User Recruitment for Enhancing Data Inference Accuracy in Sparse Mobile Crowdsensing

Wenbin Liu, Yongjian Yang, En Wang, and Jie Wu, *Fellow, IEEE*

***Abstract*—Sparse mobile crowdsensing is a practical paradigm for large sensing systems, which recruits a small number of users to sense data from only a few subareas and, then, infers the data of unsensed subareas. In order to provide high-quality sensing services under a budget constraint, we would like to select the most effective users to collect useful sensing data to achieve the highest inference accuracy. However, due to the variable user mobility and complicated data inference, it is really challenging to directly select the best user set which helps the most with data inference. From the user's side, we can obtain the probabilistic coverage according to the users' mobilities, while the probabilistic coverage cannot indicate the data inference accuracy directly. From the subarea's side, we may identify some more useful subareas under the current states (e.g., the previous sensed subareas and the current expected coverage), while these useful subareas may not be covered by the users. Moreover, both the user mobility and data inference introduce a lot of uncertainty, which yields nonmonotonicity and thus nonsubmodularity in the user recruitment problem. Therefore, in this article, we study the user recruitment problem on both the user's and subarea's sides and propose a three-step strategy, including user selection, subarea selection, and user–subarea-cross (US-cross) selection. We first select some candidate user sets, which may cover the most subareas under the budget constraint (user selection), then estimate which subareas are more useful on data inference according to the selected candidates (subarea selection), which finally guides us to recruit the best user set (US-cross selection). Extensive experiments on two real-world data sets with four types of sensing tasks verify the effectiveness of our proposed user recruitment algorithms, which can effectively enhance the data inference accuracy under a budget constraint.**

***Index Terms*—Compressive sensing (CS), local beam search (LBS), mobile crowdsensing (MCS), reinforcement learning (RL).**

Manuscript received August 19, 2019; revised November 8, 2019 and November 15, 2019; accepted November 27, 2019. Date of publication December 3, 2019; date of current version March 12, 2020. This work was supported in part by the National Natural Science Foundations of China under Grant 61772230 and Grant 61972450; in part by the Natural Science Foundation of China for Young Scholars under Grant 61702215; in part by the China Postdoctoral Science Foundation under Grant 2017M611322 and Grant 2018T110247; in part by the Changchun Science and Technology Development under Grant 18DY005; and in part by NSF under Grant CNS 1824440, Grant CNS 1828363, Grant CNS 1757533, Grant CNS 1618398, Grant CNS 1651947, and Grant CNS 1564128. *(Corresponding author: En Wang.)*

W. Liu, Y. Yang, and E. Wang are with the College of Computer Science and Technology, Jilin University, Changchun 130012, China (e-mail: liuwb16@mails.jlu.edu.cn; yyj@jlu.edu.cn; wangen@jlu.edu.cn).

J. Wu is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: jiewu@temple.edu).

Digital Object Identifier 10.1109/JIOT.2019.2957399

I. Introduction

MOBILE crowdsensing (MCS) is a promising mechanism [1], which allows a large number of users with mobile devices to address various sensing tasks, such as the monitoring of the environment [2], traffic congestion [3], and urban infrastructure status [4]. In order to provide high-quality sensing services, traditional MCS systems are built to recruit a large number of mobile users to cover most of the target sensing areas [2]–[7], which obviously costs a lot and can hardly deal with some subareas with no user. Hence, researchers have proposed to collect data from only a few subareas, and then exploit the inherent correlations among the sensing data and use data inference algorithms to deduce the data in the remaining subareas, which is called sparse MCS [8]–[12]. In this way, sparse MCS can significantly reduce the number of required users while high-quality sensing services can still be achieved.

In sparse MCS, one key issue is *user recruitment*, that is, the organizer expects to recruit a limited number of users (under budget constraints), who can collect data from a few useful subareas that are the key to data inference, in order to achieve the highest data accuracy for sensing services. Fig. 1 illustrates a general scenario of the user recruitment in sparse MCS, where the target sensing area is split into 5×4 subareas and the users are unconsciously moving among them. We prefer to recruit two effective users (under budget constraints) who collect data from the $3 + 2 = 5$ subareas they pass by during a period of time. Then, with these five useful values, we can infer the data of the remaining unsensed subareas with the highest data inference accuracy. However, due to the variable user mobility, we cannot accurately predict which subareas will be covered by the users. Moreover, without foreknowing the true values of the subareas, it is hard to predict which subareas are more helpful for the complicated data inference. Hence, exploiting the user mobility with data inference to recruit the most effective users is more challenging in sparse MCS, especially, when the number of recruitment users is limited.

The existing works mainly focus on the data inference [8]–[15] while ignoring the user mobility. In order to reflect the inherently sophisticated value distribution and the prior correlations between sensing values, the existing works mainly use the compressive sensing (CS) technology and its variants (e.g., spatio–temporal CS [8], [9], [13] and Bayesian CS [15]) as the data inference algorithms. By using the CS-based methods, these works can collect data from a few subareas and infer the full map. To select the

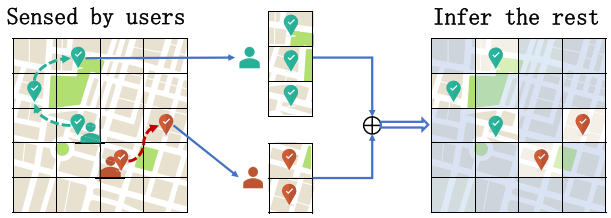


Fig. 1. Users sense data from the subareas they pass by, then upload them to infer the data of unsensed subareas.

most useful sensing subareas for data inference, they also design some subarea selection strategies. Liu *et al.* [14] and He and Shin [15] infer the current sensing data for all subareas, compare them with those collected/inferred data in the last sensing cycle, and then decide to sense the subarea which has the largest difference. Wang *et al.* [8], [9], [13] used various inference algorithms and sensed the most uncertain subarea, in which the inferred data of various algorithms have the largest variance. All these works mainly focus on selecting the effective subareas, but they ignore whether the selected subareas could be easily achieved by the users. In fact, some subareas may be useful for data inference but no users will reach there and collect data. Therefore, the more practical approach is to turn attention from subarea selection to user selection/recruitment, i.e., to select the effective users who collect data from the subareas they pass by, and then use these useful data to infer the data of the unsensed subareas.

In this article, by exploiting the data inference with user mobility, we propose several approaches to address the user recruitment problem in sparse MCS. Considering the variable user mobility and complicated data inference, it is really challenging to directly select the best user set which helps the most in data inference. From the user's side, we can only obtain the probabilistic coverage by using the mobility prediction model to predict the users' mobilities, while the probabilistic coverage cannot indicate the data inference accuracy directly. From the subarea's side, we may identify some effective subareas under certain states (e.g., the previous sensed subareas and the current expected coverage) for the CS-based data inference algorithms, while these effective subareas may not be covered by the users. Moreover, both the user mobility and data inference introduce a lot of uncertainty, e.g., a newly recruited user sensing, some abnormal data may lead to bad results, which yields nonmonotonicity and thus nonsubmodularity in the user recruitment problem. In other words, we should select the best user set from such a large number of possible user sets (due to the user mobility), each of which has a probabilistic coverage of subareas with a nonlinear utility (due to the data inference). Therefore, considering the user mobility and data inference, the user recruitment problem in sparse MCS becomes so challenging that we should study the problem on both the user and subarea sides and propose a three-step strategy that consists of *user selection*, *subarea selection*, and *US-cross selection*.

First, by using the mobility prediction, we select some candidate user sets to cover the most subareas under the budget constraint (user selection), which significantly reduces the number of candidates without considering the data inference. Specifically, we propose a local beam search (LBS) method to select the best k candidate user sets which can cover the most subareas, instead of considering directly data inference accuracy. In general, the more covered subareas will provide more information for enhancing the data inference accuracy, based on which we can roughly, but significantly, reduce the possible candidate user sets. Moreover, we aim to roughly select k candidate user sets, rather than search the best one. Thus, we propose the LBS method to cut the bad user sets while hold the good ones, which can further reduce the resource consumption and improve the time efficiency. In addition, we can use the beamwidth k in the LBS method to ensure that our kept k candidate user sets will cover the best one for data inference.

Then, according to the expected coverage by the candidate user sets, we identify the useful subareas under the current states (subarea selection), which may achieve the highest inference accuracy without considering the user mobility. Specifically, we formulate the subarea selection as a finite Markov decision process (MDP) and use a reinforcement learning (RL) method to select the useful subarea sets. The basic idea is to try out all the possible subareas and record their inference accuracies. Note that the effectiveness of subareas is determined by the different states and the selected subareas will also change the states. Hence, we use RL to deal with such interactions between the selected subareas and the states through trial and error. Compared with the existing subarea selection methods, which mainly use some indirect measures (the difference between the sensing cycles or algorithms), RL can select the more effective subareas, which directly influences the cumulative inference accuracy.

Finally, we cross the candidate user sets and useful subareas to recruit the best user set (US-cross selection). Due to the variable user mobility and complicated data inference, we first use the LBS-based user selection to reduce the large number of candidates and provide the possible states, which are then used in the RL-based subarea selection to identify the useful subareas. Considering the changed effectiveness of subareas under different states, we conduct a weighted cross between the candidate user sets and the useful subareas to select the best one. In this way, considering the user mobility and data inference, our proposed three-step strategy can deal with the user recruitment problem in sparse MCS on both user's and subarea's sides.

In summary, this article has the following contributions.

1) We formalize the user recruitment problem in sparse MCS, in order to make full use of user mobility with data inference and provide high-quality sensing services.
2) Due to the variable user mobility and complicated data inference, we propose a three-step user recruitment strategy on both user and subarea sides. We first propose an LBS method to select k user sets as candidates, which cover the most subareas. Using RL, we then identify which subareas are more effective in data inference, which finally guides us to recruit the best user set from the candidates.
3) We evaluate the proposed algorithms on two real-world data sets with four typical sensing tasks and verify the

effectiveness of our proposed algorithms in enhancing the accuracy of the inferred results.

The remainder of this article is organized as follows. First, we review related works in Section II. Then, the user recruitment problem is formulated in Section III and we present our three-step strategy in Section IV. The performance is evaluated in Section V and, finally, we conclude this article in Section VI.

II. Related Work

A. Sparse Mobile Crowdsensing

With the rapid development of mobile communications and smart devices, MCS becomes a powerful sensing paradigm, which allows users to use the smart devices carried by them to sense data from the target areas they pass by [1], [2], [16]. In order to provide high-quality sensing services, most of the existing MCS systems have to recruit a large number of users to sense data from all of the target sensing areas [6], [17]–[19]. Obviously, these systems cost a lot for user recruitment, and it is still hard to avoid that there are some subareas that have not been covered, since we may find no participants in these subareas. To deal with this problem and further reduce the costs, some researchers proposed to sense data from only a few subareas and use some data inference algorithms to infer the data in unsensed subareas, which is called sparse MCS [8].

Recently, many sparse MCS systems have been developed for various large-scale sensing systems and achieve very good performances. Rana *et al.* [20] presented a participatory urban noise mapping system, which uses the incomplete and random crowdsourcing data to infer the urban noise map by using CS. Zhu *et al.* [21] also proposed a CS approach for the traffic estimation from the data periodically collected by probe vehicles. Wang *et al.* [8]–[10], [13] formally proposed the sparse MCS paradigm and presented a framework with three stages: 1) data inference; 2) quality assessment; and 3) cell selection. They also conducted experiments with applications in temperature, humidity, air quality, and traffic monitoring to verify the effectiveness of sparse MCS. Liu *et al.* [14] presented an incentive design for the air pollution monitoring system in sparse MCS. He and Shin [15] also presented an incentive mechanism based on Bayesian CS in sparse MCS, in order to steer the crowdsourced signal map construction. With sparse MCS, these works can use only a few sensed data to infer the full sensing map with high accuracy, which can significantly reduce the sensing costs while providing high-quality services.

B. User Recruitment

In MCS, user recruitment is a foundational issue where the organizer would like to recruit the most effective users, in order to provide high-quality sensing services. Karaliopoulos *et al.* [22] considered user recruitment as a minimum cost set cover problem and proposed a greedy method to deal with it. Pu *et al.* [23] formulated an online multiple stopping problem to dynamically select users for the self-organized MCS systems. Xiao *et al.* [24] further considered the deadlines and sensing duration of tasks. Liu *et al.* [6] paid more attention to user mobility and proposed a prediction-based user recruitment strategy to effectively select users to perform more tasks. Wang *et al.* [17] also considered the cost of data uploading and proposed an efficient prediction-based solution for user recruitment. All of the above works intend to utilize the user mobilities to effectively select the best user set which can cover more target sensing areas (or complete more sensing tasks). In order to further reduce the costs, sparse MCS is presented to use data inference algorithms to infer the full sensing maps from partially sensed data.

In sparse MCS, almost all of the existing works use the CS or its variations as the data inference algorithms to infer the full map from the partially sensed data, in order to enhance the data accuracy. However, these works mainly focused on the data inference but paid less attention to user recruitment. Rana *et al.* [20] and Zhu *et al.* [21] ignored the user recruitment and mainly used the incompletely and randomly collected data to recover the full map. Some researchers ignored the user mobilities and assumed that the users in one subarea can be recruited immediately when the subarea has been selected to sense, and thus they considered the subarea selection as user recruitment. Liu *et al.* [14] and He and Shin [15] designed the incentive mechanisms to steer users to sense data from the subareas with more value differences between the last and current sensing cycles. Wang *et al.* [8], [9], [13] used several inference algorithms to deduce the full maps, and then choose the most uncertain subarea to sense, in which the inferred data of various algorithms have the largest variance. These previous works intend to select the effective subareas to sense, in order to achieve the high data inference accuracy. However, the subarea selection methods ignore whether the selected subareas could be easily covered by the users, since they have not considered the user mobilities.

III. System Model and Problem Formulation

In this section, we first present the system model. Then, the mobility prediction model and CS method are introduced briefly. Finally, we formulate the user recruitment problem in sparse MCS and provide a running example. The main notations used throughout this article are illustrated in Table I.

A. System Model

We consider a general sensing scenario where the requester wants to obtain fine-grained sensing results around a large-scale sensing area for a period of time. In order to provide high-quality sensing services, the whole sensing campaign is equally divided into some sensing cycles, denoted as $T \triangleq \{t_1, t_2, \ldots, t_\tau\}$ with $t_1 = [t_1^b, t_1^e]$. Similarly, the target sensing area is split into m subareas, denoted as $A \triangleq \{a_1, a_2, \ldots, a_m\}$, and then fine-grained sensing results of all m subareas are provided for each sensing cycle. The lengths of sensing cycles and the sizes of subareas are determined according to the requirements of the certain sensing task.[1] As an example, Zheng *et al.* [25] split the Beijing urban area into 1000×1000 m^2 subareas and would like to provide fine-grained air quality sensing services for all subareas every hour. Under such a large-scale target area, we usually have a large m

[1]If they are irregular, we may further utilize some algorithms, such as numerical interpolation method, to better capture their inherent correlations for data inference.

TABLE I
MAIN NOTATIONS

Symbol	Meaning
T, t_τ	The sensing cycle set, with τ cycles.
A, a_m	The subarea set, with m subareas.
U, u_n	The user set, with n users.
μ_u	The recruited user set.
B_u	The number of recruited users.
$Z_u(a_i, a_j, t)$, $Q_u(a_i, a_j, t)$	The probability that u moves from a_i to a_j within time t, and just at time t.
$P(\mu_u, a_j, t_s)$	The probability that μ_u will cover a_j within the s-th sensing cycle.
$E(\mu_u, s_j)$	The expected number of the covered subareas by μ_u within the s-th sensing cycle.
V, $\hat{V}$	The ground truth of m subareas and the inferred ones.
F, $\hat{F}$, F'	The ground truth matrix of m subareas in k sensing cycles, the inferred matrix and the actual sensed matrix.
C	The sensing matrix, which marks whether one subarea has been sensed.

for the fine-grained sensing results and need to recruit a large number of users, which costs a lot, and thus we introduce the sparse MCS to deal with this problem.

At each sensing cycle, we consider that there are n users, denoted as $U \triangleq \{u_1, u_2, \ldots, u_n\}$, moving around the m subareas. During a sensing cycle, users may pass by several subareas, denoted as $l_i \triangleq \{a_{i_1}, a_{i_2}, \ldots, a_{i_{|l_i|}}\}$ for user u_i. We assume that they can successfully and accurately sense data at their covered subareas if they have been recruited.[2] We also consider a budget constraint B_u on the number of recruited users, that is, we only recruit a number of B_u users for each sensing cycle and use μ_u to denote the recruited user set. These users may cover some subareas according to their mobilities (mobility prediction in Section III-B). Then, we use some historical data sensed from the previous cycles and the current data collected by μ_u to infer the data of the remaining unsensed subareas in the current sensing cycle (data inference in Section III-C).

B. Mobility Prediction via Semi-Markov Model

From the opportunistic perspective, the recruited users are unconsciously moving among the sensing subareas and we consider that they can successfully and accurately sense data at their covered subareas. Considering the time constraints of the sensing cycles and strong laws governing the mobility of humans, we use a modified semi-Markov model [6], [7], [17] to predict the time-dependent transition probabilities between the subareas as the user's mobility prediction. In this model, the subareas can be seen as the states and mobile users moving between subareas can be seen as the transition between states. Then, we can predict the probabilities that users cover each subarea within the sensing cycle, which are further used to estimate the data inference accuracy.[3]

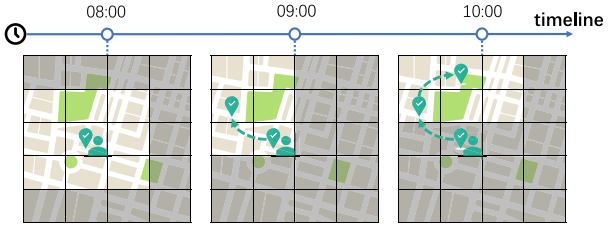


Fig. 2. Example of the mobility prediction model.

Specifically, we consider the user mobility prediction among the target sensing subareas. In order to further reduce the large number of computations, for each user, we only consider the transitions between the nearby subareas he can reach (by walk, bicycles, or vehicles) within the sensing cycles, while ignoring the other invalid subareas. Fig. 2 provides an example of our mobility prediction model, where we only consider the user's nearby subareas. The time-dependent semi-Markov kernel $Z_u(a_i, a_j, t)$, i.e., the probability that user u will move from his current subarea a_i to his next subarea a_j within time t, is defined by

$$Z_u(a_i, a_j, t) = Z\left(A_u^{n+1} = a_j, t_u^{n+1} - t_u^n \leq t | A_u^n = a_i\right) \quad (1)$$

where t indicates the time constraint and A_u is the user's moving sequence of subareas. For this probability Z_u, we only consider the nearby subareas and calculate it from the statistical results of users' historical mobility records. Furthermore, we consider the relay state transitions and obtain $Q_u(a_i, a_j, t)$, i.e., the probability that user u will move from the subarea a_i to a_j just at the time t as follows:

$$Q_u(a_i, a_j, t) = \begin{cases} \Sigma_{a_k}^{A_u} \Sigma_{t'=1}^{t} \big(Z_u(a_i, a_k, t') - Z_u(a_i, a_k, t'-1)\big) \\ \quad \times Q_u(a_k, a_j, t-t'), \quad a_i \neq a_j \\ 1 - \Sigma_{a_k, a_k \neq a_i}^{A_u} (Z_u(a_i, a_k, t) \\ \quad - \Sigma_{t'=1}^{t} \big(Z_u(a_i, a_k, t') - Z_u(a_i, a_k, t'-1)\big) \\ \quad \times Q_u(a_k, a_i, t-t')\big), \quad a_i = a_j \end{cases} \quad (2)$$

where $Q_u(a_i, a_i, 0) = 1$ and $Q_u(a_i, a_j, 0) = 0$, if $a_i \neq a_j$. Specifically, when $a_i \neq a_j$, we consider the relay state transitions as $a_i \to a_k \to a_j$ and calculate the total probability. When $a_i = a_j$, we further consider the probability that users stay at the same subareas. With the $Q_u(a_i, a_j, t)$ from mobility prediction, we obtain $p_{u_i}(a_{u_i}, a_j, t_s)$, i.e., the probability that user u_i (at the subarea a_{u_i}) can cover the subarea a_j within the sth sensing cycle as follows:

$$p_{u_i}(a_{u_i}, a_j, t_s) = 1 - \prod_{t=t_s^b}^{t_s^e} \big(1 - Q_u(a_{u_i}, a_j, t)\big). \quad (3)$$

Then, we derive the probabilities that the recruited user set μ_u can cover the subareas $a_j \in A$ within the sth sensing cycle and

[2]Note that the sensed data are usually error prone and private in MCS, which will directly influence the data inference. Actually, the sensing data quality and privacy protection are important research problems [26]–[28], while they are not the main concerns of this article. Therefore, to simplify the problem, we assume that the recruited users can successfully and accurately sense data from the subareas they pass by.

[3]Note that the mobility prediction will influence the performance of our sparse MCS, but the impact will not be significant. On the one hand, we focus on the predicted coverage of users in a certain period of time but not the more accurate mobility. On the other hand, the data inference algorithm will give us another guarantee and infer the data as accurately as possible.

also calculate the expected number of the covered subareas as follows:

$$P(\mu, a_j, t_s) = 1 - \Pi_{u_i \in \mu}\left(1 - p_{u_i}(a_{u_i}, a_j, t_s)\right) \quad (4)$$

$$E(\mu, t_s) = \sum_{a_j \in A}\left(1 - \Pi_{u_i \in \mu}\left(1 - p_{u_i}(a_{u_i}, a_j, t_s)\right)\right). \quad (5)$$

C. Data Inference via Compressive Sensing

In sparse MCS, we recruit some users for each sensing cycle and the recruited users unconsciously move among the subareas to sense and upload data. Then, we use the historical and current sensed data to infer the data of the remaining unsensed subareas via CS techniques. We model the ground truth of the full map at the sth cycle as $V^s \triangleq [v_1^s, v_2^s, \ldots, v_m^s]^T$, with the inferred values $\hat{V}^s$ and the actual sensed values V^s'. In order to measure the data inference accuracy, the error function is defined as $\mathcal{E}(V^s, \hat{V}^s)$ with the ground truth V^s and the inferred data $\hat{V}^s$ as follows:

$$\mathcal{E}\left(V^s, \hat{V}^s\right) = \sum_{i=0}^{m}\left|v_i^s - \hat{v}_i^s\right|. \quad (6)$$

Considering the last l sensing cycles and the current one, we obtain the ground-truth matrix $F \triangleq [V^{s-l}, \ldots, V^{s-1}, V^s]$, with the inferred matrix $\hat{F}$ and the actual sensed matrix F'. Given historical and current sensed data matrix F', we can use CS as the data inference algorithm to infer the unsensed data at the current sensing cycle. Mathematically, for a certain task, we infer the $\hat{F}$ from the F' based on the low-rank property [29] as follows:

$$\min \ \operatorname{rank}\left(\hat{F}\right) \quad (7)$$

$$\text{s.t.} \ \ \hat{F} \circ C = F' \quad (8)$$

where $\circ$ represents the elementwise multiplication and C marks whether one subarea has been sensed, i.e., $C[i, j] = 1$ means that the subarea a_i has been sensed at the jth cycle; otherwise, $C[i, j] = 0$. Using singular value decomposition, i.e., $\hat{F} = LR^T$, we can convert the above optimization problem from minimizing the rank of $\hat{F}$ to minimizing the Frobenius norms of L and R as in the following optimization:

$$\min \lambda\left(\|L\|_F^2 + \|R\|_F^2\right) + \left\|LR^T \circ C - F'\right\|_F^2 \quad (9)$$

where the condition $\hat{F} \circ C = LR^T \circ C = F'$ has been converted into the optimization and λ allows a tunable tradeoff between rank minimization and accuracy fitness.

In order to better capture the inherent correlations in sensing data, we further present temporal and spatial correlations [9], [29], [30] considered in the optimization

$$\min \ \ \lambda_r\left(\|L\|_F^2 + \|R\|_F^2\right) + \left\|LR^T \circ C - F'\right\|_F^2 + \lambda_t\left\|\left(LR^T\right)\mathbb{T}^T\right\|_F^2 + \lambda_s\left\|\mathbb{S}\left(LR^T\right)\right\|_F^2 \quad (10)$$

where λ_r, λ_t, and λ_s control the tradeoff between different correlations and $\mathbb{T}$ and $\mathbb{S}$ are temporal and spatial correlation matrices defined as follows.

1) $\mathbb{T}$ presents the temporal correlations among the sensing results of the same subarea at different cycles. A simple correlation matrix can be used as $\mathbb{T} = \text{Toeplitz}(0, 1, -1)$, which intuitively reflects that two continuous sensed values from the same subarea are usually similar.
2) $\mathbb{S}$ presents the spatial correlations among the sensing results of the different subareas at the same cycle. In general, the closer subareas usually have the similar sensed values. Therefore, we use the distance between two subareas to model the spatial correlations, denoted as $\mathbb{S}[i, j] = \exp(-\text{distance}(i, j)/\sigma_s^2)$. Then, we normalize the matrix $\mathbb{S}$ as $\sum_{j=1, j\neq i}^{m} \mathbb{S}[i, j] = 1$ and set $\mathbb{S}[i, i] = -1 \quad \forall i = \{1, \ldots, m\}$.

Moreover, if we have domain knowledge or historical data, we can further learn and train a more sophisticated $\mathbb{T}$, in order to capture and express more correlations, such as the periodic changes in sensing data, e.g., traffic speed. Similarly with $\mathbb{T}$, we can further capture the correlations between different but not close subareas in $\mathbb{S}$, e.g., some subareas that have similar surroundings and thus similar sensed results. Without loss of generality, we use the Toeplitz matrix and distance function to express the typical temporal and spatial correlations in sensing data. Actually, the other constraints or correlations can be easily applied in (10). Then, the alternating least squares [29] procedure is used to estimate L and R iteratively to get the optimal $\hat{F}$ (i.e., $\hat{F} = LR^T$), which converges quickly in our experiments (less than 20 iterations and costs ~ 0.5 s, which is totally acceptable in practical use).

D. Problem Formulation

Based on the above system model, mobility prediction model, and CS method, we describe our user recruitment problem for sparse MCS.

Problem (User Recruitment in Sparse MCS): Given a sparse MCS task with m subareas and τ sensing cycles, for each cycle, we recruit a total of B_u users who unconsciously move among the subareas to sense data from their covered subareas, and then use the historical and current sensed data to infer the unsensed data, with the objective of minimizing the cumulative inference errors

$$\text{minimize} \ \ \sum_{s=0}^{\tau}\mathcal{E}\left(V^s, \hat{V}^s\right) \quad (11)$$

$$\text{subject to} \ \ \left|\mu_u^s\right| \leq B_u, 0 \leq s \leq \tau. \quad (12)$$

We now provide an example to illustrate our user recruitment problem for sparse MCS in more details, as shown in Fig. 3. Consider that there are three users moving around the target sensing area, which is spilt into 5×4 subareas. User 1 will pass by three subareas while users 2 and 3 only cover two subareas. For this sensing cycle, we can only recruit $B_u = 2$ users because of the budget constraint. If we recruit users 1 and 3, we can only sense data from the left corner subareas, which may not be a good choice. If we recruit users 2 and 3, they only cover four subareas. Therefore, we would like to recruit users 1 and 2, which can sense five subareas and may achieve better data inference accuracy than other choices, and then use the five sensed results to infer the data of unsensed subareas.

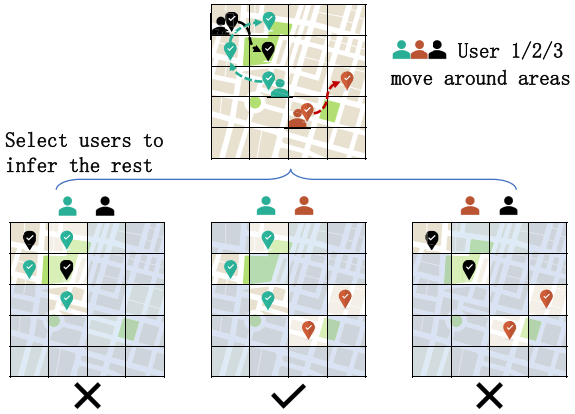


Fig. 3. Example of user recruitment in sparse MCS.

Note that this example is only used to intuitively understand our problem. Actually, not only the number and locations of the covered subareas may influence data inference accuracy but also the complicated inherent correlations among the sensing data and even the previous sensed data would also influence the current decision and data accuracy. In fact, the user recruitment problem in sparse MCS is hard to model and difficult to deal with, and we would like to discuss it in detail in the next section.

IV. User Recruitment in Sparse MCS

In this section, we focus on the user recruitment problem and elaborate on the algorithms used in the three-step strategy: 1) user selection; 2) subarea selection; and 3) US-cross selection. Before the detailed descriptions, we first present an overview of the three-step strategy to see the relationship among the three stages.

A. Overview

As introduced above, it is really challenging to directly select the best user set which helps most on data inference. The users may cover different subareas and the effectiveness of subareas is changed, which makes it hard for us to select the best user set. Also, some subareas are more useful but may not be covered by the users. Moreover, both the user mobility and data inference introduce a lot of uncertainty, e.g., a new recruited user sensing some abnormal data may lead to bad results, which yields nonmonotonicity and nonsubmodularity and makes the user recruitment more complicated. Therefore, we should deal with the user recruitment problem in sparse MCS on both the user and subarea sides, in order to select the effective user set which covers useful subareas and thus achieves better performance on data inference.

Fig. 4 shows the overview of our proposed three-step user recruitment strategy, consisting of user selection, subarea selection, and US-cross selection. The basic idea is to select some candidate user sets and useful subareas from the user's and subarea's sides, respectively, and then cross these candidate user sets and useful subareas to select the proper user set which covers effective subareas.

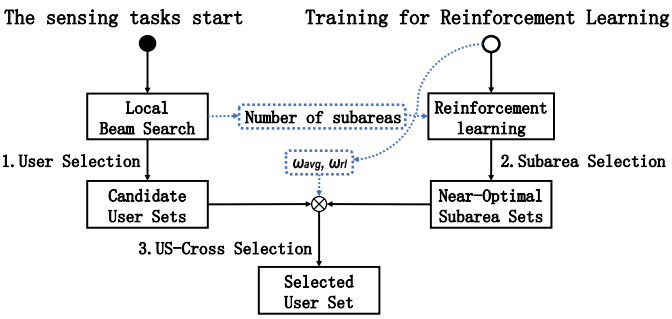


Fig. 4. Overview of the proposed user recruitment strategy in sparse MCS.

Specifically, we first propose an LBS method to select the best k user sets which may cover the most subareas. In general, the more covered subareas will provide more information for data inference and the larger beamwidth k ensures that our selected candidate user sets will cover the best one with higher probability. Then, according to the numbers of subareas covered by the candidate user sets, we use an RL method to select the useful subarea sets, which may achieve the highest inference accuracy, without considering the user covered situations. Finally, we cross these candidate user sets and effective subareas to select the best user set which covers the most effective subareas.

B. User Selection

We consider the user recruitment problem in sparse MCS on both user and subarea sides. The user selection decides the covered subareas and these subareas influence the data inference accuracy. Since the users will cover various subareas, the user selection faces a large solution space. Meanwhile, the different subarea sets covered by selected user sets may achieve different inference accuracies. It is hard to directly select the most useful user set for inference accuracy, due to the huge solution space and significant computing costs on data inference. Therefore, we would like to select some candidate user sets first, without considering the complicated subarea selection while providing good candidates.

Our user selection strategy is to select some user sets which may cover the most subareas instead of considering directly the inference accuracy. In general, the more sensed subareas will provide more information for data inference and thus achieve higher accuracy. As shown in Fig. 5, we have done some experiments to test the number of sensed subareas on four sensing tasks in two real-world data sets, i.e., the monitoring of temperature and humidity in *Sensor-Scope* [31] and PM2.5 and PM10 in *U-Air* [25] (will be elaborated in Section V). We randomly select some subareas to sense and use various data inference algorithms CS and K-nearest neighbors on temporal and spatial dimensions (KNN-T/S) [9]) to deduce other values. The results show that with the increase in the numbers of sensed subareas, the errors of the data inference algorithms will decrease (i.e., the higher accuracy). Therefore, our user selection would like to select some user sets which may cover the most subareas as the candidates, since they have the bigger chances to achieve higher inference accuracy. Then,

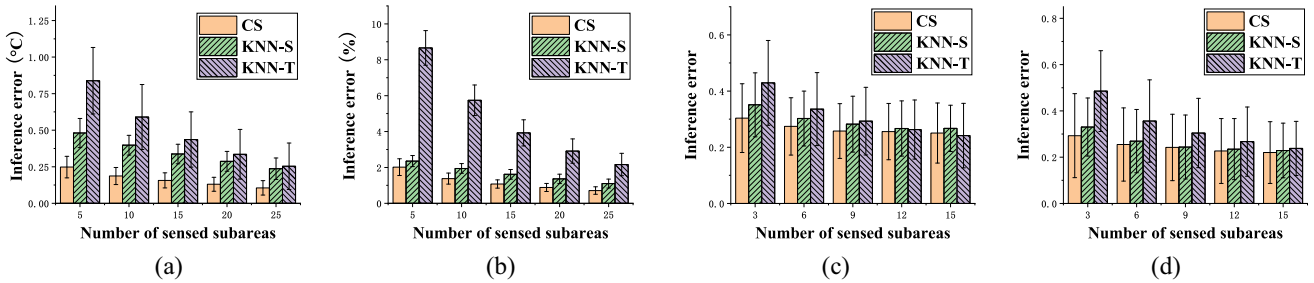


Fig. 5. Inference errors along with the increasing numbers of sensed subareas in *Sensor-Scope* and *U-Air* data sets. (a) *Sensor-Scope*: Temperature. (b) *Sensor-Scope*: Humidity. (c) *U-Air*: PM2.5. (d) *U-Air*: PM10.

Algorithm 1 LBS for User Selection

Initialization:

k, $U = \{u_1, u_2, \ldots, u_n\}$, B_u, $t_s \in T$

$S = \{\mu_1, \mu_2, \ldots, \mu_k\}$, $S' = \{\mu'_1, \mu'_2, \ldots, \mu'_k\}$,

$f(\mu) = E(\mu, t_s)$ calculates the number of covered subareas by μ in the s-th sensing cycle.

1: Init $\mu_i = \mu'_i = \emptyset \quad \forall i = 1, 2, \ldots, k$
2: *// Select B_u users*
3: **for** $t = 1, 2, \ldots, B_u$ **do**
4: *// Greedily select the best k user sets*
5: **for** μ_i in S **do**
6: **for** u_j in $U \setminus \mu_i$ **do**
7: **if** $\mu_{temp} = \mu_i \cup u_j$ not in S' **then**
8: $\mu_{min} = \arg\min f(\mu'_i) \quad \forall \mu'_i \in S'$
9: **if** $f(\mu_{temp}) > f(\mu_{min})$ **then**
10: $\mu_{min} = \mu_{temp}$
11: *// Update S from the kept S′*
12: $S = S'$
13: **return** S

we further consider the data inference and select the best one from the candidates in the next two steps.

In order to select the candidates quickly and effectively, we propose a greedy LBS (LBS) [32] method to select the best k user sets, as shown in Algorithm 1. The basic idea of our greedy LBS-based user selection algorithm is to expand and keep the best k user sets as the candidates. For each selection, we expand the kept candidates successively (line 5) by adding one unselected user into them and keep the best k expanded sets which may cover the most subareas (lines 6–10). Note that we would not hold the sets with same users in our method (line 7). Finally, we obtain the best k user sets each with B_u users for the current sensing cycle. The parameter k is called beamwidth; a larger k has a bigger chance to cover the optimal result but also costs more (not only in user selection but also in data inference).

C. Subarea Selection

Given the candidate user sets from user selection, we need to further select the best one of them which can provide the most information to help data inference. Although the user selection has significantly reduced the number of candidates, we still need to keep a large k, in order to cover the best user set for data inference. Moreover, since we cannot accurately predict how much one user set can help data inference without knowing the ground truth, it is also a big challenge to identify which user set is the most effective one. Fortunately, we can use RL to learn which subareas are more effective under the certain conditions given by the candidate user sets, and then it will guide us to further select the best user set.

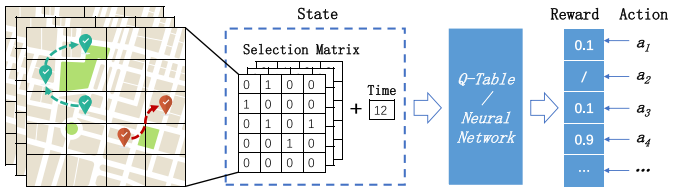


Fig. 6. State, action, and reward in subarea selection.

1) State, Action, and Reward: The basic idea of our RL-based subarea selection is to try out all the possible sensed subareas, infer the data in unsensed subareas, and record the inference accuracy by utilizing the historical data. In this way, we can learn that some subareas are more effective through trial and error, which is exactly the fundamental idea of RL. In general, RL is abstracted to take a sequence of *actions* under certain *states* so as to maximize the cumulative *rewards*, and our subarea selection can be formulated to select a sequence of *subareas to sense* (action) considering the *data already sensed* (state) so as to maximize the final *data inference accuracy* (reward), as shown in Fig. 6.

1) *State* represents the current situation, which influences the choice of action on the data inference accuracy in our problem, denoted as s. We model the state as a one-hot selection matrix for the several recent cycles and a timestamp, where the selection matrix expresses when and where we have sensed data and the timestamp helps RL to learn the temporal patterns in the sensing data.
2) *Action* is what we decide to do under a certain state. In our problem, the action is naturally the next sensed subarea, denoted as a. Note that we model the action as only one subarea instead of a set of subareas to sense, since the set space is too large and we have to add subareas one by one, just like a greedy method.
3) *Reward* represents the revenues obtained by one action under a certain state. In subarea selection, the reward is modeled as the data inference accuracy directly, denoted

as $r = \exp(-\mathcal{E}(V, \hat{V})/\sigma_e^2)$. The smaller inference error $\mathcal{E}$ means the higher accuracy, where r is closer to 1.

2) RL-Based Subarea Selection: With the state, action, and reward, we propose the RL-based subarea selection algorithm as shown in Fig. 6. Specifically, under a certain state, we use the Q-table/neural networks (NNs) to estimate the rewards for all subareas and select subarea a_4 as the action, since it has the largest reward. Actually, RL is to learn the mappings between the state–action pairs and rewards, and we will introduce them in detail as follows.

Q-table is used in the traditional RL algorithms, e.g., Q-learning, where we use a table to record the rewards of all state–action pairs, denoted as $Q_{s\times a}$. For each selection, we search the Q-table to find the action which records the largest reward, i.e., $a = \arg\max Q[s, a] \quad \forall a \in A$. Since RL further considers the future rewards of one action under a certain state, we then iteratively update Q-table according to the following equations:

$$Q[s_t, a_t] = (1-\alpha)Q[s_t, a_t] + \alpha(r_t + \gamma V(s_{t+1})) \quad (13)$$

$$V(s_{t+1}) = \max_{a_{t+1}} Q\big[s_{t+1}, a_{t+1}\big] \quad \forall a_{t+1} \in A \quad (14)$$

where s_t is the current state and $V(s_{t+1})$ represents the iterative future reward under the learning rate $\alpha \in (0, 1]$ and discount factor $\gamma \in [0, 1]$ (indicating the myopic view of the Q-learning regarding the future reward).

Neural network is a powerful tool for RL, where we use NNs instead of Q-table to estimate the rewards for all state–action pairs. In subarea selection, suppose that we have 50 subareas and only keep five cycles of selections as the state (ignore the timestamp), the size of state space achieves $2^{5\times 50}$, which is such a huge space that the traditional Q-table-based algorithms can hardly deal with by using Q-tables. Therefore, we propose to use NNs to replace the Q-table, called deep Q-learning (DQL). For each selection, we use NNs to estimate the rewards for all actions under a certain state instead of searching a large Q-table as shown in the following equations:

$$Q(s_t, a_t) = \mathbb{E}\left[r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})\right]. \quad (15)$$

For the updating or training in DQL, we use the stochastic gradient algorithm to update the neural network parameterized by θ to approximately achieve $Q_\theta(s_t, a_t) \approx Q[s_t, a_t]\ \forall s_t, a_t$. According to (13)–(15), we have the loss function:

$$L(\theta_t) = \mathbb{E}_{\langle s_t, a_t, r_t, s_{t+1}\rangle}\left[\left(r_t + \gamma \max_{a_{t+1}} Q_{\theta_t}(s_{t+1}, a_{t+1}) - Q_{\theta_t}(s_t, a_t)\right)^2\right]. \quad (16)$$

Thus

$$\nabla_{\theta_t} L(\theta_t) = \mathbb{E}_{\langle s_t, a_t, r_t, s_{t+1}\rangle}\left[\left(r_t + \gamma \max_{a_{t+1}} Q_{\theta_t}(s_{t+1}, a_{t+1}) - Q_{\theta_t}(s_t, a_t)\right)\nabla_{\theta_t} Q_{\theta_t}(s_t, a_t)\right]. \quad (17)$$

Specifically, we design the NNs with two dense layers, which can deal with the heterogeneous inputs (state) and achieve good enough performances.[4] The detailed algorithm is summarized in Algorithm 2. For each selection, we obtain the current state s_t, feed it into NNs, and obtain the outputs $Q(s_t, a_t)$ for all a_t in A (line 4). If NNs do not need to train, we directly select the action which has the largest $Q(s_t, a_t)$ (line 14). Otherwise, we use the ϵ-greedy algorithm for each selection to balance the explore and exploit, where we select the best a_t with the probability $1-\epsilon$ or randomly select an action with the probability ϵ (line 6). Then, we conduct the experience $\mathbf{e}_t$ and add it into memory pool $\mathbf{P}$ (lines 7 and 8). For the training steps, we randomly select some experience to learn and update the network parameters θ_t (lines 9 and 10). We also use the fixed Q-targets [33], which holds a target network with the parameters θ' cloned from the primary network but updates θ' periodically (lines 11 and 12). Finally, we obtain the useful subarea sets according to the numbers of covered subareas provided by user selection, which are more effective subareas and can guide us to further select the best user set.

Algorithm 2 RL for Subarea Selection

Initialization:

ϵ, s, a, r, $\mathbf{P}$, A, $|\mu_u|$

1: Init $\mathbf{P} = \emptyset$, set ϵ
2: Init two neural networks with random weights θ_t and $\theta' = \theta_t$
3: **for** $t = 1, 2, \ldots, |\mu_u|$ **do**
4: Obtain s_t and $Q(s_t, a_t) \quad \forall a_t \in A$
5: **if** *isTrain* **then**
6: Select a_t with ϵ-greedy algorithm
7: $r_t = Q(s_t, a_t)$, obtain s_{t+1}
8: $\mathbf{e}_t = \langle s_t, a_t, r_t, s_{t+1}\rangle \rightarrow \mathbf{P}$
9: **if** *isTrain_step_t* **then**
10: Train by P and update θ_t via Eq. (17)
11: **if** *isReplace_step_t* **then**
12: $\theta' = \theta_t$
13: **else**
14: $a_t = \arg\max Q(s_t, a_t) \quad \forall a_t \in A$
15: **return** $\{a_1, a_2, \ldots, a_{|\mu_u|}\}$

D. User–Subarea-Cross Selection

After the user selection and the subarea selection, we obtain k candidate user sets and several useful subarea sets according to the numbers of subareas covered by k candidate sets. The k candidates cover almost all effective user sets and the useful subarea sets tell us which subareas are more effective under the current state with certain numbers of covered subareas. Then, we do a weighted cross between the candidate user sets and the useful subarea sets as shown in the following equations:

$$W(\psi_\mu, \psi_s) = \left(1 - \vec{\psi}_s\right) \cdot \vec{\psi}_\mu \times \omega_{\text{avg}} + \vec{\psi}_s \cdot \vec{\psi}_\mu \times \omega_{rl} \quad (18)$$

where $\vec{\psi}_\mu \triangleq [P(\mu, a_0, t_s), P(\mu, a_1, t_s), \ldots, P(\mu, a_m, t_s)]^T$ indicates the probabilities that the selected user set μ can cover

[4]Other network structures and attention mechanism may further improve the performance, while it is not the main concern of this article.

TABLE II
STATISTICS OF TWO EVALUATION DATA SETS

	Datasets	
	Sensor-Scope	***U-Air***
City	Lausanne (Switzerland)	Beijing (China)
Data	Temperature-Humidity	PM2.5-PM10
Cell size	$50*30m^2$	$1000*1000m^2$
Cell number	57	36
Cycle length	$0.5h$	$1h$
Duration	$7d$	$11d$
Mean Std.	$6.04 \pm 1.87°C$ (T)	79.11 ± 81.21 (PM2.5)
	$84.52 \pm 6.32\%$ (H)	63.12 ± 48.56 (PM10)

the m subareas within the sth sensing cycle and $\vec{\psi}_s$ records the selected subareas by 1 and the unselected subareas by 0 in subarea selection.

Specifically, we first give the average weights ω_{avg} to all unselected subareas and give higher weights ω_{rl} to the selected subareas in the useful subareas. Using the RL training memory pool $\mathbf{P}$, we calculate the inference accuracy for the subareas selected by RL as ω_{rl}. Using other historical records, we obtain the average inference accuracy for all subareas ω_{avg}. Then, we allocate the weights ω_{avg} and ω_{rl} to subareas and do a weighted cross to select the user set with the largest total weights from k candidates, which represents the more covered subareas and the more effective subareas. For example, suppose that we calculate the average inference accuracy by random and RL-based subarea selection as $1 - \mathcal{E}_{\text{RAN}} = 1 - 0.2 = 0.8$ and $1 - \mathcal{E}_{RL} = 1 - 0.1 = 0.9$, and then give the weight 0.8 to all unselected subareas and 0.9 to the selected subareas. Finally, we calculate the total weights of k candidate user sets, respectively, and then decide the final selected user set with the largest weight. In this way, we utilize the useful subarea sets to select the expected effective user set from k candidates, which not only cover more subareas but also cover more effective subareas and thus may be the most helpful for data inference.

V. PERFORMANCE EVALUATION

In this section, we conduct extensive experiments based on two real-life data sets, which contain various types of sensed data, including temperature, humidity, PM2.5, and PM10.

A. Data Sets

We adopt two well-known sensed data sets, *Sensor-Scope* [31] and *U-Air* [25], to evaluate our user recruitment strategy for sparse MCS. *Sensor-Scope* [31] contains two typical types of environment readings, i.e., temperature and humidity. *U-Air* has two important types of sensed data for air quality monitoring, i.e., PM2.5 and PM10. Table II shows the detailed statistics of *Sensor-Scope* and *U-Air* and their descriptions are introduced as follows.

The *Sensor-Scope* [31] data set contains two representative types of environment readings, i.e., temperature and humidity, which were collected by static sensors in the EPFL campus. The sensing area is about 500×300 m^2 and we obtain 57 subareas each with the size of 50×30 m^2 which had continual sensing readings, as shown in Fig. 7.

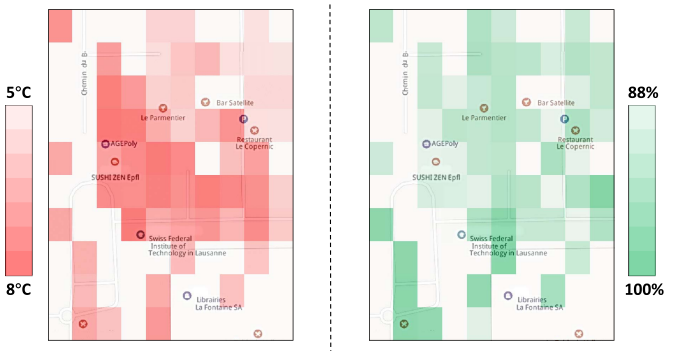


Fig. 7. Example of sensing readings in *Sensor-Scope*. (a) Temperature. (b) Humidity.

The *U-Air* [25] data set collects the important air quality data, i.e., PM2.5 and PM10, by monitor stations deployed in Beijing, China. As in [25], we obtain 36 subareas each with the size of 1000×1000 m^2. In fact, *U-Air* has an unbalanced subarea distribution, which is relatively dense in the urban areas and very sparse in the suburbs. In addition, as shown in Table II, the air quality readings have the large fluctuations and we use the air quality index category [25] instead of the original readings.[5]

Although these data were sensed by static sensors, we assume that mobile users passing by the sensing areas covered by sensors means that they can successfully sense the data by their mobile devices. Since we cannot obtain the real-life mobility traces exactly mapped by the sensed time and locations, we simply generate a large number of continuous moving trajectories in the sensing areas as the users' mobility traces, according to the widely used *Cambridge Haggle Trace Set* [34] for *Sensor-Scope* and *GeoLife* [35] for *U-Air*. The *Cambridge Haggle Trace Set* contains a total of five traces collected from office and conference environments by people carrying mobile devices over a number of days, which can be easily mapped to the EPFL campus of *Sensor-Scope*. The *GeoLife* contains the GPS data collected from phones carried by 182 users, which record a broad range of users' outdoor movements in Beijing (the same city as in *U-Air*). Thus, we consider our experiments as the mobile users moving around the sensing areas and collecting data from the subareas they pass by, which can help evaluate our proposed user recruitment strategy effectively.

In the experiments, for user mobility, we select n trajectories as the participating users who cover 0–5 subareas for each sensing cycle during the whole sensing process, and the distributions are shown in Fig. 8. Note that *Sensor-Scope* has a small size (the EPFL campus) and most users will cover five subareas within one sensing cycle, while *U-Air* has such a large size (Beijing City) that the users will cover few subareas. For data inference, we use the temporal–spatial CS method described in Section III-C. For RL, we use the first two days' data as the

[5]Six categories: good (0–50), moderate (51–100), unhealthy for sensitive groups (101–150), unhealthy (150–200), very unhealthy (201–300), and hazardous (>300).

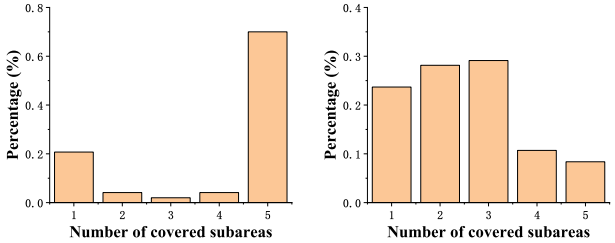


Fig. 8. Distribution of the covered subareas numbers for each user per cycle. (a) *Sensor-Scope* (b) *U-Air*.

training data and the rest for testing. In each sensing cycle, we select B_u users and use the data sensed by their covered subareas to deduce the full sensing maps and obtain the inference error $\mathcal{E}$. Then, we calculate the average $\mathcal{E}$ obtained from all cycles as the data accuracy for evaluation. In order to further reduce errors caused by random traces, we repeat this process over 10 times to get the average results for each experiment.

B. Algorithms and Configurations

We compare our user recruitment strategy with three methods: 1) MAX-LBS; 2) OPT-RL; and 3) RAN as follows.

1) *MAX-LBS* first uses a greedy LBS method to select the best k user sets, as introduced in Section IV-B. Then, from the k candidate user sets, we directly select the one which may cover the most subareas.
2) *OPT-RL* is an RL-based subarea selection algorithm. The basic idea is to try out all of the possible subareas to sense and thus learn which subarea may achieve the largest reward under certain conditions. OPT-RL has not considered if the subareas can be covered by mobile users, which can be seen as the near-optimal selection for comparison.
3) *RAN* randomly selects users and then we use the data sensed from their covered subareas to infer the full sensing maps.

For RL, we conduct a simple neural network with two fully connected layers and initialize it with random weights. Note that the training process of RL is not stable, and the random initialization strategy may influence the final results. Following the existing works, we use the experience replay, fixed Q-targets, and ϵ-greedy algorithm to deal with the problems, as discussed in Section IV-C. For the parameters, we dynamically adjust ϵ from 1 to 0.1 for the whole process of training and set the learning rate $\alpha = 0.05$ and discount factor $\gamma = 0.99$ in (13). For the RL model, we keep the selection matrix with the recent five cycles and the timestamp T from $\{0, 1, \ldots, 47\}$ as the state (input) and the neural network outputs the rewards of actions (subareas). Thus, the size of input is $5 \times 57 + 1 = 343$ and the size of output is 57. Similarly, for *U-Air*, we also keep recent five cycles, T is set as $\{0, 1, \ldots, 23\}$ and the sizes are 217 and 36. In addition, we use the Toeplitz(0,1,−1) and distance function as our temporal and spatial correlation matrices in the CS method and set $\lambda_r = 0.2$, $\lambda_{s/t} = 0.1$ for (10) and $\sigma_s = 1$ for distance function, without loss of generality.

C. Experimental Results

We evaluate the performances of our user recruitment strategy for sparse MCS on two real-life sensing tasks. First, we display a complete picture of the average inference errors which are achieved by our user recruitment strategy under two changed conditions, i.e., the number of recruited users B_u and the number of total users n, as shown in Fig. 9. We can see that the inference errors over two types of sensing tasks have the similar tendencies. Along with the increasing of B_u and n, our proposed user recruitment strategy can recruit the effective users to enhance the data accuracy (reduce the inference errors). In the next sections, we will evaluate and discuss the performances of our user recruitment strategy from the number of recruited users, the number of total users, the beamwidth, and the running times of all tested methods in detail.

1) Number of Recruited Users: We first test the average inference errors under different numbers of recruited users B_u. We change B_u from 1 to 5 while keeping the number of total users $|U| = n = 100$ and the beamwidth $k = 10$. The results are shown in Fig. 10. Note that we set B_u to a small number while keeping a large n, in order to evaluate that a small number of users can achieve a high inference accuracy, which is exactly the fundamental idea of sparse MCS.

With the increase of B_u, the inference errors drop rapidly. The reason is that more recruited users mean more covered subareas, which can provide more information for data inference and thus enhance data accuracy. Similarly, MAX-LBS can recruit the user set which covers the most subareas, and thus it achieves better performance than RAN. Moreover, our strategy always outperforms MAX-LBS, since it finds the more effective user sets from the candidates which can cover the most subareas. Meanwhile, OPT-RL actually can be seen as the upper bound because it selects subareas without considering the user mobilities, and our user recruitment strategy is very close to it.

Specifically, for the temperature, our user recruitment strategy can reduce inference errors by 14.8%–31.4% compared with RAN, and 10.3%–25.1% compared with MAX LBS, under the same number of recruited users. Meanwhile, it has only 5.2%–7.8% more inference errors than OPT-RL, which has not considered the user mobilities. Similarly, for the humidity, our strategy can give ∼24.2% and ∼14.3% less than RAN and MAX-LBS, and ∼9.4% more than OPT-RL. Actually, when we select three users to sense, we can achieve a very small inference error, i.e., 0.167 °C for temperature and 0.88% for humidity, which is totally acceptable.

For the other two types of sensing tasks, i.e., PM2.5 and PM10 in *U-Air*, we obtain similar observations with temperature and humidity in *Sensor-Scope*. Note that we use the air quality index category instead of the original readings and map the six categories into 1–6, in order to evaluate the inference errors. As shown in Fig. 10(c) and (d), our strategy has ∼25.8%/30.6% less error than RAN and 13.4%/15.6% less than MAX-LBS in PM2.5/PM10, respectively. Meanwhile, it increases inference errors by 38.3%/29.0% compared with OPT-RL. We notice that in PM2.5 and PM10, when we recruit

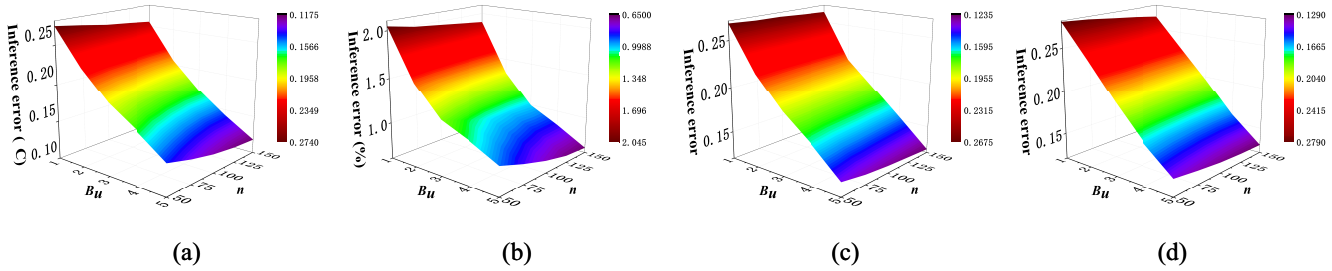


Fig. 9. Inference errors under different numbers of recruited/total users ($k = 10$). (a) Temperature. (b) Humidity. (c) PM2.5. (d) PM10.

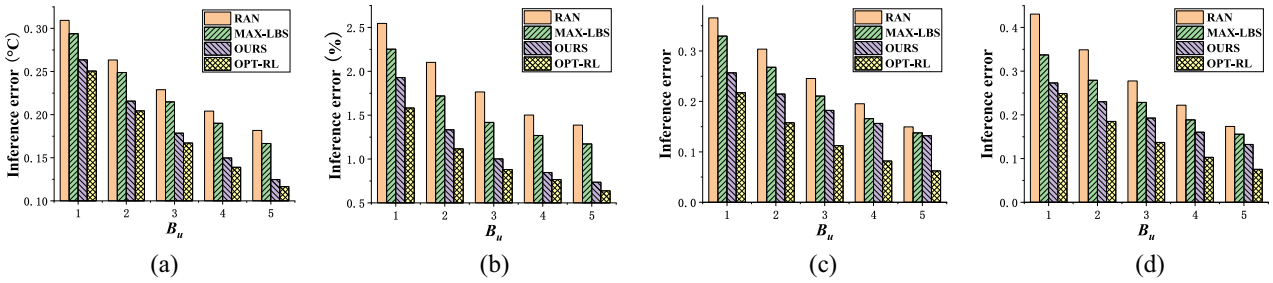


Fig. 10. Inference errors under different numbers of recruited users ($n = 100$ and $k = 10$). (a) Temperature. (b) Humidity. (c) PM2.5. (d) PM10.

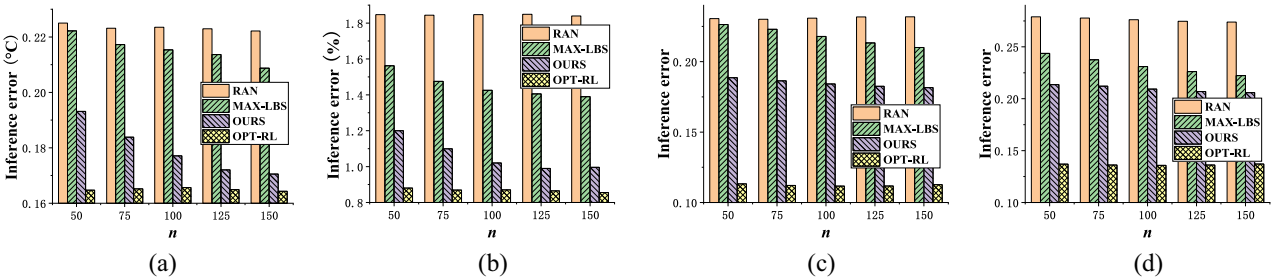


Fig. 11. Inference errors under different numbers of total users ($B_u = 3$ and $k = 10$). (a) Temperature. (b) Humidity. (c) PM2.5. (d) PM10.

more users, our user recruitment strategy reduces the inference errors but the falling rate becomes slow and even close to MAX-LBS. The reason is that we have such a large sensing area in *U-Air* and the subarea distribution is unbalanced, which is relatively dense in the urban areas and very sparse in the suburbs. Thus, some of the subareas can hardly be covered by mobile users, which results in that OPT-RL outperforms MAX-LBS and OURS under a large budget constraint, since OPT-RL does not consider the user mobilities.

2) Number of Users: Then, we evaluate the performances of our user recruitment strategy over different numbers of total users n. We change n from 50 to 150 while keeping the number of recruited users $B_u = 3$ and the beamwidth $k = 10$. The results are shown in Fig. 11.

Obviously, more users mean more user sets, which cost more running time but provide more choices for our user recruitment and thus can improve the performances, particularly, in the LBS-based user selection method. Therefore, along with the increase of the number of users, the MAX-LBS and OURS reduce the inference errors, while the RAN and OPT-RL barely change, since they have not considered the user mobilities. When we have enough users, we can select the best user set from enough choices, and thus the curves tend to get more steady.

Specifically, the RAN and OPT-RL change little while MAX-LBS and OURS decrease from 0.222 °C/0.193 °C to 0.208 °C/0.170 °C and from 1.56%/1.20% to 1.39%/0.99%, respectively. Similarly, for PM2.5 and PM10 in *U-Air*, as shown in Fig. 11(c) and (d), MAX-LBS and OURS decrease from 0.226/0.188 to 0.210/0.181 and 0.243/0.213 to 0.222/0.205, respectively. Note that the decreasing rates are slower than temperature and humidity, since *U-Air* has fewer subareas than *Sensor-Scope*, which needs fewer users to recruit. Also, the unbalanced subarea distribution indicates that the MAX-LBS and OURS have big gaps with OPT-RL.

3) Beamwidth: We also conduct some experiments on beamwidth k to further evaluate the inference error and running time. We change the beamwidth k from 1 to 10, and keep the number of recruited users $B_u = 3$ and the number of total users $n = 100$. The results are shown in Fig. 12. The inference errors of our user recruitment strategy decrease rapidly when we have a small k, which shows that our user recruitment strategy can effectively select the best user set from candidates. When k becomes larger, the inference errors begin to level off since we have kept enough candidates who have already covered the most effective one for data inference. Meanwhile, the running time over two tasks shows the linear growth all along, since the beamwidth k mainly influences the

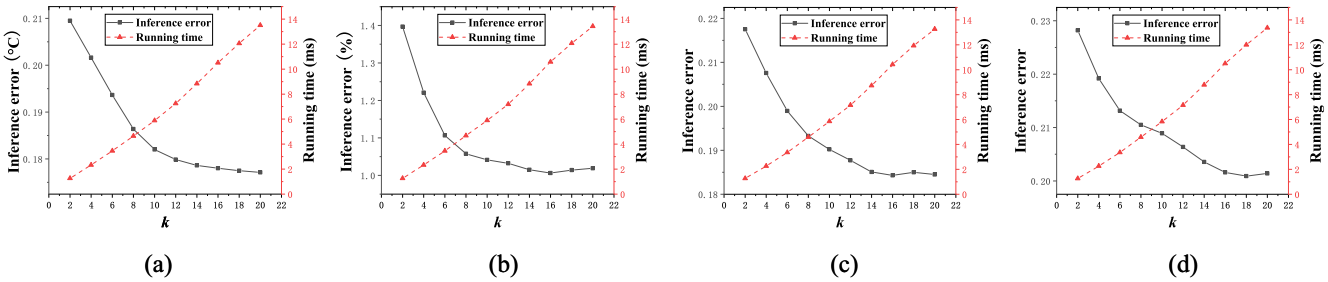


Fig. 12. Inference errors under different beamwidths ($B_u = 3$ and $n = 100$). (a) Temperature. (b) Humidity. (c) PM2.5. (d) PM10.

TABLE III
RUNTIME UNDER $B_u = 3$, $n = 100$, AND $k = 10$

	Temperature	Humidity	PM2.5	PM10
CS	0.49s	0.50s	0.35s	0.37s
RAN	0.018ms	0.019ms	0.017ms	0.016ms
MAX-LBS	5.89ms	5.90ms	5.85ms	5.84ms
OPT-RL	1.27ms	1.30ms	0.93ms	0.72ms
OURS	7.39ms	7.51ms	7.21ms	7.04ms

LBS-based user selection by holding and expanding the best k branches in LBS.

These results also verify the necessity and effectiveness of our three-step user recruitment strategy. We can see that although the user sets, which are added as a result of a larger k, may cover fewer subareas (or cover the same number of subareas), some of them are more effective on data inference. Also, as shown in Figs. 10 and 11, OPT-RL may find out the effective subareas, but they may not be covered by mobile users. Thus, our user recruitment strategy considers both user and subarea sides, which can select the best user set which covers the most effective subareas and thus enhances the data inference accuracy.

4) Running Time: Finally, we display the running times in Table III, with the setting $B_u = 3$, $n = 100$, and $k = 10$ as representative. Our experiment platform is equipped with Intel Xeon CPU E5-2630 v4@2.20 GHz and 32-GB RAM. For data inference, the CS method costs 0.35–0.50 s to infer the full sensing maps for the four tasks, which is totally acceptable in real-life deployments. For user recruitment, our proposed strategy costs only 7.0–7.5 ms, in which the LBS and RL cost $\sim$5.9 ms and $0.7-1.3$ ms, respectively. In addition, the running of computing the mobility prediction model consumes around 5–10 min. The RL method is implemented in TensorFlow (CPU version) and the training can be conducted offline, which costs $\sim$30 min for the neural network with two fully connected layers.

VI. CONCLUSION

In this article, we investigated the user recruitment problem in sparse MCS, which can recruit a small number of users to sense data from only a few subareas while inferring the data of unsensed subareas with high accuracy. Due to the variable user mobility and complicated data inference, we study the user recruitment problem on both user and subarea sides and proposed a three-step user recruitment strategy for sparse MCS. First, we presented an LBS method to select some candidate user sets. Then, we used RL to identify which subareas are more effective, which finally guides us to select the best user set from the candidates by using a weighted cross method. Extensive evaluations on two real-world data sets with four sensing tasks have verified the effectiveness of our proposed algorithms. In future work, we would like to introduce some practical mobility prediction methods and explore the privacy protection mechanism in our user recruitment solutions for sparse MCS.

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, Nov. 2011.
[2] D. Zhang, L. Wang, H. Xiong, and B. Guo, "4W1H in mobile crowd sensing," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 42–48, Aug. 2014.
[3] Z. Liu, S. Jiang, P. Zhou, and M. Li, "A participatory urban traffic monitoring system: The power of bus riders," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2851–2864, Oct. 2017.
[4] H. Aly, A. Basalamah, and M. Youssef, "Automatic rich map semantics identification through smartphone-based crowd-sensing," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2712–2725, Oct. 2017.
[5] M. H. Cheung, R. Southwell, F. Hou, and J. Huang, "Distributed time-sensitive task selection in mobile crowdsensing," in *Proc. 16th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2015, pp. 157–166.
[6] W. Liu, Y. Yang, E. Wang, Z. Han, and X. Wang, "Prediction based user selection in time-sensitive mobile crowdsensing," in *Proc. 14th Annu. IEEE Int. Conf. Sens. Commun. Netw. (SECON)*, San Diego, CA, USA, Jun. 2017, pp. 1–9.
[7] Y. Yang, W. Liu, E. Wang, and J. Wu, "A prediction-based user selection framework for heterogeneous mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 11, pp. 2460–2473, Nov. 2019.
[8] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, and A. M'hamed, "Sparse mobile crowdsensing: Challenges and opportunities," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 161–167, Jul. 2016.
[9] L. Wang *et al.*, "SPACE-TA: Cost-effective task allocation exploiting intradata and interdata correlations in sparse crowdsensing," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 2, pp. 1–28, 2018.
[10] L. Wang, W. Liu, D. Zhang, Y. Wang, E. Wang, and Y. Yang, "Cell selection with deep reinforcement learning in sparse mobile crowdsensing," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Vienna, Austria, 2018, pp. 1543–1546.
[11] W. Liu, L. Wang, E. Wang, Y. Yang, D. Zeghlache, and D. Zhang, "Reinforcement learning-based cell selection in sparse mobile crowdsensing," *Comput. Netw.*, vol. 161, pp. 102–114, Oct. 2019.
[12] W. Liu, Y. Yang, E. Wang, L. Wang, D. Zeghlache, and D. Zhang, "Multi-dimensional urban sensing in sparse mobile crowdsensing," *IEEE Access*, vol. 7, pp. 82066–82079, 2019.
[13] L. Wang *et al.*, "CCS-TA: Quality-guaranteed online task allocation in compressive crowdsensing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. (UbiComp)*, 2015, pp. 683–694.
[14] T. Liu, Y. Zhu, Y. Yang, and F. Ye, "Incentive design for air pollution monitoring based on compressive crowdsensing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

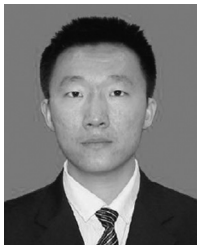
[15] S. He and K. G. Shin, "Steering crowdsourced signal map construction via Bayesian compressive sensing," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Honolulu, HI, USA, Apr. 2018, pp. 1016–1024.
[16] D. Yang, G. Xue, X. Fang, and J. Tang, "Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1732–1744, Jun. 2016.
[17] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang, "An efficient prediction-based user recruitment for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 17, no. 1, pp. 16–28, Jan. 2018.
[18] J. Wang, L. Wang, Y. Wang, D. Zhang, and L. Kong, "Task allocation in mobile crowd sensing: State-of-the-art and future opportunities," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3747–3757, Oct. 2018.
[19] H. Chen, B. Guo, Z. Yu, and Q. Han, "Crowdtracking: Real-time vehicle tracking through mobile crowdsensing," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7570–7583, Oct. 2019.
[20] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: An end-to-end participatory urban noise mapping system," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Stockholm, Sweden, 2010, pp. 105–116.
[21] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2289–2302, Nov. 2013.
[22] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2015, pp. 2254–2262.
[23] L. Pu, X. Chen, J. Xu, and X. Fu, "Crowd foraging: A QoS-oriented self-organized mobile crowdsourcing framework over opportunistic networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 4, pp. 848–862, Apr. 2017.
[24] M. Xiao, J. Wu, H. Huang, L. Huang, and C. Hu, "Deadline-sensitive user recruitment for probabilistically collaborative mobile crowdsensing," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2016, pp. 721–722.
[25] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-Air: When urban air quality inference meets big data," in *Proc. ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Chicago, IL, USA, 2013, pp. 1436–1444.
[26] S. Chang, H. Zhu, W. Zhang, L. Lu, and Y. Zhu, "Pure: Blind regression modeling for low quality data with participatory sensing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 4, pp. 1199–1211, Apr. 2016.
[27] T. Luo, J. Huang, S. S. Kanhere, J. Zhang, and S. K. Das, "Improving IoT data quality in mobile crowd sensing: A cross validation approach," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5651–5664, Jun. 2019.
[28] S. Chang, C. Li, H. Zhu, and H. Chen, "Adaptive and blind regression for mobile crowd sensing," *IEEE Trans. Mobile Comput.*, to be published.
[29] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and Internet traffic matrices," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 662–676, Jun. 2012.
[30] S. He and K. G. Shin, "Spatio-temporal adaptive pricing for balancing mobility-on-demand networks," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 10, no. 4, p. 39, 2019.
[31] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "SensorScope: Application-specific sensor network for environmental monitoring," *ACM Trans. Sensor Netw.*, vol. 6, no. 2, pp. 1–32, 2010.
[32] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Upper Saddle River, NJ, USA: Pearson Educ., 2016.
[33] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[34] S. James, G. Richard, C. Jon, H. Pan, D. Christophe, and C. Augustin. *CRAWDAD Dataset Cambridge/Haggle*. Accessed: Apr. 25, 2019. [Online]. Available: https://crawdad.org/cambridge/haggle/20090529, doi: 10.15783/C70011.
[35] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in *Proc. 18th Int. Conf. World Wide Web*, 2009, pp. 791–800.

Wenbin Liu received the B.S. degree in physics from Jilin University, Changchun, China, in 2012, where he is currently pursuing the Ph.D. degree with the College of Computer Science and Technology.

He was also a visiting Ph.D. student with the Wireless Networks and Multimedia Services Department, Telecom SudParis/Institut Mines-Telecom, Évry, France. His research interests include mobile crowdsensing and ubiquitous computing.

Yongjian Yang received the B.E. degree in automatization from the Jilin University of Technology, Changchun, China, in 1983, the M.E. degree in computer communication from the Beijing University of Post and Telecommunications, Beijing, China, in 1991, and the Ph.D. degree in software and theory of computer from Jilin University, Changchun, in 2005.

He is currently a Professor and a Ph.D. Supervisor with Jilin University, the Director of Key Lab under the Ministry of Information Industry, and the Standing Director of the Communication Academy. His research interests include the network intelligence management, wireless mobile communication and services, and wireless mobile communication.



En Wang received the B.E. degree in software engineering and the M.E. and Ph.D. degrees in computer science and technology from Jilin University, Changchun, China, in 2011, 2013, and 2016, respectively.

He is currently an Associate Professor with the Department of Computer Science and Technology, Jilin University. His current research focuses on the efficient utilization of network resources, scheduling and drop strategy in terms of buffer-management, energy-efficient communication between human-carried devices, and mobile crowdsensing.

Jie Wu (Fellow, IEEE) received the Ph.D. degree in computer engineering from Florida Atlantic University, Boca Raton, FL, USA, in 1989.

He was a Program Director of the National Science Foundation and was a Distinguished Professor with Florida Atlantic University, Boca Raton, FL, USA. He is the Director of the Center for Networked Computing and the Laura H. Carnell Professor with Temple University, Philadelphia, PA, USA. He also serves as the Director of International Affairs, College of Science and Technology. He served as the Chair of Department of Computer and Information Sciences from summer 2009 to summer 2016 and an Associate Vice Provost for International Affairs from fall 2015 to summer 2017. He regularly publishes in scholarly journals, conference proceedings, and books. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications.

Dr. Wu was a recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award. He is the Chair for the IEEE Technical Committee on Distributed Processing. He was the General Co-Chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016, and IEEE CNS 2016, as well as the Program Co-Chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was serves on several editorial boards, including the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS ON SERVICE COMPUTING, the *Journal of Parallel and Distributed Computing*, and the *Journal of Computer Science and Technology*. He was an IEEE Computer Society Distinguished Visitor and ACM Distinguished Speaker. He was a CCF Distinguished Speaker.