IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON TOWARDS SERVICE-CENTRIC INTERNET OF THINGS (IOT): FROM MODELING TO PRACTICE

Received June 27, 2018, accepted August 7, 2018, date of publication August 27, 2018, date of current version September 21, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2866641

Dynamic Mobile Crowdsourcing Selection for Electricity Load Forecasting

LIANYONG QI[1], WANCHUN DOU[2,3], WENPING WANG[2,3], GUANGSHUN LI[1], HAIRONG YU[1], AND SHAOHUA WAN[4]

[1]School of Information Science and Engineering, Qufu Normal University, Jining 276826, China
[2]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[3]Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China
[4]School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China

Corresponding author: Shaohua Wan (shaohua.wan@ieee.org)

This work was supported in part by the National Science Foundation of China under Grant 61872219 and Grant 61672276, in part by the National Key Research and Development Program of China under Grant 2017YFB1400600, and in part by the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University, Natural Science Foundation of Shandong Province under Grant ZR2018PF007.

ABSTRACT The continuous growth of mobile devices in recent years has created a variety of opportunities for people to utilize the crowdsourcing technique to execute various intelligent computing and processing tasks, e.g., electricity load forecasting. However, there are few research in the field of real-time and accurate forecasting of electricity load in a dynamic environment, and this leads to an unsatisfactory result when applying the forecasting method to the real environment. In view of this challenge, in this paper, we propose a dynamic mobile crowdsourcing selection method for electricity load forecasting considering the dynamic arrivals of both crowdsourcing tasks and candidate workers to help the crowdsourcing platform find the ideal workers for the crowdsourcing tasks. Concretely, in our method, a system model is firstly established to quantify the ability of candidate workers in executing the crowdsourcing task when both workers and tasks arrive at or leave the platform dynamically; afterwards, a dynamic worker selection method is proposed based on the ability threshold of workers and the task priority. Finally, through a set of simulated experiments, we validate the feasibility of our proposal in terms of effectiveness and efficiency when making accurate electricity load forecasting through mobile crowdsourcing technique.

INDEX TERMS Mobile crowdsourcing, electricity load forecasting, dynamic selection, threshold, priority.

I. INTRODUCTION

In recent years, people have witnessed the continuous growth of various mobile devices (e.g., mobile phones, tablet computers and smartwatches) in daily life and business applications [1]–[5]. Generally, mobile devices are universally embedded with numerous sensors, such as GPS, magnetometer and accelerometer. Therefore, data collected from these sensors can be employed collectively to satisfy the diverse demands of users [6], [7], which give birth to the mobile crowdsourcing technique. Typically, mobile crowdsourcing mainly consists of three participants, i.e., task requesters, candidate workers and a platform located in the cloud center. Thus, when a task requester gets trouble in collecting necessary information, he/she can generate a crowdsourcing task describing his/her problems and needs and then publish it to the platform. Afterwards, the platform receives the crowdsourcing task and begins to select a group of appropriate workers who can provide satisfying results to the task requester.

Mobile crowdsourcing is a promising resource sharing approach that leverages the decentralized mobile resources to accomplish the tasks that are hard to execute through the conventional ways, such as map services and translation services. As a complex activity that requires multi-party and real time data collaborations, electricity load forecasting based on mobile crowdsourcing has recently gained ever-increasing attention. Electricity load forecasting is to predict the electricity consumption in a future period by analyzing the current electricity load, which has great significance in the production, transmission and allocation of electricity

energy [7]. As an electricity company cannot monitor all the needed information about the electricity consumption condition of users, mobile crowdsourcing becomes a promising way to collect the required data for various decisions of the electricity company.

Generally, electricity load forecasting based on mobile crowdsourcing consists of the following three steps. (1) Task division and advertisement: the electricity company divides its needed data into several parts and advertises the data collection tasks to the candidate workers who are willing to provide data collection services through mobile devices. (2) Data collection: a set of chosen workers collect the needed data or information via mobile devices and report the collected data to the load prediction unit of the electricity company. (3) Consumption prediction: according to the collected data, the electricity company predicts the future electricity consumption through statistic or sampling techniques. This way, mobile crowdsourcing technique can help the electricity company to realize real time and cost-effective electricity consumption monitoring and future load prediction, which alleviates the heavy burden on the future electricity production decisions of the electricity company.

Therefore, selecting a set of appropriate workers from massive candidates is one of the most important preconditions of the success of subsequent electricity data collection and electricity consumption prediction. Many researchers have investigated this worker selection problem in the mobile crowdsourcing environment. However, existing researches often assume that the mobile crowdsourcing scenarios are static (i.e., electricity crowdsourcing tasks and candidate workers are fixed and static), while seldom consider the dynamic crowdsourcing scenarios where tasks and workers can arrive at or leave the crowdsourcing platform at will. For example, in Fig.1, a task or a worker can join or leave the platform dynamically. In this situation, the crowdsourcing problem becomes more complex as the dynamic arrivals of both tasks and workers should be taken into consideration for the purpose of finding the optimal crowdsourcing solution.

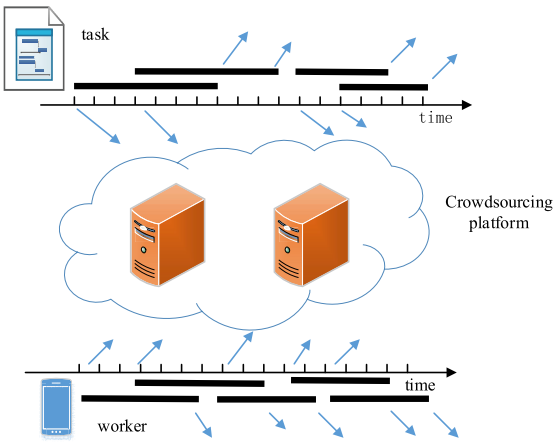


FIGURE 1. Dynamic crowdsourcing arrival model for task and worker. The task and worker join the mobile crowdsourcing system dynamically and stay in the system for a limited time.

In view of this challenge, we introduce time factor into the crowdsourcing problem and further propose a dynamic mobile crowdsourcing selection method based on time-aware ability model of workers. In summary, our contributions are three-fold.

(1) We propose an ability evaluation model to quantify the correlation between tasks and workers. The model consists of four submodels: task requester model, worker ability model, worker ability adjustment model and dynamic arrival model.

(2) We put forward a dynamic mobile crowdsourcing selection method for electricity load forecasting. The method selects the optimal workers for tasks based on the time-aware ability threshold of workers and task priority.

(3) We conduct a set of simulated experiments to validate the feasibility of our method. Experiment results show that our proposal outperforms other methods in terms of average ability value of recruited workers and task completion rate.

The remainder of this paper is organized as follows. Section II presents the system model of our dynamic crowdsourcing selection method. Problem formulation and algorithm are introduced in Section III. Experiment evaluation is made in Section IV. Related work is given in Section V and finally, in Section VI, we conclude this paper and provide some insights for the future work.

II. SYSTEM MODEL AND PROBLEM STATEMENT

In this section, we present the detailed system model and definitions for the electricity load forecasting based on dynamic mobile crowdsourcing (MC). To simplify the discussions, the symbols to be used in this paper are summarized in Table 1. Our system model mainly consists of four submodels: task requester model, worker ability model, worker ability adjustment model, and dynamic arrival model.

A. TASK REQUESTER MODEL

A task requester is a user who needs mobile crowdsourcing services from the system. We denote the set of requesters by $R = \{r_1, r_2, \ldots, r_l$. Each requester r_i can submit a series of tasks to the system and $RT_i = \{t_1, t_2, \ldots, t_{|r_i|}$ represents the set of tasks that requester r_i published to the platform. Thus the set of tasks in the platform can be defined as follows:

$$T = \bigcup_{i=1}^{l} RT_i = \{t_1, t_2, \ldots t_m\}, \quad m = \sum_{i=1}^{l} |r_i| \tag{1}$$

For each task t_i, we utilize a set of attributes to describe it, i.e., $\rho(t_i) = \{Ti_i, De_i, Ca_i, Kw_i, Am_i$. Here, Ti_i represents the task title offered by requesters; De_i represents the text description of the task (for example, a task can be described as "Please help me take three photos around the park"); the category of task is denoted by Ca_i, which indicates the kind that a task belongs to (e.g., information analysis or photo collection); Kw_i is the skills or the features that a task needs (for example, Kw_i can be a specific location in the

TABLE 1. Key terms and descriptions.

Terms	Description
R	The set of MC requesters, $R = \{r_1, r_2, \ldots, r_l\}$
RT_i	The set of tasks for i-th requester, $RT_i = \{t_1, t_2, \ldots, t_{\|r_i\|}\}$
T	The set of all tasks, $T = \{t_1, t_2, \ldots, t_m\}$
W	The set of MC workers, $W = \{w_1, w_2, \ldots, w_n\}$
F_j^{tc}	The j-th worker's familiarity to the keywords in the tc-th category
F_{jk}^{tc}	The j-th worker's familiarity to the k-th keywords in the tc-th category
S^{tc}	The keyword space of tc-th category
X_i^{tc}	The vector of i-th worker's familiarity to the keyword in tc-th category
$A_{j,i}$	The ability of j-th worker for the i-th task
WT	The matrix for all workers' abilities to all tasks
b_i	The begin time of task t_i
e_i	The end time of task t_i
l_i	The length of time to conduct the task t_i
a_j	The begin time of worker w_j
d_j	The leaving time of worker w_j
h_i	The threshold value of task t_i
p_i	The priority value of task t_i

location-finding task); the number of results that a task requires is indicated by Am_i.

B. WORKER ABILITY MODEL

A task is often executed by a worker; therefore, it's important to evaluate the correlation between the worker and the task. We define this correlation as worker ability for the task, which can be calculated by the attributes of the task and the worker's familiar degree towards these attributes.

Worker ability for a task is defined as the worker's familiarity with the task's category and keywords. For instance, if a requester submits a task "take pictures around a scenic spot", then the platform will look for workers who are familiar with the category "picture collection" by keywords such as location, camera and scenic spot. Next, we introduce several definitions about keywords of tasks.

Definition 1 (Keyword Space): Keyword space in the mobile crowdsourcing system indicates the set of all keywords from every category and is denoted by S.

Suppose that the mobile crowdsourcing system has TC categories, and S^{tc} indicates the keyword subset that the tc-th category owns. We use K to denote the maximum number of the tc-th category.

Definition 2 (Worker's Familiarity With Keyword Space): We use a $K \times TC$ matrix to define the worker's familiarity with keyword space. The vector $F_j^{tc} = [F_{j1}^{tc}, F_{j2}^{tc}, \ldots, F_{jK}^{tc}]^T$ refers to the j-th worker's familiarity with all keywords in the tc-th category, and the row vector $F_{jk} = [F_{jk}^1, F_{jk}^2, \ldots, F_{jk}^{tc}]$ represents the j-th worker's familiarity with the k-th keyword in all categories. And F_{jk}^{tc} represents the j-th worker's familiarity with the k-th keyword in the tc-th category.

Definition 3 (Keyword in Task): We use H_i to indicate the set of keywords in task t_i. It's obvious that $H_i \subseteq S^{tc}$. And we use ω_k to express the importance of the k-th keyword in defining the task. Hence the vector $X_i^{tc} = [x_1^{tc}, x_2^{tc}, \ldots, x_K^{tc}]$ can be used to describe the importance of the keywords in tc-th category for the task. The value of x_k^{tc} is defined as

$$x_k^{tc} = \begin{cases} \omega_k, & \text{k}\epsilon H_i, \\ 0, & \text{others.} \end{cases} \tag{2}$$

Without loss of generality, we set the sum of ω_k for a task as 1.0, so we have

$$\forall t_i, \sum_{k=1}^{K} x_k^{tc} = 1.0 \tag{3}$$

Definition 4 (Worker Ability): Based on the above definitions, a worker's ability can be defined as his/her familiarity with the task's category and keywords. We use $A_{j,i}$ to denote the ability of the j-th worker in executing the i-th task, hence $A_{j,i}$ can be given as follows:

$$A_{j,i} = X_i^{tc} F_j^{tc} = \sum_{k=1}^{K} x_k^{tc} f_{jk}^{tc} \tag{4}$$

The value of $A_{j,i}$ indicates the ability of worker w_j in executing task t_i in the category tc, and this formula can be applied to different kinds of mobile crowdsourcing systems. For example, if the X_i^{tc} of a task in tc-th category is [0.1, 0.3, 0, 0.4, 0, 0.2] and a worker's familiarity with the keywords in tc-th category is [0, 4, 2, 5, 0, 2], then the ability of the worker in executing the task can be calculated as: $0.1*0 + 0.3*4 + 0*2 + 0.4*5 + 0*0 + 0.2*2 = 3.6$.

In accordance with the above definitions, all the workers' abilities to all tasks can be indicated by a matrix as follows:

$$WT = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & & A_{2m} \\ \vdots & & \ddots & \vdots \\ A_{n-11} & A_{n-12} & \cdots & A_{n-1m} \\ A_{n1} & A_{n2} & & A_{nm} \end{pmatrix}$$

In the following process of worker selection, the matrix WT can be used as the basis to estimate the correlation between workers and tasks.

C. WORKER ABILITY ADJUSTMENT MODEL

In some specific mobile crowdsourcing tasks, some factors should be considered to adjust the model of worker ability. For instance, a requester creates a task querying the real-time traffic condition, the time factor turns out to be an indispensable element for the quality evaluation of the service. When the requester wants to know the pollution situation in some place, the distance turns out to be a vital element. We consider the following elements for adjusting the worker ability model.

1) DISTANCE

For some location-aware mobile crowdsourcing tasks, the requesters often have higher demands on the distance [9]. We denote the value of a worker's adjustment ability by $\hat{A}_{j,i}$, then:

$$\hat{A}_{j,i} = A_{j,i}\varphi\left(d_j\right) \tag{5}$$

Here, d_j indicates the distance between the worker and the location that a task demands. The function $\varphi(d)$ satisfies the following three properties:

1) $\varphi(0) = D_0$. D_0 indicates the initial value;
2) If $d_1 < d_2$, then $\varphi(d_1) > \varphi(d_2)$;
3) If $d_1 < D \leq d_2$, then $\varphi(d_1) > \varphi(D) \geq \varphi(d_2)$. D is the maximal boundary of the distance.

2) TIME

Some tasks tend to recruit the workers who can return results as early as possible. Therefore, a truncated exponential function [10]–[13] can be used here to define the time function as follows:

$$T_{i,j}(x) = \begin{cases} \alpha e^{-h(x)}, & x \leq T_0 \\ 0, & x > T_0 \end{cases} \tag{6}$$

Here, α is the adjusting coefficient, $h(x)$ is the time function.

3) REPUTATION

Some mobile crowdsourcing systems adopt the reputation model to punish those workers who provide poor-quality results. The reputation model mainly depends on the times that a worker provides services and the quality of returned results. It can be defined as follows:

$$RP_j = \mathrm{Q}_j\left(num_j, quality_j\right) \tag{7}$$

The better the quality ($quality_j$) is or the larger the number (num_j) is, the higher the value of RP will be. Based on the above discussions, a worker's adjusting ability can be defined as follows:

$$\hat{A}_{j,i} = A_{j,i} \times \varphi\left(d_j\right) \times T_{i,j}(x) \times RP_j \tag{8}$$

D. DYNAMIC ARRIVAL MODEL

We divide time into K slots and denote the time series as $S = \{s_1, s_2, \ldots, s_K$ where s represents the length of each time slot. We use M_k to denote the number of tasks arriving at time slot s_k and use t_i^k to represent the i-th task arriving at s_k. And N_k represents the number of workers arriving at s_k, the j-th worker arriving at s_k is denoted by w_j^k. So we have $m = \sum_{k=1}^{K} M_k$, $n = \sum_{k=1}^{K} N_k$.

When task t_i joins the crowdsourcing system, the task reports its time attribute $\sigma(t_i) = b_i, e_i, l_i$ voluntarily. Here, b_i is the beginning time of task t_i, e_i is the ending time, l_i denotes the length of time to execute the task. The available time range for task t_i is $[b_i, e_i]$; in other words, task t_i can only be executed by workers during this time range, thus we have $l_i \leq e_i - b_i$.

When worker w_j joins the crowdsourcing system, w_j reports its time attribute $\theta(w_j) = a_j, d_j$ where a_j is the beginning time of worker w_j and d_j is the leaving time of w_j. The time range for worker w_j to stay in the system is $[a_j, d_j]$. If worker w_j is selected by task t_i, then we have $a_j \leq e_i - l_i$, $d_j - a_j \geq l_i$ and $a_j \geq b_i$, which means that the beginning time of w_j is less than the difference between leaving time and execution time of task t_i, and the available time for worker w_j is larger than the execution time of task t_i.

III. DYNAMIC MOBILE CROWDSOURCING SELECTION METHOD

In this section, we formulate the electricity load forecasting problem based on dynamic mobile crowdsourcing and propose our algorithms. Concretely, in the mobile crowdsourcing system based on ability evaluation, we need a method to maximize the sum of ability values (denoted by $A^{\#}$) for selected workers under the time constraint of workers and tasks while recruiting enough workers for tasks. Therefore, the crowdsourcing selection problem can be formulated as follows.

$$\max A^{\#} = \sum_{i=1}^{m} \sum_{j=1}^{Am_i} A_{j,i} \tag{9}$$

$$s.t.\ \forall w_j \epsilon TG_i, a_j \leq e_i - l_i, a_j \geq b_i, d_j - a_j \geq l_i \tag{10}$$

$$|TG_i| = Am_i \tag{11}$$

Formula (9) denotes the maximal object function; formula (10) requires the method to satisfy the time constraint; formula (11) demands the method to select enough workers. Besides, the goal of the method should satisfy several properties such as computational efficiency, individual rationality, method validity, member equality and so on. These properties are defined as follows.

Definition 5 (Computational Efficiency): A method has computational efficiency if and only if the method can be finished in linear time.

Definition 6 (Individual Rationality): A method satisfies individual rationality iff each worker has a nonnegative ability value, i.e., $A_{j,i} \geq 0, \forall t_i \in T, w_j \in W$.

Definition 7 (Method Validity): To quantify the performance of our method, we implement an offline method in which all the time attributes of tasks and workers are given in advance. A method can satisfy the method validity iff the ratio of results between our method and the optimal offline method is O(1).

Definition 8 (Member Equality): The platform should guarantee that each task selects the candidate workers equally. Furthermore, the evaluation criterion for a worker is his/her ability value in executing the task.

If a method satisfies the abovementioned four properties (Def.5∼Def.8), it can be regarded that the method is effective and reasonable, and hence can be applied to the real crowdsourcing environment.

A. OPTIMAL SELECTION METHOD IN OFFLINE SCENARIO

Next, we first introduce the optimal worker selection method in the offline mobile crowdsourcing situation when the platform gets all the time attributes of tasks and workers in advance. Although it is impossible in reality, the result of offline crowdsourcing is still of positive significance as it be taken as the benchmark to evaluate the effectiveness of our method. The execution process of offline crowdsourcing is as follows.

In the algorithm, line 1-8 generate a queue for the allocated tasks of every worker under the time constraint. We use *workerList* to store the allocated workers for each task and set *workerList* to null at the beginning. We traverse all the workers and tasks; if time attributes of worker w_j and task t_i satisfy the constraint $a_j \leq e_i - l_i, a_j \geq b_i, d_j - a_j \geq l_i$, then we put t_i into the task list $taskList_j$ for worker w_j.

In the second part of algorithm (i.e., line 9-24), a worker selects a task with the highest ability value; then the worker is allocated to the task and put into list *taskList*. The corresponding task accepts the worker and updates its number of workers to recruit. We use *SelectedWorker* to store the workers who have been allocated to a task and set it null at first. Then we traverse all the workers. If worker w_j is in *SelectedWorker*, then we pass it. Otherwise, we pick the task t_i with the highest ability from $taskList_j$; if the size of $list_i$ is less than Am_i, then w_j will be put into $list_i$ and *SelectedWorker*; otherwise, we put $list_i$ into *workerList*. After the traversing process, if there still exist tasks calling for more workers, then the second part should be executed again until all tasks recruit enough workers or there are no available workers. At last, *workerList* is returned as the final crowdsourcing results.

The offline crowdsourcing scenario, where the time attributes of tasks and workers are known already, is the most ideal one. While in the real mobile crowdsourcing system, the time attributes of tasks and workers are often difficult to obtain in advance, which make the crowdsourcing problem more complex. In view of this challenge, next, we introduce a dynamic mobile crowdsourcing selection method based on the ability evaluation model of workers.

B. THRESHOLD CALCULATION IN DYNAMIC CROWDSOURCING SYSTEM

In the dynamic crowdsourcing system, after a task arrives at the system, the platform would recruit enough workers with high ability to execute the task within the existing time frame of the task in the system (here, we do not consider the influence of network environment among requester, platform and workers, such as network delay or disruption [14]–[16]). When a new worker joins the system, the system needs to decide whether to recruit the worker for the task. To achieve this goal, we set a threshold (denoted by h_i) for the selection criterion corresponding to task t_i. Only when the ability value of worker w_j in executing task t_i, i.e., $A_{j,i}$ is larger than threshold h_i, the task t_i will consider to select the worker.

Algorithm 1 Crowdsourcing Selection in Offline Scenarios

Input: set of tasks T, set of workers W
Output: queue for allocated workers $workerList=\{list_1, list_2, \ldots, list_m\}$
1: $workerList \leftarrow \emptyset$, $taskList \leftarrow \emptyset$;
2: **For** $j = 1$ to n **do**
3: **For** $i = 1$ to m **do**
4: **If** $a_j \leq e_i - l_i, a_j \geq b_i, d_j - a_j \geq l_i$ **then**
5: $taskList_j \leftarrow taskList_j \cup t_i$;
6: **End if**
7: **End for**
8: **End for**
9: $SelectedWorker \leftarrow \emptyset$;
10: **While** $\sum_{i=1}^{m} |list_i| < \sum_{i=1}^{m} Am_i$ **do**
11: **For** $j = 1$ to m **do**
12: **If** w_j in *SelectedWorker* **then**
13: **continue**;
14: **Else**
15: pick worker with highest value of ability in $taskList_j$,denoted as t_i;
16: **If** $|list_i| < Am_i$ **then**
17: $list_i \leftarrow list_i \cup w_j$;
18: $SelectedWorker \leftarrow SelectedWorker \cup w_j$;
19: **Else**
20: $workerList \leftarrow list_i$;
21: **End if**
22: **End if**
23: **End for**
24: **End while**
25: **Return** *workerList*

When a task t_i joins the crowdsourcing system, its initial threshold h_i is set as the average ability value of all the available workers arriving before executing t_i. Based on the definition, the initial threshold h_i is calculated as in (12).

$$h_i = \frac{\sum_{t=1}^{b_i} \sum_{j=1}^{N_k} A_{j,i}}{\sum_{t=1}^{b_i} \sum_{j=1}^{N_k} 1} \tag{12}$$

The threshold h_i can decide whether to recruit a worker for task t_i. However, the value of h_i is not fixed; this is because the probability that a task can recruit a worker decreases with time elapsing. Likewise, the closer a task is to the ending time, the lower the task's expectation on the ability value of workers will be. We denote the residual time of the task t_i as lt_i thus we can get $lt_i = e_i - t$, where t represents the current time. We update the threshold value at every time slot according to the formula in (13).

$$\bar{h}_i = \frac{h_i(lt_i - 1)}{lt_i} \tag{13}$$

When the task recruits a new worker successfully, the rate of task accomplishment increases. Correspondingly, the task's expectation on the ability value of the subsequent workers also increases; therefore, the threshold should be appropriately increased. With the above analyses, we utilize

the current threshold and the ability of the newly recruited workers to jointly generate the new threshold. The calculation formula for the new threshold is given by (14).

$$\bar{h}_i = \alpha * h_i + (1 - \alpha) * A_{j,i} \tag{14}$$

Here, $A_{j,i}$ represents the ability value of recruited workers and α is the adjustment coefficient. The threshold updating algorithm for task t_i during execution is shown as follows.

Algorithm 2 generates the set of threshold for task t_i at each time slot. We first get the initial threshold for task t_i by calculating the average ability value of the workers who arrived at the system before the time b_i. Then we calculate the threshold from time slot b_i to time slot e_i. If task t_i recruits a worker at time slot t, then the threshold becomes $H_{t-b_i+1} = [\alpha * H_{t-b_i} + (1-\alpha) * A_{j,i}] * \frac{(lt_i-1)}{lt_i}$, because H_{t-b_i+1} should be adjusted considering both the time factor and the ability of recruited workers. Otherwise, $H_{t-b_i+1} = \frac{H_{t-b_i}(lt_i-1)}{lt_i}$, which is only adjusted based on the time factor.

Algorithm 2 Threshold Updating for Task t_i

Input: task t_i, the set of workers W
Output: the set of threshold H for the task t_i
1: $\{H, sum, num\} \leftarrow \{\emptyset, 0, 0\}$;
2: **For** $t = 1$ to b_i **do**
3: **For** $j = 1$ to N_k **do**
4: $sum \leftarrow sum + A_{j,i}$, $num \leftarrow num + 1$;
5: **End for**
6: **End for**
7: $H_0 = sum/num$;
8: **For** $t = b_i$ to e_i **do**
9: **If** t_i recruits worker w_j **then**
10: $H_{t-b_i+1} = [\alpha * H_{t-b_i} + (1-\alpha) * A_{j,i}] * \frac{(lt_i-1)}{lt_i}$;
11: **Else**
12: $H_{t-b_i+1} = \frac{H_{t-b_i}(lt_i-1)}{lt_i}$;
13: **End if**
14: **End of**
15: **Return** H;

C. DYNAMIC CROWDSOURCING SELECTION

In the dynamic mobile crowdsourcing system, it is often difficult to predict the next worker's arrival time and his/her ability value. Therefore, we adopt the ability value threshold to evaluate the future workers. Concretely, if the ability value of a future worker is greater than the current threshold, then the future worker is recruited; otherwise, the future worker is rejected.

For the arrived task at time t, we first get the set of existing tasks in the system. Here, the existing tasks mean the tasks whose time section includes the time t, i.e., $b_i \leq t$ and $t \geq e_i$. The set of crowdsourcing tasks at the time t is denoted by TA_t. Then the generating algorithm for TA_t can be specified by the pseudo code in Algorithm 3. We traverse the set of tasks and put task t_i into TA_t if t_i satisfies the time constraint $b_i \leq t$ and $t \geq e_i$.

Algorithm 3 Generating TA_t at time t

Input: set of tasks T
Output: set of tasks TA_t at time t
1: $TA_t \leftarrow \emptyset$;
2: **For** $i = 1$ to |T| **do**
3: **If** $b_i \leq t$ and $t \geq e_i$ **then**
4: $TA_t \leftarrow TA_t \cup t_i$;
5: **End if**
6: **End for**
7: **Return** TA_t;

Based on the above settings, we first consider the case where there is only one task in a dynamic crowdsourcing system. In this situation, all the workers can only be selected by the unique task, and the system only needs to select the workers whose ability value is greater than the current threshold. However, it's not realistic that there is only one task; namely, there are often multiple tasks that need to recruit workers at the same time. Let's consider the example in Fig.2 where a worker joins the system at time t. In Fig.2, there are three tasks (ID: 1, 2, 3) which need to recruit workers at the same time; in this situation, how to assign the arrived workers to the appropriate task becomes the key to solve the dynamic mobile crowdsourcing problem.

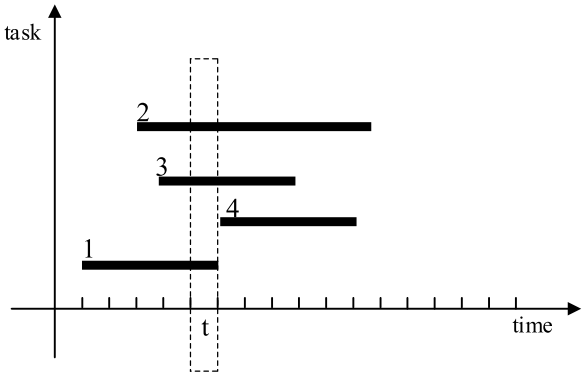


FIGURE 2. Illustration for crowdsourcing system when a worker joins. There are often several tasks in the system waiting to recruit more workers.

We use ST_j to represent the set of tasks for which the ability of worker w_j is greater than the task's threshold. A greedy algorithm to solve the dynamic crowdsourcing problem is to select a task ST_j^i with the highest ability value from ST_j for each chosen task; and subsequently, worker w_j is allocated to this task. However, this greedy strategy may lead to the "hunger" of some tasks; and ultimately, there will be insufficient workers that can be recruited by these tasks. If the ability of every worker in the recruitment process is not always the maximum for the task, then the task will not be able to recruit workers to execute it. Therefore, we adopt the priority or weight mechanism [17]–[20]. Concretely, we set a priority $p_i = \varphi(ln_i, lt_i)$ for each task t_i, where ln_i represents the number of workers who can be recruited by task t_i, and lt_i represents the remaining time for task t_i. The higher the

priority is, the larger likelihood a worker would be assigned to the task. Obviously, the priority is proportional to the value of ln_i and inversely proportional to the value of lt_i. Here, p_i represents the number of workers who can be recruited in a unit time. We take $Q_i = p_i * A_{j,i}$ as the basis for selecting appropriate participants, so we have

$$Q_i = p_i * A_{j,i} = \varphi(ln_i, lt_i) * A_{j,i} \tag{15}$$

Where $A_{j,i}$ is the ability value of the j-th worker in executing the i-th task. According to the above definition, the dynamic mobile crowdsourcing selection algorithm based on ability value assessment of workers is given as below.

Algorithm 4 specifies the process of crowdsourcing selection in a dynamic scenario. We conduct the selection process at each time slot orderly. At each time slot t, the set of tasks TA_t and the threshold for tasks in TA_t are calculated first by Algorithm 3 and Algorithm 2, respectively. For each arrived worker w_j at time t, we generate the set of tasks ST_j for worker w_j to find the tasks whose ability of worker w_j is greater than the task's threshold, while the task in ST_j must satisfy the time constraint, i.e., $a_j \le e_i - l_i, a_j \ge b_i, d_j - a_j \ge l_i$. Thus we select the task with maximum Q_i from ST_j and denote it by t_i. Afterwards, worker w_j is allocated to task t_i and put into $list_i$.

Algorithm 4 Dynamic Crowdsourcing Selection

Input: set of tasks T, set of workers W
Output: allocation queue $workerList$=$\{list_1, list_2, \ldots, list_m\}$

```
workerList ← ∅;
For t = 1 to K do
  calculate TA_t by Algorithm 3;
  update threshold of TA_t by Algorithm 2;
  If N_k > 0 then
    For j = 1 to N_k do
      ST_j ← ∅;
      For i = 1 to |TA_t| do
        If a_j ≤ e_i − l_i, a_j ≥ b_i, d_j − a_j ≥ l_i and h_i ≤ A_{j,i} then
          ST_j ← ST_j ∪ t_i;
          Q_i = φ(ln_i, lt_i) * A_{j,i};
        End if
      End for
      select task with maximum Q_i from ST_j, denoted by t_i;
      list_i ← list_i ∪ w_j;
    End for
  End if
End for
Return workerList;
```

IV. EXPERIMENTS AND EVALUATIONS

In this section, we conduct a set of simulated experiments to validate the feasibility of our proposal in terms of the capability of solving the dynamic mobile crowdsourcing problems.

A. EXPERIMENT SETTINGS

We investigated the experiment settings of 25 research studies and ascertained that most of their experiments are simulated. Therefore, in this paper, we conduct a set of simulated experiments to verify the efficiency of our proposed method. The default settings of parameters are listed in Table 2.

TABLE 2. Default settings of simulations.

Parameter name	Default value
Number of tasks	1000
Number of workers	40000
Amount of workers for a task	5
Number of time slots	25000
Time for task execution	[1, 10]
Time for worker to stay	[10, 100]

In order to evaluate the effectiveness of the dynamic crowdsourcing selection method, the average ability value and task completion rate of the recruited participants are used as evaluation criteria. The average ability value of the recruited participants is calculated as follows:

$$A_{average} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{Am_i} A_{j,i}}{\sum_{j=1}^{Am_i} 1} \tag{16}$$

Here, $A_{average}$ represents the average ability of the recruited workers. Larger value of $A_{average}$ often means better results that the recruited workers can provide as well as better performance. For those tasks which do not recruit enough workers, we set the un-recruited workers' abilities zero. For example, if a task needs to recruit 5 participants while only 3 participants with ability values of 5.3, 6.7 and 6.0 are actually recruited, then the average ability value of the recruited participants is $(5.3+6.7+6.0+0+0)/5 = 3.6$. Task completion rate is defined as the ratio of the number of tasks recruited to the total number of tasks. The tasks completion rate indicates the completion rate of the crowdsourcing method in selecting appropriate participants for tasks. Higher task completion rate implies better crowdsourcing effect that a crowdsourcing method can achieve.

We compare our proposal with Offline method (benchmark) and Greedy method (when a worker arrives at the system, it will always be assigned to the task with the highest ability value). Experiments are conducted by jdk 1.8.0 and eclipse I.D.E. All experiments are performed under Windows 10 operating system with Intel(R) Core(TM) i3-4150 3.5 GHz CPU and 8.0 GB RAM. To maintain the consistency of experiment results, each experiment is repeated 20 times.

B. EXPERIMENT RESULTS

We tested the average ability values and task completion ratio of our proposal and compare them with the other competitive methods, i.e., Offline method and Greedy method.

Concretely, the experiment parameters are set as follows: the number of workers is varied from 10000 to 40000, other parameters are set as in Table 2. Experiment results are presented in Fig.3-Fig.4.

As Fig. 3 shows, the average ability values of three methods all increase approximately linearly when the number of recruited workers rises. Furthermore, the average ability of Dynamic crowdsourcing method performs better than the Greedy method and approaches the benchmark method, i.e., Offline method. The reason is that our proposal introduces the time factor into the ability evaluation model of workers; and correspondingly, higher ''average ability'' of recruited workers are obtained finally.

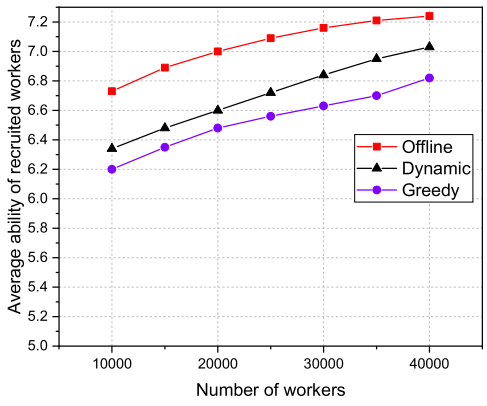


FIGURE 3. Average ability of recruited workers w.r.t. Number of workers. The ability of recruited workers increases as the number of workers increases.

Fig. 4 shows the relationships between the number of workers and the task completion rate of three crowdsourcing methods. As can be seen from the figure, the task completion rates of three methods all increase with the growth of the number of workers. Moreover, the tasks completion rate of our proposed dynamic crowdsourcing method outperforms that of the Greedy method and approaches that of the benchmark method (i.e., Offline method). The reason is that our method considers the dynamic arrivals of both tasks and workers.

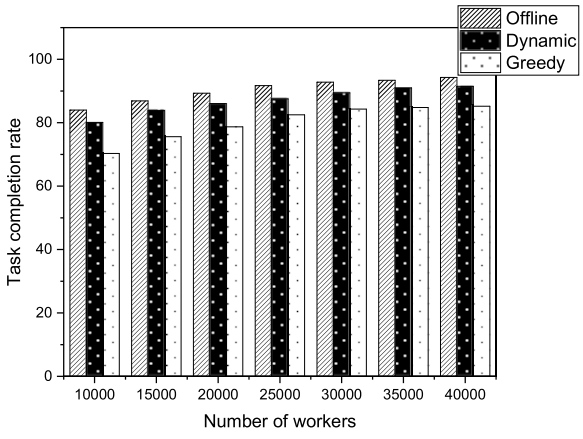


FIGURE 4. Task completion rate w.r.t. Number of workers. More workers can achieve higher task completion rate.

Time factor plays an important role in our suggested dynamic mobile crowdsourcing method. Next, for the three competitive methods, we investigate their respective relationship between the average ability of recruited workers and the number of time slots. Concretely, the experiment parameters are set as follows: the number of time slots is varied from 10000 to 25000, other parameters are set as in Table 2. Concrete experiment results are shown in Fig.5-Fig.6.

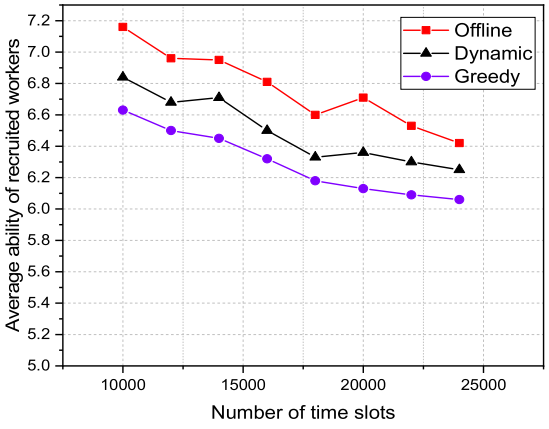


FIGURE 5. Average ability of recruited workers w.r.t. Number of time slots. As the time slots increase, the ability of recruited workers decreases.

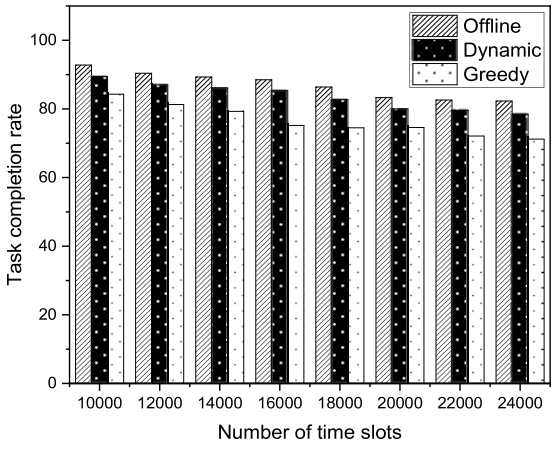


FIGURE 6. Task completion rate w.r.t. Number of time slots. The task completion rate tends to decline as the time slots become bigger.

As can be seen from Fig. 5, the average ability of recruited workers of our proposed Dynamic method decreases with the growth of the number of time slots, which renders the similar variation tendency with those of the Offline method and the Greedy method. The reason is that the increased time period often means more sparse distribution of tasks and participants as well as fewer participants that are recruited. Furthermore, our proposal performs better than the Greedy method and approaches the benchmark method in terms of the average ability of recruited workers. The reason is that our method

considers the time-aware arrivals of both tasks and workers, which suits the dynamic crowdsourcing scenarios more.

Fig. 6 shows the task completion rate of three crowdsourcing methods with respect to the number of time slots. As Fig.6 indicates, the task completion rates of three methods all decrease when the number of time slots rises, because the growth of time slots decreases the number of workers who can satisfy the task time constraints. Furthermore, our proposal outperforms the Greedy method and meanwhile approaches the Offline method in terms of the task completion rate, whose reason is the same as that of Fig.5 and hence not repeated here.

Fig.7 shows the influence of the number of tasks to the average ability of recruited workers in three crowdsourcing methods. As can be seen from the experiment figure, when the number of tasks increases, all the three methods achieve a low average ability of recruited workers; moreover, our proposed Dynamic selection method can get an approximate result to the optimal method (i.e., the Offline method) due to the dynamic property of our proposal.

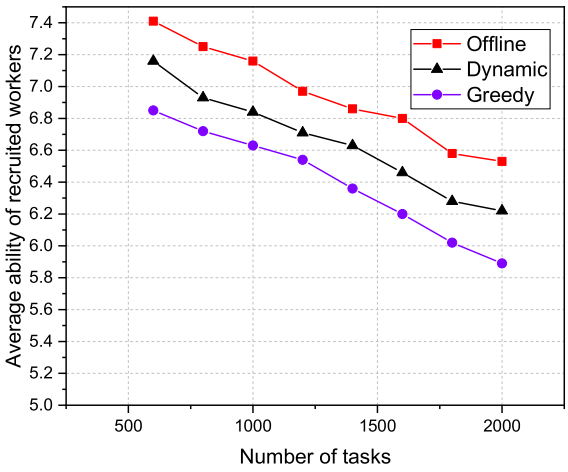


FIGURE 7. Average ability of recruited workers w.r.t Number of tasks. More tasks will bring lower average ability of recruited workers.

In Fig.8, we test the relationship between the task completion rate and the number of tasks. As can be seen from Fig.8, the Dynamic selection method can get a better task completion rate than the greedy policy. Besides, for the three methods, the task completion rate values approximately decrease when the number of tasks increases; this is because more tasks will lead to deficiency of the workers to be selected. At last, the Dynamic selection method can get an approximate result to the optimal Offline method.

C. SHORTCOMING ANALYSES

However, there are still several shortcomings in this paper, which still need further investigation.

(1) To achieve the optimal crowdsourcing solution, it is necessary to consider the candidate workers' historical data that are often sensitive [21]–[28]. Therefore, it is of necessity to improve our proposal by introducing some effective privacy-preservation strategies to guarantee the security and robustness performances [29]–[31].

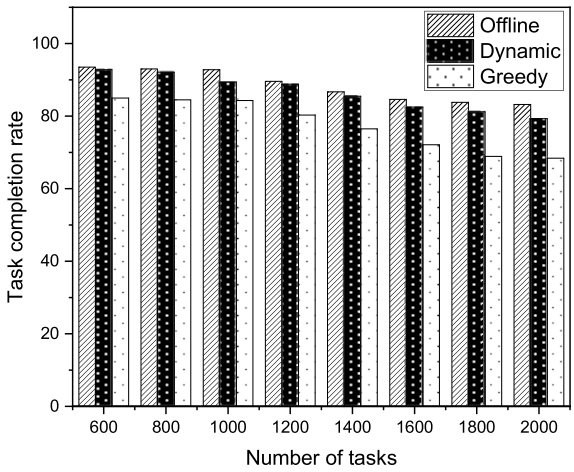


FIGURE 8. Task completion rate w.r.t Number of tasks. The bigger the number of tasks is, the lower the task completion rate will be.

(2) In this paper, we seek for the optimal mobile crowdsourcing solution mainly based on the unique evaluation criterion, i.e., the ability values of candidate workers. However, multiple criteria are often presented in the practical decision-making applications [32]–[35]. Therefore, it is of positive significance to extend our work by considering multiple evaluation criteria as well as their distinct data types (e.g., discrete [36], [37], binary [38], fuzzy [39]) simultaneously in the future research.

V. RELATED WORK

Recently, mobile crowdsourcing has been utilized to conduct the electricity load forecasting. Selecting proper workers for forecasting tasks is crucial for providing better services in the forecasting process. Many researchers have investigated this hot topic. In this section, we review the related work from the following three aspects.

A. CROWDSOURCING SELECTION BASED ON WORKERS' ABILITY

The abilities of workers are often regarded as the worker selection basis for mobile crowdsourcing applications. In [40], a self-organized mobile crowdsourcing tool called Crowdlet is introduced to recruit workers to satisfy the requester's real-time tasks. Crowdlet determines whether to select a worker based on the worker's provided service quality that is often calculated based on the worker's skills, arrival time and reward; however, reputation and device performance are not taken into consideration. Karaliopoulos *et al.* [41] present an online crowdsourcing framework to allocate workers to tasks by collecting the key information from workers and requesters, whose goal is to maximize the utility function defined in the framework. However, the utility function

only depends on network state, without considering other elements. In [42], a self-adaptive behavior-aware recruitment scheme is proposed to improve the process of recruitment. This scheme generates a reputation value for a worker by considering the worker's historical data. However, the reputation only depends on the worker's skill while neglects device capability and other elements. In [43], a crowd sensing system is designed to take the budget of workers and system into consideration. In this work, the reputation updating method utilizes a worker' willingness and provided information (e.g., the reliability of allocating a worker to a task) to generate the crowdsourcing result. This way, both the budget cost for a task and the high quality of mobile crowdsourcing results can be guaranteed finally. However, the crowdsourcing basis in this work is still not comprehensive.

B. DYNAMIC MOBILE CROWDSOURCING SELECTION

In [44], a regret minimization online learning method is proposed to generate the optimal price strategy and incentive method. In [45], two online mechanisms, i.e., OMZ and OMG are designed based on the online auction models to satisfy the dynamic arrival of workers; the goal was to select enough workers to maximize the service value before the deadline. In [46], for the dynamical crowdsourcing problem, the authors suggest that the workers can submit their personal configuration files to the system; thus the system can minimize the budget or cost while selecting enough workers for tasks. However, the auction methods in [44]–[46] often lead to low efficiency and hence cannot satisfy the quick response requirements from crowdsourcing platform or requesters. Besides, additional burden is stressed on the workers and requesters due to the auction process.

C. MOBILE CROWDSOURCING IN ELECTRICITY LOAD FORECASTING

Some researchers use mobile crowdsourcing to collect users' electricity consumption information. While few of them have explored the mobile crowdsourcing selection problem in the load forecasting. Wagy *et al.* [47] design hundreds of questions for the electricity users via mobile crowdsourcing. According to the feedback answers, a prediction model is built to evaluate the users' electricity consumption in the future. In [48], load forecasting is conducted based on the energy information collected from mobile devices. In this study, electricity users and electricity companies play the roles of workers and requesters, respectively. In [49], the users predict their electricity consumption and send the predicted results to the crowdsourcing platform or electricity companies so as to gain some remission in electricity price. However, the above methods mainly focus on how to apply the mobile crowdsourcing to conduct the electricity forecasting, while neglect to select proper workers for the forecasting tasks.

With the above analyses, we can conclude that existing researches fall short in forecasting the electricity load in the dynamic mobile crowdsourcing environment. In view of this challenge, a dynamic mobile crowdsourcing selection method for electricity load forecasting is proposed in this paper. Through considering the dynamic arrivals of both crowdsourcing tasks and candidate workers, our proposal achieves a good tradeoff between average ability value of recruited workers and task completion rate.

VI. CONCLUSION AND FUTURE WORK

Accurate prediction of future electricity consumption is of positive significance for the production, transmission and allocation of electricity energy. With the increasing popularity of mobile devices, mobile crowdsourcing has gradually become one of the most promising ways to achieve the above electricity prediction goal. However, existing researches often assume that the crowdsourcing scenarios are static, while seldom consider dynamic scenarios where the crowdsourcing tasks and candidate workers arrive at or leave the crowdsourcing platform at will. In view of this challenge, a dynamic mobile crowdsourcing selection method for electricity load forecasting is proposed in this paper, which considers the dynamic arrivals of both tasks and workers. Finally, through a set of simulated experiments, we validate the feasibility of our proposal in terms of average ability value of recruited workers and task completion rate.

In the future, we plan to apply our dynamic crowdsourcing selection method in real-world electricity load forecasting environment to further validate the feasibility of our proposal. Moreover, besides time, many other factors (e.g., location) also plays an important role in the worker selection decisions; Considering this, in the future, we will further refine our proposed mobile crowdsourcing method by introducing more factors, so as to pursue more reasonable and accurate crowdsourcing solutions.

REFERENCES

[1] R. H. Jhaveri, N. M. Patel, Y. Zhong, and A. K. Sangaiah, "Sensitivity analysis of an attack-pattern discovery based trusted routing scheme for mobile ad-hoc networks in industrial IoT," *IEEE ACCESS*, vol. 6, pp. 20085–20103, 2018, doi: 10.1109/ACCESS.2018.2822945.

[2] Z. Cai, H. Yan, P. Li, Z. Huang, and C. Gao, "Towards secure and flexible EHR sharing in mobile health cloud under static assumptions," *Cluster Comput.*, vol. 20, no. 3, pp. 2415–2422, 2017.

[3] L. Yang, Z. Han, Z. Huang, and J. Ma, "A remotely keyed file encryption scheme under mobile cloud computing," *J. Netw. Comput. Appl.*, vol. 106, pp. 90–99, Mar. 2018.

[4] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Inf. Sci.*, vol. 379, pp. 42–61, Feb. 2017.

[5] T. Wang *et al.*, "Data collection from WSNs to the cloud based on mobile Fog elements," *Future Gener. Comput. Syst.*, to be published, doi: 10.1016/j.future.2017.07.031.

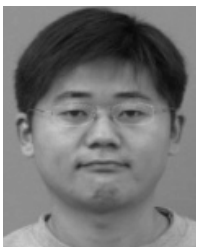
[6] L. Qi, X. Xu, W. Dou, J. Yu, Z. Zhou, and X. Zhang, "Time-aware IoE service recommendation on sparse data," *Mobile Inf. Syst.*, vol. 2016, Nov. 2016, Art. no. 4397061.

[7] T. Wang, M. Z. A. Bhuiyan, G. Wang, M. A. Rahman, J. Wu, and J. Cao, "Big data reduction for a smart city's critical infrastructural health monitoring," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 128–133, Mar. 2018.

[8] S. Mathur *et al.*, "Parknet: Drive-by sensing of road-side parking statistics," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services*, 2010, pp. 123–136.

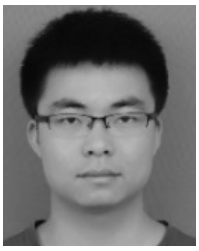
[9] L. Qi *et al.*, "'Time–Location–Frequency'–aware Internet of things service selection based on historical records," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 1, p. 1550147716688696, 2017.

[10] Y. Guo, "Mean square exponential stability of stochastic delay cellular neural networks," *Electron. J. Qualitative Theory Differ. Equ.*, vol. 2013, no. 34, pp. 1–10, 2013.
[11] L. Gao, D. Wang, and G. Wang, "Further results on exponential stability for impulsive switched nonlinear time-delay systems with delayed impulse effects," *Appl. Math. Comput.*, vol. 268, pp. 186–200, Oct. 2015.
[12] P. Li, "Some classes of singular integral equations of convolution type in the class of exponentially increasing functions," *J. Inequalities Appl.*, vol. 2017, no. 1, p. 307, 2017.
[13] B. Wang, "Exponential Fourier collocation methods for solving first-order differential equations," *J. Comput. Math.*, vol. 35, pp. 711–736, Aug. 2017.
[14] Y. Cao *et al.*, "A trajectory-driven opportunistic routing protocol for VCPS," *IEEE Trans. Aerosp. Electron. Syst.*, to be published, doi: 10.1109/TAES.2018.2826201.
[15] Y. Cao, Z. Sun, H. Cruickshank, and F. Yao, "Approach-and-roam (AaR): A geographic routing scheme for delay/disruption tolerant networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 1, pp. 266–281, Jan. 2014.
[16] Y. Cao and Z. Sun, "Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 654–677, 2nd Quart., 2013.
[17] S. Yang, X. Kong, and C. Tang, "A construction of linear codes and their complete weight enumerators," *Finite Fields Appl.*, vol. 48, pp. 196–226, Nov. 2017.
[18] Y. Wang, C. Yin, and X. Zhang, "Uniform estimate for the tail probabilities of randomly weighted sums," *Acta Mathematicae Applicatae Sinica, English Ser.*, vol. 30, no. 4, pp. 1063–1072, 2014.
[19] X. Wang and M. Wang, "Variable selection for high-dimensional generalized linear models with the weighted elastic-net procedure," *J. Appl. Statist.*, vol. 43, no. 5, pp. 796–809, 2016.
[20] S. Yang and Z.-A. Yao, "Complete weight enumerators of a class of linear codes," *Discrete Math.*, vol. 340, no. 4, pp. 729–739, 2017.
[21] T. Li, J. Li, Z. Liu, P. Li, and C. Jia, "Differentially private Naive Bayes learning over multiple data sources," *Inf. Sci.*, vol. 444, pp. 89–104, May 2018.
[22] L. Qi, X. Zhang, W. Dou, C. Hu, C. Yang, and J. Chen, "A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment," *Future Gener. Comput. Syst.*, vol. 88, pp. 636–643, Nov. 2018.
[23] P. Li, T. Li, H. Ye, J. Li, X. Chen, and Y. Xiang, "Privacy-preserving machine learning with multiple data providers," *Future Gener. Comput. Syst.*, vol. 87, pp. 341–350, Oct. 2018.
[24] W. Gong, L. Qi, and Y. Xu, "Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment," *Wireless Commun. Mobile Comput.*, vol. 2018, Apr. 2018, Art. no. 3075849, doi: 10.1155/2018/3075849.
[25] C.-Z. Gao, Q. Cheng, P. He, W. Susilo, and J. Li, "Privacy-preserving Naive Bayes classifiers secure against the substitution-then-comparison attack," *Inf. Sci.*, vol. 444, pp. 72–88, May 2018.
[26] X. Zhang *et al.*, "Verifiable privacy-preserving single-layer perceptron training scheme in cloud computing," *Soft Comput.*, pp. 1–14, May 2018, doi: 10.1007/s00500-018-3233-7.
[27] L. Qi, X. Zhang, W. Dou, and Q. Ni, "A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2616–2624, Nov. 2017.
[28] T. Wang *et al.*, "Fog-based storage technology to fight with cyber threat," *Future Gener. Comput. Syst.*, vol. 83, pp. 208–218, Jun. 2018.
[29] T. Qiu, A. Zhao, F. Xia, W. Si, and D. O. Wu, "ROSE: Robustness strategy for scale-free wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2944–2959, Oct. 2017.
[30] T. Qiu, N. Chen, K. Li, M. Atiquzzaman, and W. Zhao, "How can heterogeneous Internet of Things build our future: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2011–2027, 3rd Quart., 2018, doi: 10.1109/COMST.2018.2803740.
[31] T. Qiu, R. Qiao, and D. Wu, "EABS: An event-aware backpressure scheduling scheme for emergency Internet of Things," *IEEE Trans. Mobile Comput.*, vol. 17, no. 1, pp. 72–84, Jan. 2018.
[32] M. Wang, L. Song, and G. Tian, "SCAD-penalized least absolute deviation regression in high-dimensional models," *Commun. Statist.-Theory Methods*, vol. 44, no. 12, pp. 2452–2472, 2015.
[33] F. Xu, X. Zhang, Y. Wu, and L. Liu, "Global existence and the optimal decay rates for the three dimensional compressible nematic liquid crystal flow," *Acta Applicandae Mathematicae*, vol. 150, no. 1, pp. 67–80, 2017.
[34] P. Wang and L. Zhao, "Some geometrical properties of convex level sets of minimal graph on 2-dimensional Riemannian manifolds," *Nonlinear Anal.*, vol. 130, pp. 1–17, Jan. 2016.
[35] H. Tian and M. Han, "Bifurcation of periodic orbits by perturbing high-dimensional piecewise smooth integrable systems," *J. Differ. Equ.*, vol. 263, no. 11, pp. 7448–7474, 2017.
[36] H. Liu and F. Meng, "Some new generalized Volterra-Fredholm type discrete fractional sum inequalities and their applications," *J. Inequalities Appl.*, vol. 2016, p. 213, Dec. 2016.
[37] P. Li and G. Ren, "Some classes of equations of discrete type with harmonic singular operator and convolution," *Appl. Math. Comput.*, vol. 284, pp. 185–194, Jul. 2016.
[38] B. Zhang, "Remarks on the maximum gap in binary cyclotomic polynomials," *Bull. Math. Soc. Sci. Math. Roumanie*, vol. 59, no. 1, pp. 109–115, 2016.
[39] N. C. Kayal, P. Mondal, and T. K. Samanta, "Intuitionistic fuzzy stability of a quadratic functional equation," *Tbilisi Math. J.*, vol. 8, no. 2, Aug. 2015, doi: 10.1515/tmj-2015-0017.
[40] L. Pu, X. Chen, J. Xu, and X. Fu, "Crowdlet: Optimal worker recruitment for self-organized mobile crowdsourcing," in *Proc. IEEE Int. Conf. Comput. Commun.*, San Francisco, CA, USA, Jul. 2016, pp. 1–9.
[41] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *Proc. IEEE Conf. Comput. Commun.*, Apr./May 2015, pp. 2254–2262.
[42] Y. Zeng and D. Li, "A self-adaptive behavior-aware recruitment scheme for participatory sensing," *Sensors*, vol. 15, no. 9, pp. 23361–23375, 2015.
[43] W. Wang, H. Gao, C Liu, and K. K. Leung, "Credible and energy-aware participant selection with limited task budget for mobile crowd sensing," *Ad Hoc Netw.*, vol. 43, pp. 56–70, Jun. 2016.
[44] A. Singla and A. Krause, "Truthful incentives in crowdsourcing tasks using regret minimization mechanisms," in *Proc. 22nd Int. Conf. World Wide Web*, Toronto, ON, Canada, 2013, pp. 1167–1178.
[45] D. Zhao, X.-Y. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *Proc. IEEE Conf. Comput. Commun.*, Apr./May 2014, pp. 1213–1221.
[46] D. Zhao, H. Ma, and L. Liu. (2014). "Frugal online incentive mechanisms for crowdsourcing tasks truthfully." [Online]. Available: https://arxiv.org/abs/1404.2399
[47] M. D. Wagy *et al.*, "Crowdsourcing predictors of residential electric energy usage," *IEEE Syst. J.*, vol. 99, pp. 1–10, 2017.
[48] S. Karnouskos, "Crowdsourcing information via mobile devices as a migration enabler towards the smartgrid," in *Proc. IEEE Int. Conf. Smart Grid Commun.*, Brussels, Belgium, Oct. 2011, pp. 67–72.
[49] K. Humphreys and J. Y. Yu, "Crowdsourced electricity demand forecast," in *Proc. IEEE Int. Smart Cities Conf.*, Trento, Italy, Sep. 2016, pp. 1–6.



LIANYONG QI received the Ph.D. degree from the Department of Computer Science and Technology, Nanjing University, China, in 2011. He is currently an Associate Professor with the School of Information Science and Engineering, Qufu Normal University, China. He has published more than 40 peer-reviewed papers in international journals and conferences. His research interests include recommender systems and services computing.

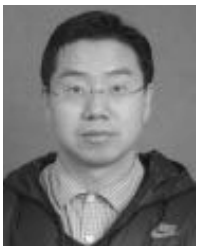
WANCHUN DOU received the Ph.D. degree in mechanical and electronic engineering from the Nanjing University of Science and Technology, China, in 2001. From 2005 to 2005 and from 2008 to 2009, he visited the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, as a Visiting Scholar. He is currently a Full Professor of the State Key Laboratory for Novel Software Technology, Nanjing University. Up to now, he has chaired three National Natural Science Foundation of China projects and published more than 60 research papers in international journals and international conferences. His research interests include workflow, cloud computing, and service computing.



WENPING WANG received the bachelor's degree in computer science and technology from Nanjing University, China. He is currently pursuing the master's degree with the Department of Computer Science and Technology, Nanjing University. His research interests include cloud computing and mobile crowdsourcing.

GUANGSHUN LI received the Ph.D. degree from Harbin Engineering University, China, in 2009. He is currently an Associate Professor of the School of Information Science and Engineering, Qufu Normal University, China. He has already published more than 30 papers. His research interests include wireless network and applications.

HAIRONG YU received the Ph.D. degree from the Huazhong University of Science and Technology, China, in 2016. She is currently a Lecturer with the School of Information Science and Engineering, Qufu Normal University, China. Her research interests include embedded systems and architecture.



SHAOHUA WAN received the Ph.D. degree from the School of Computer, Wuhan University, in 2010. Since 2015, he held a post-doctoral position with the State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology. From 2016 to 2017, he was a Visiting Scholar at the Department of Electrical and Computer Engineering, Technical University of Munich, Germany. He is currently an Associate Professor and a Master Advisor at the School of Information and Safety Engineering, Zhongnan University of Economics and Law. His research interests include massive data computing for Internet of Things and edge computing.