Received March 12, 2019, accepted April 4, 2019, date of publication April 9, 2019, date of current version April 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2909843

Intelligent Service Recommendation for Cold-Start Problems in Edge Computing

YICHAO ZHOU[1], ZHENMIN TANG[1], LIANYONG QI[2], XUYUN ZHANG[3], WANCHUN DOU[4], AND SHAOHUA WAN[5]

[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
[2]School of Information Science and Engineering, Qufu Normal University, Rizhao, China
[3]Department of Electrical and Computer Engineering, University of Auckland, Auckland, New Zealand
[4]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[5]School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan, China

Corresponding author: Lianyong Qi (lianyongqi@gmail.com)

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

ABSTRACT Memory-based collaborative filtering (i.e., MCF) is regarded as an effective technique to recommend appropriate services to target users. However, if recommendation data are very sparse in the edge environment, traditional MCF-based recommendation methods probably cannot output any recommended item (or service), i.e., a cold-start recommendation problem occurs. To cope with this cold-start problem, we propose an intelligent recommendation method named Inverse_CF_Rec. Concretely, for a target user, we first search for his/her opposite users (together referred to as ''enemy'' hereafter); afterward, we infer the possible friends of the target user indirectly according to Social Balance Theory; finally, optimal services are recommended to the target user based on the derived possible friends of the target user. The experiments are conducted on a real-world dataset WS-DREAM to validate the effectiveness and efficiency of our proposal. The experiment results show the advantages of our recommendation method in terms of recommendation accuracy and efficiency.

INDEX TERMS Service recommendation, cold-start, edge, inverse collaborative filtering, social balance theory.

I. INTRODUCTION

Recently, people have witnessed the great success of services computing technology in various intelligent and smart business applications [1]–[6]. With the continuous increment of available web services in existing communities (e.g., programmableweb.com) distributed across different edge platforms, it becomes a necessity to help a target user to find out his/her interested web services. Under this circumstance, various recommendation techniques, e.g., Memory-based Collaborative Filtering (i.e., MCF) [7] are recruited to mine the potential preferences of the target user and then make appropriate recommendation decisions. Typically, according to the traditional MCF technique, a recommender system should first search for the similar users (together referred to as ''friend'' hereafter) of the target user, and then predict missing service quality data and make recommendation decisions based on the derived friends of the target user. Generally, the abovementioned MCF-based service recommendation methods can perform very well when the historical service usage data utilized to make recommendation decisions are sufficient enough [8]–[11].

However, due to the lack of feedback incentive mechanism in edge environment, the data (e.g., historical service quality experienced by users) used to make service recommendations is often not dense enough, but very sparse [12]–[15]. In this situation, it is probable that the recommender system cannot find the similar friends of the target user as well as the similar services of the target services (i.e., services preferred by the target user). While disappointedly, existing MCF methods fail to deal with this kind of cold-start problem effectively

Considering this drawback, we propose an intelligent recommendation method based on inverse CF, named *Inverse_CF_Rec*, to smooth the cold-start recommendation

The associate editor coordinating the review of this manuscript and approving it for publication was Ying Li.

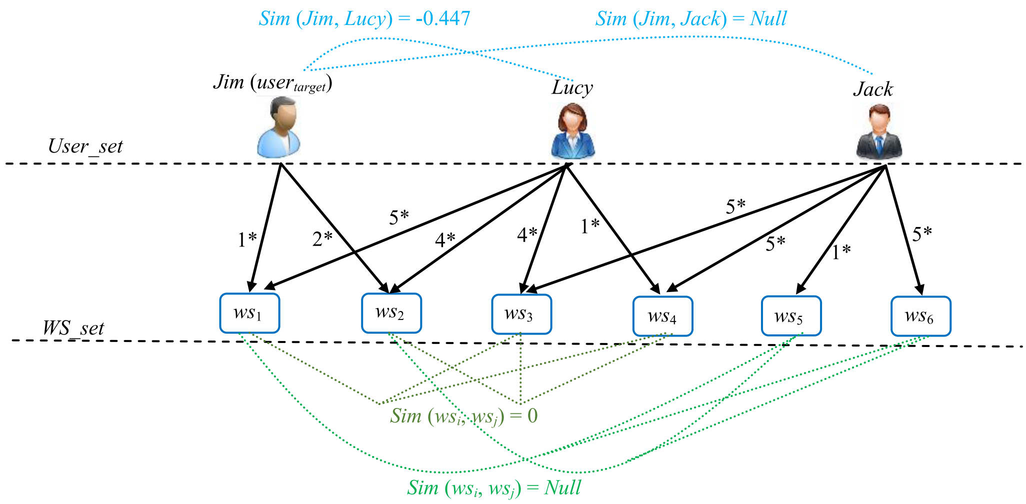


FIGURE 1. A cold-start service recommendation scenario where traditional CF approaches fail.

process in edge environment. More concretely, for a target user, we first search for his/her opposite users (together referred to as "enemy" hereafter i.e., the users who hold completely opposite preferences with the target user. This is what "inverse" means in our proposal); then we infer the target user's friends indirectly (referred to as "indirect friends" hereafter) based on the inference rules of Social Balance Theory (SBT)[16]. Finally, we predict the missing quality data of candidate services based on the derived "indirect friends" of the target user and then recommend the quality-optimal services to the target user. This is the main idea of our proposal.

The remainder of paper is structured as below. Section 2 first formulates the traditional MCF-based recommendation problems and then provides an intuitive scenario to motivate our paper. Section 3 specifies the proposed recommendation method, i.e., *Inverse_CF_Rec*. Experiments are conducted in Section 4 based on a real-world dataset, i.e., *WS-DREAM*, to validate the effectiveness and efficiency of our proposal. Related work and further discussions are presented in Section 5. Section 6 concludes the paper.

II. PROBLEM FORMULATION AND MOTIVATION

Next, we first formulate the MCF-based service recommendation problems; then we provide a motivating example to ease the understanding of readers.

A. PROBLEM FORMULATION

In this paper, a MCFbased service recommendation problem is formulated by $MCF\_SerRec(User\_set, WS\_set, Rating\_set, user_{target})$, where

(1) $User\_set = \{user_1, \ldots, user_m\}$: the set of users where m is the cardinality of $User\_set$

(2) $WS\_set = \{ws_1, \ldots, ws_n\}$: the set of web services where n is the cardinality of WS_set

(3) $Rating\_set = \{r_{i-j} | \ 1 \leq i \leq m, 1 \leq j \leq n\}$: historical user-service quality matrix, where r_{i-j} means $user_i$ ($\in User\_set$)'s rating on ws_j($\in WS\_set$). Concretely, we adopt the well-known $1^* \sim 5^*$ rating scale.

(4) $user_{target}$: the target user to whom a recommender system intends to recommend services. In this paper, $user_{target} \in User\_set$

Thus MCF recommendation problems is clarified as below: through analyzing the user-service rating data in *Rating_set*, the recommender system selects appropriate services from the candidates in *WS_set* and recommend them to $user_{target}$($\in User\_set$).

B. RESEARCH MOTIVATION

In this subsection, an intuitive scenario (shown in Fig.1) is presented to motivate this paper. In the figure, $User\_set = \{Jim, Lucy, Jack\}$ ($user_{target}$ is *Jim*), $WS\_set = \{ws_1, \ldots, ws_6\}$; the historical user-service quality data (denoted by $1^* \sim 5^*$) are also presented; ws_1 and ws_2 are referred to as "target services".

Then with the known rating data in Fig.1, we can calculate the similarity between *Jim*and $user_i$ (i.e., *Lucy* and *Jack*), denoted by $Sim(Jim, user_i)$ based on the Pearson Correlation Coefficient (i.e., domain-independent PCC) [17]. After calculation, we can get $Sim(Jim, Lucy) = -0.447$ and $Sim(Jim, Jack) = Null$. In this situation, no similar friends are present for the target user *Jim* and hence, traditional user-based CF methods cannot generate any recommended result and hence recommendation is failed.

Likewise, we can calculate the similarity between target services (i.e., ws_1 and ws_2) and other services (i.e., ws_3, ws_4, ws_5, ws_6). After calculation, we can get $Sim(ws_1, ws_3) = Sim(ws_1, ws_4) = Sim(ws_2, ws_3) = Sim(ws_2, ws_4) = 0$, while $Sim(ws_1, ws_5) = Sim(ws_1, ws_6) = Sim(ws_2, ws_5) = Sim(ws_2, ws_6) = Null$. In this situation, no similar services are present for the target services (i.e., ws_1 and ws_2); therefore, traditional item-based CF methods cannot generate any recommended result.

In terms of the above calculation, a conclusion can be drawn that the traditional MCF methods probably cannot produce a satisfying recommended list, when the available user-service quality data for recommendation decision-makings are very sparse (i.e., cold-start recommendation). Considering this drawback, we propose an intelligent recommendation method based on the inverse CF technique, i.e., *Inverse_CF_Rec* to improve the cold-start recommendation problems.

III. INVERSE CF-BASED RECOMMENDATION

Next, a novel recommendation method named *Inverse_CF_Rec* is suggested to deal with the cold-start problems in recommendation. Concretely, subsection A details the concrete algorithm of *Inverse_CF_Rec* method and subsection B analyzes the time complexity of *Inverse_CF_Rec* method

A. AN INTELLIGENT SERVICE RECOMMENDATION METHOD: Inverse_CF_Rec

The basic process of our proposal is: for a target user $user_{target}$, the recommender system first searches for his/her opposite users (i.e., "enemies") and then infers the "indirect friends" of $user_{target}$ based on the inference rules of Social Balance Theory (e.g., "enemy's enemy is a possible friend" rule, "enemy's friend is a possible enemy" rule); afterwards, the services preferred by the "indirect friends" of $user_{target}$ are recommended to $user_{target}$. More concretely, the *Inverse_CF_Rec* method mainly contains the following three steps (see Fig.2).

Step-1: Creating user-similarity tables offline. For each $user_i$ in *User_set*, calculate his/her similarity $Sim(user_i, user_j)$ with other users $user_j$ in *User_set* offline, and record $Sim(user_i, user_j)$ with a user-similarity table $Table_i$.

Step-2: Determining the indirect friends of $user_{target}$ through SBT. Search for qualified enemies of $user_{target}$ based on $user_{target}$'s user-similarity table $Table_{target}$ created in Step-1 and a predefined similarity threshold P. Afterwards, according to derived enemies of $user_{target}$ and Social Balance Theory, determine the "indirect friend" set of $user_{target}$, denoted by *Indirect_friend* ($user_{target}$).

Step-3: Service recommendation. Predict missing service quality data based on the known quality data of users in *Indirect_friend* ($user_{target}$); select the quality-optimal Top-3 candidate services and recommend them to $user_{target}$.

FIGURE 2. Three steps of Inverse_CF_Rec.

Step-1: Creating user-similarity tables offline.

Next, the user similarity values between any two entries in *User_set* are calculated. For any $user_i$ and $user_j$ in *User_set* ($user_i \neq user_j$), their similarity $Sim(user_i, user_j)$ could be calculated by the PCC formula in (1) offline.

In (1), set I denotes the service intersection rated by both $user_i$ and $user_j$; r_{i-k} and r_{j-k} represent $user_i$'s and $user_j$'s ratings on web service ws_k, respectively; $\overline{r_i}$ and $\overline{r_j}$ denote $user_i$'s and $user_j$'s average rating values, respectively. Specifically, if $user_i$ and $user_j$ have not invoked any common service before (i.e., set $I = Null$), then their similarity $Sim(user_i, user_j) = Null$ holds. As equation (1) indicates, $Sim(user_i, user_j) \in [-1, 1]$. Furthermore, the larger $Sim(user_i, user_j)$ is, the more probable that $user_i$ and $user_j$ are similar friends; on the contrary, the smaller $Sim(user_i, user_j)$ is, the more probable that $user_i$ and $user_j$ are opposite users, i.e., enemies.

$$Sim(user_i, user_j) = \frac{\sum_{ws_k \in I} (r_{i-k} - \overline{r_i}) * (r_{j-k} - \overline{r_j})}{\sqrt{\sum_{ws_k \in I} (r_{i-k} - \overline{r_i})^2} * \sqrt{\sum_{ws_k \in I} (r_{j-k} - \overline{r_j})^2}} \quad (1)$$

Next, for each $user_i \in User\_set$, we can create a user-similarity table $Table_i$ based on his/her similarity values $Sim(user_i, user_j)$ (obtained offline) with other users $user_j$ in *User_set*. Furthermore, to ease the subsequent similarity comparisons, we rank $Table_i$ offline by $Sim(user_i, user_j)$ in ascending order.

Step-2: Determining the indirect friends of $user_{\mathbf{target}}$ *through SBT.*

After Step-1, a user-similarity table of $user_{target}$ (denoted by $Table_{target}$) is achieved. Next, according to $Table_{target}$ and a pre-defined similarity threshold P ($-1 \leq P \leq -0.5$), we determine $user_{target}$'s enemy set (denoted by $Enemy\_set(user_{target})$) based on formula in (2). As the similarities $Sim(user_{target}, user_j)$ have already been calculated and recorded in $Table_{target}$ offline, the calculation of $Enemy\_set(user_{target})$ can be finished quickly.

$$Enemy\_set(user_{target}) = \{user_j | Sim(user_{target}, user_j) \leq P\} \quad (2)$$

Likewise, for any $user_j \in Enemy\_set(user_{target})$, we can determine his/her opposite enemies $user_k$ (denoted by set $Enemy\_set(user_j)$) by (2). Next, through the "enemy's enemy is a possible friend" rule of SBT, we can guess $user_k$ is probably a friend of $user_{target}$; furthermore, its credibility can be calculated by $Credibility_{friend}(user_{target}, user_k)$ in (3). Furthermore, if condition in (4) holds, $user_k$ can be taken as an "indirect friend" of $user_{target}$ and put into indirect friend set, i.e., $Indirect\_friend(user_{target})$. Here, $-P$ ($-1 \leq P \leq -0.5$) represents the threshold for friend similarity.

$$Credibility_{friend}(user_{target}, user_k) = Sim(user_{target}, user_j) * Sim(user_j, user_k) \quad (3)$$

$$Credibility_{friend}(user_{target}, user_k) \geq -P \quad (4)$$

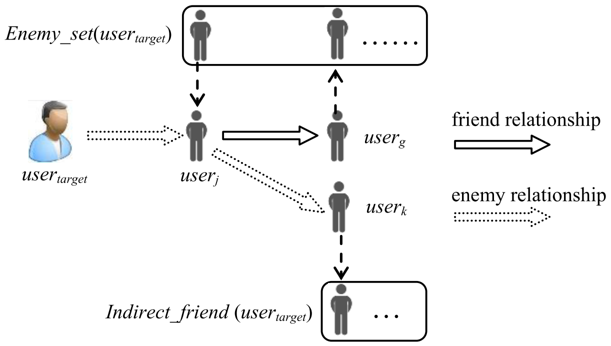


FIGURE 3. Relationships between $user_{target}$, $user_j$, $user_k$ and $user_g$.

Similarly, for any $user_j \in Enemy\_set(user_{target})$, we can determine his/her similar friends $user_g$ (denoted by set $Friend\_set(user_j)$) by (5). Then through the "enemy's friend is a possible enemy" rule of SBT, we can guess $user_g$ is an enemy of $user_{target}$; furthermore, its credibility can be calculated by $Credibility_{enemy}(user_{target}, user_g)$ in (6). Furthermore, if condition in (7) holds, then $user_g$ could be regarded as an "indirect enemy" of $user_{target}$ and put into enemy set, i.e., $Enemy\_set(user_{target})$. More intuitively, the relationships among $user_{target}$, $user_j$, $user_k$ and $user_g$ are presented in Fig.3.

$$Friend\_set(user_j) = \{user_g | Sim(user_j, user_g) \geq -P\} \tag{5}$$

$$Credibility_{enemy}(user_{target}, user_g) = Sim(user_{target}, user_j) * Sim(user_j, user_g) \tag{6}$$

$$Credibility_{enemy}(user_{target}, user_g) \leq P \tag{7}$$

Repeat the above inference process of "indirect friends" of the target user at most six times (considering the "six-degrees separation" in social network area), until the set $Indirect\_friend(user_{target})$ stays stable. Then all the users in set $Indirect\_friend(user_{target})$ are regarded as the "indirect friends" of $user_{target}$.

Step-3: Service recommendation

In Step-2, the "indirect friend" set of $user_{target}$, i.e., $Indirect\_friend$ $(user_{target})$ has been derived. Next, we recommend appropriate services to $user_{target}$ based on set $Indirect\_friend$ $(user_{target})$. Concretely, we predict the missing quality data of service *ws* over dimension *q* by $user_{target}$, denoted by $ws.q_{target}$, by the equation in (8), as shown at the bottom of this page, where $ws.q_k$ denotes the service quality of *ws* over *q* by $user_k$. Finally, the Top-3 services with the optimal predicted quality values are recommended to $user_{target}$.

Thus through Step 1 ~ Step3, a set of optimal services (at most three services) are selected and recommended to $user_{target}$, so as to solve the cold-start recommendation problems. The following pseudocode specifies the details of our proposal more formally.

B. COMPLEXITY ANALYSES

Next, the time complexity of *Inverse_CF_Rec* method is analyzed Assume the service quality matrix for recommendation contains *m* users and *n* services

Step-1: Creating user-similarity tables offline

As user-similarity tables could be created offline, the complexity is O(1).

Step2: Determining the indirect friends of $user_{\mathbf{target}}$ *through SBT*

For a target user, he/she owns *m*-1 enemy users at most; while for each enemy of the target user, $m-1$ enemies or $m-1$ friends are present at most. Therefore, to determine all the indirect friends of the target user, the time complexity is $O(m^2)$ in the worst case.

Step-3: Service recommendation.

In Step-2, we have obtained *m*-1 indirect friends of $user_{target}$ at most. While for each service (at most *n* services) never invoked by $user_{target}$, its quality should be predicted according to the obtained *m* indirect friends of $user_{target}$. Therefore, the time complexity of this step is $O(m^*n)$.

In terms of the above analyses, a conclusion can be drawn that the complexity of *Inverse_CF_Rec* method is $O(m^*n(m+n))$.

IV. EXPERIMENTS

A wide range of experiments are conducted to validate the feasibility of *Inverse_CF_Rec* method in terms of recommendation accuracy, time cost and failure rate

A. EXPERIMENT SETTINGS

The experiments are deployed on the popular dataset *WS-DREAM* [7] (including quality values of *throughput* dimension over 5825 services by 339 users). Partial quality entries are randomly removed from the dataset to simulate the cold-start problems in service recommendation; concretely, the density of recruited experiment data is denoted by parameter *r* (namely, the rest (1 - *r*)*100% data are removed from *WS-DREAM*).

Service performances can influence the satisfaction degree of users significantly [18]–[25]; therefore, we test the method performances from three perspectives: *MAE* (the smaller the better), *time cost* and *failure rate*. Besides, in order to prove the advantages of our proposal, *Inverse_CF_Rec* method

$$ws.q_{target} = \frac{\sum\limits_{user_k \in Indirect_{friend}(user_{target})} Credibility_{friend}(user_{target}, user_k)^* ws.q_k}{\sum\limits_{user_k \in Indirect_{friend}(user_{target})} Credibility_{friend}(user_{target}, user_k)} \tag{8}$$

Algorithm 1 Inverse_CF_Rec

Input:

(1) $User\_set = \{user_1, \ldots, user_m\}$: user set;

(2) $WS\_set = \{ws_1, \ldots, ws_n\}$: web service set;

(3) $Rating\_set = \{r_{i-j} | 1 \leq i \leq m, 1 \leq j \leq n\}$: user-service rating set;

(4) $user_{target}$: a target user.

Output:

$Result\_set$: web service set recommended to $user_{target}$

1: Set enemy similarity threshold, i.e., $P(-1 \leq P \leq -0.5)$
2: $Indirect\_friend(user_{target}) = \Phi$
3: $Result\_set = \Phi$
4: **for** each $user_i \in User\_set$ **do** // Step-1
5:   **for** each $user_j \in User\_set$ **do**
6:     calculate $Sim(user_i, user_j)$ by (1) offline
7:   **end for**
8:   create user-similarity table $Table_i$ offline
9: **end for**
10: Infer $Enemy\_set(user_{target})$ by $Table_{target}$ and (2) // Step-2
11: **for** each $user_j \in Enemy\_set(user_{target})$ **do**
12:   determine his/her enemies $user_k$ by (2)
13:   calculate $Credibility_{friend}(user_{target}, user_k)$ by (3)
14:   **if** condition in (4) holds
15:     **then** put $user_k$ into set $Indirect\_friend(user_{target})$
16:   **end if**
17:   determine his/her friends $user_g$ by (5)
18:   calculate $Credibility_{enemy}(user_{target}, user_g)$ by (6)
19:   **if** condition in (7) holds
20:     **then** put $user_g$ into set $Enemy\_set(user_{target})$
21:   **end if**
22: **end for**
23: repeat lines 10 $\sim$ 21 at most six times until $Indirect\_friend(user_{target})$ stays stable
24: **for** each $ws_j \in WS\_set$ **do**// Step-3
25:     **if** ws_j is never invoked by $user_{target}$
26:     **then** calculate $ws_j.q_{target}$ based on $Indirect\_friend(user_{target})$ and (8)
27:     **end if**
28: **end for**
29: put Top-3 services with the highest predicted quality into $Result\_set$
30: **return** $Result\_set$ to $user_{target}$

is compared with another three ones, i.e., *SBT-SR* [26], *WSRec* [27] *DHRF* [28].

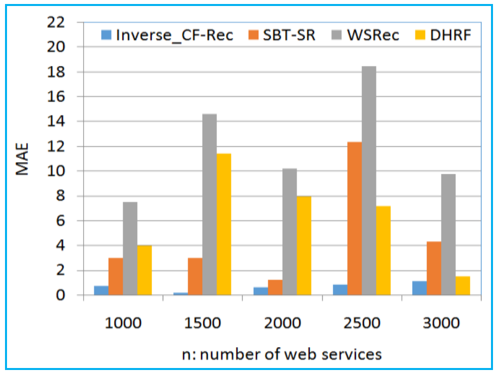
The experiment running environment is: 2.40 GHz CPU and 12.0 GB RAM. Software configurations are Windows 10 + JAVA 8 + MySQL 5.7. Each experiment was carried out ten times and their average values are reported.

B. EXPERIMENT RESULTS

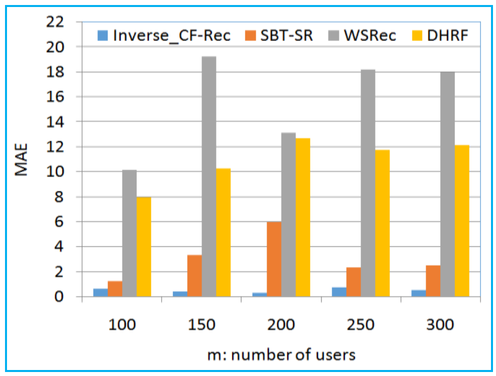
Concretely, four profiles are tested and compared. Here, m and n represent the cardinality of $User\_set$ and WS_set, respectively; r denotes the density of recruited quality matrix; P represents the threshold for enemy similarity.

Profile-1: Recommendation accuracy comparison

In this profile, we compare the recommendation accuracy values of four approaches through *MAE*. Concrete parameters are set as below: $m = 100, 150, 200, 250, 300$; n is varied from 1000 to 3000; $P = -0.6$; the user-service matrix density, i.e., $r = 5\%$. The results are presented in Fig.4.



(a)



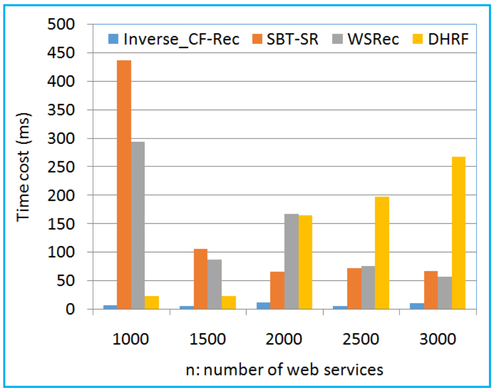
(b)

FIGURE 4. Recommendation accuracy w.r.t. *m* and *n*. (a) *m* = 100. (b) *n* = 2000.

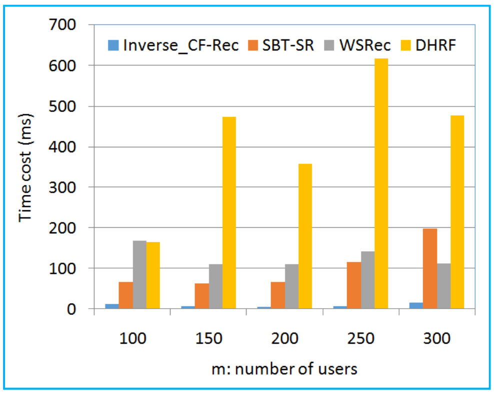
Fig.4 shows that the recommendation accuracy of *WSRec* method is the lowest (i.e., *MAE* is the largest); this is because *WSRec* method only considers $user_{target}$'s average quality and the target services' average quality, while neglecting some crucial recommendation information that can be extracted from the quality matrix. The accuracy of *DHRF* method is either not high as this method only considers the "most popular" services. While in *SBT-SR* method, only the "enemy's enemy is a possible friend" rule is employed for service recommendation; while other valuable social relationship information is overlooked and hence, the recommendation accuracy is decreased. In our proposed *Inverse_CF_Rec* approach, both the "enemy's enemy is a possible friend" rule and the "enemy's friend is a possible enemy" rule of Social Balance Theory are considered in recommendation process; therefore, the accuracy value of the derived recommended results is improved significantly and outperforms those of the rest three methods

Profile- : Recommendation efficiency comparison

Next, we compare the time cost of different recommendation method. Concrete parameter values are as below: $m =$ 100, 150, 200, 250, 300; $n =$ 1000, 1500, 2000, 2500, 3000; $P = -0.6$ and $r = 5\%$. Concrete experiment results are presented in Fig.5.



(a)



(b)

FIGURE 5. Recommendation efficiency w.r.t. *m* and *n*. (a) $m = 100$. (b) $n = 2000$.

Fig.5 shows that the efficiency of *DHRF* method is often low as much time is consumed for finding the "most popular" web services. For *WSRec* approach, much time is taken to calculate the average quality of services invoked by the target user and the average quality of the target service invoked by different users, and hence, the recommendation efficiency is not high enough. For *SBT-SR* approach, online calculation of user similarity is necessary, which brings more time cost. While in the suggested *Inverse_CF_Rec* method, most jobs (e.g., creation of user-similarity tables) is finished offline; therefore, the time cost is reduced significantly. Besides, "six-degrees separation" in social networks is considered in our proposal; therefore, the iteration times for inferring the indirect friends of $user_{target}$ are reduced, which also improves the recommendation efficiency.

Profile-3: Failure rate of Inverse_CF_Rec w.r.t. r

Social Balance Theory is an experienced theory in psychological domain. Therefore, we cannot always guarantee the applicability of SBT in service recommendation applications. In other words, *Inverse_CF_Rec* method may fail in outputting any result if the available user-service quality data are very sparse. Next, we test the failure rate of *Inverse_CF_Rec* with parameter r. Concrete parameter values are as below: $m = 100$, $n = 1000$, $P = -0.6$; r is varied from 1% to 5%. Concrete experiment results are presented in Fig.6.

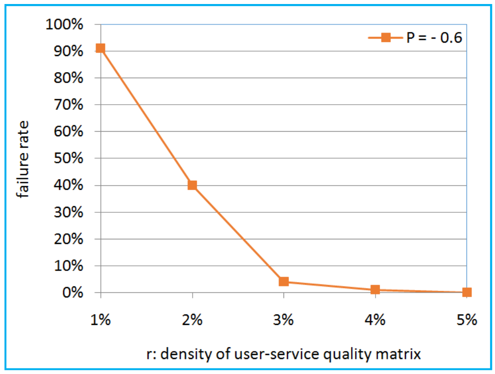


FIGURE 6. Failure rate of Inverse_CF_Rec w.r.t. *r* ($P = X - 0.6$).

Fig.6 show that the failure rate of *Inverse_CF_Rec* approach is larger than 90% when r is smaller than 1%; besides, failure rate decreases significantly when r rises. This is due to the fact that a larger r often indicates a denser user-service quality matrix as well as more valuable recommendation information. Therefore, when r grows, the failure rate decreases correspondingly. As can be seen from Fig.6, when the matrix density, i.e., r reaches 5%, the *Inverse_CF_Rec* method seldom fails.

Profile- 4: Failure rate of Inverse_CF_Rec w.r.t. P

The threshold for enemy similarity, i.e., P is crucial for the success of the *Inverse_CF_Rec* method. Thus, we investigate the correlation between the failure rate of our proposal and parameter P. Concrete parameters are set as below: $m = 100, n = 1000$, matrix density, i.e., $r = 2\%$, $P = -1, -0.9, -0.8, -0.7, -0.6$. Concrete results are shown in Fig.7.

Fig.7 indicates that the failure rate of *Inverse_CF_Rec* method decreases approximately when P grows. The reason is that when P is larger (for example, $P = -0.6$), filtering condition for enemy relationship becomes looser and hence, more qualified "enemies" of $user_{target}$ can be returned and recruited for recommendation decision-makings. Therefore, the failure rate is decreased accordingly.

C. FURTHER DISCUSSIONS

However, there are still several shortcomings in our proposed *Inverse_CF_Rec* method. First, only one quality dimension is regarded as the recommendation basis, i.e., users' ratings on services; while actually, multiple dimensions are more common in the actual decision-making problems [29]–[35]. In view of this observation, we will improve our method by introducing multiple decision-making dimensions as well as the weighting mechanism in multi-dimensional

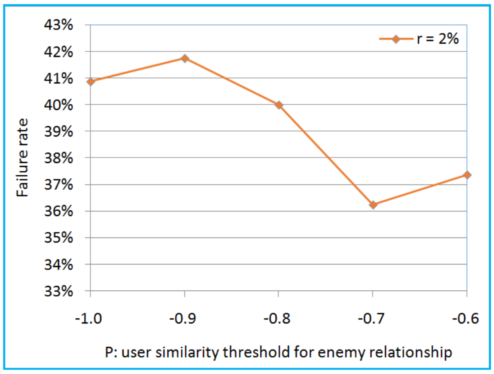


FIGURE 7. Failure rate of Inverse_CF_Rec w.r.t. P ($r = 2\%$).

scenarios [36]–[39]. Second, we only consider one data type of user rating data, without considering the diversity of rating data types such as discrete data [40]–[45], binary data [46] and fuzzy data [47]–[49]. Considering this, we will further modify *Inverse_CF_Rec* method to make it compatible with different data types of ratings.

V. RELATED WORK

Due to the easy-to-explain and domain-independent characteristics, CF technique is widely deployed in existing recommender systems. A dependable recommendation method is proposed in [50], where the traditional user-based CF technique is used for recommendation decision-makings. *SD-HCF* method is proposed in [51] where user-based and item-based CF techniques are combined to pursue a high-quality recommended list. Other similar hybrid recommendation approaches can be found in [50]. While these methods do not consider other key decision-making information, e.g., location data. Considering this drawback, user location and service location are considered in [52], in which the nearby services are recommended to the target user first.

The abovementioned recommendation methods often perform well when similar friends or similar services are present in the user-service rating network. However, when neither similar users nor similar services exist, a cold-start recommendation problem occurs. To cope with this cold-start problem, *WSRec* method is proposed in [27], where both $r_{average}(user_{target})$ (i.e., $user_{target}$'s average rating value) and $r_{average}(ws_j)$ (i.e., ws_j's average rating value) are taken as the recommendation bases. While the accuracy of *WSRec* is not high enough due to the adopted simple "average" idea. A popularity-aware recommendation method named *DHRF* is proposed in [28]. In *DHRF*, the "most popular" web services are recruited for service quality prediction and subsequent service recommendation, so as to cope with the recommendation cold-start problems. However, as an approximate strategy (i.e., service popularity) is adopted in service quality prediction, the recommendation accuracy is not as high as expected. The work [53] proposes to utilize "belief propagation" to realize more efficient task allocation. Similar to [53], in our previous work [54], Social Balance Theory is introduced into service recommendation to remedy the cold-start problem.

In this paper, we continue to improve the work in [54] to further enhance the service recommendation performance. Concretely, our improvements are three-fold: (1) "Six Degrees of Separation" in social networks domain is adopted to reduce the iteration times in finding the "indirect friends" of u_{target}, through which accuracy and time cost are improved significantly; (2) we create a set of user-similarity tables offline to record the similarity values between different users before a recommendation request arrives, so that the efficiency of online friend search is improved considerably; (3) we evaluate each recommended service quantitatively so that only the optimal service with the largest "recommended degree" is returned to the target user; this way, the recommendation accuracy is improved. Finally, through a set of experiments conducted on a real-world dataset, i.e., *WS-DREAM*, we prove the advantages of proposed *Inverse_CF_Rec* method in terms of recommendation accuracy and efficiency.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

An intelligent service recommendation method named *Inverse_CF_Rec* is proposed in this paper, to cope with the cold-start recommendation problems in edge environment with sparse data. First, we search for the opposite users (i.e., enemies) of $user_{target}$; second, we search for the friends of $user_{target}$ through the indirect friend inference rule of Social Balance Theory; third, appropriate services are recommended to $user_{target}$ based on the derived friends of $user_{target}$. Finally, we demonstrate the advantages of the proposed *Inverse_CF_Rec* method by measuring the recommendation accuracy and efficiency through extensive experiments conducted on the real-world dataset *WS-DREAM*.

As service performances often vary with service running context [55], [56], in the future, we will investigate how to improve our proposed recommendation method by taking into consideration more context factors (e.g., user location and service invocation time).

REFERENCES

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

[2] H. Gao, D. Chu, Y. Duan, and Y. Yin, "Probabilistic model checking-based service selection method for business process modeling," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 27, no. 6, pp. 897–923, Feb. 2017.

[3] X. Wang, L. T. Wang, X. Xie, J. Jin, and M. J. Deen, "A cloud-edge computing framework for cyber-physical-social services," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 80–85, Nov. 2017.

[4] X. Wang, L. T. Yang, J. Feng, X. Chen, and M. J. Deen, "A tensor-based big service framework for enhanced living environments," *IEEE Cloud Comput.*, vol. 3, no. 6, pp. 36–43, Nov./Dec. 2016.

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

[6] H. Gao, Y. Duan, H. Miao, and Y. Yin, "An approach to data consistency checking for the dynamic replacement of service process," *IEEE Access*, vol. 5, pp. 11700–11711, 2017.

[7] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware Web service recommendation by collaborative filtering," *IEEE Trans. Services Comput.*, vol. 4, no. 2, pp. 140–152, Apr./Jun. 2011.

[8] L. Qi, X. Zhang, W. Dou, and Q. Ni, "A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2616–2624, Nov. 2017.

[9] L. Yao, Q. Z. Sheng, A. H. Ngu, J. Yu, and A. Segev, "Unified collaborative and content-based Web service recommendation," *IEEE Trans. Services Comput.*, vol. 8, no. 3, pp. 453–466, May 2015.

[10] C. Yan, X. Cui, L. Qi, X. Xu, and X. Zhang, "Privacy-aware data publishing and integration for collaborative service recommendation," *IEEE Access*, vol. 6, pp. 43021–43028, 2018. doi: 10.1109/ACCESS.2018.2863050.

[11] W. Gong, L. Qi, and Y. Xu, "Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment," *Wireless Commun. Mobile Comput.*, vol. 2018, Mar. 2018, Art. no. 4680867.

[12] Y. Yin, L. Chen, Y. Xu, and J. Wan, "Location-aware service recommendation with enhanced probabilistic matrix factorization," *IEEE Access*, vol. 6, pp. 62815–62825, 2018.

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

[14] Y. Yin, F. Yu, Y. Xu, L. Yu, and J. Mu, "Network location-aware service recommendation with random walk in cyber-physical systems," *Sensors*, vol. 17, no. 9, p. 2059, 2017.

[15] L. Qi, Z. Zhou, J. Yu, and Q. Liu, "Data-sparsity tolerant web service recommendation approach based on improved collaborative filtering," *IEICE Trans. Inf. Syst.*, vol. E100D, no. 9, pp. 2092–2099, 2017.

[16] D. Cartwright and F. Harary, "Structural balance: A generalization of Heider's theory," *Psychol. Rev.*, vol. 63, no. 5, p. 277, Sep. 1956.

[17] J. L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," *Amer. Statistician*, vol. 42, no. 1, pp. 59–66, Feb. 1988.

[18] H. Gao, H. Miao, L. Liu, J. Kai, and K. Zhao, "Automated quantitative verification for service-based system design: A visualization transform tool perspective," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 28, no. 10, pp. 1369–1397, Jun. 2018.

[19] S. Deng, Z. Xiang, J. Yin, J. Taheri, and A. Y. Zomaya, "Composition-driven IoT service provisioning in distributed edges," *IEEE Access*, vol. 6, pp. 54258–54269, 2018.

[20] H. Gao, K. Zhang, J. Yang, F. Wu, and H. Liu, "Applying improved particle swarm optimization for dynamic service composition focusing on quality of service evaluations under hybrid networks," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 2, May 2018, Art. no. 1550147718761583.

[21] C. Zhang, H. Zhao, and S. Deng, "A density-based offloading strategy for IoT devices in edge computing systems," *IEEE Access*, vol. 6, pp. 73520–73530, 2018.

[22] H. Gao, W. Huang, X. Yang, Y. Duan, and Y. Yin, "Toward service selection for workflow reconfiguration: An interface-based computing solution," *Future Gener. Comput. Syst.*, vol. 28, pp. 298–311, Oct. 2018.

[23] Y. Yin, W. Xu, Y. Xu, H. Li, and L. Yu, "Collaborative qos prediction for mobile service with data filtering and SlopeOne model," *Mobile Inf. Syst.*, vol. 2017, May 2017, Art. no. 7356213.

[24] J. Yu, Z. Kuang, B. Zhang, W. Zhang, D. Lin, and J. Fan, "Leveraging content sensitiveness and user trustworthiness to recommend fine-grained privacy settings for social image sharing," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1317–1332, May 2018. doi: 10.1109/TIFS.2017.2787986.2018.

[25] W. Li, Y. Xia, M. Zhou, X. Sun, and Q. Zhu, "Fluctuation-aware and predictive workflow scheduling in cost-effective infrastructure-as-a-service clouds," *IEEE Access*, vol. 6, pp. 61488–61502, 2018.

[26] L. Qi, X. Zhang, Y. Wen, and Y. Zhou, "A social balance theory-based service recommendation approach," in *Proc. Asia–Pacific Services Comput. Conf.*, Dec. 2015, pp. 48–60.

[27] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Wsrec: A collaborative filtering based web service recommender system," in *Proc. IEEE Int. Conf. Web Services*, Jul. 2009, pp. 437–444.

[28] J. H. Wang and Y. H. Chen, "A distributed hybrid recommendation framework to address the new-user cold-start problem," in *Proc. 15th Intl Conf Scalable Comput. Commun. Associated Workshops (UIC-ATC-ScalCom)*, Aug. 2015, pp. 1686–1691.

[29] M. Wang and G.-L. Tian, "Robust group non-convex estimations for high-dimensional partially linear models," *J. Nonparam. Statist.*, vol. 28, no. 1, pp. 49–67, Oct. 2016.

[30] X. Wang and M. Wang, "Variable selection for high-dimensional generalized linear models with the weighted elastic-net procedure," *J. Appl. Statist.*, vol. 43, no. 5, pp. 796–809, Jul. 2016.

[31] P. Wang and L. Zhao, "Some geometrical properties of convex level sets of minimal graph on 2-dimensional riemannian manifolds," *Nonlinear Anal.*, vol. 130, pp. 1–17, Jan. 2016.

[32] X. Wang and M. Wang, "Adaptive group bridge estimation for high-dimensional partially linear models," *J. Inequal. Appl.*, vol. 2017, p. 158, Jun. 2017.

[33] X. Wang, S. Zhao, and M. Wang, "Restricted profile estimation for partially linear models with large-dimensional covariates," *Statist. Probab. Lett.*, vol. 128, pp. 71–76, Sep. 2017.

[34] H. Tian and M. Han, "Bifurcation of periodic orbits by perturbing high-dimensional piecewise smooth integrable systems," *J. Differ. Equ.*, vol. 263, pp. 7448–7474, Dec. 2017.

[35] P. Wang and X. Wang, "The geometric properties of harmonic function on 2-dimensional riemannian manifolds," *Nonlinear Anal., Theory, Methods Appl.*, vol. 103, pp. 2–8, Jul. 2014.

[36] S. Yang, Z. Yao, and C. Zhao, "The weight distributions of two classes of p-ary cyclic codes with few weights," *Finite Fields Their Appl.*, vol. 44, pp. 76–91, Mar. 2017.

[37] Y.-F. Wang, C.-C. Yin, and X.-S. Zhang, "Uniform estimate for the tail probabilities of randomly weighted sums," *Acta Mathematicae Applicatae Sinica*, vol. 30, no. 4, pp. 1063–1072, Oct. 2014.

[38] S. Yang, Z. Yao, and C. Zhao, "A class of three-weight linear codes and their complete weight enumerators," *Cryptogr. Commun.*, vol. 9, no. 1, pp. 133–149, Jan. 2017.

[39] J. Cai, "An implicit sigma (3) type condition for heavy cycles in weighted graphs," *ARS Combinatoria-Waterloo Winnipeg*, vol. 115, pp. 211–218, Apr. 2008.

[40] Q. Feng, F. Meng, and Y. Zhan, "Generalized Gronwall-Bellman-type discrete inequalities and their applications," *J. Inequal. Appl.*, vol. 2011, p. 47, Dec. 2011.

[41] P. Li, "Two classes of linear equations of discrete convolution type with harmonic singular operators," *Complex Variables Elliptic Equ.*, vol. 61, no. 1, pp. 67–75, May 2016.

[42] Z. Zheng, "Invariance of deficiency indices under perturbation for discrete Hamiltonian systems," *J. Difference Equ. Appl.*, vol. 19, no. 8, pp. 1243–1250, Sep. 2013.

[43] M. Han, X. Hou, L. Sheng, and C. Wang, "Theory of rotated equations and applications to a population model," *Discrete Continuous Dyn. Syst.-A*, vol. 38, no. 4, pp. 2171–2185, Apr. 2018.

[44] J. Cai and H. Li, "A new sufficient condition for pancyclability of graphs," *Discrete Appl. Math.*, vol. 162, pp. 142–148, Jan. 2014.

[45] L. L. Liu and B. Zhu, "Strong q-log-convexity of the Eulerian polynomials of coxeter groups," *Discrete Math.*, vol. 338, no. 12, pp. 2332–2340, Dec. 2015.

[46] B. Zhang, "Remarks on the maximum gap in binary cyclotomic polynomials," *Bull. Math. Soc. Sci. Math. Roumanie*, vol. 59, no. 1, pp. 109–115, 2016.

[47] L. Wang, "Intuitionistic fuzzy stability of a quadratic functional equation," *Fixed Point Theory Appl.*, vol. 2010, Dec. 2010, Art. no. 107182.

[48] X. Du and Z. Zhao, "On fixed point theorems of mixed monotone operators," *Fixed Point Theory Appl.*, vol. 2011, Dec. 2011, Art. no. 563136.

[49] B. Zhu, L. Liu, and Y. Wu, "Local and global existence of mild solutions for a class of nonlinear fractional reaction–diffusion equations with delay," *Appl. Math. Lett.*, vol. 61, pp. 73–79, Nov. 2016.

[50] S.-Y. Lin, C.-H. Lai, C.-H. Wu, and C.-C. Lo, "A trustworthy QoS-based collaborative filtering approach for web service discovery," *J. Syst. Softw.*, vol. 93, pp. 217–228, Jul. 2014.

[51] J. Cao, Z. Wu, Y. Wang, and Y. Zhuang, "Hybrid collaborative filtering algorithm for bidirectional web service recommendation," *Knowl. Inf. Syst.*, vol. 36, no. 3, pp. 607–627, Sep. 2013.

[52] D. Jiang, X. Guo, Y. Gao, J. Liu, H. Li, and N. Cheng, "Locations recommendation based on check-in data from location-based social network," in *Proc. 22nd Int. Conf. Geoinform.*, Jun. 2014, pp. 1–4.

[53] Y. Kong, M. Zhang, and D. Ye, "A belief propagation-based method for task allocation in open and dynamic cloud environments," *Knowl.-Based Syst.*, vol. 115, pp. 123–132, Jan. 2016.

[54] L. Qi, W. Dou, and X. Zhang, "An inverse collaborative filtering approach for cold-start problem in web service recommendation," in *Proc. Australasian Comput. Sci. Week*, Jan. 2017, pp. 46–54.
[55] X. Wang, W. Wang, L. T. Yang, S. Liao, D. Yin, and M. J. Deen, "A distributed HOSVD method with its incremental computation for big data in cyber-physical-social systems," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 2, pp. 481–492, Jun. 2018.
[56] X. Wang, L. T. Yang, H. Liu, and M. J. Deen, "A big data-as-a-service framework: State-of-the-art and perspectives," *IEEE Trans. Big Data*, vol. 4, no. 3, pp. 325–340, Sep. 2018.

YICHAO ZHOU received the Ph.D. degree from the School of Computer Science and Engineering, Nanjing University of Science and Technology, China, in 2018. His research interests include signal and image processing, and services computing.

ZHENMIN TANG received the Ph.D. degree from the Department of Computer Science, Nanjing University of Science and Technology, China, in 2002, where he is currently a Full Professor with the School of Computer Science and Engineering. He has already published more than 100 papers in peer-reviewed journals and international conferences. His research interests include intelligent robot systems, and image processing and analysis.

LIANYONG QI received the Ph.D. degree from the Department of Computer Science and Technology, Nanjing University, China, in 2011. He is currently an Associate Professor with the School of Information Science and Engineering, Chinese Academy of Education Big Data, Qufu Normal University, China. He has already published over 40 papers, including JSAC, TCC, TBD, and TCSS. His research interests include recommender systems and services computing.

XUYUN ZHANG received the B.S. and M.E. degrees in computer science from Nanjing University, China, in 2008 and 2011, respectively, and the Ph.D. degree from the University of Technology Sydney (UTS), Australia, in 2014. He is currently a Lecturer with the Department of Electrical and Computer Engineering, University of Auckland, New Zealand. Prior to his current appointment, he worked as a Postdoctoral Fellow of the Machine Learning Research Group of NICTA (currently Data61, CSIRO), Australia. His primary research interests include the IoT and smart cities, big data, cloud computing, scalable machine learning and data mining, data privacy and security, and Web service technology.

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

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