Robust Collaborative Filtering Recommendation With User-Item-Trust Records

Fan Wang, *Graduate Student Member, IEEE*, Haibin Zhu, *Senior Member, IEEE*, Gautam Srivastava, *Senior Member, IEEE*, Shancang Li, Mohammad R. Khosravi, and Lianyong Qi

***Abstract*— The ever-increasing popularity of recommendation systems allows users to find appropriate services without excessive effort. However, due to the unstable and complex network environment, the historical behavior data of users are quite sparse in most cases. The inherent drawbacks render preference prediction infeasible for cold-start users and have become a crucial issue to be resolved in recommendation systems. To deal with the problems, we first present a Trust-based Collaborative Filtering (TbCF) algorithm to perform basic rating prediction in a manner consistent with the existing CF methods. Then, we propose the Hybrid Collaborative Filtering Recommendation approach with User-Item-Trust Records ($\mathrm{UIT_{hybrid}}$), a novel approach that incorporates user trust into the existing CF-based methods in a harmonious way to supplement rating information. $\mathrm{UIT_{hybrid}}$ employs multiple perspectives to extract proper services and achieves a good tradeoff between the robustness, accuracy, and diversity of the recommendation. We conduct extensive real-world experiments on the Epinions data set to demonstrate the feasibility and efficiency of $\mathrm{UIT_{hybrid}}$.**

***Index Terms*— Cold-start problems, collaborative filtering (CF), rating prediction, recommendation, user trust.**

I. Introduction

WITH the support of machine-to-machine technology over the internet [1], web services facilitate human life and achieve rapid development in service-oriented society today [2], [3]. On the other hand, the explosive growth in the number of web services due to increasingly sophisticated requirements of users makes the users feel too overwhelmed to manually find appropriate services quickly. In such a background, a recommendation system performs on-demand, which provides users with more personalized service options by predicting potential preferences according to their historical behaviors and activities [4]. As a widely used tool, a recommendation system usually first predicts a user's ratings of unemployed services and then generates a recommendation list by extracting services with the highest prediction rating.

Recently, a large number of researchers have devoted themselves to the study of recommendation systems to better serve users and achieve great progress [5]. In particular, collaborative filtering (CF) methods play a crucial role because of their high accuracy and easy-to-explain characteristics, including some classical methods, such as user-based methods (UbCF), item-based methods (IbCF), and model-based methods. All of the above methods are based on a user-item rating matrix constructed by statistically collecting users' historical rating logs, in which each entry reflects a user's actual rating value for the corresponding web service. Specifically, a user-item rating value of -1 or *null* indicates that the user has never rated the web service before. Considering the huge number of candidate services satisfying users' personalized needs, along with the fact that only a small portion of them can actually be invoked, the user-item rating matrix is often large but sparse [6]. Furthermore, the unstable and complex network environment makes it difficult for us to guarantee the reliability and integrity of the rating data collected from users' historical service usage. In this situation, the sparseness of the rating matrix is further aggravated.

The sparseness of rating data often renders the above-mentioned traditional CF methods infeasible in returning a set of high-quality recommended results to a target user. In worse cases, no results are returned; i.e., a cold-start problem occurs. The rating sparseness and the resulting cold-start problems significantly decrease the robustness of the recommender systems. In this situation, it is becoming a necessity to explore more valuable information hidden in historical user-service usage in addition to the similarity relationships utilized in traditional CF recommendation methods. Fortunately, social networks provide another perspective to observe users [7]–[10]. Specifically, user-user trust relationships offer a beneficial supplement to traditional CF techniques because they provide a new perspective to evaluate whether two users

This work was supported in part by the National Natural Science Foundation of China under Grant 61872219, in part by the Natural Science Foundation of Shandong Province under Grant ZR2019MF001, in part by the Open Project of State Key Laboratory for Novel Software Technology under Grant KFKT2020B08, and in part by the Fundamental Research Funds for the Central Universities under Grant 30919011282. *(Corresponding author: Lianyong Qi.)*

Fan Wang and Lianyong Qi are with the School of Computer Science, Qufu Normal University, Rizhao 276826, China (e-mail: fanwang1997@gmail.com; lianyongqi@qfnu.edu.cn).

Haibin Zhu is with the Department of Computer Science and Mathematics, Nipissing University, North Bay, ON P1B 8L7, Canada (e-mail: haibinz@nipissingu.ca).

Gautam Srivastava is with the Department of Mathematics and Computer Science, Brandon University, Brandon, MB R7A 6A9, Canada (e-mail: srivastavag@brandonu.ca).

Shancang Li is with the Department of Computer Science and Creative Technologies, University of the West of England, Bristol BS16 1QY, U.K. (e-mail: shancang.li@uwe.ac.uk).

Mohammad R. Khosravi is with the Department of Computer Engineering, Persian Gulf University, Bushehr 75169-13817, Iran, and also with the Department of Electrical and Electronic Engineering, Shiraz University of Technology, Shiraz 71555-313, Iran (e-mail: m.r.khosravi.taut@gmail.com).

Digital Object Identifier 10.1109/TCSS.2021.3064213

is that a user's preferences may be consistent with his/her trust users' preferences. Therefore, one promising way for improving the robustness of recommendation is to incorporate user–user trust (abbreviated as user trust) relationships into existing CF-based recommender systems.

However, existing trust-based CF researches generally utilize user trust only and fail to reasonably and sufficiently integrate user trust with the traditional CF methods to better alleviate the cold-start problem [11], [12]. One fundamental reason is that their handling styles for relationships are significantly different, which prevents them from being blended harmoniously. For example, in order to implement recommendations, Yao and Jiang [6] make use of user trust to perform a random walk model, while UbCF leverages computable user similarity to predict missing ratings. The distinct manners of dealing with relationships, i.e., user trust and similarity relationships, have become major obstacles to making full use of all available relationship information since we cannot integrate trust-based methods into the existing CF-based methods in a consistent manner.

In light of the above challenges, we put forward a new trust-based collaborative filtering (TbCF) recommendation algorithm inspired by [13], within which relationships are handled in a unified manner, as in the existing UbCF and IbCF methods. Moreover, according to the proposed TbCF, we develop a novel hybrid CF recommendation model with user-item-trust records, named $\text{UIT}_{\text{hybrid}}$. The new solution not only takes user trust into consideration but also achieves higher robustness while retaining the high accuracy of CF methods. In summary, our scientific contributions are threefold.

1) Considering the effective supplement of user trust to classical CF technologies, we propose a novel trust-based CF recommendation method. In contrast with the existing trust recommendation, our method works in a concordant manner with existing CF methods, laying the foundation for their integration and complementation.
2) We develop a new hybrid CF recommendation model that systematically incorporates the proposed trust-based method into existing CF and adjusts each kind of relationship by presented dependence weight.
3) We conduct extensive real-world experiments on the Epinions data set to validate the feasibility and effectiveness of $\text{UIT}_{\text{hybrid}}$. The experiment results demonstrate that our method achieves a good tradeoff among recommendation robustness, diversity, and accuracy, even for the cold-start problem.

We organize the remainder of this article as follows. We review the related work in Section II. Section III formulates the research problem and presents the motivation of our proposal. Section IV introduces how our $\text{UIT}_{\text{hybrid}}$ takes effect in an extremely sparse context. Section V evaluates $\text{UIT}_{\text{hybrid}}$ with a range of experiments. Section VI makes a conclusion and prospects the future work of this article.

II. Related Work

In this section, some recent work will be briefly reviewed on the recommendation system to lay the groundwork for our research. In the real world, the explosive growth of web services has become a major obstacle for users to select a suitable service from massive data. Some data preparation approaches are studied in [14], but they are not conducive to convenient user service selection. Xu *et al.* [15] apply the multiobjective offloading approach in cloud infrastructure to deal with the situation of service data overload, which also achieves a general effect. Naturally, the recommendation system is on-demand, as it is an effective and convenient solution to facilitate the life of users. Therefore, this field has fostered increasingly novel research and applications. For instance, Qi *et al.* [16] employed it for Web APIs recommendation, Liu *et al.* [17] implemented citation recommendation through link prediction, and Wang *et al.* [18] utilized a tensor method for routing recommendation.

Among a great deal of recommendation studies, CF plays a significant role and draws much attention. In particular, UbCF and IbCF are the most representative CF methods due to their characteristics of accuracy and ease of explanation [19]. Jiang *et al.* [20] integrate the two classical methods and propose a hybrid CF method, which achieves a higher quality of recommendation. Considering the personalized preferences of users in requesting analyses to enhance the user experience, Wang *et al.* [21] recruit the tensor-train decomposition method to infer interests for the target user. However, the above methods only investigate user-item information, instead of incorporating the invocation context. To overcome this issue, Qi *et al.* [22] employ location scenarios to supplement valuable information and produce more comprehensive recommendations. Because of the privacy concerns exposed in the big data environment [23]–[26], increasing researchers devote themselves to privacy-preserving approaches [27], [28]. For instance, Zhang *et al.* [29] can conduct privacy protection and recommendation in edge computing networks. These approaches provide the user's private information with strong security.

However, all the above CF approaches only take effect in scenarios with sufficient data information. In general, a sparse data set often makes it challenging for the existing CF methods to find appropriate services and even results in the infeasible recommendations. We refer to these situations as cold-start problems that have become the main obstacle to deepen applications of CF. In light of this shortcoming, a promising approach is to enclose social network information, e.g., user trust, to supplement our available information. Ma *et al.* [30] and Birtolo *et al.* [31] take social trust into account to make recommendations. However, the two methods do not perform well in solving the cold-start problems. Jamali and Ester [32] propose Trustwalker to leverage user trust for cold-start issues, but it fails to take full advantage of the rating records.

In response to the above analyses, we propose a novel CF method, which integrates three perspectives harmoniously for service recommendation, i.e., user trust, user similarity, and service similarity. The full use of almost all available information of our proposal can significantly alleviate cold-start issues and achieve robust recommendations.

TABLE I
SPECIFICATION OF NOTATIONS USED IN THIS ARTICLE

Notation	Specification
U_{set}	a set that gathers all referable users.
WS_{set}	a set that gathers all referable web services.
R	a user-item rating matrix.
T	a trust relationship matrix.
u_i	the i_{th} user of a platform.
ws_i	the i_{th} web service of a platform.
$t_{u,v}$	an initial trust value of user v observed by user u.
$trust_{u,v}$	an adjusted trust value from users u to v.
$sim'(u,v)$	the initial similarity between users u and v.
$sim(u,v)$	the adjusted similarity between users u and v.
$sim'(i,j)$	the initial similarity between web services ws_i and ws_j.
$sim(i,j)$	the adjusted similarity between web services ws_i and ws_j.
$r_{u,i}$	a rating value of web service ws_i observed by user u.
$\widetilde{r_{u,i}}$	a predicted value of the missing value $\widetilde{r_{u,i}}$.
I	the subset of web services both invoked by users u and v.
U	the subset of service users invoked both web services ws_i and ws_j.
$S(u)$	the similar user set of service user u.
$S(i)$	the similar service set of web service ws_i.
$T(u)$	the trust user set of service user u.
α, β, γ	three parameters that balance the reliability of different prediction methods.

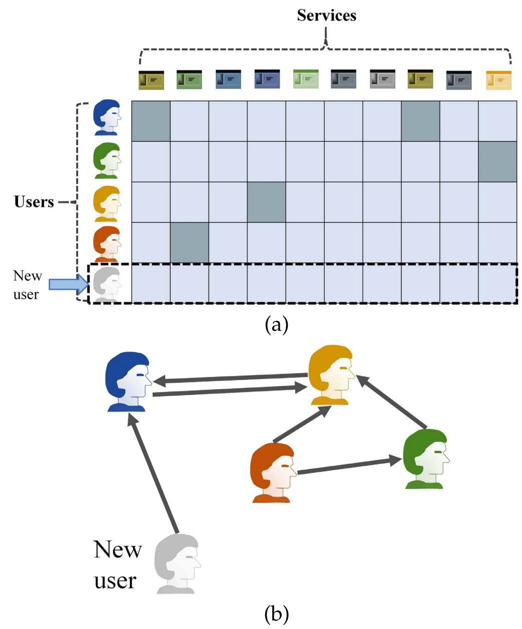

Fig. 1. Recommendation for cold-start problems: an example. (a) Invocation records of partial users. (b) User trust network.

III. Problem Definition and Motivation

First, we define the problem to be solved. Subsequently, we employ a vivid example to illustrate the motivation of our study. Table I introduces the notations to be used extensively in the remainder of this article.

A. Problem Definition

In preparation for defining the problem, we first gather all referable users into set U_{set}, i.e., $U_{\text{set}} = \{u_1, u_2, \ldots, u_m\}$, and all referable web services into set WS_{set}, i.e., $WS_{\text{set}} = \{ws_1, ws_2, \ldots, ws_n\}$. Then, the data used in our service recommendation are constructed into the following two matrices.

1) The user-item matrix denotes the user ratings for web services, as specified in (1). The matrix is constituted by m users in U_{set} and n web services in WS_{set}, where each entry $r_{u,i}$ represents a rating value from u to ws_i with range [1, 5]

$$R = \begin{array}{c} \\ u_1 \\ u_2 \\ \vdots \\ u_m \end{array} \begin{array}{c} \begin{array}{cccc} ws_1 & ws_2 & \cdots & ws_n \end{array} \\ \begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\ \vdots & \ddots & \vdots & \\ r_{m,1} & r_{m,2} & \cdots & r_{m,n} \end{pmatrix} \end{array}. \quad (1)$$

2) The user trust matrix indicates the mutual relationships between users, as formulated in (2), where $t_{u,v}$ denotes the trust relationship from users u to v, with 1 denoting trust and 0 indicating irrelevance

$$T = \begin{array}{c} \\ u_1 \\ u_2 \\ \vdots \\ u_m \end{array} \begin{array}{c} \begin{array}{cccc} u_1 & u_2 & \cdots & u_m \end{array} \\ \begin{pmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,m} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,m} \\ \vdots & \ddots & \vdots & \\ t_{m,1} & t_{m,2} & \cdots & t_{m,m} \end{pmatrix} \end{array}. \quad (2)$$

Now, we focus on the top-N recommendation based on user-item-trust records, named the $RUIT$ problem. Given a user-item rating matrix R and a user trust matrix T, we aim to find the top-N web services satisfying three conditions: 1) target user interests; 2) referring to multiple perspectives, i.e., user trust, user similarity, and service similarity; and 3) robustness. To address $RUIT$, our approaches will be elaborated in Section IV.

B. Motivation

A vivid example is depicted in Fig. 1 to elucidate the motivation of this study. As illustrated in Fig. 1(a), we adopt a matrix to illustrate the historical service usage information of partial users, where the squares with shadow indicate that the user has previously rated the services. It is obvious that the matrix implies two types of cold-start problems such that: 1) new users joining the platform without providing any rating information may cause $New\ user$ problem and 2) only a few user-item records are available, which may cause $Sparsity$ problem. Now, driven by economic interests, it is necessary to recommend some web services to increase the possibility of user invocation. According to the existing similarity-based CF approaches (UbCF and IbCF), the first step for recommendation is to extract the similar neighbors based on user-service rating records. However, two critical issues are raised in the above neighbor selection process: 1) almost no neighbors can be extracted for a user with poor rating records, as we can hardly measure the similarity in such a background and 2) the existing CF approaches failed to take any valuable information other than similarity into consideration to alleviate the cold-start problem. As a result, such inherent ills render the recommendation infeasible and seriously damage the robustness of the recommender system. In this situation, a social trust network with abundant and

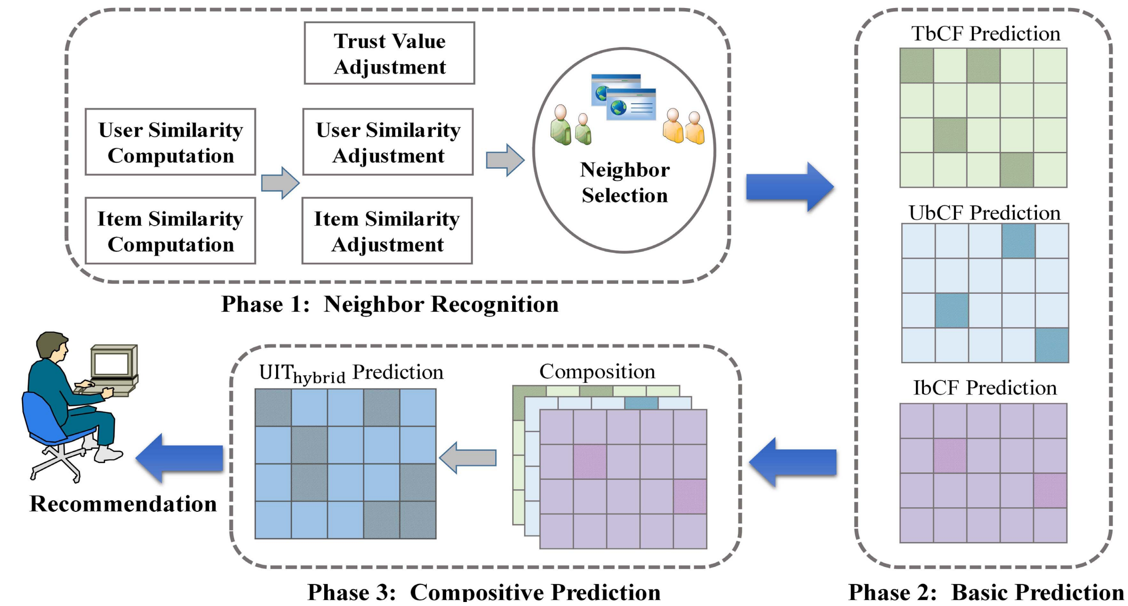

Fig. 2. Overview of UIT$_{\text{hybrid}}$.

valuable information may provide a meaningful perspective for the recommendation, as shown in Fig. 1(b).

Considering the above analyses, we supplement the existing UbCF and IbCF with user trust and propose our robust CF algorithm with user-item-trust records, named UIT$_{\text{hybrid}}$. Our proposal makes the recommendation possible and feasible for users without any historical service usage and can achieve a tradeoff between robustness and accuracy, as elaborated in Section IV.

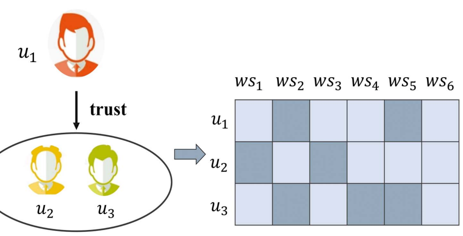

Fig. 3. User trust for recommendation: an example.

IV. UIT$_{\text{hybrid}}$: Hybrid Collaborative Filtering Recommendation Algorithm With User-Item-Trust Records

To solve the problem in Definition 1, our UIT$_{\text{hybrid}}$ is carried out in three phases (i.e., neighbor recognition, basic prediction, and compositive prediction), as illustrated in Fig. 2. In the rest of this section, we will detail these three phases for our proposal.

A. Neighbor Recognition

In this section, we elaborate on the process for neighbor recognition. We first briefly introduce the similarity computation methods based on UbCF and IbCF. Then, our *dependence weight* is presented for relation adjustment. Based on the adjusted relation, we realize the neighbor selection in the last part of this section.

1) Similarity Computation: Here, we employ the Pearson correlation coefficient (PCC) for similarity computation in our model, which is an effective similarity calculation tool with high accuracy and easy implementation.

Specifically, in UbCF, the similarity between users u and v can be calculated using the following equation:

$$\text{sim}'(u, v) = \frac{\sum_{i\in I}(r_{u,i} - \overline{r_u})(r_{v,i} - \overline{r_v})}{\sqrt{\sum_{i\in I}(r_{u,i} - \overline{r_u})^2}\sqrt{\sum_{i\in I}(r_{v,i} - \overline{r_v})^2}} \quad (3)$$

where $r_{u,i}$ represents the rating from user u to web service ws_i, and $\overline{r_u}$ indicates average service ratings adopted by user u. In addition, $I = I_u \bigcap I_v$ is constituted by web services that users u and v both invoked before. In this definition, $\text{sim}'(u, v)$ falls between [-1,1], and a larger $\text{sim}'(u, v)$ implies that the users u and v exhibit higher similarity.

In IbCF, the similarity between web services ws_i and ws_j can be calculated by

$$\text{sim}'(i, j) = \frac{\sum_{u\in U}(r_{u,i} - \overline{r_i})(r_{u,j} - \overline{r_j})}{\sqrt{\sum_{u\in U}(r_{u,i} - \overline{r_i})^2}\sqrt{\sum_{u\in U}(r_{u,j} - \overline{r_j})^2}} \quad (4)$$

where $\overline{r_i}$ is an average rating value of web services ws_i adopted by different users and $U = U_i \bigcap U_j$ is constituted by users who rated both ws_i and ws_j before. Likewise, the interval of $\text{sim}'(i, j)$ falls between $[-1, 1]$ as well.

2) Relation Adjustment: When applying user trust for recommendation, we can only refer to two absolute relations, i.e., trust and irrelevance, and ignore the trust degree between users. Consider the example in Fig. 3 with u_1 trusting u_2 and u_3. It contains a matrix, including some historical service usage records of each user, where squares with shadow represent that the user invoked the service before. According to the matrix, it is easy to find that there is no consistent service usage record between u_1 and u_2. In contrast, almost identical service usage experiences occur between users u_1

and u_3. In this situation, it is worth adopting a lower trust value from u_1 to u_2 than to u_3 as u_2's preferences are the lower convincibility for u1. However, to the best of our knowledge, traditional TbCF approaches rarely consider the trust degree to increase recommendation accuracy and effectiveness. Stemming from this observation, we present a *dependence weight* to measure the user trust degree. An enhanced trust value adjusted by *dependence weight* can be calculated using the following equation:

$$\text{trust}_{u,v} = \frac{t_{u,v}}{1 + e^{-\frac{|I_u \cap I_v|}{2}}} \tag{5}$$

where $t_{u,v}$ represents the absolute trust relation between users u and v (reflected with 0 or 1), $\text{trust}_{u,v}$ indicates u's trust degree to v, and $|I_u \cap I_v|$ is the number of services that both users rated before. According to the above trust degree measure equation, the *dependence weight* $(1/(1 + e^{-(|I_u \cap I_v|)/2}))$ will decrease the trust value when the coinvoked web service number $|I_u \cap I_v|$ is small. Because the range of $(1/(1 + e^{-(|I_u \cap I_v|)/2}))$ is within [0, 1] and the value of $t_{u,v}$ is 0 or 1, the range of $t_{u,v}$ is also [0, 1].

For the user similarity calculated by PCC in (3), our *dependence weight* can also take effect. When two dissimilar users happen to coinvoke a few web services and have similar ratings, the existing PCC always overestimates the similarity of these users. In this context, we take advantage of *dependence weight* in (6) to decrease the impact of similar experiences on only a few coinvoked services

$$\text{sim}(u, v) = \frac{\text{sim}'(u, v)}{1 + e^{-\frac{|I_u \cap I_v|}{2}}} \tag{6}$$

where $\text{sim}(u, v)$ is an adjusted user similarity, which falls between [-1,1].

As in the user similarity adjustment, the similarity between two web services is enhanced by

$$\text{sim}(i, j) = \frac{\text{sim}'(i, j)}{1 + e^{-\frac{|U_i \cap U_j|}{2}}} \tag{7}$$

where $\text{sim}(i, j)$ is an adjusted service similarity value with the interval of $[-1, 1]$, and $|U_i \cap U_j|$ is the number of users that rated both web services ws_i and ws_j before.

3) Neighbor Selection: Based on the quantified relationship calculated in Sections IV-A1 and IV-A2, we can obtain some neighbors for further recommendation. Concretely, we first sort the values generated in (5) and choose the top-K users trusted by target user u to generate a trust user set. In the same way, we arrange the values of (6) and (7) in descending order and select the top-K similar users and services, respectively. To ensure convenience, we formalize user u's trust user set as $T(u)$, user u's similar user set as $S(u)$, and web service ws_i's similar service set as $S(i)$.

B. Basic Prediction

In Section IV-A, we have obtained three neighbor sets, i.e., $T(u)$, $S(u)$, and $S(i)$. Next, we will utilize these three sets to predict ws_i's rating observed by target user u.

First, we employ the trust user set $T(u)$ for missing rating prediction using the following equation:

$$\widetilde{r_{u,i}} = \overline{r_u} + \frac{\sum_{v \in T(u)} \text{trust}_{u,v}(r_{v,i} - \overline{r_v})}{\sum_{v \in T(u)} \text{trust}_{u,v}} \tag{8}$$

where $\widetilde{r_{u,i}}$ is a predicted value of the missing entry $r_{u,i}$ in the user-item rating matrix R, while $\text{trust}_{u,v}$ is an adjusted trust value in the interval of [0, 1] calculated in (5). In summary, we design Algorithm 1 to describe the TbCF approach.

Algorithm 1 TbCF

Input:
R: user-item rating matrix;
T: user trust matrix;
K: the number of the nearest neighbors to be extracted.
Output:
$RL = \{ws_1, \ldots, ws_N\}$: a recommendation list for target user u, containing N web services.

---

initialize $T(u) = \emptyset$, $RL = \emptyset$;
for *each* $v \in U_{set}$ **do**
  if $t_{u,v} = 1$ **then**
    calculate $trust_{u,v}$ using Eq.5;
    update $t_{u,v}$ to $trust_{u,v}$ in T;
extract the $top - K$ trusted users of u into $T(u)$;
for *each* $ws_i \in WS_{set}$ **do**
  if $r_{u,i} = NULL$ **then**
    calculate $\widetilde{r_{u,i}}$ using Eq.8;
extract the $top - n$ web services into RL;
return RL;

The UbCF method makes use of the similar user set $S(u)$ to predict the unrated values, as designed in the following:

$$\widetilde{r_{u,i}} = \overline{u} + \frac{\sum_{v \in S(u)} \text{sim}(u, v)(r_{v,i} - \overline{v})}{\sum_{v \in S(u)} \text{sim}(u, v)} \tag{9}$$

where $\text{sim}(u, v)$ is an adjusted user similarity value calculated in (6).

In the same way, the IbCF method leverages the similar service set $S(i)$ to predict the missing values, as described in the following:

$$\widetilde{r_{u,i}} = \overline{r_i} + \frac{\sum_{j \in S(i)} \text{sim}(i, j)(r_{u,j} - \overline{r_j})}{\sum_{j \in S(i)} \text{sim}(i, j)} \tag{10}$$

where $\text{sim}(i, j)$ is an adjusted service similarity value calculated in (7).

C. Compositive Prediction

According to the basic prediction of the three methods above, if $S(u) = S(i) = T(u) = \emptyset$, the missing rating value $\widetilde{r_{u,i}}$ will fail to be predicted since there are no valid neighbors for prediction. We regard this situation as a failed prediction and set the unpredictable value $\widetilde{r_{u,i}}$ as an average rating of all invoked services. If there is just one nonempty neighbor set for reference, we will only apply it to make predictions. However, when we can employ more than one nonempty neighbor set for reference, utilizing only one will potentially ignore valuable information and render the result insufficiently accurate. To deal with the problem, we systematically integrate the three methods based on similar users (UbCF), similar services (IbCF), and trust users (TbCF) to produce a compositive prediction. In addition, due to the differences between users or services in the data set, the three methods may achieve different degrees of prediction accuracy. Thus, we introduce parameters α, β, and γ to balance the reliability of the respective three prediction methods. The compositive prediction of $\widetilde{r_{u,i}}$ is elaborated as follows:

$$\begin{aligned}\widetilde{r_{u,i}} = {} & \frac{\alpha}{\alpha+\beta+\gamma} \times \left(\overline{r_u} + \frac{\sum_{v\in S(u)} \mathrm{sim}(u,v)(r_{v,i}-\overline{r_v})}{\sum_{v\in S(u)} \mathrm{sim}(u,v)}\right) \\ & + \frac{\beta}{\alpha+\beta+\gamma} \times \left(\overline{r_i} + \frac{\sum_{j\in S(i)} \mathrm{sim}(i,j)(r_{u,j}-\overline{r_j})}{\sum_{j\in S(i)} \mathrm{sim}(i,j)}\right) \\ & + \frac{\gamma}{\alpha+\beta+\gamma} \times \left(\overline{r_u} + \frac{\sum_{v\in T(u)} \mathrm{trust}_{u,v}(r_{v,i}-\overline{r_v})}{\sum_{v\in T(u)} \mathrm{trust}_{u,v}}\right).\end{aligned} \tag{11}$$

It is worth noticing that, if one of the three methods does not take effect for a missing rating entry $\widetilde{r_{u,i}}$, we should set the corresponding parameter as 0 since our prediction of $\widetilde{r_{u,i}}$ does not rely on this method.

After accomplishing the prediction for all missing values, we rank the predicted values in descending order for target users and select the web services with high ratings to recommend. We denote the design above as the $\mathrm{UIT_{hybrid}}$ approach, which not only enables more accurate predictions but also makes the results robust by incorporating user trust into existing CF methods. Formally, we employ Algorithm 2 to specify the whole process of our $\mathrm{UIT_{hybrid}}$ method.

Algorithm 2 $\mathrm{UIT_{hybrid}}$

Input:
R: user-item rating matrix;
T: user trust matrix;
K: the number of the nearest neighbors to be extracted.
α, β, γ: three parameters balancing the reliability of different prediction methods.
Output:
$RL = \{ws_1, \ldots, ws_N\}$: a recommendation list for a target user, containing N web services.

```
initialize RL = ∅;
for each pair of users (u, v) where u, v ∈ U_set do
    calculate sim'(u, v) using Eq.3;
    adjust t_{u,v} to trust_{u,v} using Eq.5;
    adjust sim'(u, v) to sim(u, v) using Eq.6;
for each pair of web services (ws_i, ws_j) where
ws_i, ws_j ∈ WS_set do
    calculate sim'(i, j) using Eq.4;
    adjust sim'(i, j) to sim(i, j) using Eq.7;
extract the top − K nearest neighbors into neighbor sets;
for each u ∈ U_set do
    for each ws_i ∈ WS_set do
        if r_{u,i} = NULL then
            calculate r~_{u,i} using Eq.11;
extract the top − n web services into RL for the target
user;
return RL;
```

V. Experiments

In this section, we conduct a set of experiments based on the Epinions data set [33] to evaluate the feasibility and efficiency of $\mathrm{UIT_{hybrid}}$ and TbCF.

A. Experiment Data Set and Evaluation Methodology

The Epinions data set contains 105k users who have rated 611k items, and a total of 11 223k ratings are produced. The user-item rating record represents that users specify personalized ratings on different items according to their preferences. To formulate the rating information, we establish a user-item rating matrix, which has a density of 0.017%. The extremely sparse matrix means that a great deal of cold-start problems exists, which makes the recommendation infeasible in many situations. Thus, it is of great necessity to improve the recommendation performance and achieve high robustness in the case of poor ratings. In addition, 77k users of the data set have issued 636k trust information, which implies that users label others as reliable neighbors. Formally, we construct a user trust matrix to describe the social trust network.

In addition, we divide the known ratings of the Epinions data set into two parts: training and testing. Concretely, we randomly choose 20% of all ratings as the testing set. The training set is constituted by the remaining rating information. In this way, we predict the testing set by employing different methods and information in the training set. To demonstrate the effectiveness of our proposal, we further adopt the following four evaluation metrics.

1) MAE and RMSE: Mean absolute error (MAE) indicates the average deviation between predicted and real ratings of the target services, and root mean square error (RMSE) indicates the square root of the average square deviation between predictions and real rating values. Both metrics can evaluate the accuracy of our approach by the following equations:

$$\mathrm{MAE} = \frac{\sum_{u,i} |r^*_{u,i} - \widetilde{r_{u,i}}|}{N} \tag{12}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{u,i} (r^*_{u,i} - \widetilde{r_{u,i}})^2}{N}} \tag{13}$$

where $r^*_{u,i}$ is the real rating value of web service ws_i rated by user u, $\widetilde{r_{u,i}}$ is the predicted rating value, and N is the total number of missing entries in the user-item rating matrix.

2) Failure Rate: It is well-known that robustness plays a significant role in evaluating the effectiveness of algorithms. In a recommendation system, a robust approach can make recommendations successfully in most cases, even if the historical behavior data of users is extremely sparse. In this way, we introduce the failure rate to reflect the robustness of different models by calculating the ratio of failure prediction times (denoted as N_f) to the number of all predictions (denoted as N). In general, a lower failure rate indicates higher robustness. The failure rate is defined as

$$\text{Failure Rate} = \frac{N_f}{N} \times 100\%. \tag{14}$$

3) Coverage Rate: The coverage rate means the ratio of the different types of predicted services to the total number of web services, which is employed to measure the diversity of prediction results. In this way, a higher coverage rate represents a higher diversity. The coverage rate is defined as

$$\text{Coverage Rate} = \frac{\sum_u |\widetilde{r_u}|}{n} \times 100\% \tag{15}$$

where $\widetilde{r_u}$ includes all ratings predicted for target user u and n is the number of different web services.

Furthermore, we conduct our experiments on a machine with a 2.70-GHz CPU and 8.0 GB of RAM. The environment of our software configuration is Windows 10 with Python 3.6. We perform each experiment 100 times and adopt the average experimental results.

B. Comparison

We adopt the following three methods as comparison methods of $\text{UIT}_{\text{hybrid}}$ and TbCF.

1) *UbCF:* One of the most classical CF methods based on user similarity.
2) *HCF [34]:* The proposed model is a combined CF recommendation solution with optimized prediction order for improving the prediction accuracy. It designs the PGraph to describe the neighborhood and takes advantage of adjusted topological sorting based on the PGragh to generate the optimized order while predicting.
3) *Similarity Two-Rank-Based Core Users:* The method performs best among the four methods presented in [35], which utilizes core users specified by extracting users with the highest location weights according to user similarity degree.

The above three compared methods cover classic CF variants from diverse perspectives (i.e., benchmark user-based CF, benchmark hybrid CF, and improved CF with location weight). Therefore, the experimental comparisons with these three methods can offer an objective and comprehensive basis to evaluate the performances of our proposed $\text{UIT}_{\text{hybrid}}$ method.

In addition, we choose five groups of parameter values for our UTF_{hybrid} to evaluate the impacts of α, β, and γ on MAE, RMSE, failure rate, and coverage rate, as shown in Table II. In particular, we set top-$K = 15$ (top-K introduced in Section IV-A indicates the number of the nearest neighbors employed for each basic precision) as a representative to ensure the concision of Table II. We can easily observe that our UTF_{hybrid} model performs better when α, β, and γ are all around (1/3)) instead of others. Therefore, we set $\alpha = (1/3)$, $\beta = (1/3)$ and $\gamma = (1/3)$ for our method to conduct further experimentation.

Now, we develop five profiles to demonstrate the advantages of our proposal, including the comparison with itself and the other three competitive solutions.

TABLE II
IMPACT OF PARAMETERS

Metric	$\alpha\ \beta\ \gamma$ $1\ 0\ 0$	$\alpha\ \beta\ \gamma$ $\frac{2}{3}\ \frac{1}{3}\ 0$	$\alpha\ \beta\ \gamma$ $\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}$	$\alpha\ \beta\ \gamma$ $\frac{1}{3}\ \frac{2}{3}\ 0$	$\alpha\ \beta\ \gamma$ $0\ 0\ 1$
MAE	0.303	0.303	0.262	0.268	0.270
RMSE	0.549	0.548	0.529	0.530	0.532
Failure Rate(%)	52.977	52.156	28.747	34.805	35.626
Coverage Rate(%)	11.555	11.813	17.612	16.495	16.237

Profile 1 (Dependence Weight Performances): As stated in Section IV-A, we adopt *dependence weight* for our $\text{UIT}_{\text{hybrid}}$ approach to adjust the relationship between neighbors. To study the effect of the *dependence weight*, we evaluate performances of the four presented metrics containing two versions: one employs *dependence weight*, while the other does not. Moreover, we range top-K from 10 to 30 in steps of 5 and elaborate on the results in Fig. 4.

As illustrated in Fig. 4, benefiting from the increasingly added neighbor information, the performances of both versions are constantly improving with respect to four metrics. Moreover, we can observe that the version with *dependence weight* performs better than the version without it. This is because *dependence weight* takes effect by assigning a lower intimacy value for those actually unqualified neighbors. Consequently, our *dependence weight* can devaluate those weak but overestimated relationships and make the relativity computation between neighbors more credible in practice.

Profile 2 (Convergence Performances of UIT_{hybrid}): We evaluate the convergence performances of our $\text{UIT}_{\text{hybrid}}$ solution in this profile. In our experiment, we vary the experiment iterations from 1 to 100 and range top-K from 10 to 30 in steps of 5. The convergence performances of four metrics (including MAE, RMSE, failure rate, and coverage rate) are presented in Fig. 5. As shown in Fig. 5, we can observe that all the curves tend to converge when the repeated experiment times reach 10, and the performances become increasingly stable as repeated times increase. Thus, it is reasonable to employ the average effects of 100 iterations of experiment execution for comparison, as described previously.

Profile 3 (Accuracy Comparison): In this profile, we compare the performances of MAE and RMSE to evaluate the accuracy. Here, we range top-K from 10 to 30 in steps of 5. The experimental results are displayed in Fig. 6.

As depicted in Fig. 6, except for the similarity two-rank-based core users approach, we can observe that the prediction accuracy of the other four competing solutions all exhibit slow but steady growth (i.e., both MAE and RMSE values remain in decline) with increasing top-K. This is because the prediction can refer to more rating information when the proportion of the nearest neighbors becomes larger. The unstable accuracy of the similarity two-rank-based core users approach indicates that it only considers the preferences of a few significant users instead of the personalized interests for each target user. In addition, the performance of UbCF is also inferior because many cold-start problems occur in such a sparse context, resulting in the replacement of unpredictable ratings with average ratings of all users. The better performances of our TbCF indicate that the enhanced trust values aided by *dependence weight* takes effect so as to find the real

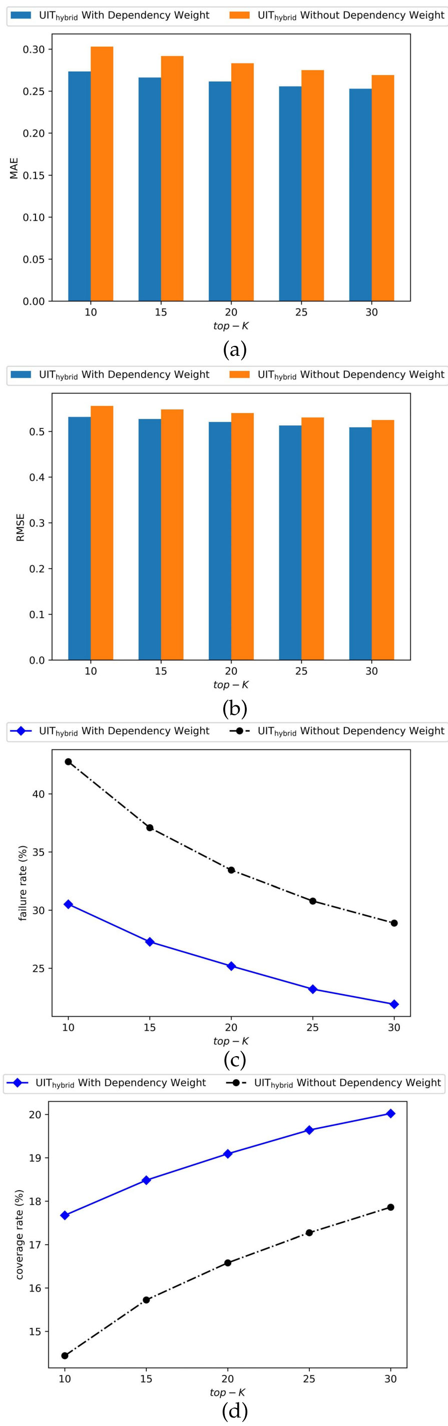

Fig. 4. Prediction performances of $\mathrm{UIT}_{\mathrm{hybrid}}$: *dependence weight*. (a) MAE. (b) RMSE. (c) Failure rate. (d) Coverage rate.

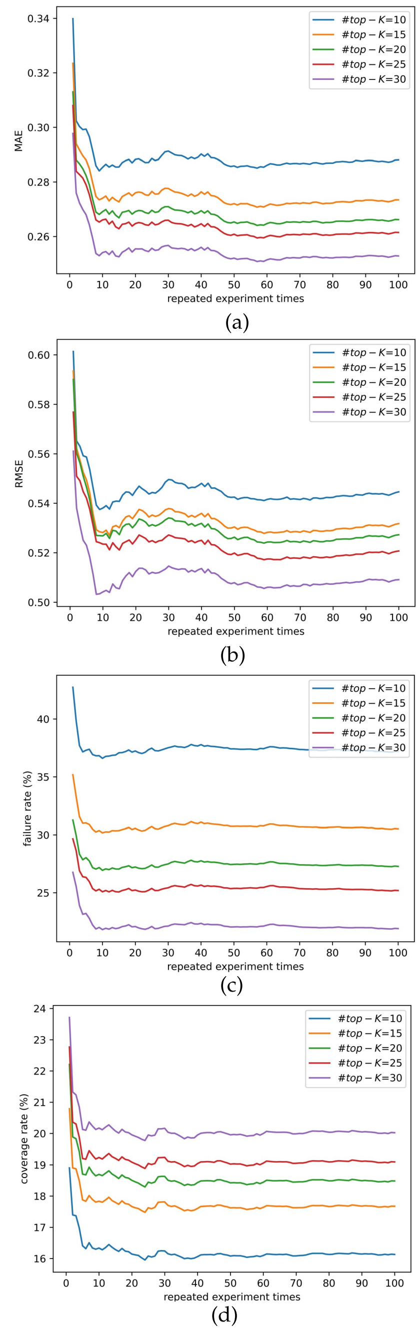

Fig. 5. Performance convergence of $\mathrm{UIT}_{\mathrm{hybrid}}$ with respect to repeated experiment times. (a) MAE. (b) RMSE. (c) Failure rate. (d) Coverage rate.

trusted users. The outstanding performances of our $\mathrm{UIT}_{\mathrm{hybrid}}$ demonstrate that user trust can provide an accurate and personalized supplement. Thus, our $\mathrm{UIT}_{\mathrm{hybrid}}$ method incorporating TbCF is harmonious and effective.

Profile 4 (Robustness Comparison): We compare the failure rate performance to evaluate the robustness in this profile. Here, we vary top-K from 10 to 30 in steps of 5. The experimental results are plotted in Fig. 7.

As depicted in Fig. 7, It can be observed that all five lines continue to decrease, which indicates that the robustness of the four approaches is enhanced with increasing top-K. This is because some originally unpredictable ratings are resolved by consulting the information of these newly added neighbors as more neighbors participate. Our TbCF performs well indicates the valuable trust information is worthy of being taken into consideration. Moreover, $\mathrm{UIT}_{\mathrm{hybrid}}$ renders more robust precision than the other four strategies, as the complementary user trust relationship makes prediction possible for those entries without similarity neighbors.

Profile 5 (Diversity Comparison): We compare the coverage rate performance to evaluate the diversity in this profile, where top-K is varied from 10 to 30 in steps of 5. We plot the evaluation results in Fig. 8.

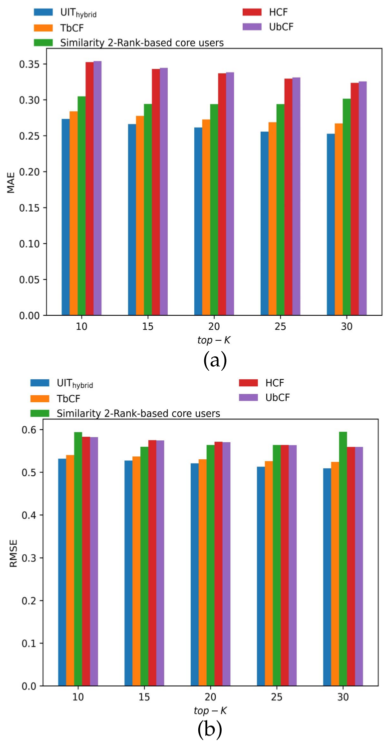

Fig. 6. Prediction accuracy comparison. (a) MAE performances. (b) RMSE performances.

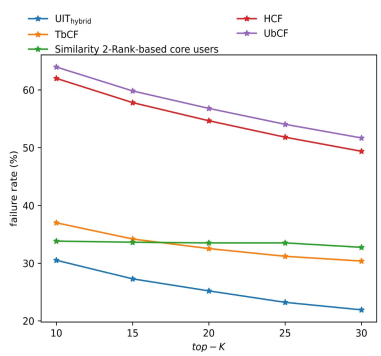

Fig. 7. Prediction robustness comparison.

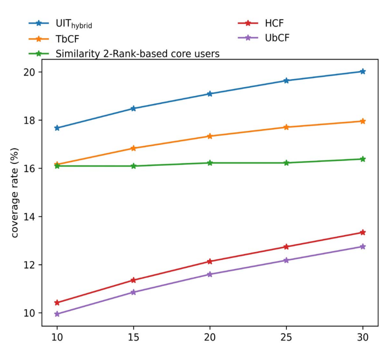

Fig. 8. Prediction diversity comparison.

As depicted in Fig. 8, only the similarity two-rank-based core users solution has not made progress in terms of predicted diversity (i.e., the coverage rate of the solution remains stable) as top-K increases. This is because the method focuses only on a small portion of core users, which offers inflexible service selections for the recommendation. As a contrast, the predicted diversity of $\text{UIT}_{\text{hybrid}}$ maintains stable growth and achieves the best performance among the presented five approaches. The reason is that our $\text{UIT}_{\text{hybrid}}$ aims to extract the most personalized services for each user. In this way, $\text{UIT}_{\text{hybrid}}$ with multiple perspectives can even take long-tail services (i.e., a considerable amount of services with rare invocation) into consideration for prediction and achieves relatively high diversity. Furthermore, the good performance of TbCF demonstrates that trust relationships can provide target users with extensive and diversified preferences compared to similarity relationships.

VI. Conclusion

In this article, we propose $\text{UIT}_{\text{hybrid}}$, a novel approach that incorporates user trust with the existing CF approach for rating prediction and service recommendation. Moreover, we develop a TbCF algorithm to lay the foundation for the incorporation. With the use of $\text{UIT}_{\text{hybrid}}$, more cold-start problems have been resolved without damaging predicted accuracy. We conduct extensive real-world experiments on a large but sparse data set to evaluate the feasibility and effectiveness of our approaches.

In our future work, we will investigate more social network information to supplement the existing sparse rating data [36], [37]. Another interesting work is combining trust-based user-item rating with the Group Role Assignment [38] and GRA with conflicting agents' methods [39] to conduct optimization for a group of items to be recommended because the user-item rating is a method of agent evaluation in role-based collaboration [40].

References

[1] Y. Wang, Y. Gao, Y. Li, and X. Tong, "A worker-selection incentive mechanism for optimizing platform-centric mobile crowdsourcing systems," *Comput. Netw.*, vol. 171, Apr. 2020, Art. no. 107144.

[2] Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y. Zhang, "Blockchain empowered arbitrable data auditing scheme for network storage as a service," *IEEE Trans. Services Comput.*, vol. 13, no. 2, pp. 289–300, Mar./Apr. 2020, doi: 10.1109/TSC.2019.2953033.

[3] Y. Xu, C. Zhang, G. Wang, Z. Qin, and Q. Zeng, "A blockchain-enabled deduplicatable data auditing mechanism for network storage services," *IEEE Trans. Emerg. Topics Comput.*, early access, Jun. 29, 2020, doi: 10.1109/TETC.2020.3005610.

[4] L. Qi, Q. He, F. Chen, X. Zhang, W. Dou, and Q. Ni, "Data-driven Web APIs recommendation for building Web applications," *IEEE Trans. Big Data*, early access, Feb. 24, 2020, doi: 10.1109/TBDATA.2020.2975587.

[5] W. Zhong *et al.*, "Multi-dimensional quality-driven service recommendation with privacy-preservation in mobile edge environment," *Comput. Commun.*, vol. 157, pp. 116–123, May 2020, doi: 10.1016/j.comcom.2020.04.018.

[6] J. Yao and W. Jiang, "Utilizing the bidirectional effect of evolutive trust-rating for recommendation in E-commerce," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov.*, Oct. 2018, pp. 1015–1022, doi: 10.1109/SmartWorld.2018.00178.

[7] C. Zhou *et al.*, "Modeling methodology for early warning of chronic heart failure based on real medical big data," *Expert Syst. Appl.*, vol. 151, Aug. 2020, Art. no. 113361.

[8] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Trans. Depend. Sec. Comput.*, vol. 15, no. 4, pp. 577–590, Jul./Aug. 2016.

[9] S. Zhang, Q. Liu, and Y. Lin, "Anonymizing popularity in online social networks with full utility," *Future Gener. Comput. Syst.*, vol. 72, pp. 227–238, Jul. 2017.

[10] Q. Liu, G. Wang, F. Li, S. Yang, and J. Wu, "Preserving privacy with probabilistic indistinguishability in weighted social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 5, pp. 1417–1429, May 2017.

[11] N. Lathia, S. Hailes, and L. Capra, "Trust-based collaborative filtering," in *Proc. IFIP Int. Conf. Trust Manage.* Boston, MA, USA: Springer, 2008, pp. 119–134.

[12] T. Duricic, E. Lacic, D. Kowald, and E. Lex, "Trust-based collaborative filtering: Tackling the cold start problem using regular equivalence," in *Proc. ACM Conf. Recommender Syst.*, 2018, pp. 446–450, doi: 10.1145/3240323.3240404.

[13] F. Zhang, L. Bai, and F. Gao, "A user trust-based collaborative filtering recommendation algorithm," in *Proc. Inf. Commun. Secur.*, Berlin, Germany, 2009, pp. 411–424.

[14] X. Wang, L. T. Yang, H. Li, M. Lin, J. Han, and B. O. Apduhan, "NQA: A nested anti-collision algorithm for RFID systems," *ACM Trans. Embedded Comput. Syst.*, vol. 18, no. 4, pp. 1–21, Aug. 2019.

[15] X. Xu *et al.*, "Multiobjective computation offloading for workflow management in cloudlet-based mobile cloud using NSGA—II," *Comput. Intell.*, vol. 35, no. 3, pp. 476–495, Aug. 2019.

[16] L. Qi *et al.*, "Finding all you need: Web APIs recommendation in Web of things through keywords search," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 5, pp. 1063–1072, Oct. 2019.

[17] H. Liu, H. Kou, C. Yan, and L. Qi, "Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph," *Complexity*, vol. 2020, pp. 1–15, Apr. 2020.

[18] X. Wang, L. T. Yang, L. Kuang, X. Liu, Q. Zhang, and M. J. Deen, "A tensor-based big-data-driven routing recommendation approach for heterogeneous networks," *IEEE Netw.*, vol. 33, no. 1, pp. 64–69, Jan. 2019.

[19] H.-G. Rong, S.-X. Huo, C.-H. Hu, and J.-X. Mo, "User similarity-based collaborative filtering recommendation algorithm," *Tongxin Xuebao/Journal Commun.*, vol. 35, no. 2, pp. 16–24, 2014.

[20] C. Jiang, R. Duan, H. K. Jain, S. Liu, and K. Liang, "Hybrid collaborative filtering for high-involvement products: A solution to opinion sparsity and dynamics," *Decis. Support Syst.*, vol. 79, pp. 195–208, Nov. 2015.

[21] X. Wang, L. T. Yang, Y. Wang, X. Liu, Q. Zhang, and M. J. Deen, "A distributed tensor-train decomposition method for cyber-physical-social services," *ACM Trans. Cyber-Physical Syst.*, vol. 3, no. 4, pp. 1–15, Oct. 2019.

[22] L. Qi, X. Wang, X. Xu, W. Dou, and S. Li, "Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing," *IEEE Trans. Netw. Sci. Eng.*, early access, Jan. 27, 2020, doi: 10.1109/TNSE.2020.2969489.

[23] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 144–153.

[24] Z. Sun, Y. Wang, Z. Cai, T. Liu, X. Tong, and N. Jiang, "A two-stage privacy protection mechanism based on blockchain in mobile crowdsourcing," *Int. J. Intell. Syst.*, Jan. 2021, doi: 10.1002/int.22371.

[25] T. Liu, Y. Wang, Y. Li, X. Tong, L. Qi, and N. Jiang, "Privacy protection based on stream cipher for spatiotemporal data in IoT," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 7928–7940, Sep. 2020.

[26] Q. Liu, P. Hou, G. Wang, T. Peng, and S. Zhang, "Intelligent route planning on large road networks with efficiency and privacy," *J. Parallel Distrib. Comput.*, vol. 133, pp. 93–106, Nov. 2019.

[27] X. Xu, X. Liu, Z. Xu, C. Wang, S. Wan, and X. Yang, "Joint optimization of resource utilization and load balance with privacy preservation for edge services in 5G networks," *Mobile Netw. Appl.*, vol. 25, no. 2, pp. 713–724, Apr. 2020, doi: 10.1007/s11036-019-01448-8.

[28] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 2, pp. 766–775, Apr. 2020.

[29] Y. Zhang, J. Pan, L. Qi, and Q. He, "Privacy-preserving quality prediction for edge-based IoT services," *Future Gener. Comput. Syst.*, vol. 114, pp. 336–348, Jan. 2021.

[30] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, New York, NY, USA, 2009, pp. 203–210.

[31] C. Birtolo, D. Ronca, and G. Aurilio, "Trust-aware clustering collaborative filtering: Identification of relevant items," in *Artificial Intelligence Applications and Innovations*. Berlin, Germany: Springer, 2012, pp. 374–384.

[32] M. Jamali and M. Ester, "*TrustWalker* a random walk model for combining trust-based and item-based recommendation," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 397–406, doi: 10.1145/1557019.1557067.

[33] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proc. ACM Conf. Recommender Syst. (RecSys)*, New York, NY, USA, 2007, pp. 17–24.

[34] Z. Zhou, B. Wang, J. Guo, and J. Pan, "QoS-aware Web service recommendation using collaborative filtering with PGraph," in *Proc. IEEE Int. Conf. Web Services*, Jun. 2015, pp. 392–399.

[35] G. Cao and L. Kuang, "Identifying core users based on trust relationships and interest similarity in recommender system," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2016, pp. 284–291.

[36] J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, and F. Xia, "Community-diversified influence maximization in social networks," *Inf. Syst.*, vol. 92, Sep. 2020, Art. no. 101522.

[37] T. Cai, J. Li, A. S. Mian, R. Li, T. Sellis, and J. X. Yu, "Target-aware holistic influence maximization in spatial social networks," *IEEE Trans. Knowl. Data Eng.*, early access, Jun. 17, 2020, doi: 10.1109/TKDE.2020.3003047.

[38] H. Zhu, M. Zhou, and R. Alkins, "Group role assignment via a Kuhn–Munkres algorithm-based solution," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 3, pp. 739–750, May 2012.

[39] H. Zhu, "Avoiding conflicts by group role assignment," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 4, pp. 535–547, Apr. 2016.

[40] H. Zhu and M. Zhou, "Role-based collaboration and its kernel mechanisms," *IEEE Trans. Syst., Man Cybern. C, Appl. Rev.*, vol. 36, no. 4, pp. 578–589, Jul. 2006.