collected some important air quality data, such as PM2.5 and PM10, by some monitor stations deployed in Beijing, China. As shown in Table I, the air quality readings in *U-Air* fluctuate greatly, and we thus use the air quality index category[4] and compare the error rates.

The ***TaxiSpeed*** [25] data set collected the traffic speed data for road segments by using the GPS devices deployed on taxis in Beijing, China. We consider the road segments as the subareas and use the MAE to evaluate the data accuracy.

Note that all of these three data sets were collected by static sensors, while recently, we can use mobile devices to collect the same data from the subareas. Moreover, these five selected sensing tasks, including temperature, humidity, PM2.5, PM10, and traffic speed, are typical urban sensing tasks and also in need of predictions. Therefore, we use these data sets to verify the effectiveness of the proposed algorithms.

B. Comparison Algorithms

In order to effectively utilize the sparse sensed data to conduct predictions, we first present the Bipartite-Graph-based Matrix Completion algorithm with Temporal-Spatial constraints (referred as "TS-BGMC"). We mainly compare it with the following algorithms:

- BGMC, which is the same matrix completion algorithm but ignores the temporal-spatial constraints.
- KNN-S/T, which uses the weighted average of K data sensed from the nearest subareas (spatial/temporal correlations) as the inferred value of the current subarea in each sensing cycle. KNN-S/T is modified from the famous K-Nearest Neighbors (KNN) algorithm.

Since the subarea selection is not the main point of this work, we randomly sense some subareas in each sensing cycle, and send the sensed data to data inference algorithms to recover the unsensed data. With the recovered sensing matrix, we then conduct the Neural-Network-based Continuous Conditional Random Field (referred as "NN-CCRF") to do the near-future prediction. Note that in this paper, we mainly focus on predicting the near future sensing map from the sparse sensed data, which is such a difficult scenario that most of the existing works on urban prediction cannot work well, and we thus mainly compare it with the following algorithms:

- LSTM, which only uses the LSTM component to predict the near-future full map. Without the SDAE component, LSTM mainly captures the temporal relationships but ignores the spatial correlations among the sensing data.
- LINEAR, which uses the Linear Regression model to predict the near-future full map. It assumes that the future results are linearly related to the historical sensed data.

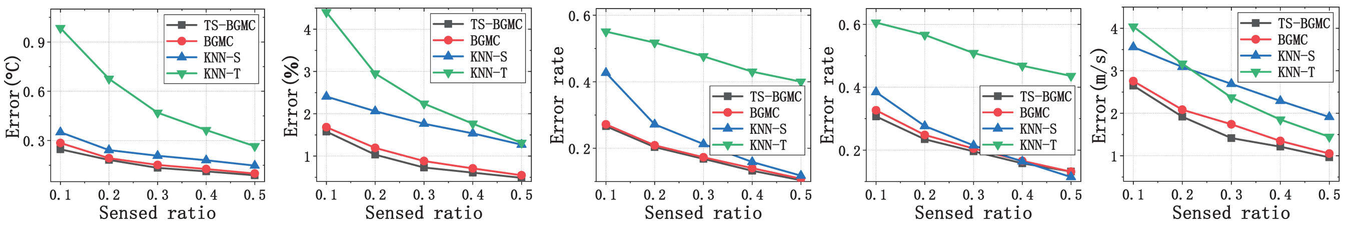
In addition, the structure of the neural networks is not the main point of this work, and we conduct a three layers LSTM with 32 hidden states and also build a 4 layers SDAE component in our evaluations. For the training process, we use the first 20% of the data to train both TS-BGMC and NN-CCRF. In order to conduct sufficient training data, we repeatedly use the data set by randomly selecting the sensed subareas for each sensing cycle. We also learn the rank k from the training data for each sensing task.

C. Evaluation Results

1) Matrix Completion: We first test the average inference accuracy under different sensed ratios from 0.1 to 0.5. Note that the sensed ratio 0.1 means that only 10% subareas will be sensed randomly at each sensing cycle, which is already very sparse for urban sensing tasks. As shown in Fig. 6, we can see that the inference errors of all compared methods decrease along with the increases of the sensed ratios. Obviously, the reason is that the more sensed data contains the more information, which can help the matrix completion. Our TS-BGMC can perform the best in most times, since the temporal-spatial constraints can really help the matrix completion. Meanwhile, the BGMC has a close performance with TS-BGMC, since both of them use the same method to recover the matrices. Moreover, the KNN-S usually achieves better performances than KNN-T, which shows that the spatial correlations in matrix completion may be more important than temporal relationships and both of them should be considered in the near-future prediction.

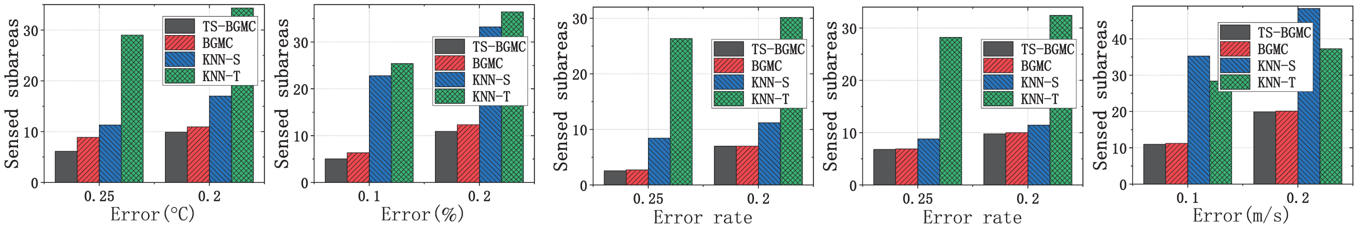
Furthermore, in order to show the effectiveness of the proposed matrix completion algorithms, we constrain the accuracy and test the number of sensed subareas over three

[4] Six categories [24]: Good (0-50), Moderate (51-100), Unhealthy for Sensitive Groups (101-150), Unhealthy (150-200), Very Unhealthy (201-300), and Hazardous (>300).



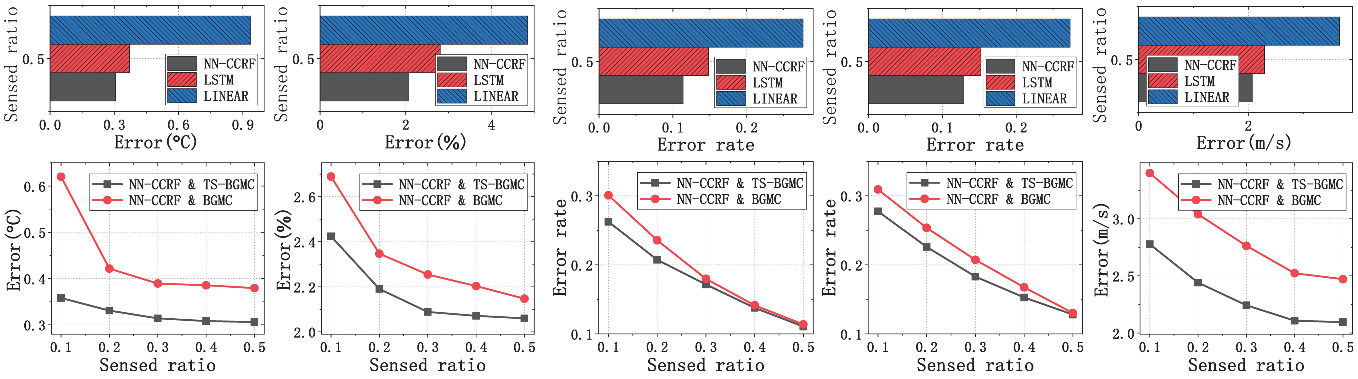
(a) Temperature (b) Humidity (c) PM2.5 (d) PM10 (e) Traffic Speed

Fig. 6: Inference accuracy under different sensed ratios over *Sensor-Scope*, *U-Air*, and *TaxiSpeed*.



(a) Temperature (b) Humidity (c) PM2.5 (d) PM10 (e) Traffic Speed

Fig. 7: Number of sensed subareas under different inference accuracy over *Sensor-Scope*, *U-Air*, and *TaxiSpeed*.



(a) Temperature (b) Humidity (c) PM2.5 (d) PM10 (e) Traffic Speed

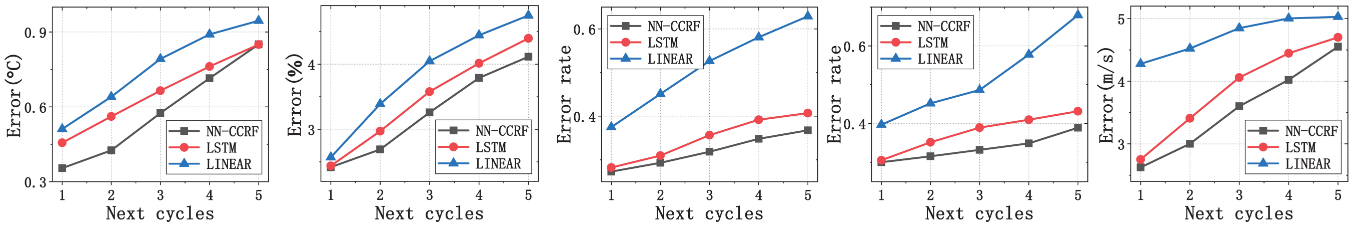
Fig. 8: Prediction accuracy under different sensed ratios over *Sensor-Scope*, *U-Air*, and *TaxiSpeed*.

data sets with five tasks. The results are shown in Fig. 7, which actually have the similar tendency with Fig. 6. Under different data accuracy, the TS-BGMC always needs to sense the least subareas, while BGMC usually has to sense more subareas than TS-BGMC. Also, the KNN-S and KNN-T usually need to sense the most subareas, especially when the required errors are low (the high accuracy requirements). Therefore, both of the Figs. 6 and 7 can verify the effectiveness of TS-BGMC.

2) Near-Future Prediction: With the accurate and complete sensing matrices recovered from the matrix completion part, we then evaluate the effectiveness of our near-future prediction part. We first set the sensed ratio to 0.5 and use the matrix recovered by TS-BGMC to evaluate the performance of next-cycle predictions. As shown in Fig. 8 (a-e) (upper part), NN-CCRF always outperforms the LSTM and LINEAR. We then change the sensed ratio from 0.1 to 0.5 and evaluate the NN-CCRF on different matrix completion algorithms. Since KNN-S and KNN-T perform poorly on the near-future prediction, we mainly compare the TS-BGMC and BGMC with NN-CCRF, as shown in Fig. 8 (lower part). We can see that the TS-BGMC with NN-CCRF obviously outperforms the BGMC, which verifies that our temporal-spatial constraints in TS-BGMC can not only help on matrix completion but also hold the strong temporal-spatial correlations for prediction.

Furthermore, we also change the predicted near future cycles from 1 to 5, with the sensed ratio 0.1. As shown in Fig. 9, our proposed NN-CCRF can always achieve the best performance and clearly outperform the LSTM, which shows that our NN-CCRF can effectively capture the temporal-spatial correlations. Moreover, the LINEAR method performs poorly, since the temporal-spatial correlations in urban sensing tasks are more complex than the simple linear relationships.

3) Running Time: Finally, we report the running time of the proposed methods, as shown in Table II. Our experiment platform is equipped with Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz and 32 GB RAM, and we implement the urban prediction scheme using Pytorch. The TS-BGMC costs $0.33 - 0.62$s to recover a complete matrix for each sensing cycle. The LSTM component costs $1.00 - 2.13$ms and the NN-CCRF costs $0.06 - 0.12$s. The running time are totally acceptable in practical applications. In addition, the training of LSTM and SDAE can be conducted offline, which cost $\sim$10 minutes.



(a) Temperature (b) Humidity (c) PM2.5 (d) PM10 (e) Traffic Speed

Fig. 9: Prediction accuracy for near future sensing cycles over *Sensor-Scope*, *U-Air*, and *TaxiSpeed*.

TABLE II: Running time for main methods

	Tem.	Hum.	PM2.5	PM10	Tra.
TS-BGMC	0.45s	0.45s	0.33s	0.34s	0.62s
LSTM	2.10ms	2.13ms	1.52ms	1.52ms	1.00ms
NN-CCRF	0.12s	0.12s	0.06s	0.06s	0.10s

VII. Conclusion

In this paper, we investigate the urban prediction problem via Sparse MCS, which can use the sparse sensed data to predict the fine-grained full map of the future sensing data. To effectively utilize the sparse sensed data, we first present TS-BGMC to recover the historical sensing matrix and provide sufficient data for prediction. Then, we further propose NN-CCRF for urban prediction, in which we first conduct a LSTM component to learn the non-linear temporal relationships, then build a SDAE component to learn the pairwise spatial correlations, and finally predict the fine-grained future sensing map. Extensive evaluations have been conducted on three real-world data sets with five types of sensing tasks, and the results verify that our proposed algorithms can achieve a high prediction accuracy with the sparse sensed data.

Acknowledgment

This work is supported by the National Natural Science Foundations of China under Grant No. 61772230 and No. 61972450, National Natural Science Foundations of China for Young Scholars No. 61702215, Natural Science Foundations of Jilin Province No. 20190201022JC, National Science Key Lab Fund project No. 61421010418, and this research is also supported in part by NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS 1651947, and CNS 1564128.

References

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

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

[3] H. Aly, A. Basalamah, and M. Youssef, "Automatic rich map semantics identification through smartphone-based crowd-sensing," *IEEE Transactions on Mobile Computing*, no. 10, pp. 2712–2725, 2017.

[4] H. Chen, B. Guo, Z. Yu, and Q. Han, "Crowdtracking: Real-time vehicle tracking through mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7570–7583, Oct 2019.

[5] Y. Yang, W. Liu, E. Wang, and J. Wu, "A prediction-based user selection framework for heterogeneous mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2460–2473, 2019.

[6] W. Liu, Y. Yang, E. Wang, and J. Wu, "Dynamic user recruitment with truthful pricing for mobile crowdsensing," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, July 2020.

[7] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, and A. M'hamed, "Sparse mobile crowdsensing: challenges and opportunities," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 161–167, 2016.

[8] 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 Transactions on Intelligent Systems and Technology*, vol. 9, no. 2, pp. 1–28, 2017.

[9] W. Liu, L. Wang, E. Wang, Y. Yang, D. Zeghlache, and D. Zhang, "Reinforcement learning-based cell selection in sparse mobile crowdsensing," *Computer Networks*, vol. 161, pp. 102–114, 2019.

[10] K. Xie, X. Li, X. Wang, G. Xie, J. Wen, and D. Zhang, "Active sparse mobile crowd sensing based on matrix completion," in *Proceedings of the 2019 International Conference on Management of Data*, ser. SIGMOD '19. New York, NY, USA: ACM, 2019, pp. 195–210.

[11] W. Liu, Y. Yang, E. Wang, and J. Wu, "User recruitment for enhancing data inference accuracy in sparse mobile crowdsensing," *IEEE Internet of Things Journal*, pp. 1–1, 2019.

[12] T. Liu, Y. Zhu, Y. Yang, and F. Ye, "Alc2: When active learning meets compressive crowdsensing for urban air pollution monitoring," *IEEE Internet of Things Journal*, 2019.

[13] S. He and K. G. Shin, "Steering crowdsourced signal map construction via bayesian compressive sensing," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, April 2018, pp. 1016–1024.

[14] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, p. 717, 2009.

[15] B. Recht, "A simpler approach to matrix completion," *Journal of Machine Learning Research*, vol. 12, no. Dec, pp. 3413–3430, 2011.

[16] 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 *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010, pp. 105–116.

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

[18] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 662–676, 2012.

[19] F. Yi, Z. Yu, F. Zhuang, X. Zhang, and H. Xiong, "An integrated model for crime prediction using temporal and spatial factors," in *IEEE International Conference on Data Mining*, 2018, pp. 1386–1391.

[20] F. Yi, Z. Yu, F. Zhuang, and B. Guo, "Neural network based continuous conditional random field for fine-grained crime prediction," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 4157–4163.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

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

[24] Y. Zheng, F. Liu, and H. P. Hsieh, "U-air:when urban air quality inference meets big data," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 1436–1444.

[25] J. Shang, Y. Zheng, W. Tong, E. Chang, and Y. Yu, "Inferring gas consumption and pollution emission of vehicles throughout a city," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 1027–1036.