

Research Article

TDSRC: A Task-Distributing System of Crowdsourcing Based on Social Relation Cognition

Zhenlong Peng,^{1,2,3,4} Xiaolin Gui^{1,3} Jian An^{1,3}, Ruowei Gui,^{1,3} and Yali Ji^{1,3}

¹School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China

²TSL Business School, Quanzhou Normal University, Quanzhou 362000, China

³Key Laboratory of Computer Network of Shaanxi Province, Xi'an 710049, China

⁴High Educational Engineering Research Center of Fujian Province for E-Commerce Intelligent Based on Cloud Computing and Internet of Things, Quanzhou 362000, China

⁵Xi'an Jiaotong University Shenzhen Research School, High-Tech Zone, Shenzhen 518057, China

Correspondence should be addressed to Xiaolin Gui; xlgui@mail.xjtu.edu.cn and Jian An; anjian@mail.xjtu.edu.cn

Received 28 August 2018; Accepted 4 December 2018; Published 5 February 2019

Guest Editor: Nelson Baloian

Copyright © 2019 Zhenlong Peng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Crowdsourcing significantly augments the creativity of the public and has become an indispensable component of many problem-solving pipelines. The main challenge, however, is the effective identification of malicious participators while distributing crowdsourcing tasks. In this paper, we propose a novel task-distributing system named *Task-Distributing system of crowdsourcing based on Social Relation Cognition* (TDSRC) to select qualified participators. First, we divided the tasks into categories according to task themes. Then, we constructed and calculated the Abilities Set (AS), Abilities Values (AVs), and the Friends' Abilities Matrix (FAM) by using the historical interactive texts between a given task publisher (requester) and its friends. When a requester distributes a task, TDSRC can generate the candidate participators' sequence based on the task needs and FAM. Finally, the best-matched friends in the sequence are selected as the task receivers (solvers), thus producing a personal FAM to disseminate the tasks. The experimental results indicate that (1) the proposed system can accurately and effectively discover the requester's friends' abilities and select appropriate solvers and (2) the natural trust relationship in the social network reduces fraudsters and enhances the quality of crowdsourcing services.

1. Introduction

Crowdsourcing systems [1] have become a powerful, scalable, cost-effective method for promptly completing tasks, enabling requesters to allocate large-scale tasks to a crowd and obtaining results by leveraging mass wisdom. A solver crowd is typically large, anonymous, transient, and unprofessional, so that it is challenging to establish a trust relationship between a requester and solvers [2]. Some solvers may not have required abilities for tasks or they may want to obtain the reward without carefully performing the tasks, which significantly influences the quality of the task outputs [3–6].

Many works have documented recently in order to improve the crowdsourcing quality. For example, Howe [1]

proposed the *golden standard data* paradigm. Also, Eickhoff et al. [7] and Cao et al. [8] proposed another popular methodology, “*simple majority voting*.” Jeffrey et al. [9, 10] leveraged the behavioral traces captured from online solvers to predict the crowdsourcing quality. The method of weighing results based on a solver's historical performance was well established by [5, 6, 11]. Some researchers leverage social relationship in crowdsourcing system [12–14]. However, most of these prior studies have assumed that a crowdsourcing platform has information of all the solvers, and these solvers can be considered as an entire large and stable resource set. The common processing flow is shown in Figure 1(a) where the platform matches the task needs with all the participants. There are two defects in the platform: (1) the tasks can only be distributed by the platform once, and

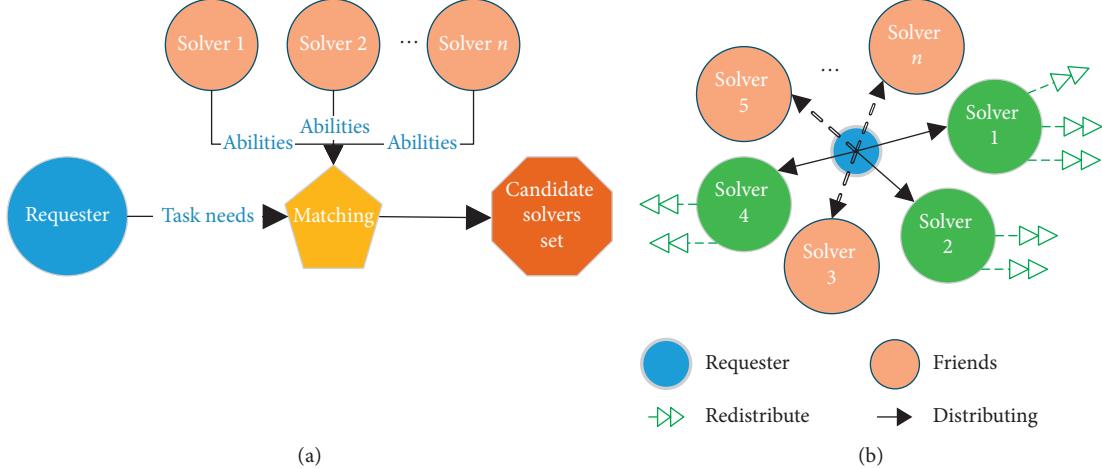


FIGURE 1: (a) Common node selection of crowdsourcing; (b) illustration of TDSRC.

the distribution process cannot be iterated by the individual solver and (2) all the potential participants can only be the registered users on the platform. In addition, the method of historical performance-based easily incurs a “cold start” because some new solvers have no any historical records.

Considering an alternative scenario in which any participant does not have to register in the crowdsourcing platform, we proposed a novel model called the Task-Distributing system of crowdsourcing based on Social Relation Cognition (TDSRC) where a requester can distribute the crowd tasks to some of his friends without obtaining information about all information of potential participants (e.g., friends’ friend). The requester needs only to generate a task and distribute it to his relevant friends. Iteratively, the friends can play the role of the next requester and redistribute the task in their social networks without any extra burden (as shown in Figure 1(b)). By introducing social relation cognition (SRC) into crowdsourcing, we establish a trust relationship that is considered to be challenging in a common crowdsourcing platform [2].

This study has the following contributions:

- (i) A method that enables a requester to efficiently distribute a task to more suitable solvers is proposed, and the accuracy of task distribution is promoted
- (ii) The social relationship is used to create and distribute a crowdsourcing task in his friend circle directly without obtaining global information (e.g., the set of all candidate solvers) which is often difficult to get
- (iii) The system can effectively avoid cold start problem which exists in performance-based methods

The remainder of this paper is organized as follows: The related studies are described in Section 2. Necessary definitions are described in Section 3. Feature discovery and candidate solver selection are discussed in Section 4. The process and simulation are described in Section 5. Section 6 summarizes this work and explores possibilities for future studies.

2. Related Works

Crowdsourcing has been attracted considerable attention since it was proposed approximately ten years ago. Lease et al. [15] indicated that quality control must be considered if the crowdsourcing quality needs to be improved. Eickhoff et al. [7] pointed out that (1) filtering low-quality solvers decreases malicious solvers but causes longer completion times and that (2) a solver’s reliability cannot be efficiently ensured by the solver’s acceptance ratio of the previous tasks. In general, the selection of crowdsourcing nodes and the guarantee of quality of service are always core issues. Many researchers have made great contributions to different aspects in this field. The following related studies are briefly reviewed as follows.

2.1. Quality Control. Howe [1] proposed the *golden standard data* paradigm. According to the paradigm, certain questions (named *golden standard data*), which are elaborately predesigned with definite baselines, are advocated by careful insertion into the tasks without being identified by solvers. By comparing the solvers’ responses to these baselines, a requester can identify unqualified solvers and precisely aggregate all task results. The main flaw of this approach is that the design of golden data is generally challenging and costly. Another popular methodology, “*simple majority voting*” [7, 8], has been extensively discussed. This method classifies solvers’ responses and aggregates the results according to the largest number of votes of the classification. Although this method is simple, it fails to identify a deceitful participant. The basic principle state of historical performance-based methods in [5, 6, 11] is that better historical performances are correlated with a greater impact on the aggregate results of the solvers. As a useful complementary technique, Jeffrey et al. [9, 10] leveraged the behavioral traces captured from online solvers to predict the crowdsourcing quality. The behavioral characteristics of the participants are highly correlated with the quality of the crowdsourcing. However, historical performance-based methods, for example, fail to consider the matched degree

between the task requirements and the potential solvers' abilities. Some solvers may have better performance for a particular type of task than another [3]. In addition, they easily incur a "cold start" because they require sufficient historical data to build an effective model.

Moreover, some challenges are encountered when crowdsourcing complex tasks. Some tasks (e.g., picture editing) may generate a substantial amount of traffic in task distribution and results collection, which hinders the ability to attract participants due to the large cost of energy and money [16–18]. Therefore, the participants must collaborate with each other (e.g., applying the "*store-carry-forward*" routing pattern [18] to upload the results' data). Considering these facts, an alternative efficient manner for improving the task quality is to allocate the task to appropriate solvers rather than using complete random distribution.

Many researchers have employed social relationships in crowdsourcing. A social relation is considered as an essential and significant attribute of a human being; numerous methodologies are employed to establish social relationships [19–21]. The famous theory of "*six degrees of separation*" [22] maintains that people are six or fewer steps from each other and that a chain of "a friend of a friend" statements can be made to connect any two people using no more than six steps. The trust relationship in society is broadly applied to personalized recommendation systems [23, 24], software crowdsourcing [25], image annotation [26], and so on. Rahman et al. [12] proposed a framework that can create a large ad hoc social network and construct an incentive based on context-aware. This framework can solve many daily life problems such as finding lost individuals, handling emergency situations, helping pilgrims to perform ritual events based on location and time, and sharing geotagged multimedia resources within the crowd. Assem et al. [13] proposed a framework for summing up the crowd mobility patterns in cities using Location-Based Social Networks (LBSNs) data which is a spatial-temporal dataset crawled from Twitter based on nonnegative matrix factorization and Gaussian kernel density estimation. This framework utilizes a temporal functional to discover the correlation between the locations and crowd, and the framework can help in better allocating resources based on the expected crowd mobility. Gan et al. [14] proposed a novel game-based incentive mechanism for multiresource sharing based on social network, and a combination of task allocation process, profit transfer process, and reputation updating process is involved in the incentive to satisfy the truthfulness and individual rationality. Yang et al. [27] introduced a novel approach named the social incentive mechanism to incentivize the social friends of the participants who perform the sensing tasks. The incentive leveraged the social ties among participants to promote global cooperation.

However, the most prior studies focused on obtaining an optimal aggregate result by identifying and excluding frauds after analyzing the collected results of a crowd, which fails to partially remove the deceivers at the earliest time (e.g., the node selection stage). The researches of introducing social relations into crowdsourcing mainly focus on the coverage of sensing region based on the participants' location

[12, 13, 28] and motivating participants by utilizing social relationships. They only use the mutual influence between friends [14, 27] and do not classify and quantify the ability of friends.

2.2. Solver Selection. The pioneering literature of solver selection in a social network was described in a study from Lappas et al. [29], in which the authors proposed a model to identify a group of individuals who can function as a team to minimize the communication cost. Zhao Dong et al. [30] designed two online mechanisms based on an online auction model. Under certain constraints (e.g., budget and time), the mechanisms can select proper solvers for different tasks and maximize the value of services. Considering the mobility of the mobile terminals, based on a time-sensitive task and a delay-tolerant task, Guo Bin et al. [31] proposed a framework named "ActiveCrowd" for multitask-oriented solver selection in large-scale mobile crowdsourcing scenarios, which applied the "greedy enhanced genetic algorithm" to achieve optimal or near-optimal solutions for minimizing the total distance and the burden, respectively, for tasks and solvers. According to the constraints of tasks, Zhang et al. [32] provided an incentive mechanism that enables a requester to actively assign most valuable tasks to the solvers. Bozzon et al. [33] proposed a model to select the *top-K* experts in a social network when a set of task needs is received. Considering both the profile information and the social activities, the model matches the expertise needs to candidate experts by formulating them as vectors. In contrast, with other team formation methods, Wang et al. [34] proposed an approach to build a collaborative team in a non-cooperative social network, which assumed that individuals are selfish and pursue the maximization of their profits. Montelisciani et al. [35] highlighted some critical issues to structure a team formation with the aim of identifying suitable solvers in crowdsourcing natural language processing (NLP). Qing Liu et al. [36] devised four incentive mechanisms for selecting a team of solvers to accomplish some complex tasks. The authors addressed the team formation problems by formulating them as a task allocation and pricing mechanism design problem.

However, the majority of these authors assumed that the requester (or crowdsourcing platform) can obtain all potential solvers in advance, which is typically impossible and unnecessary in reality. The TDSRC proposed in this paper can accommodate the lack of information about the candidate solvers and needs only routine interactive information between the requester and friends. Based on the "*six degrees of separation*" [37, 38], the trust chain between the requester and solvers can be well established and iteratively transmitted. Relative to strangers, people always prefer to believe people with whom they are more familiar. Deception among friends is relatively lower, and the crowdsourcing results become more precise and reliable [39]. Therefore, using the social relationship, the TDSRC facilitates building a trust chain between the requesters and the solvers and then enhances the accuracy and credibility of task distribution.

3. Preliminary Definitions

We aim to apply the social relations of the requester in the crowdsourcing system. The first step is to discover and quantitate friend features. In this study, friends' abilities have the same meaning as friends' features and include interest, hobbies, personality, characteristics, and integrity.

3.1. Social Network. A social network is a social structure that consists of many nodes that typically refer to individuals or organizations. Such a network links various people or organizations regardless of whether they have a close relationship [37]. The interaction among individual members in the social network form relatively stable relations and influence people's social behaviors [38].

In the book "Networked: The New Social Operating System" [40], published in 2012, Lee Rainie and Barry Wellman described the social network revolution, mobile revolution, and Internet revolution as the three revolutions that influenced human society in the new period.

A social network is formed by nodes and the connections between these nodes. Commonly, nodes consist of different types of properties [22]. The social network in this study refers to any social network. A requester is the center of a network, and an edge is a one-way connection that indicates the friend features evaluated by the center node.

The participant node is denoted as $p = \langle A \rangle$, where $A = \{a_1, a_2, \dots, a_n\}$ represents the attributes of the node. The social network is denoted as $G = \langle P, E \rangle$, where $P = \{p_1, p_2, \dots, p_m\}$ denotes the friend nodes of the central node p and $E = \langle e_{i,j} \rangle$, where

$$e_{p,j(m \times n)} = \begin{cases} x \text{ with a direct connection between } p \text{ and } p_j, \\ 0 \text{ no direct communication between } p \text{ and } p_j, \end{cases} \quad (1)$$

where x represents the strength degree of communication between p and p_j , and zero indicates that p_j has no communication.

3.2. Definitions Based on SRC. Each node has unique properties, such as hobbies and professional competence. A node typically evaluates the abilities of his friends, such as the specific interests of the friends and the friends that are suitable for specific tasks. p is a node in a social network, and p has m friends. Requesters and solvers are referred to as participants.

For the convenience of reading, the important and frequent notations used in this paper are illustrated in Table 1.

Definition 1 (ability). Ability denotes the qualities that are needed to complete a project or task. An ability is denoted as a in this study.

Definition 2 (abilities set (AS)). This set has all types of abilities to complete a crowdsourcing task. The AS is $A = \{a_1, a_2, \dots, a_n\}$ in our system. AS is a global factor that should be shared in this system.

TABLE 1: Frequently used notations.

Notation	Description
a	An ability
p	A participant node
AS	Abilities set
ASS	Subset of AS
AV	Ability value
ACR	Abilities coverage rate
QF	Qualities factor
$F_{p,p_j}^{a_k}$	The communication times in a sampling time between node p and his friend p_j for the ability (topic).
HI	Ho a_k nesty index
FAV	Friends' abilities vector, the ability values of a friend in different task classification
FAM	Friends' abilities matrix, the ability values of all friend in different task classification
e_{p,p_j}	The FAV that node p gives to his friend p_j
$C_{p,p_j}^{a_k}$	The communication times between p and his friend p_j for ability a_k .

Definition 3 (abilities subset (ASS)). This subset consists of the elements from the AS.

Definition 4 (abilities value (AV)). This digital denotation corresponds to the AS. We denote it as C . For example, $C_p = \{C_p^{a_1}, C_p^{a_2}, \dots, C_p^{a_n}\}$ denotes the abilities of node p . The original value is set between zero and one by p , and the default value is zero.

Definition 5 (abilities coverage rate (ACR)). The ACR is the proportion of the actual AS of the solvers to the demanded AS of the requester. We use δ to denote it as follows:

$$\delta = \frac{AS_{so} \cap AS_{se}}{AS_{se}}, \quad (2)$$

where AS_{so} denotes the AS of the solvers and AS_{se} represents the demanded AS of the solvers.

The ACR indicates the match degree between the solvers and the task. For example, if the government wishes to conduct a public poll, certain characteristics of the informants, such as knowledge, background, location, job category, sexuality, income, and party category, may substantially influence the results. The larger the ACR, the more typical are the results.

Definition 6 (qualities factor (QF)). The QF is the comprehensive valuation given by all friends of a solver after the solver finishes a crowdsourcing task. The QF can be denoted as Q . Hypothesis s_j is the total number of tasks that the friend p_j invited p to perform. After the tasks are completed, p_j gives p a valuation according to the performance of every task. The valuation is represented as y_z , ($z = 1, 2, s_j$) and $y_z \in [0, 1]$, and QF is denoted as Q_p^j :

$$Q_p^j = \frac{\sum_{z=1}^{s_j} w_z^j * y_z^j}{s_j}, \quad j = 1, 2, \dots, m, \quad (3)$$

where w_z^j indicates the weight of the task Z of the friend j and $w_z^j \in [0, 1]$.

Definition 7 (communication). Communication represents the interaction times between a node and its friends in the social network. A short message, telephone, and information receiving and sending on social software can be counted in communication. We use F_{p,p_j} to denote the total communication times in a sampling time between node p and his friend p_j .

Definition 8 (honesty index (HI)). This index is a weighted average of the QF evaluated by all a solver's friends. We denoted it as h_p , and it is a global variable. For example, h_p denotes the total evaluation that all friends of node p give to p :

$$h_p = \frac{\sum_{j=1}^m (W_p^j * Q_p^j)}{m}, \quad (4)$$

where W_p^j denotes the weight of the friend j to node p , which is generally set to one.

Definition 9 (friends' abilities vector (FAV)). A solver, as the central node in his social network, gives the AVs to one of his friends based on the AS according to their communications. For example, the FAV that node p gives to his friend p_j is denoted as e_{p,p_j} :

$$e_{p,p_j} = [x_{p,p_j}^1, x_{p,p_j}^2, \dots, x_{p,p_j}^n]. \quad (5)$$

Definition 10 (friends' abilities matrix (FAM)). The FAM of a node is a matrix that consists of all the node's FAVs. For example, the FAM of node p is denoted as E_p :

$$E_p = \begin{bmatrix} e_{p,1} \\ e_{p,2} \\ \vdots \\ e_{p,m} \end{bmatrix}. \quad (6)$$

4. Feature Discovery and Candidate Solver Selection

As previously discussed, we redefine the node p in the social network as a triple:

$$p = \langle h_p, C_p, E_p \rangle, \quad (7)$$

where h_p indicates the HI, C_p denotes the AVs, and E_p represents the FAM.

4.1. Computing and Updating the AVs. where d represents the total sampling times, and $C_{p,p_j}^{a_k}$ denotes the communication times between p and his friend p_j for ability a_k . The AVs update once in every sampling period, and $C_p^{a_k} \in [0, 1]$.

4.2. Computing and Updating the FAM. where F_{p,p_j} denotes the total communication times between node p and his friend p_j , and $F_{p,p_j}^{a_k}$ represents the times for the ability (topic) a_k .

Based on formula (4) and algorithms 1 and 2, node p can reconstruct itself as the following form:

$$p = \langle h_p, \{C_p^{a_1}, C_p^{a_2}, \dots, C_p^{a_n}\}, \begin{bmatrix} x_{p,1}^1 & x_{p,1}^2 & \dots & x_{p,1}^n \\ x_{p,2}^1 & x_{p,2}^2 & \dots & x_{p,2}^n \\ \vdots & \vdots & \dots & \vdots \\ x_{p,m}^1 & x_{p,m}^2 & \dots & x_{p,m}^n \end{bmatrix} \rangle. \quad (8)$$

4.3. Candidate Nodes Selection for Task Distribution

4.3.1. Definition 11 (candidate nodes (CNs)). The CNs comprise a friend subset (FSS) whose AVs match the task's demands.

When p wants to distribute a task, all he needs to do is select the task topics and set the weight for each topic. If the task is associated with a location, his friends are filtered based on the location. Then, the TDSRC generates the CNs by algorithm 3.

The topics (abilities) of the task should be set by node p ; two main parameters must be set: ASS and the weight of this subset. Assuming that $\text{ASS} = \{a_{u1}, a_{u2}, \dots, a_{up}\}$, the corresponding weight is $\{w_{u1}, w_{u2}, \dots, w_{up}\}$, the node number in CN is α , and $u1, u2, u3 \in \{1, 2, \dots, n\}$.

4.4. Quick Task-Distribution Mode Based on Abilities Coverage. According to algorithm 3, the CNs of p can be determined, and then p can push the task forward to the CNs. As shown in Figure 2, the social network of p is surrounded by a red dotted line. The CNs of p may be $\{p_1, \dots, p_m\}$, and p does not push the task to $\{p_3, p_4, p_5\}$, whose backgrounds are gray. The friend who receives the task can complete the task or redistribute the task in his social network in the same manner. The processes can be repeated until the task is completed.

According to the concept of "six degrees of separation", a task can be sent to anybody in the world by transferring six times [23–26]. Every time, we let a participant push the task to α friends in his social network (the value of α can be changed according to the demand). As a result, the distribution accuracy of the TDSRC is higher than that of a random distribution, and friends can avoid interference by irrelevant information.

5. Frameworks and Simulation

5.1. Framework and Process of the System. The modules and flow of the distribution system are illustrated in Figure 3. Assumption: $P4$ is the requester who wants to distribute a task. The main processing flow may be expounded in the following steps:

Step 1. Requester $P4$ extracts the historical contents and records between his friends and himself.

Step 2. $P4$ statistically analyzes the contents and records, selects suitable abilities from the AS, and sets relevant weights to generate the task requirements.

Input: communication times of p and his friends for different keywords;
Output: ability value for node p ;
n: number of ability types;
m: number of p 's friends;

- (1) **For** $k = 1, 2, \dots, n$
- (2) $C_p^{a_k} = (C_p^{a_k} / \sum_{k=1}^n C_p^{a_k})$; //Normalization of the initial AVs for node p .
- (3) **End for;**
- (4) $k = 1; j = 1;$
- (5) **Do while** $k \leq n$
- (6) **Do while** $j \leq m$
- (7) $C_p^{a_k} = (C_p^{a_k} + (\sum_{j=1}^m C_{p,p_j}^{a_k} / \sum_{j=1}^m \sum_{k=1}^n C_{p,p_j}^{a_k}) / d)$; //Dynamic updating of AVs of p .
- (8) $j = j + 1;$
- (9) **End Do;**
- (10) $k = k + 1;$
- (11) **End Do;**

ALGORITHM 1: Dynamic updating of the AVs.

Input: communication times of p and his friends for different keywords;
Output: the ability value for p 's friends;

- (1) $k = 1; j = 1;$
- (2) **Do while** $k \leq n$
- (3) **Do while** $j \leq m$
- (4) $x_{p,p_j}^k = h_p * (x_{p,p_j}^k + (F_{p,p_j}^{a_k} / \sum_{j=1}^m F_{p,p_j}) / d);$
- (5) $j = j + 1;$
- (6) **End Do;**
- (7) $k = k + 1;$
- (8) **End Do;**

ALGORITHM 2: Dynamic updating of the FAM.

Input: FAM of p and task demand;
Output: the candidate sequence for p to distribute the task;

- (1) **For** $i = 1$ to m ;
- (2) $WA_i = \sum_{i=1}^{up} w_{ui} * x_{p,ui}^{ui}$
- (3) **End For;**
- (4) Sorting WA_i from large to small, assuming the first α numbers are $\{WA_{p1}, WA_{p2}, \dots, WA_{p\alpha}\}$, where $p1, p2, \dots, p\alpha \in \{1, 2, \dots, m\}$
- (5) $CNs = \{p_{p1}, p_{p2}, \dots, p_{p\alpha}\};$

ALGORITHM 3: Discovering the CNs.

Step 3. As the center, $P4$ reconstructs his social network and generates the triple, as shown in formula 7.

Step 4. $P4$ generates CNs using algorithm 3, and some best-matched friends in CNs are chosen as the solvers.

Step 5. The solvers iteratively undertake or redistribute the task.

Step 6. $P4$ evaluates the responses of the friends.

Step 7. $P4$ updates relevant data in his tables.

Step 8. Friends are regarded as the next requesters if they redistribute the task in their social network. These steps are repeated (as shown in Figure 2, second distribution).

5.2. Simulation. In recent years, WeChat has become the most popular social network in China. In 2017, the number of monthly active users reached 963 million, which is 20% more than the previous year [41]. By the end of 2016, the WeChat public platforms published an average of 518 articles, each of which was read approximately 3603 times and won 17 praises [42]. Thus, WeChat has excellent transmission capacity. Regarding privacy protection, any individual in WeChat is restricted to viewing his contents and records through the WeChat system, which is suitable to our system. TDSRC simulates the process of information diffusion in a friends circle in WeChat.

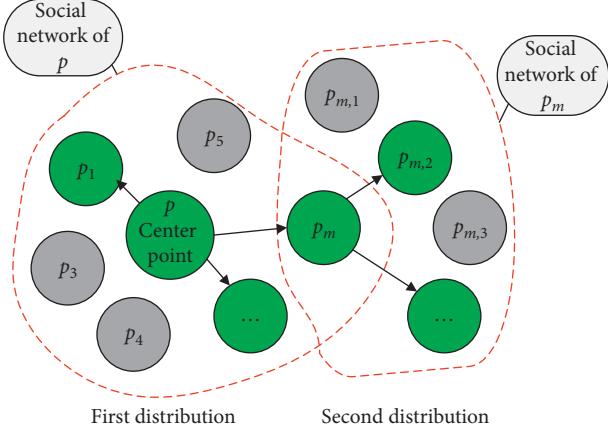


FIGURE 2: Task releasing mode based on social relationship.

5.2.1. Data Preparing. A dataset for our scheme does not exist, and conducting a comparable and real experiment to examine our scheme is challenging. The study by Bozzon et al. [33] employed a perspective that is similar to our perspective. The authors selected the *top-K* experts in a social network who fulfilled the task needs, and all potential experts were regarded as a stable and whole resource set. However, the set of candidate experts cannot be built in our system, which prevents the outcome of the two methodologies from being directly comparable. Therefore, to validate the feasibility of our scheme, we leverage web crawler technology to grab the data, e.g., task categories, time, and other data for about 8 weeks on ZhuBajie [43], which is an actual and well-known crowdsourcing platform in China. Then we simulate data according to those data. Similar to [33], in which the experts' needs were classified into seven domains (namely, *computer engineering, location, movies and TV, music, science, sport, technology, and video games*), we approximately categorize the tasks into ten types by investigating ZhuBajie. The tasks are designated $A = \{1, 2, \dots, 10\}$. Several keywords are extracted in every type, as shown in Table 2.

Assumption: Node p has 100 friends numbered from 1 to 100. The data are sampled once every three months. Ten topics (abilities) exist, as shown in Table 1. The communication times between p and his friends range from 0 to 300. The abilities' interactive times follow a Poisson distribution. Several topics are randomly selected from the ten topics, and the *FAM* of p is calculated and shown in Table 3. Only 20 friends are included in the table due to length restrictions. The numbers in the table indicate the communication times with different friends for different topics in a sampling cycle. This table can also be denoted as E_p (formula 6).

5.2.2. Abilities Discovery of Friends. The data in Table 3 cover only one sampling period. We also count the communication times in five sampling periods. The *AVs* of p can be calculated by algorithm 1, and the results after the data are normalized by formula 7 are shown in Table 4. From Table 4, we can easily determine the largest value. Column 1 and column 10 contain the largest amount of data, which indicates that p is good at (i.e., interested in) abilities 1 and 10.

The differences between one sampling and five samplings are shown in Figure 4. Only Nos. 14, 28, 42, 85, and 100 are randomly selected as the examples.

As shown in Figure 4–7, Figure 4 is similar to Figure 5, and Figure 6 is similar to Figure 7. We can conclude that the number of times for topic 1 is large, whereas the number of times for topic 2 and topic 3 is small, which implies that the *AVs* of p is relatively stable and that p likes topic 1 and he probably is interested or skilled in topic 1.

5.2.3. CN Selection. Because a requester intends to distribute a crowdsourcing task, he should select ASS and the weight of ASS. Using algorithm 3, for every ability, p can select the highest priority of ten (or other number according to the demands) friends to perform or redistribute a task. In the experiment, the CNs obtained by algorithm 3 are listed in Table 5. If p needs to release a crowdsourcing task of topic 1, he should send the task to his 92nd, 78th, 2nd, 46th friends, and so on.

The experimental results with multiple topics/abilities are shown in Table 6.

The simulations reveal that the TDSRC can successfully count the communication times according to the AS and calculate the *AVs* and *FAM*. These parameters can be simultaneously updated according to the sampling period. For any task, the TDSRC can correctly determine the most appropriate CNs by matching the abilities' demands and the friends. A CN can complete and redistribute the task in his social network, and all procedures can be iterated until the task constraints (e.g., time constraints) are violated.

5.2.4. Time Efficiency of Task Distribution. The time efficiency of task distribution is very significant mainly for delay-sensitive tasks. Therefore, we randomly selected three different types of tasks: Sports, Business, Public welfare, and Manufacturing (Nos. 1, 2, 9, and 10 in Table 2). We applied three simulation experiments, i.e., random distribution method, full distribution method, and TDSRC distribution method. In the experiment, we have 2,000 friends. We assume that the success of the task distribution is that we receive valid task execution results from 50 friends. We, respectively, selected 200 friends for the random method and TDSRC method, and full distribution means that the task is distributed to all friends. In the experiment, we also assumed that when the task ability requirement falls in the top 50% of the friend's ability matrix (*FAM*), it means that the friend will perform the task. The time spent on the task means the average time spent of the 50 friends. The experiment results for different methods are shown in Figure 8.

As can be seen from Figure 8, the random strategy takes the longest time because it cannot accurately find the most appropriate participants. The results of the full strategy are almost the same as the TDSRC strategy, which shows that TDSRC can accurately find the suitable task workers almost as much as the full strategy. However, the number of samples selected by TDSRC is only one-fifth of the full strategy, which means TDSRC brings much less interference to unrelated friends than full strategy. In addition, most task

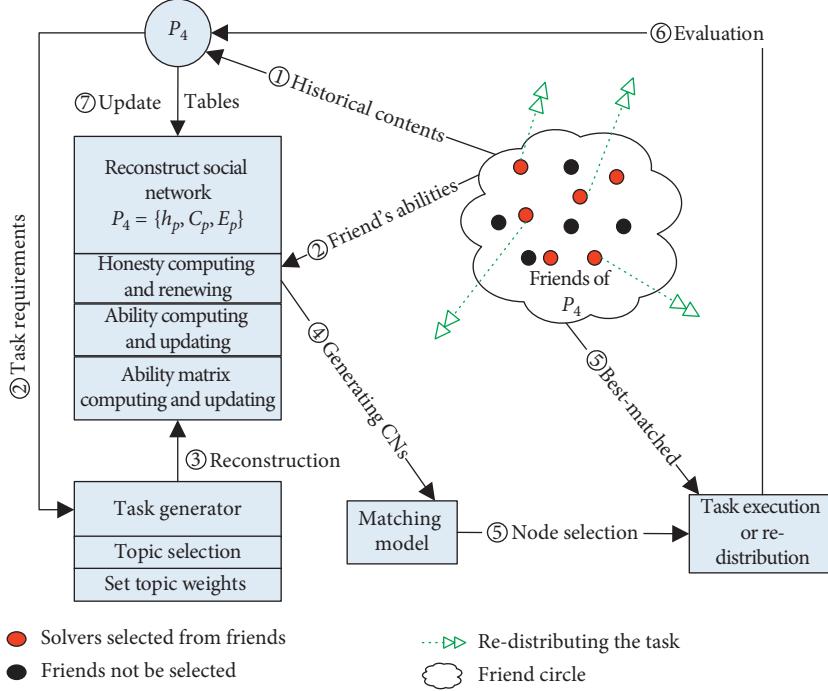


FIGURE 3: Crowdsourcing flow based on SRC in the social network.

TABLE 2: Task topics and relevant keywords.

Ability no.	Topic/ability (AS)	Keywords
1	Sports	Basketball, football, badminton, table tennis, running...
2	Business administration	Finance, law, register, HR, patent, taxation...
3	Part-time jobs	Part-time, promotion, typewriting, shopping, sharing...
4	Brand ideas	Logo, packaging, AD, VI, cartoon, video production...
5	Health and medicine	Ill, sickness, disease, cold, infect, food, drink, contagion, health...
6	Software	APP, test, website, H5, UI, develop, 3D VR...
7	Scientific research	Poll, mark, questionnaire, survey, research investigating, science, lab, experiment...
8	Personal lives	Translation, shoot, camera, travel, name, photo...
9	Public welfare	Donation, pollute, opinion, jam, safe, reward, old, young...
10	Manufacturing	Hardware, industry, machine, intelligent...

distribution is accompanied with some incentives, and the TDSRC strategy can save more costs than full distribution strategy.

6. Conclusion and Future Work

Adequate qualified participation is one of the most crucial factors that determine whether a crowdsourcing system can

TABLE 3: FAM of p (20 friends, one sampling cycle).

Friends	Abilities									
	1	2	3	4	5	6	7	8	9	10
1	157	19	25	57	103	126	98	78	149	134
2	51	36	12	45	21	31	22	27	27	74
3	62	18	65	77	27	108	57	21	77	56
4	61	22	65	96	63	68	58	75	58	73
5	178	40	110	94	86	164	53	88	109	161
6	196	129	48	65	91	57	121	158	101	120
7	5	3	3	3	2	5	1	2	1	5
8	52	35	29	20	56	69	17	27	20	31
9	106	15	71	55	13	60	65	71	92	57
10	44	11	22	48	50	59	3	18	43	46
11	108	87	46	61	105	103	62	51	89	160
12	10	7	1	9	2	7	3	7	8	9
13	48	44	19	77	72	93	44	25	47	69
14	143	23	122	144	32	68	51	81	105	95
15	112	96	15	76	18	98	88	107	98	94
16	54	63	59	82	54	69	29	64	98	64
17	11	2	6	3	5	6	3	3	4	7
18	117	114	123	70	47	114	127	125	73	202
19	71	58	50	73	26	46	5	59	50	29
20	31	0	22	24	28	18	27	11	10	19

achieve perfection. We expand participants' coverage to location, attributes, background knowledge, social relations, and credibility. The TDSRC can dynamically and automatically discover participants' abilities according to the routine communication between requesters and friends and then reconstruct their social networks to facilitate task distribution. This study is the first investigation of tasks distribution by leveraging the trust chain and transmission capabilities implied in a friends circle. The TDSRC not only

TABLE 4: Statistics of AVs for various sampling times (normalized).

Sampling times	AVs (ability from 1 to 10)
1	(0.1274, 0.0751, 0.0703, 0.1183, 0.0853, 0.1204, 0.0840, 0.0906, 0.0996, 0.1289)
5	(0.1422, 0.0789, 0.0781, 0.0787, 0.0943, 0.1245, 0.0902, 0.0800, 0.1096, 0.1236)

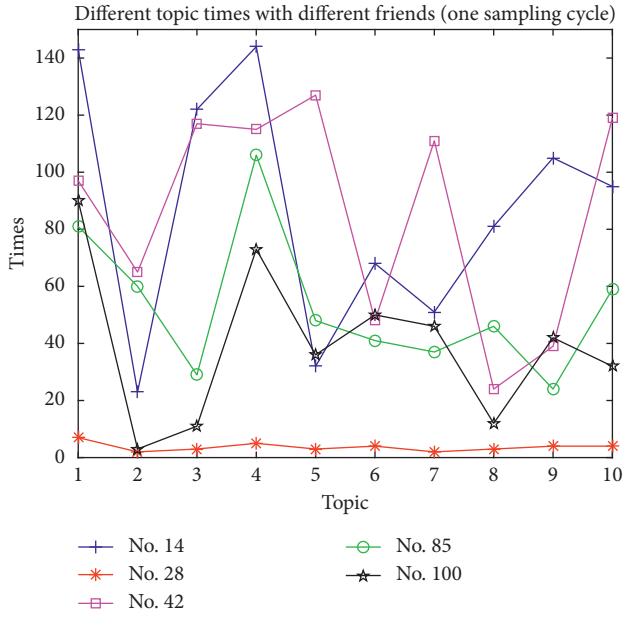


FIGURE 4: One sampling period.

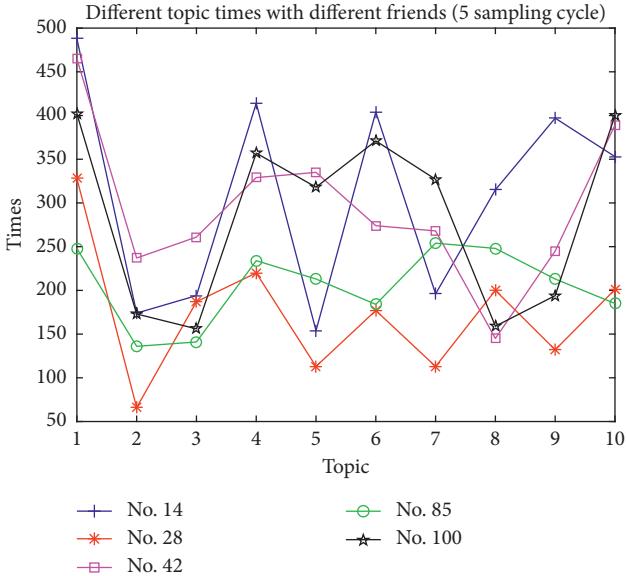


FIGURE 5: Five sampling periods.

improves the rapidity, precision, and extensy of task distribution but also protects privacy and avoids building a set of all candidates. The simulation results verify the effectiveness of the TDSRC. However, several issues warrant future investigation.

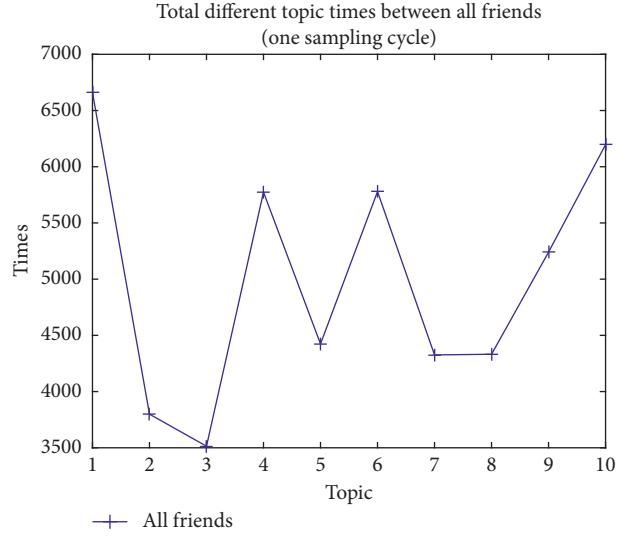


FIGURE 6: One sampling period.

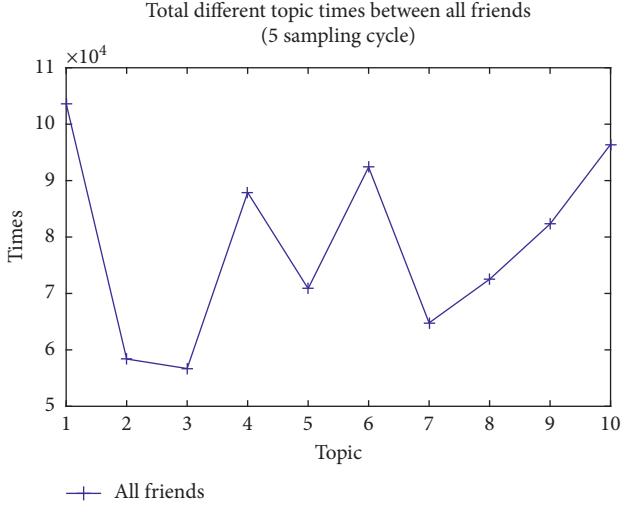


FIGURE 7: Five sampling periods.

6.1. Time Factor of Keywords. In this study, we employ communication content without considering the time factor, which is significant (e.g., a keyword that appeared one month ago is more important than a keyword that appeared six months ago). The TDSRC becomes more complex if the time factor is considered. We can compromise by setting different weights for different sampling periods. The nearer the time, the more important the content.

6.2. Weight Values of Friend Evaluation. Many weights should be set in the TDSRC (e.g., formulas 3 and 4). Different weights produce different results. Setting the weights is a topic worthy of further discussion. In our system, we use default values, which can typically be manually set by the central node. In the future, we will attempt to employ machine-learning methods to automatically set these weight values.

TABLE 5: Selection results of CNs with single ability.

Topic/ability no.	Candidate sequence nodes (CNs)									
1	92	78	2	46	94	30	73	49	1	8
2	24	46	83	52	92	1	59	6	95	15
3	31	75	98	83	52	46	78	11	80	92
4	52	30	31	57	66	83	53	42	69	51
5	69	68	46	30	75	2	78	25	51	96
6	78	30	83	66	31	88	92	97	10	52
7	30	50	92	3	80	51	68	83	10	46
8	78	83	46	52	69	66	75	30	97	6
9	30	52	97	75	80	66	12	42	83	92
10	2	46	80	32	57	43	94	12	52	83

TABLE 6: CN selection results with multiple abilities.

Topic/ ability no.	Weight respective	Candidate sequence nodes (CNs)									
1, 4	0.4, 0.6;	92	30	52	2	31	78	94	46	83	68
5, 9	0.8, 0.2;	69	30	75	46	2	68	57	24	42	31
3, 8	0.3, 0.7;	78	83	46	52	75	11	34	10	69	92
2, 6	0.9, 0.1;	83	46	24	52	92	1	59	30	43	6
7, 10	0.6, 0.4;	80	46	30	92	83	2	68	12	25	57
6, 9	0.5, 0.5	30	52	66	83	97	92	75	80	31	2

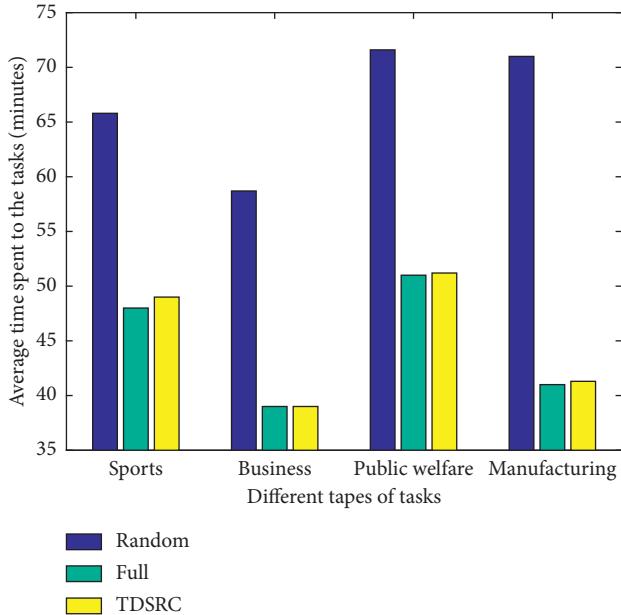


FIGURE 8: Time efficiency under different distribution strategies.

6.3. Varied Interactive Data. In this study, we considered only the contextual information. In reality, extratextual elements, such as voice, pictures, and emoji, are also popular in WeChat. Such elements play an increasingly important role in expressing emotions among friends. To take advantage of all information, AI technologies, e.g., speech recognition and image understanding, should be incorporated to enhance the complexity of the TDSRC. We plan to conduct extensive research in this area in the future.

Data Availability

The data used in the study come from ZhuBaJie (a crowdsourcing network in China; <https://www.zbj.com/>).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the NSFC (no. 61472316) and Science and Technology Program of Shenzhen under grant JCYJ20170816100939373 and partly by the Technology Project of Shaanxi Province (grant no. 2017ZDXM-GY-011), Key Projects of Science and Technology Plan in Fujian Province under grant 2016H0029, Fujian Social Science Project (grant FJ2018B022), and Key Projects of Quanzhou Technology Bureau (nos. 2018C004, 2012Z102, and 2013Z123).

References

- [1] J. Howe, “The rise of crowdsourcing,” *Wired Magazine*, vol. 14, no. 5, pp. 1–4, 2006.
- [2] D. R. Karger, S. Oh, and D. Shah, “Iterative learning for reliable crowdsourcing systems,” in *Proceedings of 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011*, Curran Associates Inc, Granada, Spain, December 2011.
- [3] Z.-Q. Zhang, J.-S. Pang, X.-Q. Xie, and Y. Zhou, “Research on crowdsourcing quality control strategies and evaluation algorithm,” *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 36, no. 8, pp. 1636–1649, 2013.
- [4] S. Han, P. Dai, P. Paritosh, and D. Huynh, “Crowdsourcing human annotation on web page structure: infrastructure design and behavior-based quality control,” *ACM Transactions on Intelligent Systems and Technology*, vol. 7, no. 4, 2016.
- [5] J. Gao, X. Liu, B. C. Ooi, H. Wang, and G. Chen, “An online cost sensitive decision-making method in crowdsourcing systems,” in *Proceedings of 2013 ACM SIGMOD Conference on Management of Data, SIGMOD 2013*, pp. 217–228, Association for Computing Machinery, New York, NY, USA, June 2013.
- [6] D. Yue, G. Yu, D. Shen, and X. Yu, “A weighted aggregation rule in crowdsourcing systems for high result accuracy,” in *Proceedings of 12th IEEE International Conference on Dependable, Autonomic and Secure Computing, DASC 2014*, pp. 265–270, Institute of Electrical and Electronics Engineers Inc, Dalian, Liaoning, China, August 2014.
- [7] E. Carsten and A. P. D. Vries, “How crowdsourcable is your task?,” in *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining*, pp. 11–14, Hong Kong, China, June 2011.
- [8] C. C. Cao, J. She, Y. Tong, and L. Chen, “Whom to ask?,” *Proceedings of the VLDB Endowment*, vol. 5, no. 11, pp. 1495–1506, 2012.
- [9] J. M. Rzeszotarski and A. Kittur, “CrowdScape: interactively visualizing user behavior and output,” in *Proceedings of 25th Annual ACM Symposium on User Interface Software and Technology, UIST 2012*, pp. 55–62, Association for Computing Machinery, Cambridge, MA, USA, October 2012.

- [10] J. M. Rzeszotarski and A. Kittur, "Instrumenting the crowd: using implicit behavioral measures to predict task performance," in *Proceedings of 24th Annual ACM Symposium on User Interface Software and Technology, UIST'11*, pp. 13–22, Association for Computing Machinery, Santa Barbara, CA, USA, October 2011.
- [11] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on amazon mechanical Turk," in *Proceedings of Human Computation Workshop 2010, HCOMP 2010*, pp. 64–67, Association for Computing Machinery, Washington, DC, USA, July 2010.
- [12] M. A. Rahman and M. S. Hossain, "A location-based mobile crowdsensing framework supporting a massive ad Hoc social network environment," *Ieee Communications Magazine*, vol. 55, no. 3, pp. 76–85, 2017.
- [13] H. Assem, T. S. Buda, and D. O'Sullivan, "RCMC: recognizing crowd-mobility patterns in cities based on location based social networks data," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 5, 2017.
- [14] X. Gan, Y. Li, W. Wang, L. Fu, and X. Wang, "Social crowdsourcing to friends: an incentive mechanism for multi-resource sharing," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 3, pp. 795–808, 2017.
- [15] M. Lease, "On quality control and machine learning in crowdsourcing," in *Proceedings of 2011 AAAI Workshop, WS-11-11*, pp. 97–102, AI Access Foundation, San Francisco, CA, USA, August 2011.
- [16] Y.-X. Zhao and Q.-H. Zhu, "Evaluation on Crowdsourcing reasearch: current status and future direction," *Information Systems Frontiers*, vol. 11, no. 1, pp. 1–18, 2012.
- [17] H. Tangmunarunkit, J. Kang, Z. Khalapyan et al., "Ohmage: a general and extensible end-to-end participatory sensing platform," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 3, 2015.
- [18] D. Zhao, H. Ma, S. Tang, and X.-Y. Li, "COUPON: a co-operative framework for building sensing maps in mobile opportunistic networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 392–402, 2015.
- [19] B. Zhang, R. Yong, M. Li, J. Pan, and J. Huang, "A hybrid trust evaluation framework for E-commerce in online social network: a factor enrichment perspective," *IEEE Access*, vol. 5, pp. 7080–7096, 2017.
- [20] V. Sharma, I. You, R. Kumar, and P. Kim, "Computational offloading for efficient trust management in pervasive online social networks using osmotic computing," *IEEE Access*, vol. 5, pp. 5084–5103, 2017.
- [21] J. An, X.-L. Gui, W.-D. Zhang, J.-H. Jiang, and J. Zhang, "Social relation cognitive model of mobile nodes in the Internet of Things," *Chinese Journal of Computers*, vol. 35, no. 6, pp. 1164–1174, 2012.
- [22] D. Alderson, J. C. Doyle, and L. Li, "Towards a theory of scale-free graphs: definition, properties, and implications," *Internet Mathematics*, vol. 4, pp. 431–523, 2005.
- [23] M. Eirinaki, M. D. Louta, and I. Varlamis, "A trust-aware system for personalized user recommendations in social networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 4, pp. 409–421, 2014.
- [24] S. Deng, L. Huang, G. Xu, X. Wu, and Z. Wu, "On deep learning for trust-aware recommendations in social networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 5, pp. 1164–1177, 2017.
- [25] N. Li, W. Mo, and B. Shen, "Task recommendation with developer social network in software crowdsourcing," in *Proceedings of 23rd Asia-Pacific Software Engineering Conference, APSEC 2016*, pp. 9–16, IEEE Computer Society, Hamilton, New Zealand, December 2016.
- [26] A. Sorokin and D. Forsyth, "Utility data annotation with amazon mechanical Turk," in *Proceedings of 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*, IEEE Computer Society, Anchorage, AK, USA, June 2008.
- [27] G. Yang, S. He, Z. Shi, and J. Chen, "Promoting cooperation by the social incentive mechanism in mobile crowdsensing," *Ieee Communications Magazine*, vol. 55, no. 3, pp. 86–92, 2017.
- [28] Z. Xu, H. Zhang, Y. Liu, and L. Mei, "Crowd sensing of urban emergency events based on social media big data," in *Proceedings of 13th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2014*, pp. 605–610, Institute of Electrical and Electronics Engineers Inc, Beijing, China, September 2014.
- [29] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 467–476, Paris, France, June 2009.
- [30] D. Zhao, X.-Y. Li, and H. Ma, "Budget-feasible online incentive mechanisms for crowdsourcing tasks truthfully," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 647–661, 2016.
- [31] B. Guo, Y. Liu, W. Wu, Z. Yu, and Q. Han, "ActiveCrowd: a framework for optimized multitask allocation in mobile crowdsensing systems," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, pp. 392–403, 2017.
- [32] X. Zhang, Z. Yang, Y. Liu, J. Li, and Z. Ming, "Toward efficient mechanisms for mobile crowdsensing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1760–1771, 2017.
- [33] A. Bozzon, M. Brambilla, S. Ceri, M. Silvestri, and G. Vesci, "Choosing the right crowd: expert finding in social networks," in *Proceedings of International Conference on Extending Database Technology*, pp. 637–648, Genoa, Italy, March 2013.
- [34] W. Wang, J. Jiang, A. Bo, Y. Jiang, and C. Bing, "Toward efficient team formation for crowdsourcing in noncooperative social networks," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–15, 2016.
- [35] G. Montelisciani, D. Gabelloni, G. Tazzini, and G. Fantoni, "Skills and wills: the keys to identify the right team in collaborative innovation platforms," *Technology Analysis and Strategic Management*, vol. 26, no. 6, pp. 687–702, 2014.
- [36] Q. Liu, T. Luo, R. Tang, and S. Bressan, "An efficient and truthful pricing mechanism for team formation in crowdsourcing markets," in *Proceedings of IEEE International Conference on Communications*, pp. 567–572, Sydney, Australia, June 2014.
- [37] M. Martínez-Núñez and W. S. Pérez-Aguiar, "Efficiency analysis of information technology and online social networks management: an integrated DEA-model assessment," *Information and Management*, vol. 51, no. 6, pp. 712–725, 2014.
- [38] A. Ziberna and V. Vehovar, "Using social network to predict the behavior of active members of online communities," in *Proceedings of 2009 International Conference on Advances in Social Network Analysis and Mining, ASONAM 2009*, pp. 119–124, IEEE Computer Society, Athens, Greece, July 2009.
- [39] S. Rocca and M. B. Brewer, "Social identity complexity," *Personality and Social Psychology Review*, vol. 6, no. 2, pp. 88–106, 2016.
- [40] L. Rainie and B. Wellman, *Networked: The New Social Operating System*, The MIT Press, London, UK, 2012.

- [41] Sohu, *The Monthly Active User of WeChat will Catch Up with QQ*, 2018, http://www.sohu.com/a/167658834_437281.
- [42] Aoshuang, *We Chat Influence Report for 2016*, 2018, <http://mydown.yesky.com/news/87280487.html>.
- [43] ZhuBajie, *ZhuBajie-A crowdsourcing Platform*, 2018, <http://www.zbj.com>.

