

Received August 13, 2020, accepted August 26, 2020, date of publication September 3, 2020, date of current version September 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3020825

QoS Prediction of Web Services Based on Reputation-Aware Network Embedding

XUECHUN WANG¹, PENG HE^{1,2}, JIANXIONG ZHANG¹, AND ZHAN WANG¹

¹School of Computer and Information Engineering, Hubei University, Wuhan 430062, China

²Research Center of Educational Informatization Engineering and Technology, Wuhan 430062, China

Corresponding author: Peng He (penghe@hubu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1400602, in part by the National Natural Science Foundation of China under Grant 61572371 and Grant 61702377, in part by the Science and Technology Innovation Program of Hubei Province under Grant 2018ACA133, and in part by the Educational Commission of Hubei Province under Grant Q20171008.

ABSTRACT As the emergence of numerous services with similar functions, it is very helpful to recommend personalized services for users, and urgent to accurately predict the QoS(Quality-of-Service) values of Web services. Collaborative Filtering (CF) is a commonly-used method to handle above issues. However, it faces two common issues: data sparsity problem and trustworthiness issue, which greatly reduces its prediction accuracy. To address this problem properly and systematically, we introduce the network embedding learning into the QoS prediction process and propose an improved QoS prediction method based on the reputation-aware network embedding learning. Firstly, a two-phase K-means clustering is adopted to filter untrustworthy users. Next, the reputation of trustworthy users is calculated, and an attributed user-service bipartite network is constructed between trustworthy users and services while considering the user reputation. Then the reputation-aware network embedding is adopted to learn the hidden representations of users. Finally, user-based CF is adopted to predict the unknown QoS values. The experimental results show that our method has a significant improvement in accuracy compared with other methods.

INDEX TERMS Web service, QoS prediction, reputation-aware, network embedding.

I. INTRODUCTION

With the rapid development of service-oriented computing and big data, more and more services are available for users to choose. It is a primary task to efficiently recommend services that meet the personalized requirements for users from a large number of functional equivalent web services. QoS is a set of properties describing the non-functional characteristics of web services, such as cost, response time, throughput, reliability, availability, etc [1]. At present, QoS has become an important evidence for users to choose services, so the accurate prediction of QoS has attracted much attention.

Collaborative Filtering (CF) is one of the most widely used methods for QoS prediction [2], especially neighborhood-based CF [3]–[11]. The key step of CF is to calculate the similarity between users (or services), and then predict the unknown QoS based on the historical QoS values provided by similar users (or services). It is not difficult to find that the similarity calculation of this method depends on

historical data. In other words, the quality of historical data has a great influence on prediction results. If the data is provided by untrustworthy users, the prediction performance will be greatly affected. Therefore, it is necessary to consider user reputation and then further find out similar users (or services). To address the trustworthiness problem, many researchers proposed a series of QoS prediction methods based on user reputation [1], [2], [12]. On the other hand, the model-based CF [13]–[14] is another popular method for QoS prediction, which builds a model based on users' historical QoS performance to predict the unknown QoS values. Considering the problem of data sparsity, [15] proposed a matrix factorization model that integrates QoS time series, and [16] proposed a Location-aware Low-rank Matrix Factorization for QoS prediction.

Recently, researchers have proposed many state-of-the-art methods by utilizing representation learning [17], random walk [18] or deep learning [19] to alleviate the sparsity problem and retrieve appropriate neighborhood. Considering the complexity of service invocations, [20] developed a general context-aware matrix-factorization approach, which

The associate editor coordinating the review of this manuscript and approving it for publication was Peng-Yong Kong ¹.

can make full use of implicit and explicit contextual information.

Nevertheless, few methods have properly and systematically addressed two common issues:

1. Relieve the problem of data sparsity and retrieving appropriate neighborhood information;
2. Effectively alleviate the influence of unreliable data on prediction accuracy.

Motivated by these, we propose a QoS prediction method based on reputation-aware network embedding, which is called RANEP (Reputation-Aware Network Embedding-based QoS Prediction) for short. The advantages of our method are two-fold. First, it introduces a network embedding method to find more similar neighbors, which makes full use of user-service bipartite network information and alleviates the problem of data sparsity. Second, it first integrates user reputation into the random walk process of network embedding, which can help retrieve more trustworthiness and similar neighborhood information for the similarity computation, thus alleviates the impact of unreliable data on prediction accuracy.

In summary, the main contribution of this paper is summarized as follows:

- To extract appropriate neighborhood information, we employ a network embedding method to explore more similar neighbors, which takes full advantage of user-service bipartite network information and effectively solve the problem of data sparsity as much as possible.
- To resolve the trustworthiness issue and provide accurate prediction results. We incorporate user reputation into the random walk process of network embedding. In this way, RANEP properly addresses the issue of trustworthiness and help identify more trustworthy and similar neighbors for the target users.
- To verify the effectiveness of the proposed QoS prediction method in our context, we carried out a large-scale experiments on the public dataset.

The rest of this paper is organized as follows. Section 2 is a review of the relevant work. Section 3 presents the detailed introduction of our method. Section 4 shows experimental results and Section 5 concludes this paper.

II. RELATED WORK

There are two main types of CF methods: memory-based CF and model-based CF. The former uses the existing QoS values of similar users (or services) to predict the unknown QoS values, whereas model-based CF approaches build a model that have been trained using historical QoS data.

A. MEMORY-BASED CF QoS PREDICTION APPROACHES

In memory-based method, Zheng *et al.* [21] used a hybrid CF approach (UIPCC) to predict the QoS values of services, and carried out a large-scale real world experiment to verify the effectiveness and feasibility of this method. Hu *et al.* [22] integrated the time information into the similarity measurement process, and proposed a novel time-aware

QoS prediction method. Meanwhile, in order to alleviate the problem of data sparsity, [22] adopted a hybrid personalized random walk algorithm to infer more indirect similar users and similar services, which effectively improves the prediction accuracy.

Although the above methods improve the accuracy of QoS prediction to some extent, they directly use the historical QoS data provided by all users, ignoring the existence of untrustworthy users in practice. Thus, one of the problems with CF-based method is invalid data [23]. For avoiding this problem, some researchers put forward a series of QoS prediction methods based on user reputation. Qiu *et al.* [12] proposed a reputation-aware QoS prediction method RAP, which calculates the user's reputation based on the historical QoS values, and excludes the data contributed by untrustworthy users through reputation ranking, and finally only the data provided by trustworthy users is used for prediction. However, these reputation evaluation methods are very sensitive to parameter settings, and once the parameter settings are not appropriate, the prediction accuracy will be greatly affected. To solve this problem, Wu *et al.* [2] proposed a QoS prediction method CAP based on two-phase K-means clustering algorithm, which identifies untrustworthy users through user clustering and predicts QoS values based on trustworthy clustering information. Although CAP is not sensitive to parameter settings and significantly improves the robustness of prediction, there is still room for improvement in prediction performance because CAP only uses reliable similar users for prediction without considering the information of similar services. Su *et al.* [1] proposed a trust-aware approach TAP for reliable personalized QoS prediction, the author uses K-means clustering algorithm and Beta reputation system based approach to calculate the user's reputation, then determines the set of trustworthy similar users for target users. At the same time, the author employs K-means clustering to identify similar services, and finally uses the information of trustworthy similar users and similar services to predict. These methods consider the impact of user reputation on the accuracy of QoS prediction, but do not consider the problem of data sparsity.

B. MODEL-BASED CF QoS PREDICTION APPROACHES

Matrix factorization is a model-based collaborative filtering method, which is favored by many researchers for its high accuracy and scalability. Xu *et al.* [24] proposed a matrix factorization method based on user reputation, and proved the accuracy of the method by experiments. Qi *et al.* [13] proposed a MF based method, which integrates both user and service neighborhood information to predict personalized QoS values. Zhu *et al.* [14] extended the traditional matrix factorization model by using new technologies such as data conversion, online learning and adaptive weights. Although matrix factorization has made some progress in prediction accuracy, its prediction performance will be affected greatly when the data is extremely sparse. Li *et al.* [15] proposed a time-aware matrix factorization (TMF) model, which provides two-phase

QoS predictions. It first attempts to employ an adaptive matrix factorization model on a sparse QoS dataset to make QoS prediction, which can improve the QoS prediction accuracy. Ryu *et al.* [18] proposed a Location-based Matrix Factorization using Preference Propagation method (LMF-PP), which integrates both invocation similarity and neighborhood similarity into the process of preference propagation to cope with the data scarcity. Wu *et al.* [20] proposed a general context-aware matrix-factorization method to model the interactions of users-to-services and environment-to-environment simultaneously. In this way, both implicit and explicit contextual factors can be fully utilized.

Latent factor model is another popular model-based CF method. Wu *et al.* [25] proposed a deep latent factor model by sequentially connecting multiple latent factor models. Luo *et al.* [26] proposed a second-order Latent factor-based QoS-predictor with acceptable computational cost. In [27] the author proposed a posterior-neighborhood-regularized Latent factor model for QoS prediction, which can achieve high prediction accuracy and efficiency. Wu *et al.* [28] proposed a data-characteristic-aware latent factor (DCALF) model to implement highly accurate QoS predictions, which can appropriately predict QoS values according to the characteristics of given QoS data.

Besides, there are many other models. For example, Zhuang *et al.* [17] proposed a new representation learning framework for service recommendation, and proved that the use of network embedding technology can fully explore the potential relationship between users and items, which not only overcomes the shortcomings of the traditional model, but also greatly improves the recommendation accuracy. Zhu *et al.* [29] proposed a new context-aware reliability prediction approach, which solves the problem of data sparsity by constructing context-aware reliability models. Jin *et al.* [19] proposed a novel deep learning model for QoS prediction, which uses the CNN to retrieve the potential non-linear feature relationships and achieve accurate predictions.

C. HYBRID CF QoS PREDICTION APPROACHES

Memory-based CF methods use the historical QoS values to make prediction and face two common issues: Sparsity and scalability. Model-based approaches construct a global model based on the all QoS values in the user-service matrix, but they may take a lot of time to build and update models. Hybrid CF approaches combine the memory-based and model-based approaches to solve the limitations of the aforementioned CF approaches and improve prediction performance [30]. Chen *et al.* [31] combined the model-based and memory-based CF algorithms for Web service recommendation to improve the recommendation accuracy. The method named CNMF proposed by Zhang *et al.* [32] employed a covering-based clustering algorithm to identify appropriate neighborhood and incorporate both users' and services' neighborhood information into MF model to fully utilize neighborhood information. Ding *et al.* [33] proposed a time-aware service recommendation. They first improved the CF

method for time-aware user similarity estimation. Then combined the proposed similarity-enhanced CF and the ARIMA model, which can comprehensively capture the time feature of user similarity and accurately predict the unknown QoS values in the future under QoS instantaneity.

In this paper, we propose RANEP. To our knowledge, it is the first attempt which adopts the reputation-aware network embedding approach to fully extract useful information from heterogeneous information network and effectively alleviate the impact of unreliable data on prediction accuracy.

III. QoS PREDICTION APPROACH BASED ON REPUTATION-AWARE NETWORK EMBEDDING

In this section, we introduce network embedding learning while considering user reputation, and propose a novel QoS prediction method based on reputation-aware network embedding RANEP. We first adopt a two-phase k-means clustering to identify untrustworthy users. Next, the reputation of trustworthy users is calculated, and an attributed user-service bipartite network is constructed between trustworthy users and services while considering the reputation of users. Then the reputation-aware network embedding is adopted to learn the latent representations of users. Finally, according to the obtained user representations, we further use user-based CF to predict the unknown QoS values. The proposed method not only fully mines the high-order implicit relations between users (or services) from the historical data, but also considers the interference of untrustworthy users to some extent, thus effectively alleviates the problem of data sparsity and the impact of unreliable data on prediction accuracy.

The main framework of our approach is illustrated in Figure 1, includes the following procedures: (1) A two-phase K-means clustering for excluding untrustworthy users: In the first phase, we use K-means algorithm to cluster the historical QoS values of Web services, and the clustering results are used to calculate the user untrustworthy index. In the second phase, K-means clustering is carried out on the untrustworthy index values of users, and the untrustworthy user set is finally obtained. (2) Attributed user-service bipartite network modeling: According to the relationship between users and services, we first construct a user-service bipartite network, and then we calculate the reputation of trustworthy users, and use the reputation as attributes of user nodes. (3) Reputation-aware network embedding: To obtain the user's representation vector, we further perform ANE (Attributed Network Embedding) on the attributed user-service bipartite network. (4) QoS prediction: Given a target user, we return its Top-k similar users based on the obtained representation vectors previously, and predict the missing QoS values according to the preferences of similar users.

A. TWO-PHASE K-MEANS CLUSTERING

To resolve the trustworthiness issue and obtain accurate Web service QoS predictions, several methods have been

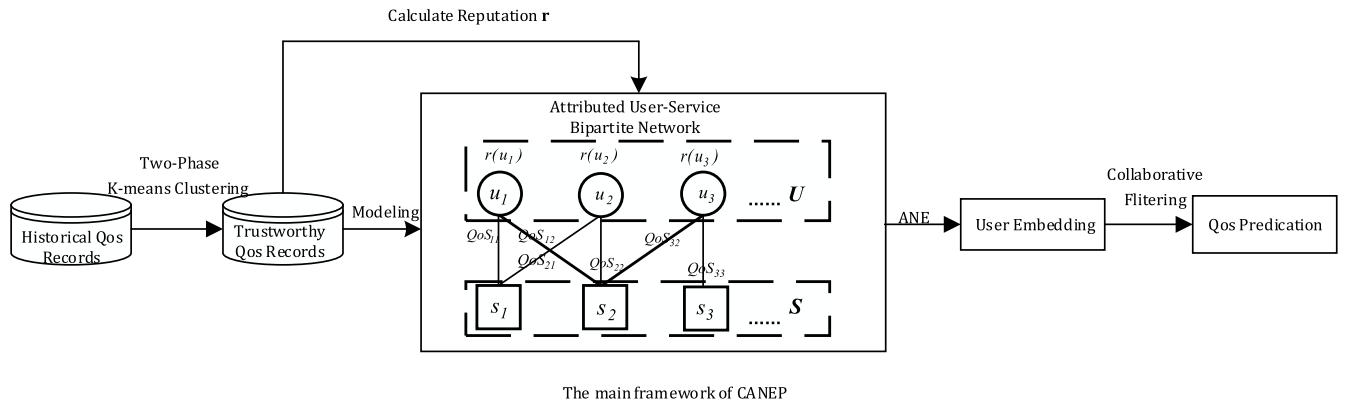


FIGURE 1. The main framework of RANEP.

proposed [12], [23]. As described in the second section, they are very sensitive to parameter settings. So in this paper, two-phase K-means clustering is used to identify untrustworthy users and exclude the data provided by these users as processed in [2].

1) Phase I is QoS values clustering: For each service, we use K-means algorithm to cluster all its QoS values provided by users. After clustering, the users who belong to the cluster that has the minimum number of elements are viewed as candidate untrustworthy users. Note that, the definition of untrustworthy user refers to [2]. The parameter K in the clustering algorithm determines the number of groups to be partitioned, C_j^k denotes the k_{th} cluster of service j , where k is the cluster index and $0 \leq k < K$, and U_j^{\min} represents the candidate untrustworthy user set on this service:

$$U_j^{\min} = \{u | u \in C_j^i, i = \operatorname{argmin}_{0 \leq t < K} |C_j^t|\} \quad (1)$$

Here we use a $m \times n$ matrix A consisting of m users and n services, to record the user untrustworthy index. The initial value of each entry of matrix A is set as 0. If user i is identified as a candidate untrustworthy user on service j , then $a_{i,j}$ is set as 1. Repeat the clustering process and update matrix A for each service until the QoS values of all services have been clustered.

2) Phase II is user clustering: After clustering all QoS values, we obtain the set of user untrustworthy index by using the following equation:

$$W = \{w_u | w_u = \sum_{j=0}^{n-1} a_{u,j}, 0 \leq u < m\} \quad (2)$$

where w_u represents the number of times that user u is identified as a candidate untrustworthy user. In order to get the set of untrustworthy users, we use K-means with $K = 2$ to cluster the users in W again. That is, all users are partitioned into two sets, C_w^0 and C_w^1 . The untrustworthy user set \bar{U} is defined as follows:

$$\bar{U} = \{u | w_u \in C_w^i, i = \operatorname{argmin}_{t \in \{0,1\}} |C_w^t|\} \quad (3)$$

where $\operatorname{argmin}_{t \in \{0,1\}} |C_w^t|$ returns the cluster index t that holds C_w^t have a smaller number of elements between 0 and 1.

By using the method presented above, we can identify the set of untrustworthy users easily.

B. ATTRIBUTED USER-SERVICE BIPARTITE NETWORK

1) USER REPUTATION CALCULATION

Assume there is a $m \times n$ matrix \mathcal{M} , which represents the interactions between m users and n services. Each entry $q_{i,j}$ of \mathcal{M} denotes the QoS value of service j obtained by user i . The reputation of user i is calculated by:

$$r_i^{k+1} = 1 - \frac{d \sum_{j \in I_i} |q_{i,j} - \operatorname{avg}_j^{k+1}|}{|I_i|} \quad (4)$$

where I_i is the set of service invoked by user i , avg_j denotes the aggregated evaluation value for service j , and d is the damping factor in $(0, 1)$. In Eq. 4, $\operatorname{avg}_j^{k+1}$ can be calculated by:

$$\operatorname{avg}_j^{k+1} = \frac{1}{|U_j|} \sum_{i \in U_j} q_{i,j} \cdot r_i^k \quad (5)$$

where U_j denotes the set of users who invoke service j , and r_i^k is the reputation value of user i calculated in the last iteration. We use the following equation to initialize the calculation process. That is to say, the initial reputation of each user is set to 1:

$$\operatorname{avg}_j^1 = \frac{1}{|U_j|} \sum_{i \in U_j} q_{i,j} \quad (6)$$

2) ATTRIBUTED USER-SERVICE BIPARTITE NETWORK MODELING

Although some researchers have attempted to infer more similar users (or services) through random walk algorithm [22], they predict the unknown QoS values based on the historical QoS values contributed by similar users and services, ignoring the existence of untrustworthy users.

As a ubiquitous data structure, bipartite network is usually used to model the relationship between two kinds of entities.

In a bipartite network, although edges only exist between nodes of different types, there is an implicit relationship between nodes of the same type. For example, in a user-item bipartite network for recommendation, the implicit relationship between users can indicate that they are interested in the same item. Therefore, in this paper we construct an attributed user-service bipartite network. That is, using the bipartite network to model the interactions between users and services, and user's reputation is considered as the attribute of its corresponding node.

Let $G = \{U, S, R, E\}$ be an attributed user-service bipartite network, where U and S denote the set of the two types of nodes respectively, and R denotes users' reputation. $U = \{u_1, u_2, u_3, \dots, u_m\}$, $S = \{s_1, s_2, s_3, \dots, s_n\}$. $E \subseteq U \times S$ denotes the edges between users and services, where $e_{ij} = QoS_{ij}$ represents the QoS value of service j obtained from user i . Figure 2 shows a simple example of the interactions between three users and three services. The reputation of u_1, u_2 and u_3 are $r(u_1), r(u_2)$ and $r(u_3)$ respectively.

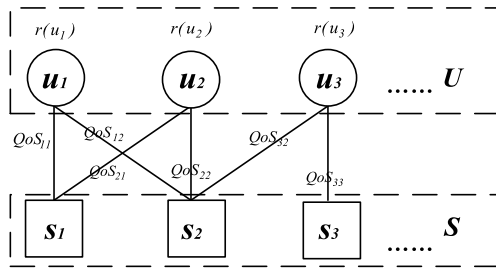


FIGURE 2. A simple example of user-service bipartite network.

C. REPUTATION-AWARE NETWORK EMBEDDING

As we all know, the real-world network is often very sparse. Just because two users do not invoke the same service does not mean that they are not similar. As shown in Figure 3, in fact, although u_1 and u_3 do not share a common service,

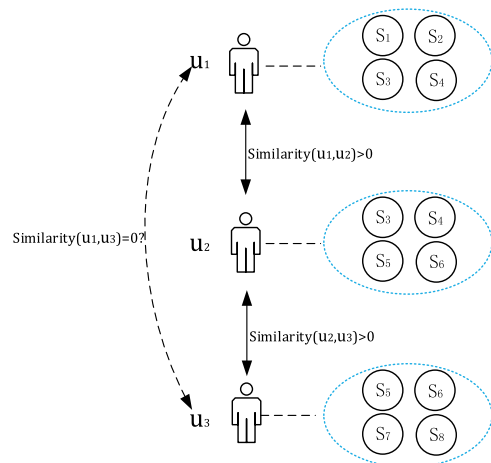


FIGURE 3. Transitive similarity.

there is a transitive similarity relationship between them due to the existence of u_2 . Most of the previous research work use PPC to calculate their similarity directly, ignoring the impact of transitive similar user. At the same time PCC requires an ample amount of quality information. This is often not realistic [32]. To deal with the sparsity problem, Ryu et al. [18] propose a novel service recommendation method, which models a bipartite graph between users and items, and perform a random walk on this graph to identify more trustworthy and similar neighbors for the target users. Inspired by the success of this method, in this paper, we will model a user-service bipartite graph and introduce the reputation-aware network embedding to fully explore the potential relationship between users.

Bipartite Network Embedding (BINE) is proposed to learn node representations in bipartite network [34]. By performing a biased and self-adaptive random walk, it keeps the long-tail distribution of node in the original bipartite network. It jointly models both the explicit relations (i.e., observed links) and high-order implicit relations (i.e., unobserved but transitive links) in learning the representations for nodes. Through bipartite network embedded learning, we can find more transitive reliable similar users for a target user.

1) MODELING EXPLICIT RELATIONS

In a user-service bipartite network, if there is an observed edge e_{ij} between node u_i and node s_j , indicating an explicit relationship between them. Thus their co-occurring probability is defined as:

$$P(i, j) = \frac{w_{ij}}{\sum_{e_{us} \in E} w_{us}} \tag{7}$$

where w_{ij} is the weight of edge e_{ij} .

The local proximity between two nodes in the embedding space is defined as:

$$\hat{P}(i, j) = \frac{1}{1 + \exp(-\vec{u}_i^T \vec{s}_j)} \tag{8}$$

where \vec{u}_i and \vec{s}_j are the embedding vectors of nodes u_i and s_j respectively. To model explicit relationships, we need to minimize the difference between $P(i, j)$ and $\hat{P}(i, j)$.

$$\begin{aligned} \text{minimize } O_1 = KL(P||\hat{P}) &= \sum_{e_{ij} \in E} P(i, j) \log\left(\frac{P(i, j)}{\hat{P}(i, j)}\right) \\ &\propto - \sum_{e_{ij} \in E} w_{ij} \log \hat{P}(i, j) \end{aligned} \tag{9}$$

By minimizing the objective function, two directly connected nodes in the original network tend to be closer to each other in the embedding space, thus preserving the explicit structural information.

2) MODELING IMPLICIT RELATIONS

The influence of each node in a network is various. To some extent, this can be expressed in terms of the degree of node. However, in some practical situations, only the degree of

node cannot reflect its real influence. For example, in our context, a user who invokes many services may have a very low reputation, due to the abnormal QoS values. As a result, the influence of the user will be greatly reduced. As shown in Figure 4, assume u_1 is the current node, $r(u_i)$ and $d(u_i)$ denote the reputation and degree of u_i . It is not hard find that although $d(u_2)$ is less than $d(u_5)$, the reputation $r(u_2)$ is much greater than $r(u_5)$, and $d(u_2) \times r(u_2) > d(u_5) \times r(u_5)$. Generally speaking, u_2 may tend to be more valuable than u_5 , so u_2 is more likely to be recommended to u_1 as a similar neighbor. Therefore, in this paper, we propose to combine the degree and reputation of nodes as its influence to model the implicit relations. Before modeling implicit relations, we need to construct corpus of node sequences.

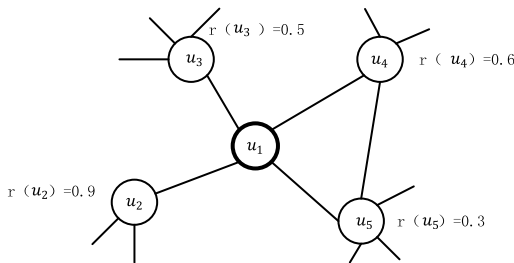


FIGURE 4. A snapshot of the network.

a: CONSTRUCTING CORPUS OF NODE SEQUENCES

The 2nd-order proximity between two vertices is defined as:

$$w_{ij}^U = \sum_{k \in S} w_{ik} w_{jk}; \quad w_{ij}^S = \sum_{k \in U} w_{ki} w_{kj} \quad (10)$$

where w_{ij} is the weight of edge e_{ij} . So we can use $|U| \times |U|$ matrix $W^U = [w_{ij}^U]$ and $|S| \times |S|$ matrix $W^S = [w_{ij}^S]$ to represent the two homogeneous networks, respectively. Next, we perform a biased and self-adaptive random walk on two homogeneous networks respectively, to generate corpus of node sequences. In our context, for the service network, the influence of each service node is associated with its degree, while for the user network, the influence of each user node is determined by its degree and reputation. The more influent a node is, the more likely a random walk will start from it. In this way, we integrate user reputation into the random walk process, and seek similar users with higher reputation for each target user to alleviate the impact of untrustworthy users on prediction accuracy. Algorithm 1 shows the process of random walk. Where $maxT$ and $minT$ are the maximal and minimal length of random walks starting from each node respectively. D^R is the corpus generated from the node set U (or S), $d(i)$ is the degree of the node i , and $r(u)$ is the reputation of user u calculated in section 3.2.1.

b: IMPLICIT RELATION MODELING

After performing biased random walks on the two homogeneous networks respectively, we obtain two corpus of node sequences. Next we employ the Skip-gram model [35] on the

Algorithm 1 WalkGenerator ($G, U, S, maxT, minT, p$)

Input: weight matrix of the bipartite network G , the set of user node U and the set of service node S , maximal walks per node $maxT$, minimal walks per node $minT$, walk stopping probability p

Output: The set of node sequences: D^U, D^S

```

1 Calculate  $W^R$  w.r.t. Equation (10);
2 foreach node  $u_i \in U$  do
3   Calculate node's reputation  $r(u_i)$  w.r.t. Equation (4)(5)(6);
4   Calculate node's influence:
        $\mathbf{H}(u_i) = d(u_i) \times r(u_i)$ ;
5    $l = \max(\mathbf{H}(u_i) \times maxT, minT)$ ;
6   for  $i = 0$  to  $l$  do
7      $D_{u_i} = BiasedRandomWalk(W^U, u_i, p)$ ;
8     Add  $D_{u_i}$  into  $D^U$ ;
9 foreach node  $s_j \in S$  do
10   $l = \max(d(s_j) \times maxT, minT)$ ;
11  for  $i = 0$  to  $l$  do
12     $D_{s_j} = BiasedRandomWalk(W^S, s_j, p)$ ;
13    Add  $D_{s_j}$  into  $D^S$ ;
14 return  $D^U, D^S$ 

```

two corpora to learn node embedding vector. Given a node sequence Q and a node u_i , each node is associated with a context vector $\vec{\theta}_i$ (or $\vec{\vartheta}_i$) to denote its role as a context. The objective functions for corpus D^U and D^S are:

$$\text{maximize } O_2 = \prod_{u_i \in Q \wedge Q \in D^U} \prod_{u_c \in C_Q(u_i)} P(u_c | u_i) \quad (11)$$

$$\text{maximize } O_3 = \prod_{s_j \in Q \wedge Q \in D^S} \prod_{s_c \in C_Q(s_j)} P(s_c | s_j) \quad (12)$$

where $C_Q(u_i)$ (or $C_Q(s_j)$) denotes the context nodes of node u_i (or s_j) in sequence Q .

The conditional probability is defined as:

$$P(u_c | u_i) = \frac{\exp(\vec{u}_i^T \vec{\theta}_c)}{\sum_{k=1}^{|U|} \exp(\vec{u}_i^T \vec{\theta}_k)}, \quad P(s_c | s_j) = \frac{\exp(\vec{s}_j^T \vec{\vartheta}_c)}{\sum_{k=1}^{|S|} \exp(\vec{s}_j^T \vec{\vartheta}_k)} \quad (13)$$

By optimizing the objective function defined in Equations (11) and (12), the nodes with similar context will be closer in the embedding space.

c: JOINT OPTIMIZATION

In order to preserve both explicit and implicit relations in bipartite network, we combine their objective functions to form a joint optimization framework:

$$\text{maximize } L = \alpha \log O_2 + \beta \log O_3 - \gamma O_1 \quad (14)$$

where parameter α , β and γ are hyper-parameters used in the joint optimization framework to combine different components.

After performing the reputation-aware network embedding, we can obtain user representation vector \vec{u}_i .

D. USER-BASED COLLABORATIVE FILTERING

For each user u_i , we use its representation vector \vec{u}_i learned above to calculate the similarity. Thus, the similarity between u_i and u_j is defined as:

$$\text{Sim}(u_i, u_j) = \frac{\vec{u}_i \cdot \vec{u}_j}{\|\vec{u}_i\| \times \|\vec{u}_j\|} \quad (15)$$

For a target user u , we choose the Top- k users with the highest similarity as its similar users $S(u)$. Then, we further make QoS values prediction according to the preference of similar users by the following equation:

$$\hat{q}_{u,s} = \bar{q}_u + \frac{\sum_{a \in S(u)} \text{sim}(u, a)(q_{a,s} - \bar{q}_a)}{\sum_{a \in S(u)} \text{Sim}(u, a)} \quad (16)$$

where \bar{q}_u denotes the average QoS value observed by the user u .

IV. EXPERIMENTS

In this section, we conduct experiments on real-world web service QoS data, and aim to answer the following three research questions:

RQ1 How does RANEP perform compared with the baselines?

RQ2 Is the introduction of user's reputation helpful to the learning of users' representation vector?

RQ3 How do the key parameters affect the performance of RANEP?

A. DATASET

To evaluate the performance of QoS prediction method proposed in this paper, we used a commonly-used public dataset from WS-Dream collected by Zheng [36], [37], which includes real-world QoS values (i.e., response-time) from 339 users on 5825 web services. We first remove a certain number of QoS values from the user-service matrix, and use the parameter *matrix density* to indicate the density of removed matrix. Meanwhile, we randomly select p percent of users as untrustworthy users and substitute their contributed values by randomly generated values. A large number of experiments are carried out on the modified dataset, the performances of different methods are compared, and the effects of different parameters are evaluated.

B. EVALUATION MEASURES

The mean absolute error (MAE) and root mean squared error (RMSE) are commonly used to measure the accuracy of prediction methods.

NMAE is defined as:

$$\text{NMAE} = \frac{\text{MAE}}{\sum_{u,s} q_{u,s}/N} \quad (17)$$

RMSE is defined as:

$$\text{RMSE} = \sqrt{\frac{\sum_{u,s} |q_{u,s} - \hat{q}_{u,s}|^2}{N}} \quad (18)$$

where N is the number of all predicted values, $q_{u,s}$ represents the real QoS value, and $\hat{q}_{u,s}$ denotes the predicted value.

C. BASELINES

In order to validate our work, we introduce the following six methods as the baselines.

- UIPCC: A hybrid collaborative filtering method that combines data contributed by similar users and similar services [21].

- RAP: A reputation-aware prediction method, which calculates the user's reputation based on historical data, and excludes the data of untrustworthy users through the ranking of the reputation [12].

- TAP: A personalized trust-aware QoS prediction method, which uses K-means clustering algorithm and Beta distribution based method to calculate the user's reputation. The service clustering is used to identify similar services, and the data contributed by trustworthy similar users and similar services are used to make predictions [1].

- CNMF: A Covering-based quality Prediction method via Neighborhood-aware Matrix Factorization, which employs a covering-based clustering method to retrieve appropriate neighborhood information and utilizes neighborhood information on both users and services to improve the prediction accuracy [32].

In addition, we also construct two variants of RANEP.

- RANEP-init: The case that, line 4 in algorithm 1, only associates user's reputation with its influence when network embedding learning is done on the user-service bipartite network.

- NEP-D: The case that, line 4 in algorithm 1, only associates the degree of user node with its influence when network embedding learning is done on the user-service bipartite network.

D. CASE STUDY

In this section we report the results aiming at answering the three research questions mentioned above.

RQ1 How does RANEP perform compared with the baselines?

In this experiment, let *Matrix density* varies from 5% to 14% by a step of 1% and similar to [1], the percentage of untrustworthy user is set as 10%. According to the analysis in [2], in the first phase K-means clustering, the parameter K determines the number of groups to be partitioned. When K is set between 5 and 8, the approach can achieve better performance. So K is set as 5 in our context. *Top-k* in Eq. (17) is set as 20, indicating that 20 users are returned as similar users. In addition, For the user-service bipartite network embedding, according to [34], the number of negative samples is set as 4, the window size is set as 5, the walking stop probability is 0.15, and the loss trade-off parameters $\alpha = 0.01$, $\beta = 0.01$, $\gamma = 4$ and the learning rate λ is 0.15. For a fair comparison, all parameters of the benchmarks follow their default settings. As for CNMF, the parameter θ that control how much it relies on users' and services' neighborhood information is set

TABLE 1. Prediction accuracy comparison.

Metrics	Method	matrix density (the percentage of untrustworthy user is 10%)									
		5%	6%	7%	8%	9%	10%	11%	12%	13%	14%
NMAE	UIPCC	1.371	1.336	1.303	1.240	1.210	1.184	1.131	1.101	1.059	1.046
	CAP	0.706	0.708	0.703	0.689	0.669	0.669	0.657	0.660	0.639	0.645
	TAP	0.658	0.643	0.635	0.614	0.599	0.601	0.594	0.584	0.579	0.577
	CNMF	0.629	0.613	0.611	0.598	0.585	0.572	0.556	0.558	0.546	0.532
	RANEP	0.544	0.536	0.529	0.521	0.513	0.505	0.497	0.488	0.481	0.477
Improve .vs CNMF(%)		13.51%	12.56%	13.42%	12.88%	12.31%	11.71%	10.61%	12.54%	11.90%	10.34%
RMSE	UIPCC	2.477	2.415	2.382	2.328	2.275	2.289	2.224	2.157	2.066	1.926
	CAP	1.979	1.951	1.925	1.871	1.826	1.784	1.725	1.691	1.653	1.671
	TAP	1.707	1.682	1.650	1.656	1.633	1.652	1.618	1.623	1.594	1.556
	CNMF	1.497	1.474	1.460	1.447	1.433	1.415	1.401	1.372	1.340	1.327
	RANEP	1.203	1.189	1.172	1.167	1.153	1.140	1.135	1.126	1.123	1.103
Improve .vs CNMF(%)		19.60%	19.31%	19.76%	19.38%	19.54%	19.46%	18.99%	17.96%	16.18%	16.85%

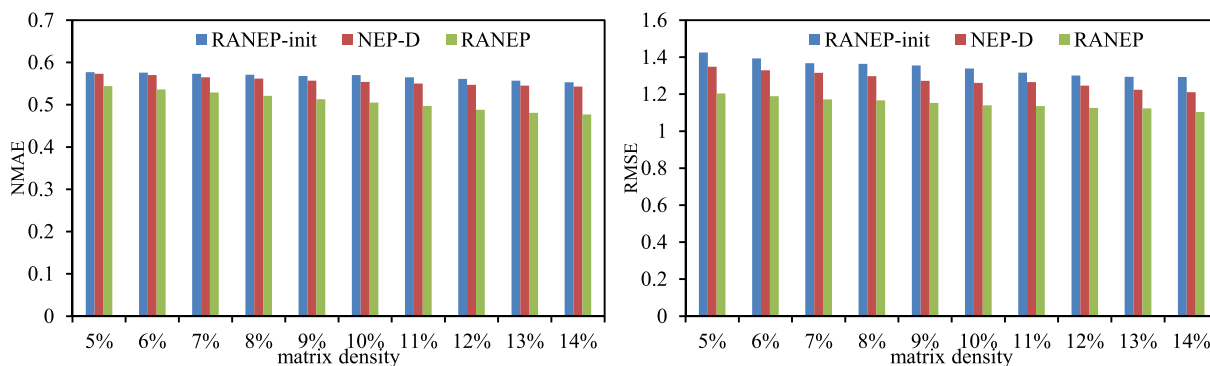


FIGURE 5. A comparison between RANEP with its two variants.

as 0.6, the number of neighbor users and neighbor services utilized for prediction is 20 and the dimensionality d is set as 20.

Table 1 shows the prediction results of five methods under different *matrix density*.

First of all, compared with all baseline methods, RANEP achieves the best performance on both MAE and RMSE than selected baselines under all densities. That is to say, the method RANEP proposed in this paper can obtain higher prediction accuracy. Moreover, the superiority of the proposed RANEP becomes more significant with less training data. In particular, the NMAE of RAENP is 0.544 when matrix density is 5% and the NMAE of four baselines are all more than 0.629. As the best baseline, our method improves prediction accuracy by 13.51% over CNMF. Besides, the improvement ratios over CNMF are about 18.7% in terms of RMSE. The reason for this result may be that RANEP adopts the attributed network embedding learning, which alleviates the impact of data sparsity on the prediction accuracy. Second, Table 1 also shows that UIPCC has the highest NMAE and RMSE. Since UIPCC does not consider the effect of untrustworthy users and may employ their contributed values to predict the missing QoS value. The experimental results again verify the necessity of filtering untrustworthy users in QoS prediction.

As the *matrix density* increases, it is clear that the QoS prediction accuracy of five methods is improved indicated by the

smaller NMAE and RMSE values. The result is completely consistent with the conclusions proposed in the literature [2]. A possible explanation is that the greater the density of the matrix, the richer the interaction information between users and services.

RQ2 Is the introduction of user’s reputation helpful to the learning of users’ representation vector?

This section investigates the impact of user’s reputation on the results of QoS prediction. Here keeping the same parameter settings as in the previous experiment, we further make a comparison between RANEP with its two variants in terms of prediction accuracy. The results are shown in figure 5. Clearly, RANEP performs better than the two variants in all cases, indicated by the obviously shorter bars in the histogram, followed by NEP-D, and RANEP-init performs the worst. A possible reason is that the incorporation of degree and reputation factor increases the capability of the prediction models. Meanwhile, in Table 2, the p -values indicate that we have to reject the *null hypothesis* that the sets of NMAE values are drawn from the same distribution in terms of the Wilcoxon signed-rank test (p -value = 0.005 < 0.05). In other words, there is a statistically significant difference between RANEP and the two variants from the perspective of prediction accuracy.

Therefore, we can conclude that, in the process of random walk, comprehensive consideration of the user’s node degree and reputation as its influence is helpful to the learning of

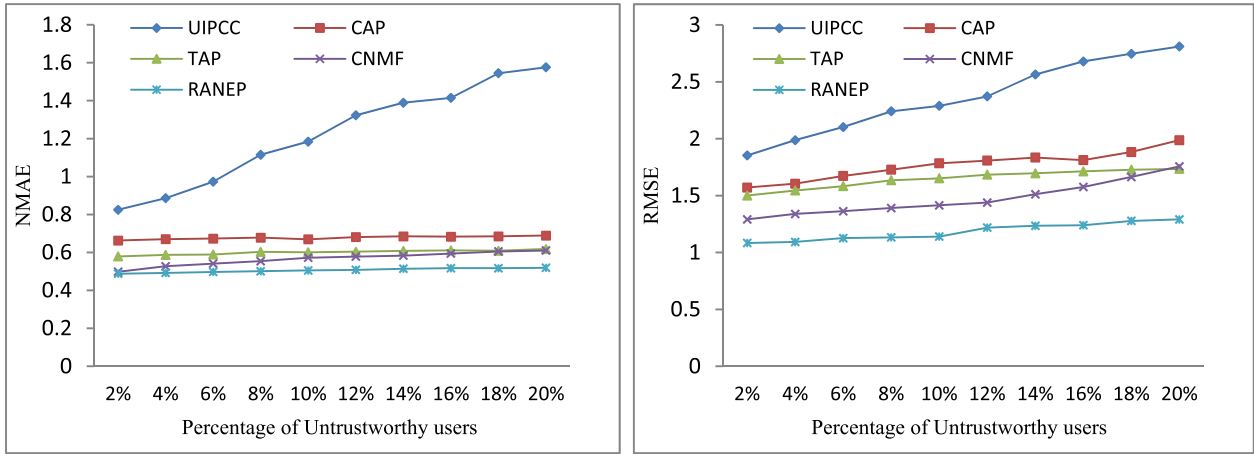


FIGURE 6. Robustness evaluation.

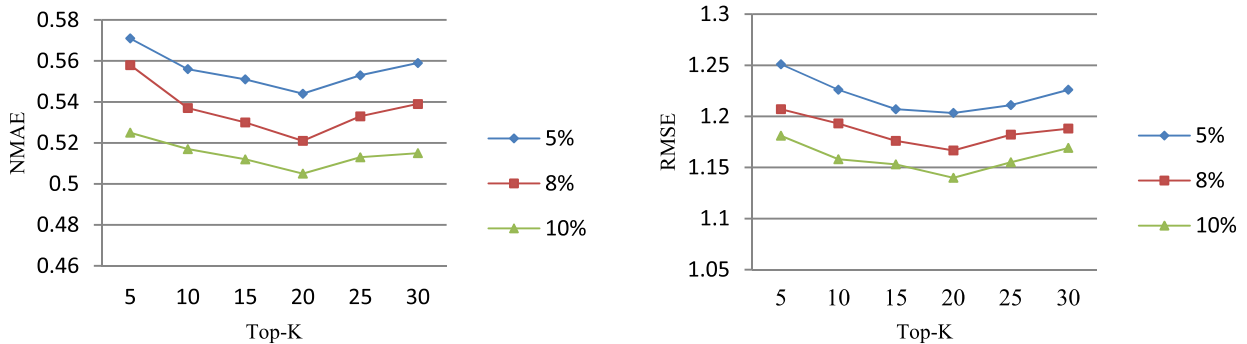


FIGURE 7. Impact of top-k.

TABLE 2. The wilcoxon signed-rank test (sig. p – value = 0.05).

	p-value
RANEP vs RANEP-init	0.005
RANEP vs NEP-D	0.005

user’s representation vector, so as to achieve the improvement of prediction accuracy.

RQ3 How do the key parameters affect the performance of RANEP?

In this section, we will analyze the robustness of RANEP by changing the key parameters. First, we vary the percentage of untrustworthy users from 2% to 20% by a step of 2%, to learn the influence of untrustworthy users on prediction accuracy. The *matrix density* is set as 10%, and the settings of other parameters are the same as those in **RQ1**. In Figure 6, as the percentage of untrustworthy user increases, the experimental results show that: 1) UIPCC is very sensitive to the impact of unreliable data due to the lack of identifying untrustworthy users, while the growth of NMAE and RMSE values of the other methods is very gently. 2) Although CNMF performs well when the percentage of untrustworthy users is small, its RMSE is even higher than TAP when the

proportion of untrustworthy users reaches 20%. Compared to CNMF, we can find that the growth of RANEP is relatively small. The reason may be that CNMF ignores the trustworthiness issue. 3) RANEP always has the smallest NMAE and RMSE values, and the NMAE of it only increases by 0.031 throughout. A slight growth in NMAE value indicates that our method is robust to the proportion of untrustworthy users.

Note that, the other parameter worth discussing is Top-*k*, which refers to the number of similar users used for the prediction. To study the impact of Top-*k*, we increase the value of Top-*k* from 5 to 30 with a step of 5. The other parameter settings are the same as those in **RQ1**. Figure 7 shows the prediction accuracy under three matrix density conditions: 5%, 8% and 10% respectively. The experimental results show that the NMAE and RMSE value initially decrease with the increase of *k*, and begins to increase when *k* is roughly over 20.

This validates that a suitable Top-*k* value does have an effect on the accuracy of prediction, too large or too small is not a good suggestion. This is because fewer similar users are returned to the target user, which may result in insufficient information available. At the same time, too large Top-*k* value also affects the performance since the set of similar users that contains devalued data is employed to make prediction. Besides, Figure 7 also shows that the optimal value of Top-*k* is

not influenced by *matrix density*. Thus, the parameter Top-*k* can be set as 20 in our context, and the prediction results have optimum performance.

V. CONCLUSION

In this paper, we introduce the network embedding method into the QoS prediction process, and propose a Web service QoS prediction method based on the reputation-aware network embedding. Firstly, we adopt the two-phase K-means clustering to identify untrustworthy users. Next, we calculate the reputation of trustworthy users, and construct an attributed user-service bipartite network while considering the reputation of users. Then we adopt the reputation-aware network embedding method to learn the hidden representations of users. Finally, we use collaborative filtering to predict the unknown QoS values. This method makes full use of user-service bipartite network information and user reputation information. Thus it can fully mine the high-order implicit relations between users (or services) from the historical data, and find more potential reliable similar users for the target users. It not only considers the impact of untrustworthy users on the prediction accuracy, but also alleviates the problem of data sparsity to some extent. A large number of experiments show that our method is more accurate and reliable than other methods.

Our method doesn't solve the problem of cold start, and we can't predict the QoS values of users who don't invoke any services. In the future, we will integrate the geographic location information of users and services for prediction to mitigate the impact of the cold start problem.

REFERENCES

- [1] K. Su, B. Xiao, B. Liu, H. Zhang, and Z. Zhang, "TAP: A personalized trust-aware QoS prediction approach for Web service recommendation," *Knowl.-Based Syst.*, vol. 115, pp. 55–65, Jan. 2017.
- [2] C. Wu, W. Qiu, Z. Zheng, X. Wang, and X. Yang, "QoS prediction of Web services based on two-phase K-Means clustering," in *Proc. IEEE Int. Conf. Web Services*, New York, NY, USA, Jun. 2015, pp. 161–168.
- [3] G. Zou, M. Jiang, S. Niu, H. Wu, S. Pang, and Y. Gan, "QoS-aware Web service recommendation with reinforced collaborative filtering," in *Proc. Int. Conf. Service-Oriented Comput.* New York, NY, USA: Springer, 2018, pp. 430–445.
- [4] T. Cheng, J. Wen, Q. Xiong, J. Zeng, W. Zhou, and X. Cai, "Personalized Web service recommendation based on QoS prediction and hierarchical tensor decomposition," *IEEE Access*, vol. 7, pp. 62221–62230, 2019.
- [5] J. Li, J. Wang, Q. Sun, and A. Zhou, "Temporal influences-aware collaborative filtering for QoS-based service recommendation," in *Proc. IEEE Int. Conf. Services Comput.*, Honolulu, HI, USA, Jun. 2017, pp. 471–474.
- [6] Q. Xie, S. Zhao, Z. Zheng, J. Zhu, and M. R. Lyu, "Asymmetric correlation regularized matrix factorization for Web service recommendation," in *Proc. IEEE Int. Conf. Web Services*, San Francisco, CA, USA, Jun. 2016, pp. 204–211.
- [7] Z. Chen, L. Shen, and F. Li, "Exploiting Web service geographical neighborhood for collaborative QoS prediction," *Future Gener. Comput. Syst.*, vol. 68, pp. 248–259, Mar. 2017.
- [8] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "Predicting Web service QoS via matrix-factorization-based collaborative filtering under non-negativity constraint," in *Proc. 23rd Wireless Opt. Commun. Conf. (WOCC)*, Newark, USA, May 2014, pp. 1–6.
- [9] Y. Xu, J. Yin, W. Lo, and Z. Wu, "Personalized Location-Aware QoS Prediction for Web Services Using Probabilistic Matrix Factorization," in *Pro. Web Information Systems Engineering, Nanjing, China*, vol. 2013, pp. 229–242.
- [10] P. He, J. Zhu, Z. Zheng, J. Xu, and M. R. Lyu, "Location-based hierarchical matrix factorization for Web service recommendation," in *Proc. IEEE Int. Conf. Web Services*, Anchorage, AK, USA, Jun. 2014, pp. 297–304.
- [11] X. Wu, B. Cheng, and J. Chen, "Collaborative filtering service recommendation based on a novel similarity computation method," *IEEE Trans. Services Comput.*, vol. 10, no. 3, pp. 352–365, May 2017.
- [12] W. Qiu, Z. Zheng, X. Wang, X. Yang, and M. R. Lyu, "Reputation-aware QoS value prediction of Web services," in *Proc. IEEE 10th Int. Conf. Services Comput.*, Santa Clara, Ca, USA, Jun. 2013, pp. 41–48.
- [13] K. Qi, H. Hu, W. Song, J. Ge, and J. Lu, "Personalized QoS prediction via matrix factorization integrated with neighborhood information," in *Proc. IEEE Int. Conf. Services Comput.*, New York, NY, USA, Jun. 2015, pp. 186–193.
- [14] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "Online QoS prediction for runtime service adaptation via adaptive matrix factorization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 10, pp. 2911–2924, Oct. 2017.
- [15] S. Li, J. Wen, F. Luo, and G. Ranzi, "Time-aware QoS prediction for cloud service recommendation based on matrix factorization," *IEEE Access*, vol. 6, pp. 77716–77724, 2018.
- [16] X. Zhu, X.-Y. Jing, D. Wu, Z. He, J. Cao, D. Yue, and L. Wang, "Similarity-maintaining privacy preservation and location-aware low-rank matrix factorization for QoS prediction based Web service recommendation," *IEEE Trans. Services Comput.*, early access, May 23, 2019, doi: [10.1109/TSC.2018.2839741](https://doi.org/10.1109/TSC.2018.2839741).
- [17] F. Zhuang, Z. Zhang, M. Qian, C. Shi, X. Xie, and Q. He, "Representation learning via dual-autoencoder for recommendation," *Neural Netw.*, vol. 90, pp. 83–89, Jun. 2017.
- [18] D. Ryu, K. Lee, and J. Baik, "Location-based Web service QoS prediction via preference propagation to address cold start problem," *IEEE Trans. Services Comput.*, early access, Apr. 2, 2018, doi: [10.1109/TSC.2018.2821686](https://doi.org/10.1109/TSC.2018.2821686).
- [19] Y. Jin, K. Wang, Y. Zhang, and Y. Yan, "Neighborhood-aware Web service quality prediction using deep learning," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, p. 222, Dec. 2019.
- [20] H. Wu, K. Yue, B. Li, B. Zhang, and C.-H. Hsu, "Collaborative QoS prediction with context-sensitive matrix factorization," *Future Gener. Comput. Syst.*, vol. 82, pp. 669–678, May 2018.
- [21] 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. 2011.
- [22] Y. Hu, Q. Peng, and X. Hu, "A time-aware and data sparsity tolerant approach for Web service recommendation," in *Proc. IEEE Int. Conf. Web Services*, Anchorage, AK, USA, Jun. 2014, pp. 33–40.
- [23] S. H. Ghafouri, S. M. Hashemi, and P. C. K. Hung, "A survey on Web service QoS prediction methods," *IEEE Trans. Services Comput.*, early access, Mar. 16, 2020, doi: [10.1109/TSC.2020.2980793](https://doi.org/10.1109/TSC.2020.2980793).
- [24] J. Xu, Z. Zheng, and M. R. Lyu, "Web service personalized quality of service prediction via reputation-based matrix factorization," *IEEE Trans. Rel.*, vol. 65, no. 1, pp. 28–37, Mar. 2016.
- [25] D. Wu, X. Luo, M. Shang, Y. He, G. Wang, and M. Zhou, "A deep latent factor model for high-dimensional and sparse matrices in recommender systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Aug. 15, 2019, doi: [10.1109/TSMC.2019.2931393](https://doi.org/10.1109/TSMC.2019.2931393).
- [26] X. Luo, M. Zhou, S. Li, Y. Xia, Z.-H. You, Q. Zhu, and H. Leung, "Incorporation of efficient second-order solvers into latent factor models for accurate prediction of missing QoS data," *IEEE Trans. Cybern.*, vol. 48, no. 4, pp. 1216–1228, Apr. 2018.
- [27] D. Wu, Q. He, X. Luo, M. Shang, Y. He, and G. Wang, "A posterior-neighborhood-regularized latent factor model for highly accurate Web service QoS prediction," *IEEE Trans. Services Comput.*, early access, Dec. 24, 2019, doi: [10.1109/TSC.2019.2961895](https://doi.org/10.1109/TSC.2019.2961895).
- [28] D. Wu, X. Luo, and M. Shang, "A data-aware latent factor model for Web service QoS prediction," in *Advances in Knowledge Discovery and Data Mining*, vol. 99, 2020, p. 1.
- [29] J. Zhu, P. He, Q. Xie, Z. Zheng, and M. R. Lyu, "CARP: Context-aware reliability prediction of black-box Web services," in *Proc. IEEE Int. Conf. Web Services*, Honolulu, HI, USA, Jun. 2017, pp. 17–24.
- [30] Z. Zheng, L. Xiaoli, M. Tang, F. Xie, and M. R. Lyu, "Web service QoS prediction via collaborative filtering: A survey," *IEEE Trans. Services Comput.*, early access, May 18, 2020, doi: [10.1109/TSC.2020.2995571](https://doi.org/10.1109/TSC.2020.2995571).
- [31] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, "Personalized QoS-aware Web service recommendation and visualization," *IEEE Trans. Services Comput.*, vol. 6, no. 1, pp. 35–47, 1st Quart., 2013.

- [32] Y. Zhang, K. Wang, Q. He, F. Chen, S. Deng, Z. Zheng, and Y. Yang, "Covering-based Web service quality prediction via neighborhood-aware matrix factorization," *IEEE Trans. Services Comput.*, early access, Jan. 9, 2019, doi: 10.1109/TSC.2019.2891517.
- [33] S. Ding, Y. Li, D. Wu, Y. Zhang, and S. Yang, "Time-aware cloud service recommendation using similarity-enhanced collaborative filtering and ARIMA model," *Decis. Support Syst.*, vol. 107, pp. 103–115, Mar. 2018.
- [34] M. Gao, L. Chen, X. He, and A. Zhou, "BiNE: Bipartite network embedding," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 715–724.
- [35] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 3111–3119.
- [36] Z. Zheng, Y. Zhang, and M. R. Lyu, "Distributed QoS evaluation for real-world Web services," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Washington, DC, USA, Jul. 2010, pp. 83–90.
- [37] Y. Zhang, Z. Zheng, and M. R. Lyu, "Exploring latent features for memory-based QoS prediction in cloud computing," in *Proc. IEEE 30th Int. Symp. Reliable Distrib. Syst. (SRDS)*, Madrid, Spain, Oct. 2011, pp. 1–10.



PENG HE was born in 1988. He received the Ph.D. degree from Wuhan University, China. His current research interests include service-oriented software engineering, software quality analysis, and defect prediction.



JIANXIONG ZHANG is currently pursuing the master's degree with Hubei University. His current research interests include representation learning and service recommendation.



XUECHUN WANG is currently pursuing the master's degree with the School of Computer and Information Engineering, Hubei University. Her research interests include representation learning, neural networks, and service computing.



ZHAN WANG is currently pursuing the master's degree with Hubei University. His research interests mainly include data mining and machine learning.

...