Received August 6, 2017, accepted September 19, 2017, date of publication September 26, 2017, date of current version October 25, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2756826

A Cooperative-Based Model for Smart-Sensing Tasks in Fog Computing

TING LI[1], YUXIN LIU[1], LONGXIANG GAO[2], AND ANFENG LIU[1]

[1]School of Information Science and Engineering, Central South University, Changsha, 410083, China
[2]School of Information Technology, Deakin University, Melbourne, VIC 3216, Australia

Corresponding author: Yuxin Liu (e-mail: yuxinliu@csu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772554, Grant 61379110, Grant 61572526, and Grant 61572528, and in part by the National Basic Research Program of China, 973 Program, under Grant 2014CB046305.

ABSTRACT Fog computing (FC) is currently receiving a great deal of focused attention. FC can be viewed as an extension of cloud computing that services the edges of networks. A cooperative relationship among applications to collect data in a city is a fundamental research topic in FC. When considering the green cloud, people or vehicles with smart-sensor devices can be viewed as users in FC and can forward sensing data to the data center. In a traditional sensing process, rewards are paid according to the distances between the users and the platform, which can be seen as the existing solution. Because users with smart-sensing devices tend to participate in tasks with high rewards, the number of users in suburban regions is smaller, and data collection is sparse and cannot satisfy the demands of the tasks. However, there are many users in urban regions, which makes data collection costly and of low quality. In this paper, a cooperative-based model for smartphone tasks, named a cooperative-based model for smart-sensing tasks (CMST), is proposed to promote the quality of data collection in FC networks. In the CMST scheme, we develop an allocation method focused on improving the rewards in suburban regions. The rewards to each user with a smart sensor are distributed according to the region density. Moreover, for each task there is a cooperative relationship among the users; they cooperate with one another to reach the volume of data that the platform requires. Extensive experiments show that our scheme improves the overall data-coverage factor by 14.997% to 31.46%, and the platform cost can be reduced by 35.882%.

INDEX TERMS Fog computing, cloud computing, smart-sensing tasks, costs, coverage.

I. INTRODUCTION

Cloud computing uses high-speed network infrastructures to provide services related to storage and computation in a modern society [1]–[7]. It continues to evolve as a domain that offers those services by managing a pool of data in a city. Such data can contribute to an application platform. An adequate utilization of these data for the application platform is a major concern for the cloud-computing paradigm [8], [9]. Similarly, evolved fog computing results in a distribution of applications and services, and people or vehicles with smart-sensing devices (such as smartphones) can be viewed as mobile devices (or users) in fog computing when supporting the cloud and the Internet of Things [8], [10]–[12]. This cloud-computing platform contains many data centers and applications and can publish tasks in the networks. Then, those smart-sensing devices in Fog computing can sense the data for the published tasks and participate in a variety of tasks [13]–[16]. Therefore, these smart-sensing devices (or users) are an important part of the fog computing process.

In a modern society, almost everyone can own one or more smart-sensing devices, such as smartphones, that can sense the surrounding environment [17]–[20]. Moreover, smart devices for vehicles are more rich and powerful. Vehicles and people with smart-sensor devices are defined as users. Those users can participate in the tasks by collecting the data of the city that the tasks require and delivering these data to the cloud data center (DC), thus contributing to society [21]–[23]. Smartphones [8], [18], [19] are developing into significant applications processors in cloud and fog computing networks [24]–[29]. In the data collection process, users must sense and deliver these data [30]–[33]; thus, a variety of resources is consumed [33]–[35]. Energy is one of the important resources consumed by users participating in these tasks [30], [36]–[37]. Consequently, the energy issue has become a main area of interest for researchers in Green Computing [15]–[30]. However, because those users must consume other resources, such as communication resources or time, the task publisher (or applications) must pay the users rewards to encourage them to participate in the tasks [15], [23]. Once the application platform obtains sufficient data packets, it will pay the users rewards [15], [23]. In other words, the rewards to users are the costs of the application platforms. In this paper, we focus on reducing those costs.

Each user can sense the data in their surroundings in fog computing networks [1], [5], [21], [24]–[29]. In the traditional model, the devices are immobile, distributed about the city and sensing the surrounding data [31]. Those fixed devices are assumed to consume a certain amount of energy. Many papers have discussed how to reduce the consumption of energy [5], [16], [23], [30], [33], [40]. However, this type of sensor application still consumes network resources. With the rapid development of 4G and 5G wireless networks [13], smart-sensing devices such as smartphones have become an important part of people's lives. Therefore, in this paper, those smart-sensing devices can be considered users with the property of mobility. They can sense the data of their surroundings and forward those data to the tasks published by the application platform to contribute to the cloud and fog computing. The users are significant; they will improve service quality and reduce resource consumption in green computing. Some researchers focus on smart sensing such as sensor applications and proposed several solutions and schemes [18], [19], whereas others largely focus on security and privacy [41]–[44]. However, there remain several problems such as the following.

(1) Reducing the costs of application platforms. With a reasonable reward strategy for users, data of high quality can be collected at a lower cost. Therefore, formulating the optimization policy for rewards is a challenging issue. In the previous studies, the rewards are distributed equally, or distributed according to the distance, which results in large costs for the application platform, and the quality of rewards is poor. For example, in applications such as smog, if one utilizes the previous calculation methods for rewards, the number of users in the urban regions can be very large. However, in smog applications, even when a small proportion of users participate in the data collection process, the collected data volume can be very large. Moreover, in a small region, the data on the smog grains vary only slightly. Therefore, in a small region, the smog data might only require one sample; additional data from this region would be redundant. A large volume of data packets does not improve the quality of service of the smog application, whereas the costs of the application are increased. However, for suburban regions, the number of users is small, and data collection in this type of region requires more energy and resources. This pattern can lead to a loss of consumers or to decreasing application profits, which is a challenge for the applications. Thus, reducing the costs of the platform and earning more profits depends upon the policy of rewards. The quality of data collection is a significant issue and is discussed below.

(2) Quality of data collection. The quality of data collection is a main component of the studies, but data collection also must collect sufficient data. However, different from previous studies, in this participatory sensing network, the data collection persons (users in this paper) lack a relevant application background; therefore, the tasks must be simple to some degree. Those tasks cannot select the data according to quality as is normally done by some specific applications. Therefore, in this type of application, another important performance index of the collected data quality is the coverage factor of the data [45], [46]. The lack of data in some regions can seriously negatively affect the applications. For example, in applications such as traffic data, a lack of data on critical congested road segments might cause drivers great inconvenience. However, few studies focus on the quality of data collection in smart-sensor applications.

Moreover, what makes the issue more serious is that there is a balance between the costs of the application platform and the quality of data. Generally, to improve the quality of data collection, user rewards should be improved to promote user participation. A variety of data can help to improve the service quality of applications. However, when collecting data, improving user rewards can result in the rapid growth of platform costs and threaten the viability of applications. Therefore, determining how to formulate a reasonable reward policy for users to minimize costs, optimize data collection and provide consumers with maximum service quality is a significant challenge. In this paper, a Cooperative-based Model for Smart-sensing Tasks (CMST) in fog computing is proposed to promote the rewards to smart users in the suburban regions in fog computing and balance data collection in a city. Listed below are the contributions of this paper:

(1) A reward allocation method is proposed that can improve data quality and reduce the cost of an application platform in a city. In Cooperative-based Model for Smart-sensing Tasks (CMST), the rewards for each participating user are allocated according to the density of their region. For a task, the rewards to those participating users (also called the costs of the application platform) in the suburban regions are more valuable compared with those in the urban regions because users are sparsely distributed in the suburban regions. Therefore, the data is difficult to collect. Thus, the data in the suburban regions are more valuable. The rewards in the suburban regions should be greater compared with those in the urban regions. Thus, based on the reward allocation method, the application platform will pay more rewards to users who have collected key data. Therefore, when improving the overall quality of data collection, the costs of data collection can be reduced on a large scale.

(2) A new evaluating indicator for data collection, the data coverage factor, is proposed to evaluate the quality of data collection. The CMST method proposed in this paper can ensure that the collected data deliver high coverage. In the CMST scheme, based on the premise of reducing the costs of

the application platform, the quality of data collection is also ensured. This paper evaluates the quality of data collection according to the data coverage indicator. By adjusting the rewards dynamically, we can improve the data collection rate in the regions with lower coverage and incentivize the users in those regions with higher probability to collect data to ensure the coverage of data collection. Thus, the scheme in this paper cannot reduce the costs but can ensure the satisfactory quality of the collected data.

(3) The performance of the method compared with previous methods is evaluated in this paper. Through our extensive simulation study, we demonstrate that the method proposed in this paper can reduce the costs of the application platform and that the quality of data collection can be improved. Compared with the previous data collection method, the costs of data collection can be reduced by as much as 35.882%, and the coverage of data can be enhanced by 14.997% to 31.46%.

The remainder of this paper is organized as follows: in section 2, related works are reviewed. In section 3, the system model and problem statement are described. In section 4, the CMST schemes are proposed. Section 5 provides the analysis and performance of the CMST scheme based on the experimental results. We conclude the paper in section 6.

II. RELATED WORK

Extensive studies have been done on the topic of cloud and fog computing and inference [1], [5], [21], [24]–[29]. These studies have expanded to various fields, such as energy consumption [5], [16], [23], [30], [33], [40], security [41]–[44], delays [11], [12], [15], [32], [34], [43], coverage [45], [46] and power management [10]. In the process of cloud and fog computing, the issue of data collection is significant [14], [45]. Among the users of fog and cloud computing [24]–[29], there is a cooperative relationship [33]. Participating smartphones should deliver the data to the data center (DC) [35]; the smartphones can be considered the edge networks in cloud and fog computing. With fog and cloud computing, the data packets in the cloud can be utilized in several aspects, for example, in wireless networks [38], [39], [47]. Wireless sensors can be deployed in the city with the help of cloud and fog computing; the data can be gathered from these sensors. Once the data collection process is completed, the data is delivered through Wi-Fi, 4G or 5G networks to the data center; it will be stored there, forwarded to the tasks, and contribute to society [26].

Cloud computing is first defined in paper [48], which also presents candidates for future enhancement of this emerging technology. Then, papers [29] and [49] discuss the relationship between the Internet of Things (IoTs) and other emerging technologies, including big data analytics, cloud computing and fog computing. Paper [27] indicates that fog computing extends cloud computing at the edges of networks and provides several characteristic definitions of fog computing. Moreover, [28] elaborates the advantages and motivations of fog computing, analyzes the applications and connects fog computing to vehicular networks. The paper also discusses the issues of security and privacy in fog computing, which is an important issue in networks. Paper [21] not only connects the Internet of Things with fog computing but also discusses the delay issue in fog computing. Paper [1] discusses applications scheduling in the fog computing process and focuses on the influence of user mobility on application performance.

Few of the previous studies on fog computing discuss the issue of user rewards and contribution to profits. Although sensor applications can sense the surrounding data, their sensor range is limited, and those applications have limited lifetimes, which does not fit the demands of the green cloud. Therefore, in this paper, smartphone users can be viewed as sensor applications with the characteristic of mobility. We focus on balancing the rewards to those users and collecting the data in the whole city to make more contributions to society.

III. SYSTEM MODEL AND PROBLEM STATEMENT

A. SYSTEM MODEL

In the system model, suppose that the set of tasks published by the application is defined as $\mathbb{M}$, and $\mathbb{M} = \{1, 2, \ldots\ldots, \mathtt{m}\}$, where $\mathtt{m}$ represents the number of tasks. The set of users is defined as $\mathbb{N}$, and $\mathbb{N} = \{1, 2, \ldots\ldots, \mathtt{m}\}$, where $\mathtt{m}$ represents the number of uses that can participate in several tasks and upload the sensor data required by the application according to the wireless data center (DC). Because the application publishes the different types of tasks via wireless networks with smartphones connecting to the wireless networks, the users can select several tasks to accomplish within the sensing range. In fog computing, the users in the wireless networks can sense all the published tasks and can select a reasonable number of them in which to participate. In this paper, there is one wireless data center in the city. The general graph is shown in Figure 1.

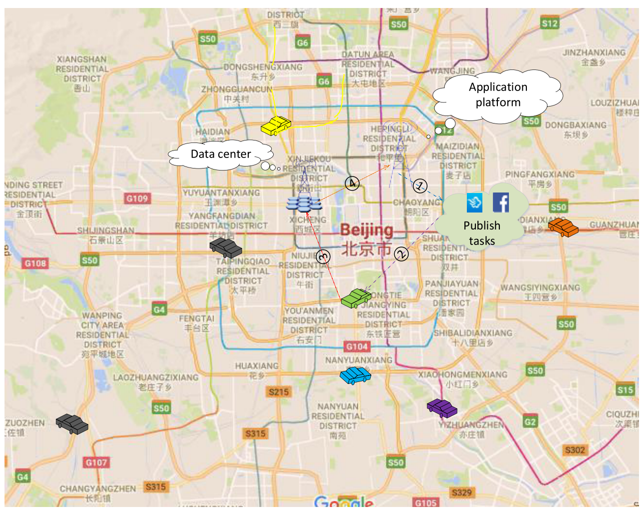

FIGURE 1. System platform.

In the first step, the application platform publishes tasks, in the second step, the users participate in some of those tasks, in the third step, the users will deliver the collection information to the data center and in the last step, the data center will forward the processed data to the application platform.

In the previous delivery model, users are assumed to participate in the tasks independently; therefore, there are no coalition relationships among users. Consequently, some tasks with high rewards might attract more users and those with lower rewards might have fewer users. To address this problem, we set a cooperative relationship among the participating users. The users will tend to take part in the tasks with high rewards. They collect the sensor data based on the location and then deliver those data to the DC. After the DC receives the data packets of task m, it then forwards those data to the application platform. The application platform can provide consumers services to obtain profits based on the collected data of the participating users.

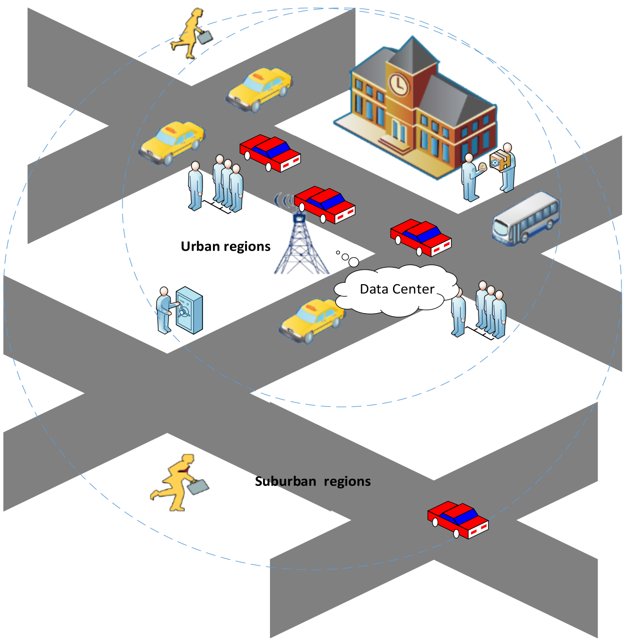

FIGURE 2. **Density distribution of users in a city.**

To encourage users to participate in the tasks, the application platform must reward those users for their contribution to the tasks. In this paper, the rewards for each user are defined to have a relationship with the density degree of a city. A city is divided into several regions according to the density degree, which is defined as $\mathbb{D} = (d_1, d_2, d_3, \ldots\ldots, d_k)$, as shown in Figure 2. In reality, the denser the region, the greater the number of users that can participate in the task. However, in a sparse region, there are fewer users; therefore, the data packets in that region are fewer than in the denser region—and, therefore, the data packets in the dense region are not as valuable as are those in the sparse region. In this paper, the rewards for a user are related to the location region in a city. If a user n has participated in task m, then the rewards to user n are defined as $\mathbb{R}_{m,n}$. The calculation method is defined in section 4.

Figure 2 shows user density in a city. As seen, in the urban regions of a city, users with sensor devices are densely distributed, whereas in the suburban regions of a city, users are sparsely distributed.

B. PROBLEM STATEMENTS

The application scenario considered in this paper is the following: in fog computing, the application platform publishes several tasks via wireless networks, the vehicles in the city receive those tasks, and the vehicles select some of them to accomplish. In this paper, we define that once user n has participated in task m, the user must return at least s data to the DC. The DC will receive those data packets from user n and then deliver them to the application platform. The application platform can obtain profits via the data and will pay the users rewards according to their contributions.

For task m, the related profit can increase with the contribution of all the participant users; we assume the profit of task m has a threshold $\mathscr{L}$. If the total contribution has reached threshold $\mathscr{L}$, then the task is finished. The profit for task m is defined as $\mathcal{U}_m$. When the profits of task m are less than the threshold, then the profits can be expressed as $\mathcal{V}_m$. The calculation methods are defined in section 4.

$$\mathcal{U}_m = \begin{cases} \mathcal{V}_m, & \textit{the total profits are less than } \mathscr{L} \\ \mathscr{L}, & \textit{the total profits reach to the } \mathscr{L} \end{cases} \tag{1}$$

Equation (1) defines the total profits of the task. Therefore, for the software platform, the total profit of all the tasks can be given by the following equation:

$$\mathcal{J} = \sum\nolimits_1^m \mathcal{U}_m \tag{2}$$

where $\mathcal{J}$ expresses the total profits of all the tasks on the platform.

Note that the users will not be willing to participate in the tasks without any rewards; after the total profits of task m have reached threshold $\mathscr{L}$, the application should pay the users according to the contributions of each participating user. Many users can participate in task m; thus, there exists a coalition relationship among the users. After task m is finished, the application will pay the rewards to the coalition group, not to the users directly. The coalition group will distribute the rewards to the users according to their contribution.

Many papers have researched data collection methods [25], [28]. However, the rewards for each participating user are not necessarily reasonable. Some data cannot be collected. Compared with data in the city, data in the suburbs is more valuable to the software. Moreover, users in the suburbs should be paid more than those in the city are paid. Previous methods did not consider this situation. Previous methods pay users according to distance or equally, an approach that might result in collecting less weather data in the suburbs.

Therefore, in this paper, to inspire users to collect data in the suburbs, we defined the platform costs (also called the rewards for the users) according to the region density of a city. The reward distribution is shown in Figure 3.

With the collaborative relationship among the users, the quality of data for the application platform can be

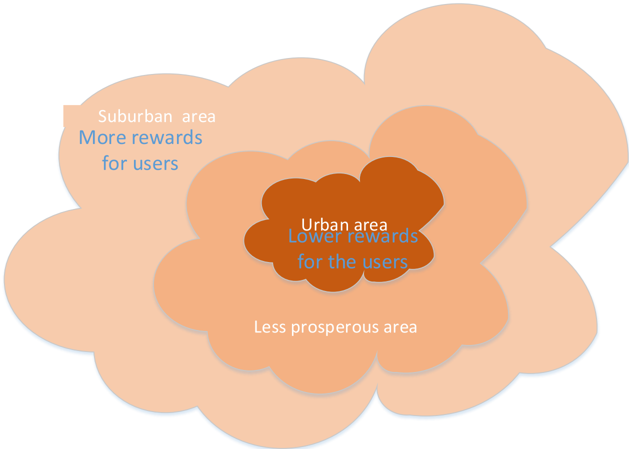

FIGURE 3. Rewards distribution.

TABLE 1. Notations.

Symbol	Description
$\mathbb{M}$	Set of published tasks in fog computing
$\mathbb{N}$	Set of users in fog computing
$\mathbb{D}$	Density degree of the city
$\mathbb{R}_{m,n}$	Rewards for user
$\mathcal{V}_m$	Profits for task m
$\mathscr{b}$	Contribution threshold for task m
$\mathcal{T}_n$	List of whether user n participates in the tasks
$\mathfrak{C}_{m,n}$	Contributions of user n
$\mathcal{Q}_m$	Coverage factor
$\mathcal{H}_m$	Collation relationship among the smartphones
$\mathbb{s}$	Data minimum
$distance(n)$	Distance between the users and the software platform
u_{m1}	Coverage factor
$number(\mathrm{p})$	Number of geographical points
$\mathcal{G}$	Total rewards

improved in the experiments.

In this paper, we also compared the coverage factor. To obtain more rewards, some users might focus on the suburbs if the distances to the suburban region and the urban area are the same.

(1) cost reduction

$$\mathcal{R} = \frac{\sum_1^m \mathbb{R}_n - \mathfrak{C}}{\sum_1^m \mathbb{R}_n}$$

where $\sum_1^m \mathbb{R}_n$ represents total platform application costs from task 1 to m, and $\mathfrak{C}$ represents the platform costs in the previous scheme, for example, distance or equal distribution.

(2) coverage factor of the data collection

The rewards for each user are distributed according to their contributions, and the contribution of each user is connected with the density distribution. To reach high rewards, the users might focus on the suburbs, and the coverage factor of the data collection can be improved. For a task, the calculation methods of the coverage factor are shown in the following equation.

$$O_m = \frac{d}{number(\mathrm{p})}$$

where $number(\mathrm{p})$ indicates the number of selected geographical points in a city, and d is the number of points that the user is in from which the data can be collected.

IV. CMST SCHEMES

A. OVERVIEW

The main contribution of CMST schemes is to consider the cooperative relationship among the users and improve the distribution methods of rewards to users. At the same time, the coverage factor can be improved. The coverage factor $\mathfrak{C}$ expresses the performance of data collection, which means that the wider the coverage factor, the more effective the scheme. Table 1 summarizes the notations used in this paper.

The vehicles in a city are assumed to be the users in fog computing. They can participate in published tasks and

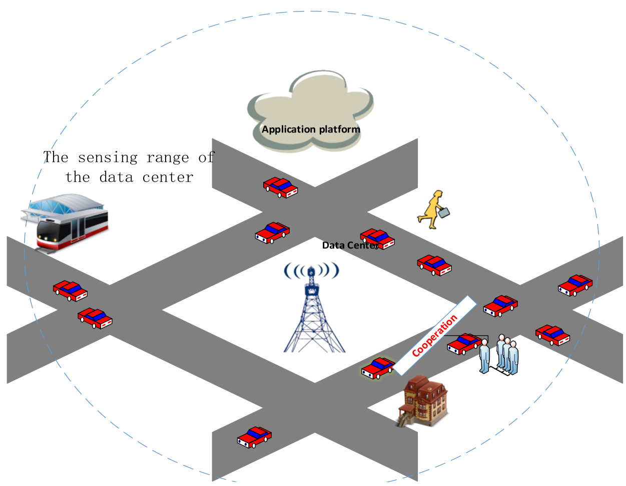

FIGURE 4. Sensing range of the data center.

contribute to the software platform. Those collected data are delivered to the DC in the city. Then, the DC forwards those data to the software, which publishes the tasks. The application platform can obtain profits via the data and pay the participating vehicles with rewards according to their contribution. The data center can receive the data packets throughout the city. The sensing range of the data center is shown in Figure 4.

In the following subsections, we describe the rewards of the user model, the profits of the application platform model, the coverage factor and the collaborative method.

B. USER REWARDS MODEL

In reality, there are plenty of users with sensor devices, those sensor devices can collect the information of the surrounding environments. Based on the large population, the information in a region can be collected completely to a large extent. Therefore, in this paper, the quality of the collected information is not taken into consideration.

The application platform publishes several tasks in fog computing. By participating in the tasks and accomplishing the tasks, users can obtain rewards from the software platform via the DC. In this paper, the user rewards are assumed to have a relationship with the density degree in a city. The more suburbs that users focus on, the more rewards they will be paid because the data in the suburbs is valuable. If user n has participated in task m, then the calculation method for the rewards is defined as follows.

$$\mathbb{R}_{m,n} = \frac{d_n}{\sum_1^k d_j} \cdot \mathbb{R}_m / number(\mathbb{N}_{d_n}) \tag{3}$$

where d_n indicates which region user n is in, $\sum_1^k d_j$ indicates the calculation results of all of the region divisions, and $\mathbb{R}_m$ is a constant different for each user that is defined at the beginning of task m. The $number(\mathbb{N}_{d_n})$ represents the number of participating users in the d_n region. Equation (3) yields the rewards for each user. As seen in Equation (3), participating users can obtain more rewards in the suburban region. To obtain more rewards, the users might focus on the suburban region if the distances to the denser region and the suburban region are the same. Therefore, the quality of collected data for the application platform can be improved in general. The distribution of rewards is shown in Figure 5 as follows.

A user n can participate in more than one task. The application platform can publish several tasks, and the users in the city can participate in some of them. In this paper, the list $\mathcal{T}_n = (t_{1,n}, t_{2,n}, \ldots\ldots, t_{m,n})$ indicates whether user n participates in task m. If the value of $t_{m,n}$ is 1, then user n selects task m; otherwise, user n does not take part in task m.

A user cannot select more than i tasks because many users selecting a given task can cause data redundancy. A user n will choose the tasks with high current rewards.

Task m will accept contributions from participating users until the associated profits reach threshold $\mathscr{L}$. Users can select another task the next time.

The pseudo code for user rewards is presented in algorithm 1.

C. APPLICATION PROFIT ALGORITHM

In this subsection, the profit of the application platform model is discussed.

Users participate in the tasks and contribute to the software platform. In this paper, the contributions of user n to task m are assumed to have a relationship with the location of user n because in the sparse region of a city, the users are fewer; therefore, the contributions of participating users are more valuable to the software platform compared with those collected data of participating users in the denser region. In this paper, the contributions of user n to task m are defined

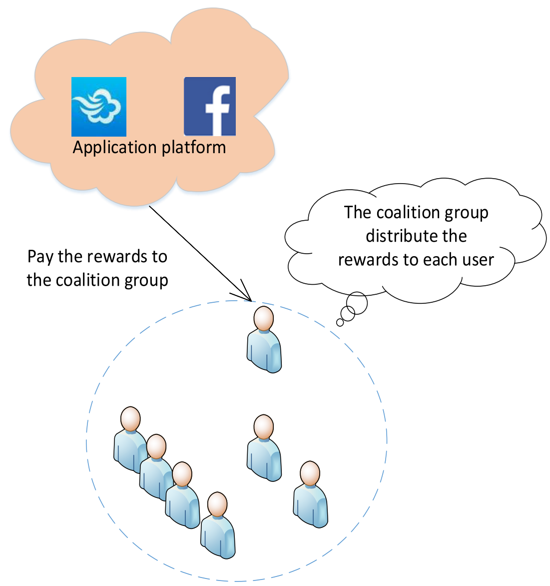

FIGURE 5. **Distribution of rewards for users.**

Algorithm 1 Algorithm Running in User n Rewards

Initialize:
Initialize the user location data;
Initialize the tasks published by the software platform.
While task number$< m$
  If the contribution of this task is less than the threshold
    User n receives the rewards of published tasks;
    Storage;
  End if
End while
For the list of storage tasks
  User n uses Quicksor to select random number of tasks
  with high rewards and returns the data to the DC
  Get the total number of users in this region division
  $\mathbb{R}_{m,n} = \frac{d_n}{\sum_1^k d_j} \cdot \mathbb{R}_m$ // the rewards calculation
End for

in the following equation.

$$\mathfrak{C}_{m,n} = \frac{d_n}{\sum_1^k d_j} \cdot \propto_m \tag{4}$$

where $\mathfrak{C}_{m,n}$ is the contribution of user n to task m, $\frac{d_n}{\sum_1^k d_j}$ has been defined in Equation (4), and $\propto_m$ is a fixed value for task m. These values differ from task to task.

Users participate in the task to obtain rewards and contribute to the task at the same time. The users tend to participate in tasks with high rewards. However, all users can take part in task m until the total contribution reaches threshold $\mathscr{L}$. Therefore, the profit of a task m is defined by the following equation:

$$\mathcal{U}_m = \begin{cases} \sum_1^n \mathcal{T}_n \mathfrak{C}_{m,n}, & \sum_1^n \mathcal{T}_n \mathfrak{C}_{m,n} < \mathscr{b} \\ \mathscr{b}, & \sum_1^n \mathcal{T}_n \mathfrak{C}_{m,n} \geq \mathscr{b} \end{cases} \tag{5}$$

where $\mathcal{U}_m$ is the profits of task m, $\mathscr{b}$ is the upper bound of the total contribution, and $\sum_1^n \mathcal{T}_n \mathfrak{C}_{m,n}$ is the total contributions of the users when the contributions are less than upper bound $\mathscr{b}$. $\mathcal{T}_n$ indicates whether user n is participating in task m. The $\mathfrak{C}_{m,n}$ in Equation (5) is defined in Equation (5). The application platform can acquire profits via collecting the data of participating users. Equation (5) shows that the more users participate, the more quickly the task can finish the process.

The application platform can obtain the profits via the collected data.

The pseudo code for profits is given in algorithm 2.

Algorithm 2 Algorithm for Computing Profits

Initialize:
Initialize all the notation here
For a task m
  For user number less than n
    For total contributions less than threshold $\mathscr{b}$
      Read in the user contributions to this task m;
      The profit is $\mathcal{T}_n \mathfrak{C}_{m,n}$; // the $\mathcal{T}_n$ indicates user n //whether to participate in the task.
    End for
  End for
End for

D. MAXIMIZE THE COVERAGE FACTOR OF THE DATA

In this paper, we also define a method to maximize the coverage rate of data in the city and verify the performance of the CMST scheme.

Some tasks published by the application platform are assumed to require data with a wider range, like the weather tasks. Therefore, the platform must achieve greater coverage in the city.

In previous schemes, the users either obtain rewards based on a distance factor or they are equally distributed, which might cause inequitable reward distribution. Moreover, the distribution of data collection is imperfect. Data availability in the suburbs might be lower, or some of it might not be forwarded to the software platform. Therefore, in a city, the coverage of data collection for task m is imperfect.

Therefore, in this paper, the city is divided into several regions according to the density degree, and the rewards to each user have a relationship with his region. To obtain more rewards, the users might focus on the suburban region if they are convenient to it. Moreover, the coverage of data collection for tasks can be improved in general. The coverage factor for a task m is defined in the following equation.

$$\mathcal{Q}_m = \frac{number(\mathbb{d})}{number(\mathbb{D})} \tag{6}$$

where the $number(\mathbb{D})$ indicates the number of geographical points, and the $number(\mathbb{d})$ indicates the number of points that have participating users. Equation (6) can obtain the coverage data of each task.

The pseudocode for the coverage factor is given in algorithm 3.

Algorithm 3 Algorithm to Maximize Coverage Factor

Initialize:
Get the location data of all participating users for task m
Initialize all the notation here
Divide the city into several regions
Select several graphical points in the city
Switch the user is in
  Case the region d_1
    number1++;
  Case in the region d_2
    number2 ++;
  Case in the region d_3
    number3 ++; // in this paper, the city is divided into
  //three regions according to the density degree.
End switch
For k < number of geographical points
  For the user $< n$
    If there exist users in the sensor range of this geographical point
      Then d++;
    End if;
  End for;
End for;
Calculate the number of users in the previous calculation list
$\mathrm{Q}_m = \frac{number(\mathbb{d})}{number(\mathbb{D})}$ // calculate the coverage factor

E. USER COLLABORATIVE ALGORITHM

In this subsection, the collaborative model of users is introduced.

For a task m, the model requires many users to participate, and the model obtains data from the users. Therefore, users participating in the tasks must cooperate with each other until the contributions of all users reach threshold $\mathscr{b}$. The users will obtain rewards from the software platform. Therefore, among the participating users, there is a collaborative relationship.

In the cooperative approach, any user n in user list $\mathfrak{N}$ can be considered a player. For any task m, we define the coalition as $\mathcal{H}_{\mathscr{m}} = (\mathscr{h}_1, \mathscr{h}_2, \ldots\ldots, \mathscr{h}_n)$, where the $\mathscr{h}_n = 1$ in the list indicates that user n is a member of coalition $\mathscr{m}$. The total number of participating users for task $\mathscr{m}$ is defined as $\mathcal{X}(\mathcal{H}_{\mathscr{m}})$. In other words, once user n has participated in task $\mathscr{m}$, he must convey at least s bits of data to the software platform for this user coalition. The contributions of the participating users cannot be greater than threshold $\mathscr{b}$;

the definition equation has been defined previously. In this paper, to ensure both the quality of collected data and that all the tasks can be completed, each user can participate in as many as tasks as desired.

The coalition relationship among the users can be defined as the following equation:

$$\mathcal{Z}_m = total\,(\mathbb{R}_m) = \wp\mathcal{U}_m = \begin{cases} \sum_1^n \mathcal{T}_n \mathfrak{C}_{m,n}, & \sum_1^n \mathcal{T}_n \mathfrak{C}_{m,n} < \mathscr{b} \\ \mathscr{b}, & \sum_1^n \mathcal{T}_n \mathfrak{C}_{m,n} \geq \mathscr{b} \end{cases} \tag{7}$$

where $\wp\mathcal{U}_m$ in Equation (7) indicates the total contributions, and $\wp$ is a factor that relates the rewards to users and the profits of the application platform. In fog computing, the application platform publishes several tasks, and users select some of them to participate in. The users can contribute via collecting data and cooperating with each other until the total contribution reaches the threshold.

V. PERFORMANCE ANALYSIS AND EXPERIMENTAL RESULTS

A. OVERVIEW

In this section, we will prove the effectiveness of CMST schemes by theoretical analysis and extensive experiments. In section 5.2, the calculation methods for rewards to users and the coverage factor of data collection are given to evaluate the performance of the CMST scheme. In section 5.3, the performance of CMST schemes is analyzed via experiments and simulations.

All simulation programs are implemented in C++ with Visual Studio 2013. In the experiment, the vehicles appear as users in fog computing, and they can sense the published tasks. Obviously, the vehicles in the urban region are more populous than those in the suburban region, which is consistent with the distribution characteristic of users discussed above.

To prove the effectiveness of the CMST scheme, in the experiments, we use the dataset of T-Drive trajectory, which is provided by MSRA [31]. The dataset of T-Drive trajectories contains the GPS trajectories of approximately 10,357 taxis in the period between Feb. 2 and Feb. 8, 2008, in Beijing. Those T-Drive data can be viewed as users in a city. The number of GPS points in the trajectory dataset is approximately 15 million, and the total trajectory distances of the datasets reach up to 9 million kilometers. The distribution graph is shown in the following graph. The vehicles with sensor devices can be regarded as the sensor devices which can collect information of surroundings. And this dataset is utilized in the simulations bellow.

Figure 6 shows that the distribution of users is dense in the urban area and sparse in the suburban region.

B. PERFORMANCE ANALYSIS

In this subsection, we discuss the performance of the CMST scheme in terms of user rewards and the data coverage factor. The calculation methods are defined as follows.

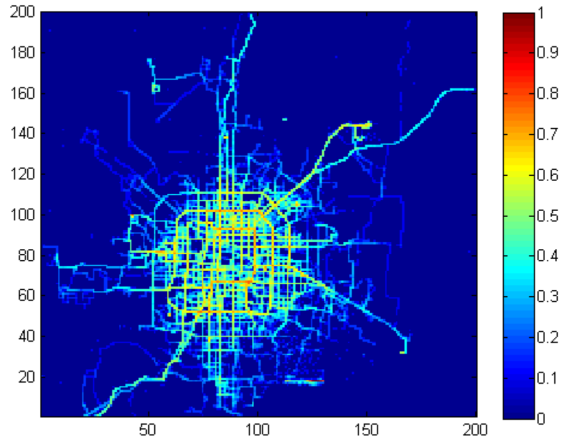

FIGURE 6. Distribution of vehicles (users).

$\mathcal{T}_n = (\ell_{1,n}, \ell_{2,n}, \ldots\ldots, \ell_{m,n})$ indicates whether a user n participates in task m. If he takes part in task $\ell_{m,n}$, then the value of $\ell_{m,n}$ is 1. Otherwise, the value of $\ell_{m,n}$ is 0. Therefore, for user n, the total rewards can be expressed in the following equation.

$$\mathfrak{N}_n = \sum\nolimits_1^m \mathbb{R}_{m,n} \cdot \mathcal{T}_n \tag{8}$$

where $\mathcal{R}_{m,n}$ indicates the rewards to user n for task m. In this paper, the rewards to users are also the costs of the application platform. In previous reward distribution methods, the rewards are distributed according to the distance between the participating users and the tasks. The calculation methods of previous rewards for each user are shown as the following equation.

$$\mathcal{Y}_n = \sum\nolimits_1^m \left(1 - \frac{distance\,(n)}{SD\,(n)}\right) \cdot \mathcal{T}_n \cdot \mathbb{R}_m \tag{9}$$

where $distance(n)$ indicates the distance of user n to the software platform, and $SD(n)$ is the sensor range of user n.

The calculation methods of reduction of costs are shown in the following equation.

$$\text{cost} = \frac{\mathcal{Y}_n - \mathfrak{N}_n}{\mathcal{Y}_n} \tag{10}$$

In this paper, considering the reality, the sensor range can cover entire networks. All users can participate in the tasks if they wish.

Therefore, for user n, the comparison of the two calculation methods for rewards can be expressed as follows:

$$\mathcal{Z} = \frac{\mathcal{Y}_n}{\mathfrak{N}_n} = \frac{\sum_1^m \mathbb{R}_{m,n} \cdot \mathcal{T}_n}{\sum_1^m \frac{distance(n)}{SD(n)} \cdot \mathcal{T}_n} \tag{11}$$

Alternatively, the rewards for each user are equally distributed; in other words, each participating user can receive the same rewards when the task has been finished. This situation is also compared with the distribution methods of rewards in this paper, shown in subsection 5.3.

We then compare the coverage factor of data collection in the CMST scheme with that of the previous scheme.

For a task m, the coverage factor in this paper is defined in equation (6).

C. PERFORMANCE IN CMST MODELS

1) COST PERFORMANCE

In this subsection, we choose several vehicles with rich trajectory datasets as sensor devices of users in the fog computing process. The application platform publishes several tasks, and they will pay the participating user rewards for user contributions to the tasks, which can be described as costs of the application platform. The performance of cost reduction approaches for the CMST scheme is compared in this section under different numbers of users.

In the previous schemes, the costs of the application platform (also called the rewards to the users) are defined to have a relationship with the distance between the participating users and the software platform.

In the experiment, Beijing is divided into three regions according to the density degree; therefore, $\mathbb{D} = (d_1, d_2, d_3)$.

The contributions of users in the urban regions are assumed greater than those of users in the suburban regions. Therefore, the rewards for the users in the urban regions are greater than are those in the suburban regions. This disparity can cause an issue wherein the data in the suburban regions cannot be collected, and the costs of the application platform are large. In the CMST scheme, the costs of the application platform (the rewards to the users) are based on the density degree of a city. The application platform will pay more rewards to the users in the suburban regions for their contributions to the overall tasks. This method can inspire users to collect the data in the suburban regions.

A comparison of the costs for each user of the application platform with 20 users is shown in Figure 7.

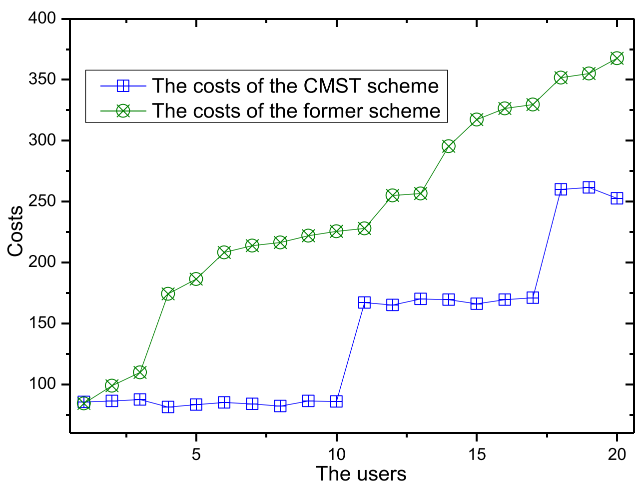

FIGURE 7. Costs of the application platform for each user when the number of user is 20.

In Figure 7, it is clear that the costs of application platforms based on the previous scheme are greater than are those of the CMST scheme because the users with sensor devices focus on regions in which they can earn more rewards. Likewise, the rewards in the urban regions are greater than are those in the suburban regions, which can create more users in the urban regions. However, in the CMST scheme, the rewards to users (costs of the application platform) are connected with region density. Because there are fewer users in the suburban regions, their contributions are more valuable compared with those in the urban regions. The rewards for the users with sensor devices are definitely greater, and the rewards in the urban regions are less than are those in the suburban regions. Therefore, generally, when the number of users is 20, the costs of an application platform can be reduced based on the CMST scheme.

In the simulation process, no number limitation exists on tasks in which the users can participate. The more tasks the users participate in, the more the application platform must pay to those users.

Then, based on the three different density divisions, when the number of users is 20, the comparisons of application platform costs are shown in the Figure 8.

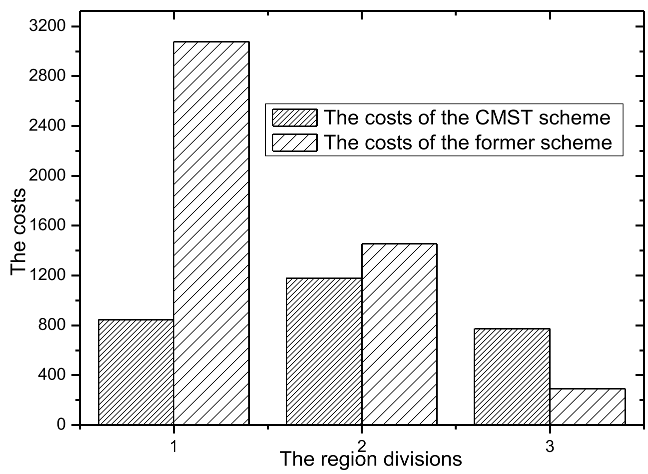

FIGURE 8. Costs of the application platform in each region with 20 users.

Figure 8 shows that in the previous scheme, application platform costs in the d_1 and d_2 regions are greater than are those in the CMST scheme, particularly in the d_1 regions. In the urban regions, the number of users is greater than that of users originally in the suburbs, and the rewards in urban regions are greater, which causes large application platform costs in the urban regions. However, as seen in the CMST scheme, the platform costs can maintain a balance to some extent. With the CMST scheme, the costs of the application platform can be reduced.

To further verify the cost-reduction performance, we then compare the costs of an application platform with the number of users at 40, 80 and 100. Comparisons are shown in Figure 9 for 40 users.

Figure 9 shows that when the number of users is 40, the costs of the application platform of the CMST scheme are generally less than those of the previous scheme. This difference exists because, based on the previous calculation

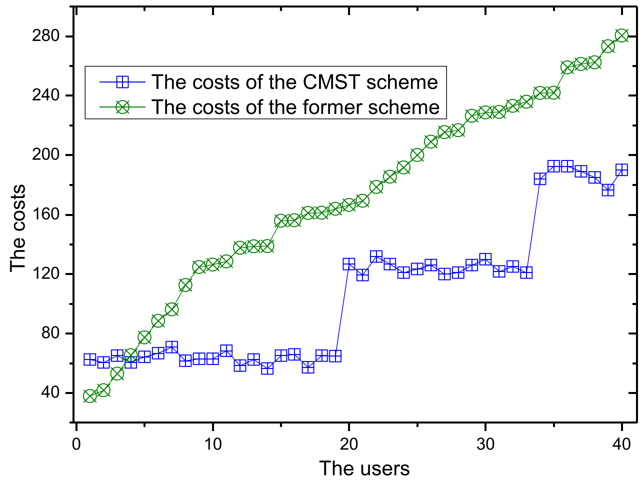

FIGURE 9. Costs of the application platform for each user with 40 users.

methods of the costs, users in the d_1 regions can obtain more rewards. Therefore, users with sensor devices in the d_2 or d_3 regions might focus on the d_1 regions if they are convenient, which could cause data redundancy in the urban regions and a shortage of data in the suburban regions.

Data distributions of data collected by sensor devices are uneven, but the application platform will still spend more on paying the participating user rewards, as shown in Figure 9.

In the CMST scheme, rewards to the users (also called platform costs) with sensor devices are based on the region density. Those users in the suburban regions make more contributions to the integrity of the data for the application platform because users in the suburban regions are fewer, and the data in the suburban regions is difficult to collect. The CMST scheme can balance the distributions of the collected data, and the application platform can spend less on paying the sensor devices in fog computing. When the number of users is 40, the costs of the platform can be reduced by 38.154%.

The costs of the platform in each density region are compared for 40 users in Figure 10 as follows.

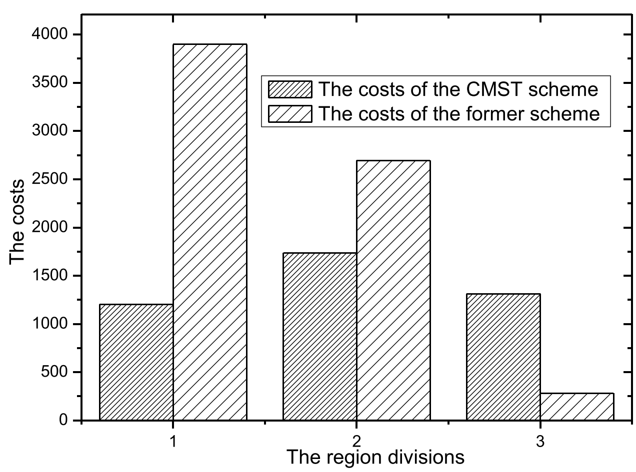

FIGURE 10. Costs of the application platform in each region with 40 users.

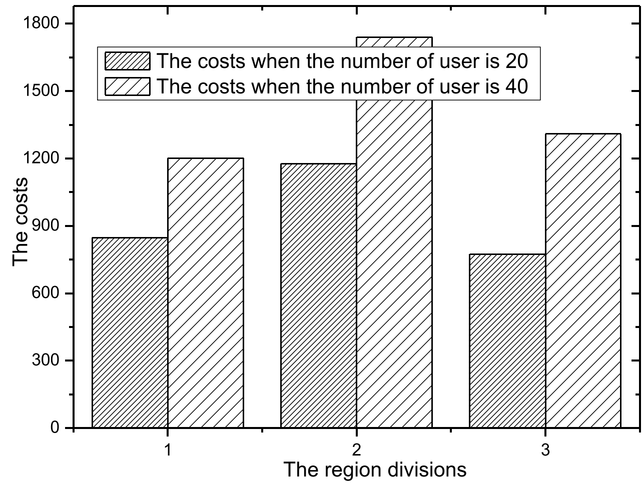

FIGURE 11. Cost comparison of the application platform.

In Figure 10, the distribution of costs in the d_1 regions of the previous scheme is clearly much greater than those in the d_2 and d_3 regions. There is a high probability that the application platform costs more, but the data collected by the users with sensor devices is repeated and redundant. The performance in the previous scheme is not satisfactory. The CMST scheme can reduce the costs of the application platform by improving the rewards in the suburban regions and inspiring the users with sensor devices to participate, which is clearly shown in Figure 10.

Moreover, Figure 10 is compared with Figure 8 to illustrate the differences in cost when the number of users is increased in the CMST scheme. The comparisons are shown in the following graph.

Figure 11 shows that with the increased number of users, the platform costs also increase.

A comparison of the application platform costs for 60 users is shown in Figure 12.

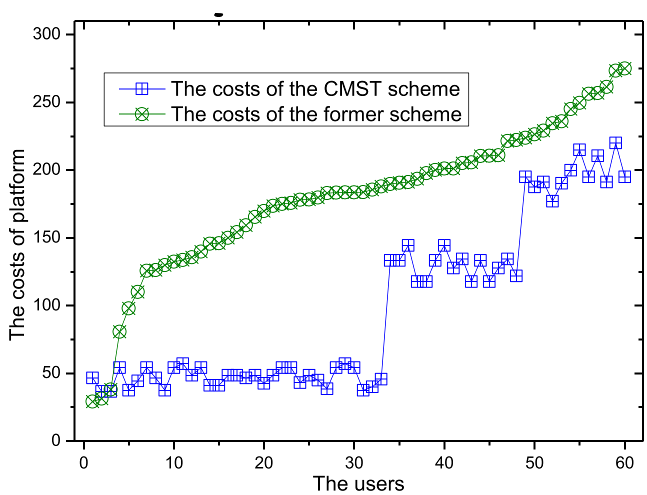

FIGURE 12. Costs of the application platform for each user with 60 users.

Figure 12 shows that the application platform costs can be considerably reduced. As the number of users increases,

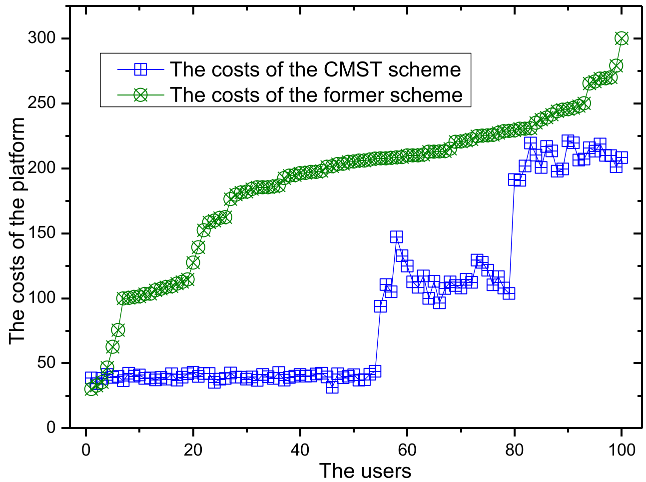

FIGURE 13. Costs of the application platform for each user with 100 users.

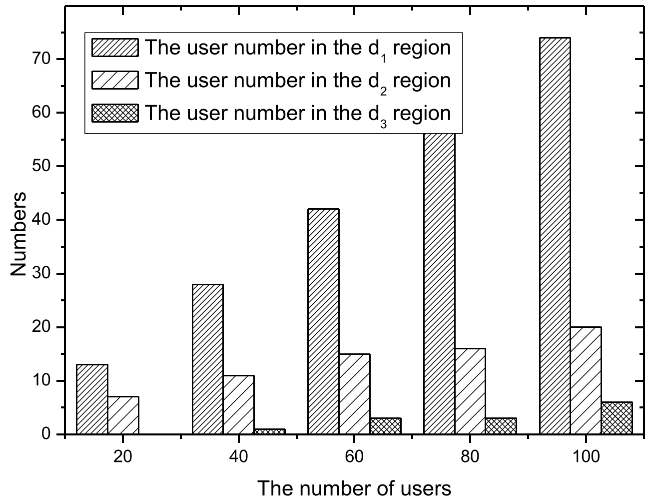

FIGURE 15. Number of users in each region according to the density divisions in the previous scheme.

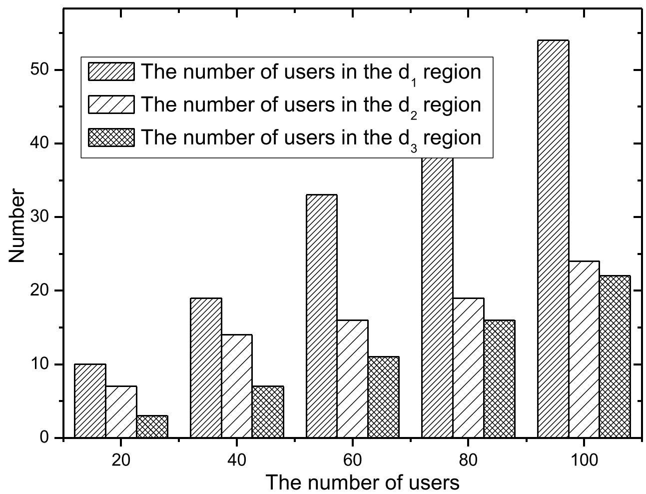

FIGURE 14. Number of users in each region according to the density divisions in the CMST scheme.

more users focus on the urban region to achieve high rewards in the previous scheme; thus, the platform costs will increase substantially. Moreover, the CMST scheme inspires the users to participate in the tasks in the suburban regions; considering actual user distributions, the costs of the platform can apparently be reduced.

Based on the two schemes, platform cost comparisons are shown in Figure 13 for 100 users.

Clearly, based on the CMST scheme, application costs can be reduced for the reason illustrated above.

To further verify the performance of the CMST scheme, we then compared the rewards to the users when the numbers of users are 80 and 100. We also compared the results with the previous calculation of rewards. The comparison is shown in Figure 14.

Figures 14 and 15 show that the number of users in the previous schemes in the d_1 region is growing faster, which imposes more costs on the platform. There are fewer users with sensor devices in the d_3 regions; in that case, the data collection cannot reach a balance. Figure 14 shows that the number of users in each region division is becoming steady. Although the number of users in the d_1 region remains greater than that of users in the d_2 and d_3 regions, there are no greater differences between them. Based on the CMST scheme, the data collected in the suburban region can be delivered to the software platform and solve the issue of uneven data distribution.

Whether the number of users is 80 or 100, the cost tendency of the application platform is the same. After utilizing the CMST scheme, platform costs can be reduced by 35.882%.

2) COVERAGE FACTOR

This section compares the data collection distributions.

The profits of each published task are calculated based on the data packets that participating users forward to the DC. In fog computing, users tend to participate in the tasks with high rewards, and the rewards in the suburban region are greater than those in the urban region. Therefore, this technique can solve the issue of uneven data distribution.

Based on the region division and the calculation methods defined above, task profits can also be described as user contributions and can be evaluated for 20 tasks.

If the user participates in a task, then he must forward at least a certain number of data items to the software platform, a number that is defined in the experiments. A large data packet is defined as 512 bytes. When the number of users is 20, based on the distribution of region divisions, the profits of the 20 tasks are shown in Figure 16.

Figure 16 shows that when the user number is 20, based on the previous calculation scheme, user contributions are distributed in the d_1 and d_2 regions. However, in the d_3 region, the profits are obviously zero because in the d_3 region, there are no participating users. Therefore, the contributions in this region are zero. Additionally, the application platform cannot

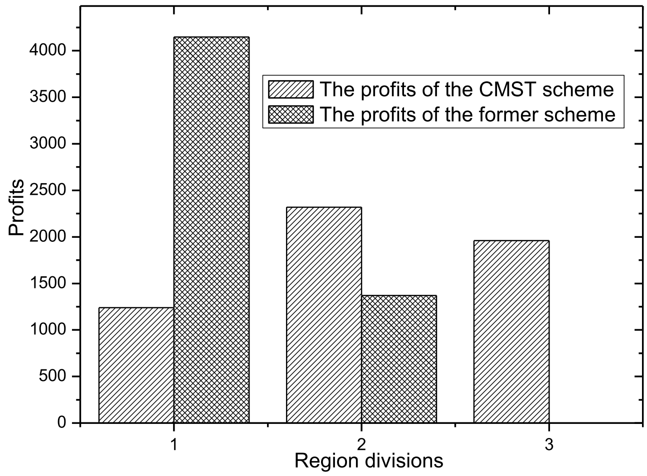

FIGURE 16. **Distributions of profits according to the region divisions with 20 users.**

obtain the data in the d_3 region. In the previous scheme, user rewards are connected with the distance between the users and the software platform. Therefore, to attain more rewards, users will focus on the urban regions, which can cause the phenomenon of uneven data collection in Figure 16. Additionally, the platform will receive a vast amount of repeat data in the urban regions and less data in the suburban regions, which does not occur in fog computing. In the CMST scheme, the rewards to each user are related to the region density. Users participating in the suburban regions can obtain more rewards compared with users in the urban regions. Therefore, as shown in Figure 16, user contributions are more stable compared with those of the previous scheme.

The profits of the region divisions are shown in the following graph for 40 users.

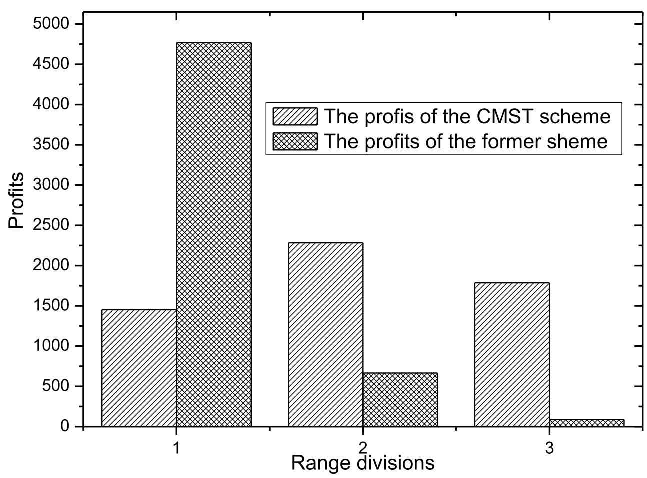

FIGURE 17. **Distribution of profits according to the region divisions with 40 users.**

Figure 17 shows that when the number of users is 40, the disparity of profits (also called user contributions) between the urban and suburban regions is growing. However, based on the CSMT scheme, the disparity between the urban regions and the suburban regions becomes smaller; therefore, the application platform can obtain the collection data of users from the whole city.

The probability of a user (that is, a vehicle in the city) coming to the urban regions is greater than that of their coming to the suburban regions. Therefore, with the number of users growing, there remain participating users in the urban regions in the CMST scheme.

To further verify the performance of the profits based on the CMST scheme, the graph below shows the profits of application platforms in different density regions for 100 users.

As seen in Figure 18, the distribution of profits based on the previous scheme is unrealistic. The data collected in the suburban region are too few. Additionally, in the CMST scheme, the profits from each region can generally maintain a balance, which can verify the performance of the CMST scheme in this paper.

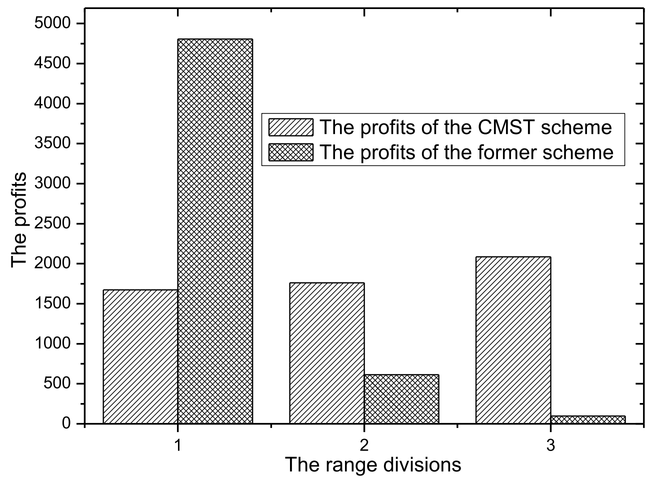

FIGURE 18. **Distribution of profits according to the region divisions with 100 users.**

The coverage factor is also compared in this subsection.

In fog computing, the application platform publishes several tasks, and users distributed in different regions can select and participate in them. Some tasks require the data at a city level, for example, weather software. The task of the weather software must obtain data for the whole city for analysis; in other words, this type of task requires the coverage factor to be maximized. Therefore, to verify the coverage factor of those tasks in this paper, we compare the coverage factor of the CMST scheme with that of the previous scheme.

To calculate the coverage factor, several geographical points are selected to analyze the coverage factor of the schemes. These geographical points have a sensor range. Therefore, if participating users exist in their sensing range, the data in this region can be collected, which indicates that this region can be covered. However, in a set of geographical points, there might be no participating users. In this situation, the coverage factor will not be greater. Thus, based on the calculation methods illustrated above, the coverage factor of the city can be obtained.

In the experiments, the sensor range of the geographical points is defined as 1800 m in this paper. The geographical points in the experiments are randomly chosen; there are 24 such points. In each density division, the number of points is 8. When the number of users is 20, based on the CMST scheme and the previous scheme, the comparisons of the coverage factor for each task can be obtained, as shown in Figure 19.

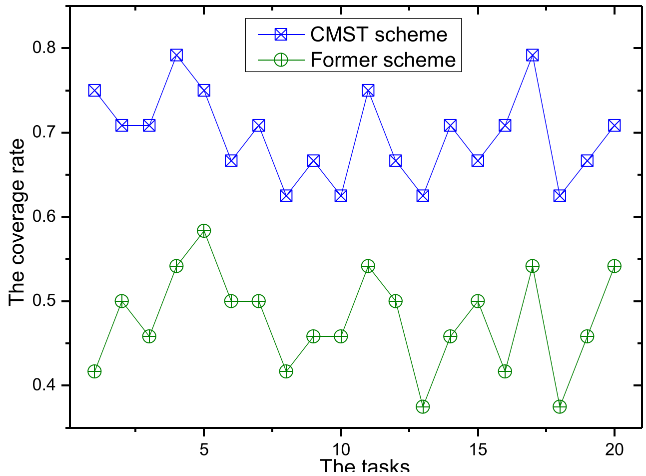

FIGURE 19. **Coverage factor of the data with 20 users.**

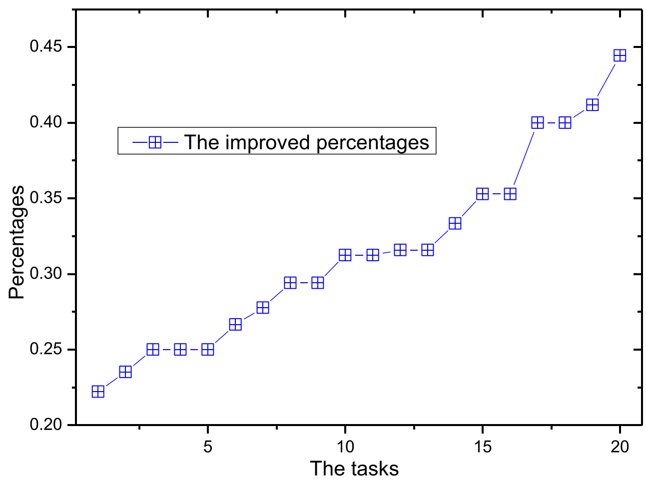

FIGURE 20. **Improved percentages of the coverage factor with 20 users.**

Figure 19 shows that in general, the coverage factor of the CMST scheme is greater than that of the previous scheme because some of the participating users might focus on the suburbs to earn more rewards. Therefore, the coverage factor in the suburbs can be improved. Thus, in general, the total coverage of the data collection task can be improved. Based on the previous scheme, when the number of users is 20, no users participate in the suburban regions; thus, the coverage factor for a task will be markedly less compared with the coverage factor of the CMST scheme. With the CMST scheme, the coverage factor of the data collections for a task can be improved.

The improved percentages of the coverage factor with 20 users can be obtained, as shown in Figure 20.

The improved percentages with 20 users are sorted from larger to smaller. From Figure 20, the improved percentages of the coverage factor for each task can be obtained.

To further verify the performance of the coverage factor, we then compare the coverage factor of the CMST scheme with the previous scheme with the number of users equal to 40, 60, 80 and 100. The comparison results are shown in the following graphs.

Figure 21 shows a comparison with 40 users of the coverage factor for the CMST scheme and the previous scheme. The coverage rate of the CMST scheme is greater than that of the previous scheme because users tend to take part in the tasks with more rewards. If users will not collect data in the suburban regions, then those regions cannot be covered, and the data in this type of region cannot be delivered to the software platform. In the CMST scheme, the users are encouraged by high rewards in the suburban regions; therefore, the number of covered points in the CMST scheme in the suburban regions is greater than in the previous scheme, as shown in Figure 21.

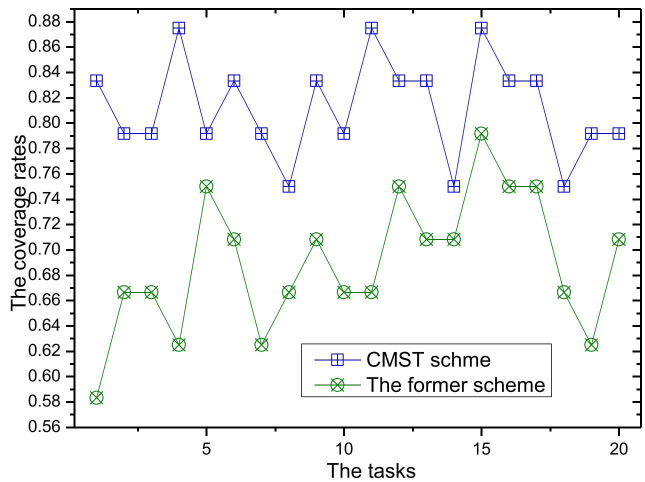

FIGURE 21. **Coverage factor of the data with 40 users.**

The improved percentages of the coverage factor for the CMST scheme and the previous scheme are calculated and sorted from large to small. The comparisons are shown in Figure 23.

Compared with Figure 20, the improved percentages of the coverage factor for each task in Figure 22 are decreased because with the increase in number of users, a smaller number of users are present in the suburban regions than in the previous schemes. Therefore, the coverage percentage can be improved. Nevertheless, it still cannot fit the demands of the software platform because the coverage of data remains lower. In the CMST scheme, with the number of users growing, users can cover more regions, particularly the suburban regions. The performance of the CMST scheme remains greater than the previous scheme.

The coverage factor of the CMST scheme is then compared with that of the previous scheme, when the number of users is 100. The comparisons are shown in the following graph.

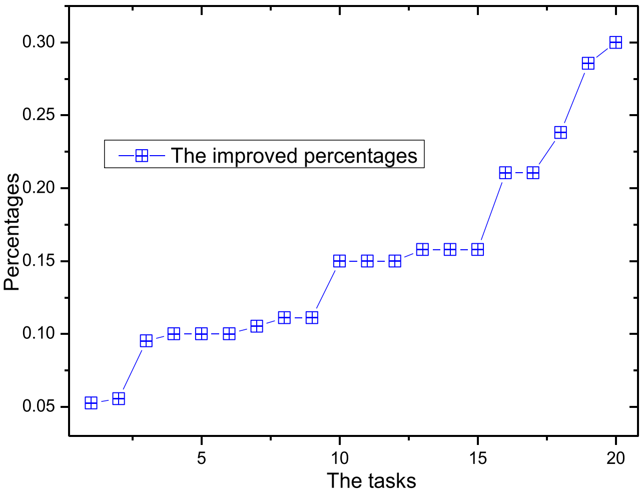

FIGURE 22. Improved percentages of the coverage factor based on 40 users.

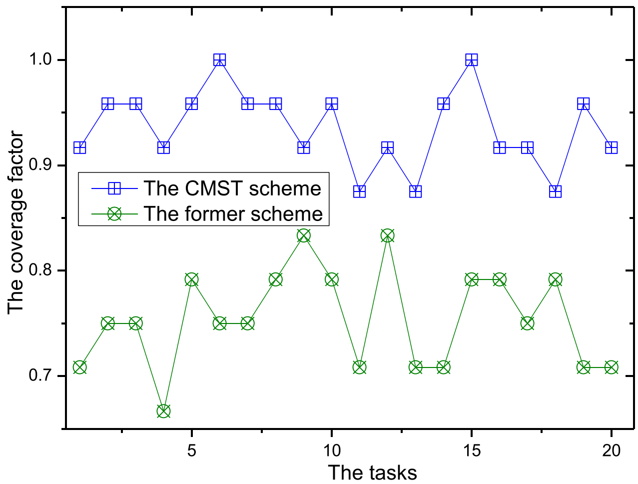

FIGURE 23. Coverage factor of the data with 100 users.

Figure 23 shows that the data-collection coverage performance can reach an ideal situation based on the CMST scheme. Additionally, the performance of the previous scheme cannot satisfy the coverage demands of the tasks. The distance between the software platforms is shorter, and the rewards to the users are greater in the previous scheme. Therefore, users focus on the urban regions, resulting in the data in the suburban regions not being collected. Moreover, the coverage factor cannot be improved at a large scale.

The improved percentages of the coverage factor are shown in Figure 24 with 100 users. Figure 24 shows that the performance tendency is the same as that shown in Figures 20 and 22. Compared with Figure 22, the improved performance of coverage factors is similar because when the number of users increases, the users focus on the urban regions to attain more rewards than in the previous scheme. This focus can cause a growing number of users in the urban regions, and the data in the suburban regions cannot be collected for lack of users.

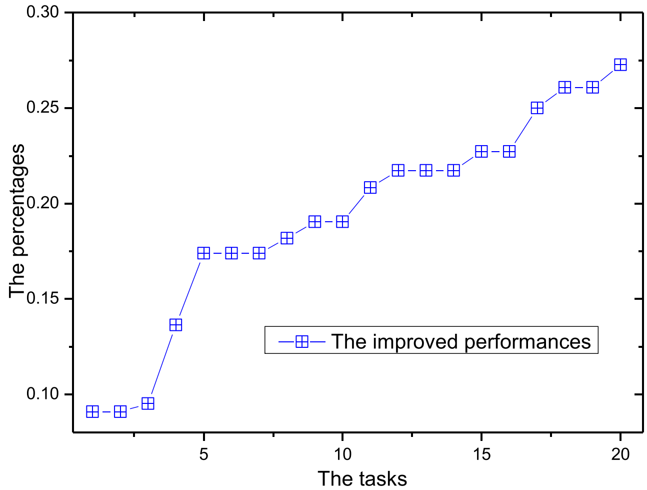

FIGURE 24. Improved percentages of the coverage factor based on 100 users.

Regardless of the increase in number of users, the performance of the previous scheme remains unsatisfactory. Moreover, in the CSMT scheme, some users might focus on the suburban regions for the increased rewards; thus, the data in these regions can be delivered to the tasks published by the software platform. Therefore, with the increased number of users, the coverage factor of the CMST scheme increases in the experiments, which is clearly shown in the above comparisons.

VI. CONCLUSION

In this paper, we propose a cooperative-based model for users in fog and cloud computing to participate in tasks published by an application platform. In the formulation process, the participating users can contribute to the tasks, and the tasks will pay those users rewards. The rewards to the users in this paper are defined to have a relationship with region density. Users in the urban regions will obtain fewer rewards compared with the users in the suburban regions. The experiments show that with the proposed CMST scheme, the tasks published by the application platform can acquire a greater volume of useful data.

With the advance of Fog computing, the forms of many practical tasks have changed. The tasks of the application platform are distributed via wireless networks. Therefore, users in a city can sense those published tasks. With the users' contributions, the tasks can generate profits and bring benefits to society. The rewards to users are the key requirement for cooperation. The CMST scheme provides a better solution to user cooperation, which can achieve better practical performance.

REFERENCES

[1] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-aware application scheduling in fog computing," *IEEE Cloud Comput.*, vol. 4, no. 2, pp. 26–35, Mar./Apr. 2017.

[2] M. A. Khan, "A survey of security issues for cloud computing," *J. Netw. Comput. Appl.*, vol. 71, pp. 11–29, Aug. 2016.

[3] Y. Liu, A. Liu, S. Guo, Z. Li, Y.-J. Choi, and H. Sekiya, "Context-aware collect data with energy efficient in cyber–physical cloud systems," *Future Generat. Comput. Syst.*, to be published, doi: 10.1016/j.future.2017.05.029.
[4] S. Yu, Y. Tian, S. Guo, and D. O. Wu, "Can we beat DDoS attacks in clouds?" *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 9, pp. 2245–2254, Sep. 2014.
[5] L. Gao *et al.*, *Fog Computing and Its Applications in 5G*. Switzerland: Springer, 2017, pp. 571–593.
[6] H. Li, D. Liu, Y. Dai, and T. H. Luan, "Engineering searchable encryption of mobile cloud networks: When QoE meets QoP," *IEEE Wireless Commun.*, vol. 22, no. 4, pp. 74–80, Aug. 2015.
[7] J. Wang, A. Liu, T. Yan, and Z. Zeng, "A resource allocation model based on double-sided combinational auctions for transparent computing," in *Peer-to-Peer Networking and Applications*. U.K.: Spring, 2017, doi: 10.1007/s12083-017-0556-6.
[8] S. Peng, S. Yu, and A. Yang, "Smartphone malware and its propagation modeling: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 925–941, 2nd Quart., 2014.
[9] M. Amjad, M. K. Afzal, T. Umer, and B.-S. Kim, "QoS-aware and heterogeneously clustered routing protocol for wireless sensor networks," *IEEE Access*, vol. 5, pp. 10250–10262, 2017, doi: 10.1109/ACCESS.2017.2712662.
[10] X. Chen, M. Ma, and A. Liu, "Dynamic power management and adaptive packet size selection for IoT in e-healthcare," *Comput. Elect. Eng.*, to be published, doi: 10.1016/j.compeleceng.2017.06.010.
[11] Y. Liu *et al.*, "FFSC: An energy efficiency communications approach for delay minimizing in Internet of Things," *IEEE Access*, vol. 4, pp. 3775–3793, 2016.
[12] Y. Xu, A. Liu, and C. Huang, "Delay-aware program codes dissemination scheme in Internet of everything," *Mobile Inf., Syst.*, vol. 2016, Nov. 2016, Art. no. 2436074. [Online]. Available: http://dx.doi.org/10.1155/2016/2436074
[13] Z. Su and Q. Xu, "Content distribution over content centric mobile social networks in 5G," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 66–72, Jun. 2015.
[14] A. Liu, X. Liu, T. Wei, L. T. Yang, S. C. Rho, and A. Paul, "Distributed multi-representative re-fusion approach for heterogeneous sensing data collection," *ACM Trans. Embedded Comput. Syst.*, vol. 16, no. 3, p. 73, 2017. [Online]. Available: http://dx.doi.org/.10.1145/2974021
[15] J. Wang, C. Hu, and A. Liu, "Comprehensive optimization of energy consumption and delay performance for green communication in Internet of Things," *Mobile Inf. Syst.*, vol. 2017, Mar. 2017, Art. no. 3206160.
[16] Q. Zhang and A. Liu, "An unequal redundancy level-based mechanism for reliable data collection in wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, p. 258, Dec. 2016.
[17] T. Li, M. Zhao, A. Liu, and C. Huang, "On selecting vehicles as recommenders for vehicular social networks," *IEEE Access*, vol. 5, pp. 5539–5555, 2017.
[18] G. A. Spina, F. Roberts, J. Weppner, P. Lukowicz, and O. Amft, "CRNTC+: A smartphone-based sensor processing framework for prototyping personal healthcare applications," in *Proc. 7th Int. Conf. Pervasive Comput. Technol. Healthcare*, May 2013, pp. 252–255.
[19] A. Kos, T. Tomažic, and A. Umek, "Evaluation of smartphone inertial sensor performance for cross-platform mobile applications," *Sensors*, vol. 16, no. 4, p. 477, 2016.
[20] B. Di, T. Wang, L. Song, and Z. Han, "Collaborative smartphone sensing using overlapping coalition formation games," *IEEE Trans. Mobile Comput.*, vol. 16, no. 1, pp. 30–43, Jan. 2017.
[21] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *Proc. Int. Conf. Future Internet Things Cloud*, Aug. 2014, pp. 464–470.
[22] Z. Chen, A. Liu, Z. Li, Y.-J. Choi, H. Sekiya, and J. Li, "Energy-efficient broadcasting scheme for smart industrial wireless sensor networks," *Mobile Inf. Syst.*, vol. 2017, Jan. 2017, Art. no. 7538190. [Online]. Available: https://doi.org/10.1155/2017./7538190
[23] J. Wang, A. Liu, and S. Zhang, "Key parameters decision for cloud computing: Insights from a multiple game model," *Concurrency Comput., Pract. Exper.*, vol. 29, no. 19, 2017, Art. no. UNSP e4200, doi: 10.1002/cpe.4200.
[24] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K.-K. R. Choo, and M. Dlodlo, "From cloud to fog computing: A review and a conceptual live VM migration framework," *IEEE Access*, vol. 5, pp. 8284–8300, 2017.
[25] V. K. Sehgal *et al.*, "Smart human security framework using Internet of Things, cloud and fog computing," in *Intelligent Distributed Computing*. Switzerland: Springer, 2015, pp. 251–263.
[26] V. Stantchev *et al.*, "Smart items, fog and cloud computing as enablers of servitization in healthcare," *Sensors Transducers*, vol. 185, no. 2, pp. 121–128, 2015.
[27] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st Ed. MCC Workshop Mobile Cloud Comput. (MCC)*, 2012, pp. 13–16.
[28] I. Stojmenovic and S. Wen, "The Fog computing paradigm: Scenarios and security issues," in *Proc. Fed. Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Sep. 2014, pp. 1–8.
[29] V. Pande, C. Marlecha, and S. Kayte, "A review-fog computing and its role in the Internet of Things," *Int. J. Eng. Res. Appl.*, vol. 6, no. 10, pp. 2248–9622, 2016.
[30] A. Liu, Q. Zhang, Z. Li, Y.-J. Choi, J. Li, and N. Komuro, "A green and reliable communication modeling for industrial Internet of Things," *Comput. Elect. Eng.*, vol. 58, pp. 364–381, Feb. 2017.
[31] O Yaghmazadeh, F. Cicoira, D. A. Bernards, S. Y. Yang, Y. Bonnassieux, and G. G. Malliaras, "Optimization of organic electrochemical transistors for sensor applications," *J. Polym. Sci. B, Polym. Phys.*, vol. 49, no. 1, pp. 34–39, 2015.
[32] Z. Chen, A. Liu, Z. Li, Y.-J. Choi, and J. Li, "Distributed duty cycle control for delay improvement in wireless sensor networks," *Peer-to-Peer Netw. Appl.*, vol. 10, no. 3, pp. 559–578, 2017.
[33] X. Liu *et al.*, "Distributed cooperative communication nodes control and optimization reliability for resource-constrained WSNs," *Neurocomputing*, vol. 270, pp. 122–136, Dec. 2017. [Online]. Available: https://doi.org/10.1016/j.neucom.2016.12.105
[34] X. Chen, Y. Xu, and A. Liu, "Cross layer design for optimizing transmission reliability, energy efficiency, and lifetime in body sensor networks," *Sensors*, vol. 17, no. 4, p. 900, 2017, doi: 10.3390/s17040900.
[35] T. Li, A. Liu, and C. Huang, "A similarity scenario-based recommendation model with small disturbances for unknown items in social networks," *IEEE Access*, vol. 4, pp. 9251–9272, 2016.
[36] C. Huang, M. Ma, Y. Liu, and A. Liu, "Preserving source location privacy for energy harvesting WSNs," *Sensors*, vol. 17, no. 4, p. 724, 2017, doi: 10.3390/s17040724.
[37] J. Gui and K. Zhou, "Flexible adjustments between energy and capacity for topology control in heterogeneous wireless multi-hop networks," *J. Netw. Syst. Manage.*, vol. 24, no. 4, pp. 789–812, 2016.
[38] Q. Liu and A. Liu, "On the hybrid using of unicast-broadcast in wireless sensor networks," *Comput. Elect. Eng.*, to be published. [Online]. Available: http://dx.doi.org/10.1016/j.compeleceng.2017.03.004
[39] X. Liu, A. Liu, Q. Deng, and H. Liu, "Large-scale programing code dissemination for software-defined wireless networks," *Comput. J.*, vol. 60, no. 10, pp. 1417–1442, 2017, doi: 10.1093/comjnl/bxx014.
[40] Y. Liu *et al.*, "APMD: A fast data transmission protocol with reliability guarantee for pervasive sensing data communication," *Pervasive Mobile Comput.*, to be published, doi: 10.1016/j.pmcj.2017.03.012.
[41] X. Liu, M. Dong, K. Ota, L. T. Yang, and A. Liu, "Trace malicious source to guarantee cyber security for mass monitor critical infrastructure," *J. Comput. Syst. Sci.*, to be published. [Online]. Available: http://dx.doi.org/10.1016/j.jcss.2016.09.008
[42] J. Long, M. Dong, K. Ota, and A. Liu, "A green TDMA scheduling algorithm for prolonging lifetime in wireless sensor networks," *IEEE Syst. J.*, vol. 11, no. 2, pp. 868–877, Jun. 2017.
[43] A. Liu, X. Liu, Z. Tang, L. T. Yang, and Z. Shao, "Preserving smart sink-location privacy with delay guaranteed routing scheme for WSNs," *ACM Trans. Embedded Comput. Syst.*, vol. 16, no. 3, p. 68, 2017. [Online]. Available: http://dx.doi.org/10.1145/2990500
[44] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, X. S. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," *IEEE Trans. Depend. Sec. Comput.*, vol. 13, no. 3, pp. 312–325, May/Jun. 2016.
[45] Y. Xu, X. Chen, A. Liu, and C. Hu, "A latency and coverage optimized data collection scheme for smart cities based on vehicular ad-hoc networks," *Sensors*, vol. 17, no. 6, p. 888, 2017, doi: 10.3390/s17040888.
[46] S. He, J. Chen, X. Li, X. Shen, and Y. Sun, "Mobility and intruder prior information improving the barrier coverage of sparse sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 6, pp. 1268–1282, Jun. 2015.

[47] X. Liu, G. Li, S. Zhang, and A. Liu, "Big program code dissemination scheme for emergency software-define wireless sensor networks," in *Peer-to-Peer Networking and Applications*. U.K.: Spring, 2017, doi: 10.1007/s12083-017-0565-5.
[48] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generat. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, 2009.
[49] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

TING LI is currently pursuing the degree with the School of Information Science and Engineering, Central South University, China. Her research interests include services-based network, crowd sensing networks, and wireless sensor networks.

YUXIN LIU is currently pursuing the degree with the School of Information Science and Engineering, Central South University, China. Her research interests are services-based networks, crowd sensing networks, and wireless sensor networks.

LONGXIANG GAO received the Ph.D. degree in computer science from Deakin University, Australia. He is currently a Lecturer with the School of Information Technology, Deakin University, to teach database, web development, and networking units for both Bachelor's and Master's students. Before joined Deakin University, he was a Post-Doctoral Research Fellow with IBM Research and Development, Australia. At IBM Research and Development, he had been the Core Team Member to develop a crisis event analysis, computing and reporting system to Australia Red Cross, and this project has been selected as the feature project of IBM. His research interests include data processing, mobile social networks, and fog computing.

Dr. Gao has over 30 publications, including patent, monograph, book chapter, journal, and conference papers. Some of his publications have been published in the top venue, such as the IEEE TMC, the IEEE IoT, the IEEE TDSC, and the IEEE TVT. He was the Chief Investigator for over ten research projects. He received the 2012 Chinese Government Award for Outstanding Students Abroad (Ranked No.1 in Victoria and Tasmania consular districts). He is active in the IEEE Communication Society. He has served as the TPC co-chair, publicity co-chair, organization chair, and TPC member for many international conferences.

ANFENG LIU received the M.Sc. and Ph.D. degrees in computer science from Central South University, China, in 2002 and 2005, respectively. He is a Professor with the School of Information Science and Engineering, Central South University, China. His major research interests include cloud computing, fog computing, and wireless sensor network. He is also a member of the China Computer Federation.