NDMF: Neighborhood-Integrated Deep Matrix Factorization for Service QoS Prediction

Guobing Zou, Jin Chen, Qiang He, Kuan-Ching Li, *Senior Member, IEEE*, Bofeng Zhang, *Member, IEEE*, and Yanglan Gan

***Abstract*—Quality of service (QoS) has been mostly applied to represent non-functional properties of Web services and differentiate those with the same functionality. How to accurately predict service QoS has become a key research topic. Researchers have employed neighborhood information into matrix factorization (MF) for service QoS prediction in recent years. However, they are restricted to traditional matrix factorization that may incur a couple of limitations. 1) Conventional MF for QoS prediction linearly combines the multiplication of the latent feature representation of users and services through inner product, failing to fully capture the implicit features of user and service. 2) Most of approaches integrate user or service neighborhood as heuristics into MF model, where either location context or historical invocation records are used to calculate similar users or services. Nevertheless, combining both of them together in a collaborative way is ignored for neighborhood selection that has yet to be properly explored. To deal with the challenges, we propose a novel approach for service QoS prediction called *N*eighborhood-integrated *D*eep *M*atrix *F*actorization (NDMF), which integrates user neighborhood selected by a collaborative way into an enhanced matrix factorization model via deep neural network (DNN). We implement a prototype system and conduct extensive experiments on public and real-world large Web service dataset with almost 2,000,000 service invocations called WS-DREAM which is widely used in service QoS prediction. The experimental results demonstrate that our proposed approach significantly outperforms state-of-the-art ones in terms of multiple evaluation metrics.**

***Index Terms*—Web services, QoS prediction, matrix factorization, deep neural network, neighborhood selection.**

Manuscript received January 3, 2020; revised June 25, 2020 and September 20, 2020; accepted September 24, 2020. Date of publication September 28, 2020; date of current version December 9, 2020. This work was partially supported by National Key Research and Development Program of China (2017YFC0907505), Shanghai Natural Science Foundation (18ZR1414400), and National Natural Science Foundation of China (61772128). The associate editor coordinating the review of this article and approving it for publication was Y. Diao. *(Corresponding author: Yanglan Gan.)*

Guobing Zou, Jin Chen, and Bofeng Zhang are with the School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China (e-mail: gbzou@shu.edu.cn; cj1125@shu.edu.cn; bfzhang@shu.edu.cn).

Qiang He is with the Department of Computer Science and Software Engineering, Swinburne University of Technology, Melbourne, VIC 3122, Australia (e-mail: qhe@swin.edu.au).

Kuan-Ching Li is with the School of Computer Science and Information Engineering, Providence University, Taichung 43301, Taiwan (e-mail: kuancli@pu.edu.tw).

Yanglan Gan is with the School of Computer Science and Technology, Donghua University, Shanghai 201620, China (e-mail: ylgan@dhu.edu.cn).

Digital Object Identifier 10.1109/TNSM.2020.3027185

I. Introduction

WEB SERVICES are widely used in service-oriented architecture (SOA) to rapidly integrate and develop software applications, since they are self-described and platform-independent components with the characteristics of loose coupling, reusability and composability. It accelerates the interoperable machine-to-machine interaction and greatly promotes the advancements on service discovery, selection, composition and recommendation. Along with the popularity of cloud and edge computing paradigm, more and more services are published on the Internet by software vendors and consumed by service requesters. Consequently, the number of Web services grow exponentially in recent years. However, as the overwhelming explosion on the number of Web services, many of them share the same or similar functionality that leads to be a labor-intensive challenging task for inexperienced service requesters to choose their desired services from a large-scale service repository, as faced with multiple candidate services of similar functionalities, when building their service-oriented Web applications. Therefore, recommending satisfactory Web services for a target user from those functionally equivalent or similar ones has been a critical issue that needs to be addressed.

Quality of Service (QoS) as a non-functionality criterion has been widely applied as a key factor to differentiate those Web services with the same functionality. In many cases, however, it is uneasy to obtain QoS of a Web service invoked by a target user. The main reason is two-fold: On one hand, the quality of a service invoked by a user depends on the contextual information of both the user and service itself. For example, a Web service invoked by different users holds different QoS values because of the changes of user geographical location and network environment. On the other hand, it is a time-consuming and resource-consuming task for service providers to monitor QoS information for each service invocation, leading to the large sparsity of user-service invocations due to very few historical QoS values. That is, it still has difficulty in recommending services to a target user due to vacant service QoS. To deal with this challenge, many efforts have been done to predict the unknown service QoS by exploiting the known service QoS invocations.

Among the existing QoS prediction approaches, collaborative filtering (CF) is the most widely used technique. CF-based QoS prediction approaches can be divided into two categories, including memory-based and model-based. Memory-based CF aims at predicting service QoS via similar users or services that can be calculated by historical invocation records [1]–[3]. Moreover, there are also some memory-based investigations [4], [5] for service QoS prediction, where contextual information such as location and temporal information is taken into account to eliminate those dissimilar users or services. In this way, it optimizes the neighborhood selection and boosts the procedure of QoS prediction. However, this kind of approaches are vulnerable to data sparsity of user-service invocations, resulting in lower QoS prediction accuracy.

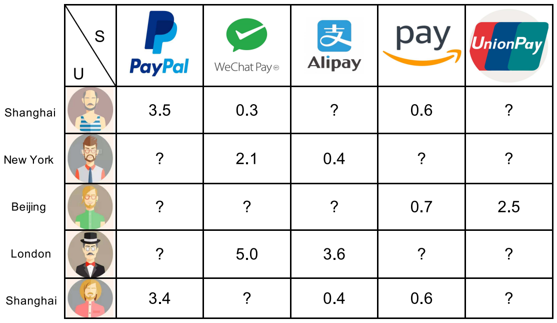
To alleviate the influence on data sparsity, model-based CF approaches usually employ matrix factorization (MF) technique to learn latent factors of users and services from historical QoS invocations. By modeling the user-service invocation relationship with inner product of the user and service latent vector, it generally receives better QoS prediction performance than memory-based approaches. Furthermore, researchers also utilize neighborhood information as the auxiliary heuristics, i.e., those users/services similar to the target user/service, to reinforce the model training of matrix factorization and improve the accuracy of MF-based QoS prediction [6]–[9].

Currently, state-of-the-art MF-based approaches for service QoS prediction take historical QoS invocations as input, calculate user or service neighborhood for training matrix factorization model, and finally predict service QoS values. However, two problems arise in this paradigm.

First, traditional MF for QoS prediction linearly combines the multiplication of the latent vector of users and services through inner product, failing to fully capture the latent information of the user and service latent feature vectors. Although the variations of MF may be applied to improve the performance of model training by incorporating user and service bias terms into the interaction function [10], there is still lack of consideration on deeply exploiting user and service latent information for achieving better QoS prediction accuracy. Consequently, it cannot inherently avoid the disadvantages of interaction function for service QoS prediction.

Second, it is observed that neighborhood selection is not only aware of contextual information such as user/service geographical location, but also user-service interactions such as QoS records of historical invocations. Nevertheless, there is still no such consideration of combining both of them together for performing better selection of user and service neighborhood selection as heuristics into MF model.

To address the above two issues, we propose a novel collaborative approach for service QoS prediction named Neighborhood-integrated Deep Matrix Factorization (NDMF). Specifically, we use a deep neural network (DNN) to match the complex non-linear interaction relationship between the latent features of user and service, which can overcome the limitation of inner product in traditional matrix factorization. Simultaneously, we reinforce the neighborhood selection by fusing user geographical location and user-service invocation QoS. To test the performance of service QoS prediction, extensive experiments are conducted on a public and large-scale real-world dataset called WS-DREAM [11], involving 5,825 real-world Web services in 73 countries and 339 service users in 30 countries. Comparing our proposed approach with several state-of-the-art competing ones, the results demonstrate the effectiveness and efficiency of our proposed approach NDMF in multiple evaluation metrics for service QoS prediction.



	U \ S	PayPal	WeChat Pay	Alipay	pay	UnionPay
Shanghai		3.5	0.3	?	0.6	?
New York		?	2.1	0.4	?	?
Beijing		?	?	?	0.7	2.5
London		?	5.0	3.6	?	?
Shanghai		3.4	?	0.4	0.6	?

Fig. 1. The motivating example of user-service invocation QoS for online payment service.

The main contributions of this article are summarized as follows:

- We propose a novel collaborative framework for service QoS prediction via deep neural network. Compared to the traditional MF-based approaches, the advantage is that we can more deeply reveal the implicit features of user and service by complex non-linear interaction function, which leads to better performance of service QoS prediction.
- We propose a comprehensive approach for measuring the relevance among users by fusing both user geographical information and user-service invocation QoS, which achieves better performance on user neighborhood selection than traditional approaches.
- We design and implement a prototype system and conduct extensive experiments on WS-DREAM dataset. The experimental results validate the effectiveness of our proposed approach NDMF, which is superior to existing competing approaches in terms of accuracy of service QoS prediction.

The remainder of this article is organized as follows. Section II illustrates the motivation of collaborative QoS prediction. Problem formulation is presented in Section III. Section IV elaborates our approach of NDMF for service QoS prediction. Section V shows and analyzes the experimental results. Section VI reviews the related work. Finally, Section VII concludes this article and discusses the future works.

II. Motivating Example

In this section, we initially give a motivating example in e-commerce application that demonstrates the procedure of neighborhood-based MF approach for service QoS prediction.

Fig. 1 illustrates a motivating example with five users and five Web services, where the users come from the same or different cities and services have the same functionality of online payment. The QoS matrix contains response time of Web services experienced by users and it can be denoted as $R = [r_{u,i}]_{m \times n}$, where m is the number of users and n is the number of Web services. Each entry $r_{u,i}$ in QoS matrix, $1 \leq u \leq m, 1 \leq i \leq n$, is the QoS value of Web service i

invoked by user *u*. An "?" entry represents an unknown QoS value to be predicted. We denote each user and service as u_x, i_x by its sequence such as u_1 for the first user and i_1 for the first service. According to [7], if users are located in a close physical place, they usually share similar IT infrastructure and routing protocols. Then, they would have similar QoS experiences when they invoke the same Web services. For example, u_1 and u_5 are both located in Shanghai, they invoked i_1, i_4 and received similar QoS experiences. Nevertheless, u_2 and u_4 are from different countries, and they have totally different QoS when invoking the same services i_3.

By calculating the similarity of users based on their observed QoS values [3], [12], we can find user neighborhood for further service QoS prediction. Motivated by the existing work, we integrate users' geographical locations by using Autonomous System (AS) into the neighborhood selection to improve the accuracy of service QoS prediction.

After achieving the neighbors in a collaborative way, traditional neighborhood-integrated matrix factorization performs the procedure of modeling the relationship of feature vectors of users and services with an inner product function. Once it reaches convergence, a complete user-service invocation matrix can be generated without vacant QoS value. As discussed above, the traditional matrix factorization model has not accurately performed the task of service QoS prediction, since it mainly takes a linear strategy to combine the latent feature of users and services. To solve the issue, we upgrade the traditional matrix factorization model by deep neural network and an improved neighborhood selection to further boost the accuracy of service QoS prediction.

III. Problem Formulation

In this section, we first focus on the understanding of service QoS prediction problem by a set of formal definitions, and then clearly demonstrate what the solution is to a service QoS prediction problem.

Definition 1 (Web Service): A Web service can be described as a five-tuple $i = <n, f, d, q, l>$, where *n*, *f* and *d* represent service name, functionality description and domain tag. *q* is the service dimension and *l* indicates the location information of a service.

For service QoS prediction, we mainly focus on the non-functional features of a Web service including QoS dimension *q* and its location information *l*, rather than service functionality and domain features.

Definition 2 (Service User): Given a service user *u*, it can be described as a two-tuple $u = <id, l>$. *id* is the identity label of *u* and *l* is the location information.

Generally, the location information of a service user mainly includes IP, AS and country. The neighborhood set *N*(*u*) is defined as a set of users who hold the same or similar QoS experiences when invoking Web services.

Definition 3 (User-Service Invocation Record): Given a user set *U* and a service set *I*, a user-service invocation record is defined as a three-tuple $<u, i, r_{u,i}>$, where $u \in U$ is a service user, $i \in I$ is a Web service, and $r_{u,i}$ is the QoS value when *u* invokes *i*.

By the invocations of Web services, all of the user-service invocation records can be represented as a QoS matrix, denoted as *R*. Each row represents the QoS of a user who invokes all of the Web services, and each column represents QoS of a Web service that is invoked by all of the users. Note that if an entry of a user-service invocation matrix is equal to vacant, indicating that a user has not ever invoked this service. In such case, it needs to be predicted for further use, which is defined as below.

Definition 4 (QoS Prediction Problem): Given a user set *U*, a service set *I* and all observed QoS invocation records *R*, the QoS prediction problem is defined as a five-tuple $Q = <U, I, R, u, i>$, where *u* is a target user, *i* is a target service and $r_{u,i}$ has no invocation record that needs to be predicted.

The solution to a QoS prediction problem is $<u, i, \hat{r}_{u,i}>$. It indicates the predicted QoS when a target user invokes a target service. By predicting missing QoS values on each service in *I*, we have their predicted QoS values. In terms of the ranking of predicted QoS values, a subset of Web services with the equivalent or similar functionality can be recommended to a target user. To achieve this objective, we propose a neighborhood-integrated deep matrix factorization approach called NDMF that is illustrated and elaborated in the subsequent section.

IV. Approach

In this section, we discuss NDMF in detail. We first present the overall framework, then elaborate the neural user-service feature interaction model, and subsequently introduce the collaborative neighborhood selection. Finally, we fuse these two components together and train the model for service QoS prediction.

A. The Overall Framework of the Approach

Starting from the historical Web service invocation records, the framework and its procedure is shown in Fig. 2. It mainly consists of two crucial components, including neural user-service feature interaction modeling and collaborative neighborhood selection.

When modeling the neural user-service feature interaction, we first use the identity of a user and a service as the inputs, by transforming them into a high-dimensional and sparse binarized vectors with one-hot encoding, then learn them as dense vector representations, respectively. After that, a non-parametric operation is performed on users' and services' dense feature vectors to concatenate latent representations that are fed into the deep neural network of user-service interaction to derive the non-linear relationship between users and services. The collaborative user neighborhood selection finds out user neighborhood by combining user geographical context and user-service historical QoS records for the model training. Consequently, NDMF model trained by integrating the implicit user-service features and user neighborhood can predict service QoS value.

Specifically, NDMF is divided into two parts as illustrated in Fig. 3. The left part is used to model neural user-service feature interaction and learn an effective non-linear relationship

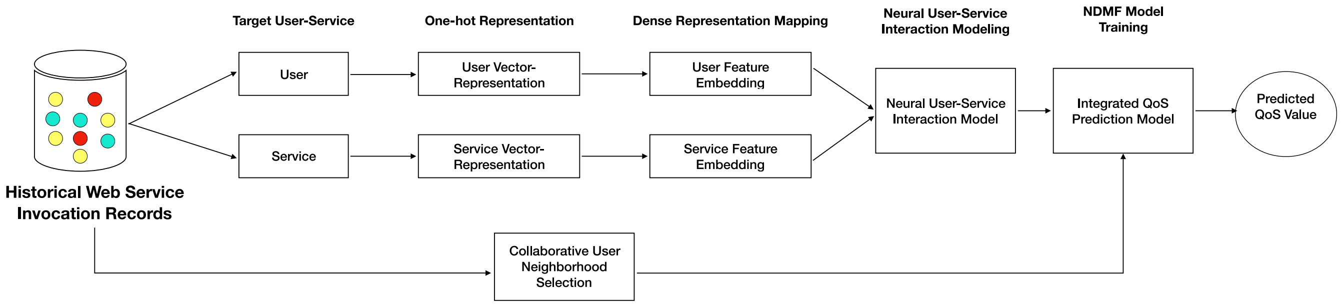

Target User-Service
One-hot Representation
Dense Representation Mapping
Neural User-Service Interaction Modeling
NDMF Model Training
User
Service
User Vector-Representation
Service Vector-Representation
User Feature Embedding
Service Feature Embedding
Neural User-Service Interaction Model
Integrated QoS Prediction Model
Predicted QoS Value
Historical Web Service Invocation Records
Collaborative User Neighborhood Selection

Fig. 2. The overall framework of the approach.

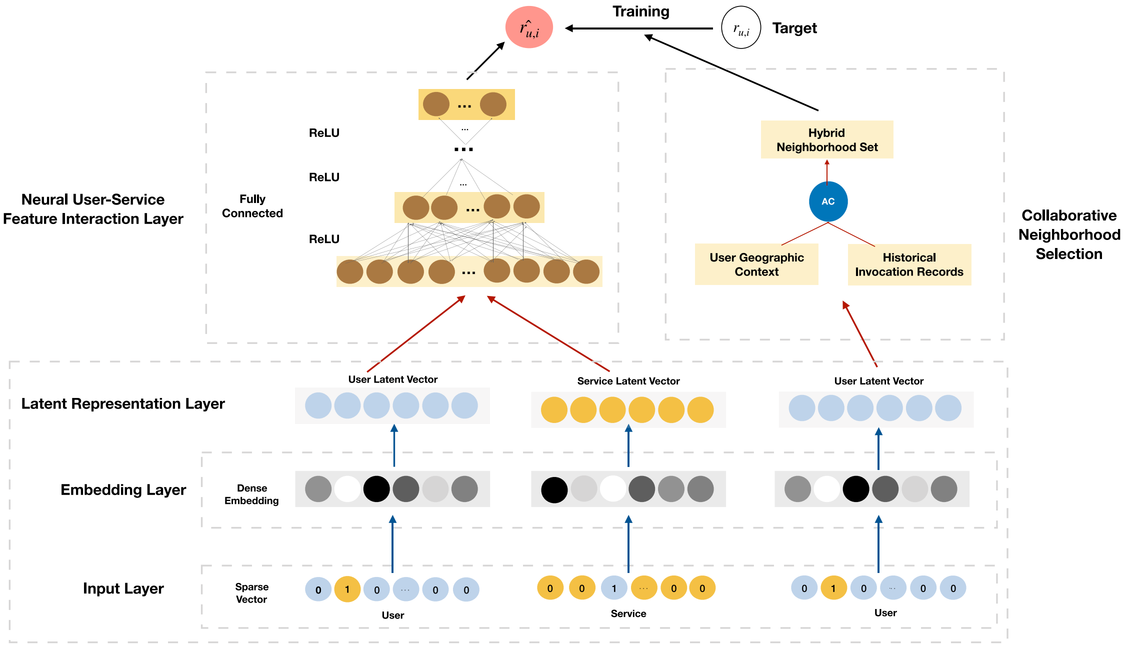


Fig. 3. QoS prediction integrating user-service deep neural interaction feature and hybrid collaborative user neighborhood.

function between users and services and the right part aims at choosing user neighborhood in a collaborative way.

B. Neural User-Service Feature Interaction

MF associates each user and item with a real-valued vector of latent features. Let p_u and q_i denote the latent vector for user u and service i, respectively. It predicts an interaction $\hat{y}_{u,i}$ as the inner product of p_u and q_i:

$$\hat{y}_{u,i} = f(u, i|p_u, q_i) = p_u^T q_i = \sum_{k=1}^{K} p_{u,k} q_{i,k} \qquad (1)$$

where K denotes the dimension of the latent space. It is observed that MF models the two-way interaction of user and service latent factors, assuming each dimension of the latent space is independent of another one and they are linearly combined with the same weight. As such, MF can be deemed as a linear model of latent factors.

To permit a full non-linear treatment of matrix factorization, we adopt a neural user-service feature interaction model to extract implicit features of user-service interaction as shown in Fig. 3. It consists of multi layers with the deep neural network, where the output of one layer serves as the input of the next one. The predictive model of neural user-service feature interaction can be represented as:

$$\hat{y}_{u,i} = f\left(P^T v_u^U, Q^T v_i^I | P, Q, \Theta_f\right) \qquad (2)$$

where $P \in \mathbb{R}^{M \times K}$ and $Q \in \mathbb{R}^{N \times K}$, indicating the latent factor matrix for users and services, respectively; v_u^U and v_i^I are the initial representations of user u and service i; Θ_f denotes model parameters of the feature interaction function f. Since f is defined as a multi-layer deep neural network, it can be formulated as:

$$\begin{aligned} & f\left(P^T v_u^U, Q^T v_i^I\right) \\ & \quad = \phi_{out}\left(\phi_X\left(\dots \phi_2\left(\phi_1\left(P^T v_u^U, Q^T v_i^I\right)\right)\dots\right)\right) \qquad (3) \end{aligned}$$

where ϕ_{out} and ϕ_x denote the mapping function for output layer and the x-th neural user-service feature interaction layer, respectively.

To draw the predictive model, we mainly apply two procedures, including latent feature mapping and neural feature interaction between user and service.

1) User-Service Latent Feature Mapping Layer: NDMF maps target user and service from sparse high-dimensional vectors into dense low-dimensional vectors. It can be partitioned into two parts, i.e., input layer and embedding layer.

The Input Layer identifies a user and a service and generates its initial feature representation, by transforming it into a binarized sparse vector with one-hot encoding. It is a high-dimensional zero vector with a specified dimension that is set to be one and represents the corresponding user or service. Suppose that if there are 3 users total, then the dimensionality of one-hot vector $d = 3$, then the user with $id = 1$ can be represented as $u_1 = [1, 0, 0]$ and the user with $id = 2$, 3 can be represented as $u_2 = [0, 1, 0]$ and $u_3 = [0, 0, 1]$, respectively.

After transforming user and service into sparse high-dimensional vectors, they are further mapped into low-dimensional dense vectors with Embedding Layer. The Embedding Layer is regarded as a special fully-connected layer without bias term. Similar to Word2Vec [13], Doc2Vec [14] and GloVe [15], our embedding takes user/service high-dimensional sparse vectors as input and turns out dense low-dimensional vector. The mapping process is formalized as:

$$p_u = f_e\left(U_e^T h_u\right) \tag{4}$$

$$q_i = f_e\left(I_e^T h_i\right) \tag{5}$$

where h_u and h_i represent one-hot encoding vectors of user u and service i, respectively; U_e and I_e represent user's and service's embedding weight matrix; f_e is the activation function.

Vectors in the Latent Representation Layer are all dense and low-dimensional, which represent users' and services' latent features, and they can be adapted through the process of back propagation. Since the task is to predict the QoS of any user invoking any service, we must utilize the latent features of both specified user and service. They are concatenated as input for the neural user-service feature interaction layer to learn a non-linear relationship function between users and services. The concatenation of two latent features is expressed as:

$$x = \Phi(p_u, q_i) = [p_u, q_i] \tag{6}$$

where Φ represents the concatenation operation, p_u and q_i denotes the embedding vector of a user and service, and x is the input vector in neural user-service feature interaction layer.

2) Neural User-Service Feature Interaction Layer: Upon the joint latent feature, a multi-layer fully connected deep neural network is trained and used to predict the unknown QoS of a user invoking a service [10]. Each layer in the neural user-service feature interaction layer can be customized to mine certain latent structures of interactions. In this way, more complex and non-linear interactions between p_u and q_i can be learned, rather than only linear inner product in traditional MF training. The forwarding procedure in neural user-service feature interaction layer is expressed as:

$$\begin{aligned} z_1 &= \phi(p_u, q_i) = [p_u, q_i] \\ \phi(z_1) &= a_2\left(W_2^T z_1 + b_2\right) \\ &\vdots \\ \phi_x(z_{x-1}) &= a_x\left(W_x^T z_{x-1} + b_L\right) \end{aligned}$$

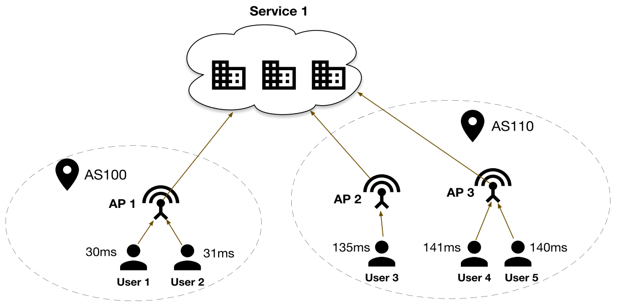


Fig. 4. User neighborhood selection within different autonomous systems.

$$\hat{y}_{u,i} = \sigma\left(h^T \phi(z_{x-1})\right) \tag{7}$$

where W_x, b_x, and a_x denote the weight matrix, bias vector and activation functions, respectively. Here, Rectified Linear Unit (ReLu) is applied as the activation function. Furthermore, the typical tower pattern [10], is used to as the architecture in our deep neural network for user-service feature interaction shown in Fig. 3. The output of the neural user-service feature interaction layer is the predictive QoS for the target user on target service $\hat{r}_{u,i}$, where user neighborhood is taken into account during the model training.

C. Collaborative Neighborhood Selection

To further improve the accuracy of service QoS prediction, neighborhood selection in a collaborative way is performed by combining user geographical locations and historical QoS of user-service invocations.

1) Location-Based Neighborhood Selection: Autonomous system (AS) as a small unit decides which routing protocol should be used in the system on the Internet. In many application scenarios, AS can be a kind of simple network or a network group controlled by one or more common network administrators such as a university or a company.

It is a consensus that users who share the same access point would have similar network latency when invoking a Web service, and users with different access points in the same AS would have minor differences when invoking a Web service. Through many statistics and calculations, it's proved that users with smaller network distances or within closer physical regions indeed have more similar QoS values [7]. For example, Web services can be invoked by users with different geographical locations as shown in Fig. 4, a real-world Web service invocation scenario with multiple ASs and access devices. Suppose that User 1, User 2 are located in AS100, while User 3 and User 4, User 5 are located in AS110 with different access devices. The arrows among users, access points and the cloud represent service invocations, and the numbers around the users represent the time measurements of network latency of user-service invocations. We can see from Fig. 4, User 1 and User 2 with AP-1 in AS100 have the similar network latency, while User 3 with AP-2, User 4, User 5 with AP-3 in AS110 have the similar network latency between different APs when invoking Service 1, although there are variable network latencies with those users distributed in different ASs.

By this observation, the users who share the same AS are selected as the neighborhood in terms of users' physical and network locations. However, in some cases, AS can span considerably wide geographical areas. In such case, although users are located in the same AS, they may be a long distance away with different network environment, leading to diverse QoS experiences and disturbing the neighborhood selection. In order to reduce inaccurate neighborhood selection, user-service historical invocations and Adaptive Corrector (AC) mechanism are also applied to exclude these latent inaccurate neighbors by AS-based neighborhood selection.

2) Memory-Based Neighborhood Selection: Although Pearson Correlation Coefficient (PCC) is the most widely used approach to calculate similarity, it remains some disadvantages analyzed in [3] to select user neighborhood.

Sun *et al.* [3] argue that PCC does not take the differences of QoS values given by different users into account. To overcome this issue, we apply ratio-based similarity (RBS) [12] method for memory-based neighborhood selection. The similarity between two users is calculated by:

$$Sim(u,v) = \frac{\sum_{i \in I} \frac{min(r_{u,i}, r_{v,i})}{max(r_{u,i}, r_{v,i})}}{|I|} \tag{8}$$

where $r_{u,i}$ and $r_{v,i}$ are the QoS values when user *u* and *v* invoke service *i*; $min(r_{u,i}, r_{v,i})$ is the minimum value of $r_{u,i}$ and $r_{v,i}$; $max(r_{u,i}, r_{v,i})$ is the maximum value of them; $I = I_u \cap I_v$ is the co-invoked services set of user *u* and *v*; $|I|$ denotes the number of services in *I*.

After calculating the user similarity with RBS, we have user neighborhood in terms of QoS values of service invocations.

3) Adaptive Corrector (AC): After achieving the two user neighborhood sets from physical location and QoS values of service invocations, we apply AC to determine the final collaborative user neighborhood.

The process of merging two user neighborhood sets is shown in Algorithm 1. If we aim to select *k* number of neighbors for a target user *u*, we separately find *top-k* similar users by location-based and memory-based approach. There are two possibilities in collaborative neighborhood selection by AC. One possibility is that if the size of location-based neighborhood set $N_l(u)$ is equal to *k*, then memory-based similarity set $N_m(u)$ is used through intersection operation to generate *N*(*u*). Subsequently, we check the size of *N*(*u*). If $len(N(u)) < k$, $N_l(u)$ is united into *N*(*u*); In such case, if $len(N(u))$ is still smaller than *k*, we take the subset of $N_m(u)$ to make the size of *N*(*u*) equal to *k*, where those selected neighborhood users are not included in *N*(*u*) and have the highest similarity with the target user *u*. Another possibility is that if the size of $N_l(u) < k$, then we combine it and the subset of $N_m(u)$ into $N^k(u)$, where those selected users from $N_m(u)$ hold the highest similarity with the target user.

Algorithm 1: Adaptive Corrector of Neighborhood Selection

```
Input:   (1) Ni(u) //neighborhood of location-based approach (top-k)
         (2) $N_m(u)$ //neighborhood of memory-based approach (top-k)
         (3) k //the predefined number of neighborhood
Output:  $N^k(u)$ //collaborative neighborhood set (top-k)
if $len\ (N_l(u)) = k$ then
    $N(u) = N_l(u) \cap N_m(u)$
    if $len(N(u)) < k$ then
        $N(u) = N(u) \cup N_l(u)$
    end
    if $len(N(u)) < k$ then
        $N^k(u) = N(u) \cup N_m^2(u), where\ \forall u_n \in N_m^2(u), u_n \notin N(u),$
                  $and\ len(N_m^2(u)) = k - len(N(u)$
    end
end
if $len(N_l(u)) < k$ then
    $N(u) = N_l(u)$
    $N^k(u) = N(u) \cup N_m^2(u), where\ \forall u_n \in N_m^2(u), u_n \notin N(u),$
              $and\ len(N_m^2(u)) = k - len(N(u))$
end
```

D. Model Training and QoS Prediction

In this section, we first elaborate how to train the model by integrating the neural user-service feature interaction with the collaborative neighborhood selection. Upon the trained model, we describe QoS prediction for service recommendation.

1) Loss Function and Model Optimization: Matrix factorization is one of the most popular and effective techniques for predicting missing values by revealing the latent features. In traditional rating-based recommender systems, MF maps both users and items to a joint latent factor space of dimensionality *d*, such that ratings are modeled as inner products in that space. The premise behind MF is that there are a few potential factors that affect the user's preference on items. In our NDMF framework, it divides an $m \times n$ user-service QoS matrix *R* into two parts *U* and *S* with the dimensionality of latent factors *d*:

$$R \approx U^T S \tag{9}$$

where $U \in \mathbb{R}^{d \times m}$ and $S \in \mathbb{R}^{d \times n}$ represent user and service latent matrices, respectively.

The objective function used to approximate the original user-service QoS matrix *R* with *U* and *S* by minimizing the following term is expressed:

$$\underset{U,S}{min} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} \left\| R_{i,j} - U_i^T S_j \right\|_F^2 \tag{10}$$

where $||\cdot||$ denotes the *Frobenius norm* [16] that calculates the difference between real value R_{ij} and the corresponding estimated value calculated through $U_i^T S_j$, where *U* and *S* represent the matrices of user and service latent feature vectors, respectively. Since each user generally invoked a small number of services, the QoS matrix *R* is always sparse. Therefore, (10) can be improved as:

$$\underset{U,S}{min} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{i,j} \left(R_{i,j} - U_i^T S_j \right)^2 \tag{11}$$

where $I_{i,j}$ is the indicator function that it is equal to 1 if user U_i invoked service S_j or 0 otherwise. To obtain the optimal *U* and *S* approximate to *R*, two regularization terms related to *U* and *S* are introduced for the purpose of overfitting avoidance, which is expressed:

$$\underset{U,S}{min} \mathcal{L}(R,U,S) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{i,j} \left(R_{i,j} - U_i^T S_j \right)^2 + \frac{\lambda_1}{2} ||U||_F^2 + \frac{\lambda_2}{2} ||S||_F^2 \tag{12}$$

where λ_1 and λ_2 are two tunable parameters that control the regularization degree. It is desired to minimize the sum of squared errors in (12). Since it is non-convex objective function, it is difficult to find the global minimum value [17]. Instead, the stochastic gradient descent technique [16] is employed to find the sub-optimal solution with (13):

$$\begin{aligned} U_i' &= U_i - \gamma \frac{\partial \mathcal{L}}{\partial U_i} \\ S_j' &= S_j - \gamma \frac{\partial \mathcal{L}}{\partial S_j} \end{aligned} \tag{13}$$

where $\gamma > 0$ represent the learning rate.

In our NDMF framework, to further utilize the user neighborhood information, NDMF integrates the following term into the basic objective function:

$$min \sum_{j \in N(u)} \left\| U_i - U_j \right\|_F^2 \tag{14}$$

where U_i and U_j are the user latent feature vectors from user embedding matrix U; U_i denotes the latent feature vector of the target user *i* and $U_j,\ j \in N(u)$ denotes its latent feature vector of neighborhood. It aims to minimize the latent difference between each user and the corresponding neighborhood to facilitate personalized QoS prediction, which can be regarded as a means of correcting and optimizing the representations of user latent feature vectors.

By incorporating (12) and (14), the objective function of NDMF framework is transformed into:

$$\begin{aligned} \min_{U,S} \mathcal{L}'(R, U, S) = {} & \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{i,j} \left(R_{i,j} - U_i^T S_j \right)^2 \\ & + \frac{\lambda_1}{2} ||U||_F^2 + \frac{\lambda_2}{2} ||S||_F^2 \\ & + \frac{\alpha}{2} \sum_{i=1}^{m} \sum_{j \in N(u)} \left\| U_i - U_j \right\|_F^2 \end{aligned} \tag{15}$$

where $\alpha > 0$ is a tunable parameter that controls the importance of difference between target user and the corresponding neighborhood.

Similarly, we employ the gradient descent method to approximately optimize the NDMF objective function and expectantly reach the minimum $\mathcal{L}'$:

$$\begin{aligned} \frac{\partial \mathcal{L}'}{\partial U_i} = {} & \sum_{j=1}^{n} I_{i,j} \left(R_{i,j} - U_i S_j \right) \left(-S_j \right) + \lambda_1 U_i \\ & + \alpha \sum_{j \in N(u)} \left(U_i - U_j \right) \end{aligned} \tag{16}$$

$$\frac{\partial \mathcal{L}'}{\partial S_j} = \sum_{i=1}^{m} I_{i,j} \left(R_{i,j} - U_i S_j \right) \left(-U_i \right) + \lambda_2 S_j \tag{17}$$

Given (9), (16) and (17), we fuse the objective function of neural user-service feature interaction and the collaborative neighborhood selection, and then employ the stochastic gradient descent technique [16] to find the sub-optimal solution.

Due to the non-convexity of the loss function in NDMF, gradient-based optimization methods can only find sub-optimal solutions. In the procedure of model optimization, we first train the neural user-service feature interaction model from scratch with random initializations until convergence, and then use this model parameters as the initialization of NDMF's parameters with the optimizer RMSProp [18].

TABLE I
STATISTICS OF SERVICE DATASET

Item Name	Value
Users	339
Services	5825
Service Invocations	1,974,675
Users' Regions	31
Users' AS	137
Services' Regions	74
Services' AS	992
Services' Providers	2699

2) QoS Prediction: Once completing the model training procedure, we generate the fine-trained NDMF model. When performing the task of service QoS prediction, it has to specify a target user and service as inputs, respectively. By applying the NDMF model to predict the missing QoS for those entries where users did not invoke the corresponding services, we can recommend the same or similar Web services.

V. EXPERIMENTS

A. Service Dataset and Experimental Setup

We conduct experiments on widely-used the WS-DREAM dataset,[1] which is public and has large number of real-world Web services collected and maintained by Zheng *et al.* [2]. It contains 1,974,675 historical QoS records of service invocations (both response time and throughput) historical service invocation records originating from 339 users on 5,825 services, which provides location information about users and services. The statistics of dataset is illustrated in Table I. The QoS dataset is represented in the form of a user-service matrix, where a row represents the QoS of a user who invokes all of the services, and a column represents the QoS of a service that is invoked by all of the users.

To validate the performance of our proposed approach, we conducted various experiments under different matrix densities with 5%, 10%, 15%, 20% (both response time and throughput), which include randomly selected number of 98,734, 197,468, 296,201, 394,935 observations as training set and 200,000 observations in each specified density as testing set.

All the experiments are carried out on a workstation with a NVIDIA Geforce 1080Ti GPU, Intel Xeon Gold 6132 CPU@ 2.60 GHz, and the components of the NDMF approach are implemented by Python 3.7.4 with Pytorch 1.1.0.

B. Evaluation Metrics

Mean absolute error (MAE) and root mean square error (RMSE) used as the two evaluation metrics to measure the accuracy of service QoS prediction among the competing approaches in the experiments.

MAE is defined as:

$$MAE = \frac{\sum_{u,i} \left| r_{u,i} - \hat{r}_{u,i} \right|}{N} \tag{18}$$

where $\hat{r}_{u,i}$ and $r_{u,i}$ are the predicted and ground truth value of the target user invoking a service, respectively; *N* is the number of the predicted QoS values. Since MAE is linear to the

[1] https://wsdream.github.io/

TABLE II
PARAMETER SETTINGS

Parameter	Value	Description
k	5	Number of neighborhoods
λ_1	0.0001	Parameter of the norm of U
λ_1	0.0001	Parameter of the norm of S
α	0.00001	Parameter of the norm of neighborhood-based regulation
d	8	Dimensionality value
dep	<16,256,128,64,32,16,8>	Depth of DNNs

deviation of QoS prediction, all the individual differences are weighted equally in the average. It is obvious that the smaller MAE is, the better QoS prediction accuracy it indicates.

RMSE measures the deviations between those predicted QoS and their corresponding observed QoS, which is then squared and averaged for calculating the square root. It is defined as:

$$RMSE = \sqrt{\frac{\sum_{u,i}\left(r_{u,i} - \hat{r}_{u,i}\right)^2}{N}} \qquad (19)$$

RMSE represents a relatively high weighting to large errors because they are squared before they are averaged by the number of samples.

In our experiments, MAE reflects the overall accuracy of QoS prediction, which averages absolute deviations to the ground truth QoS values. Compared with MAE, RMSE is very sensitive to individual outliers by representing a relatively high weighting to large errors on predicted QoS values. They are the most widely-used evaluation metrics in service QoS prediction [1]–[3], [5], [7], [19] and other quantitative prediction tasks [20]–[22], since they can intuitively show the error between the predicted value and the ground truth one.

C. Competing Approaches

In the experiments, we compared with twelve competing approaches, including UMEAN, IMEAN, UPCC [1], IPCC [23], UIPCC [24], NMF [25], PMF [26], LACF [4], NIMF [8], NAMF [7], NeuMF [10], LRMF [19].

- *UMEAN (User Mean):* It is a user-based QoS prediction approach. It averages all the QoS values of the services that the target user invokes as the predicted result. It is the baseline approach.

- *IMEAN (Item Mean):* It is a service-based QoS prediction approach. It averages all the QoS values of the target service that is invoked by users as the prediction result. It is the baseline approach.

- *UPCC:* It is a user-based CF approach using PCC for service QoS prediction. It is required to find a set of similar users as the neighborhood of the target user with PCC. The prediction result combines the average QoS by UMEAN and the deviation migration based on the found similar users.

- *IPCC:* It is a service-based CF approach using PCC for service QoS prediction. It selects the most similar services as the neighborhood of the target service with PCC. The prediction result combines the average QoS by IMEAN and the deviation migration based on the found similar services.

- *UIPCC:* It is a comprehensive CF approach for service QoS prediction by the combination of UPCC and IPCC, which utilizes a parameter to adjust the weighting of UPCC and IPCC. It provides a fundamental approach by simultaneously considering similar users and services for predicting the missing QoS.

- *NMF:* It is a non-negative matrix factorization approach that makes service QoS prediction by non-negative factorized factors without considering neighborhood information. It is the basic traditional model-based approach for service QoS prediction that is used as the foundation of follow-up ones by incorporating more heuristic information.

- *PMF:* It is a probabilistic matrix factorization approach which introduces probability model to optimize matrix factorization model for service QoS prediction. It is a model-based representative approach by updating the traditional matrix factorization.

- *LACF:* It is a location-aware collaborative filtering approach for service QoS prediction that uses both the locations of users and services. It is a typically memory-based approach that takes location context into account when calculating the similar users/services.

- *NIMF:* It is a representative neighborhood-integrated matrix factorization approach that was the first one merging similar users into matrix factorization for service QoS prediction. It applies PCC to identify user neighborhood based on historical service invocation QoS. It is similar with our proposed approach, where neighborhood information is incorporated into the model training. However, the significant differences in our research not only include neighborhood selection by combining both historical service invocation QoS and location context, but also perform model learning by deep neural network, rather than traditional matrix factorization.

- *NAMF:* It is a network-aware matrix factorization approach which also integrates users' neighborhood information into matrix factorization for service QoS prediction. It is also a representative approach by incorporating user neighbors into model training by traditional matrix factorization. Unlike the NIMF, it selects user neighborhood based on their geographical locations, while both historical service invocation QoS and location context are considered in our approach NDMF.

- *NeuMF:* It is an advanced neural collaborative filtering approach that combines multi-layer perceptron and matrix factorization for recommender systems. It is a very influential deep learning approach that is used to rank the items for a target user and solve regression problems such as service QoS prediction.

- *LRMF:* It is a location and reputation aware matrix factorization approach that integrates users' location information and reputation into matrix factorization for service QoS prediction. It calculates user reputation to minimize the influence of those dishonest users who distribute excellent QoS for their own services and bad QoS for other competing services. It is a state-of-the-art approach based on traditional matrix factorization.

D. Experimental Results and Analysis

To validate the effectiveness of our proposed NDMF approach for service QoS prediction, we compare it with

TABLE III
EXPERIMENTAL RESULTS OF SERVICE QOS PREDICTION AMONG COMPETING APPROACHES IN RESPONSE TIME

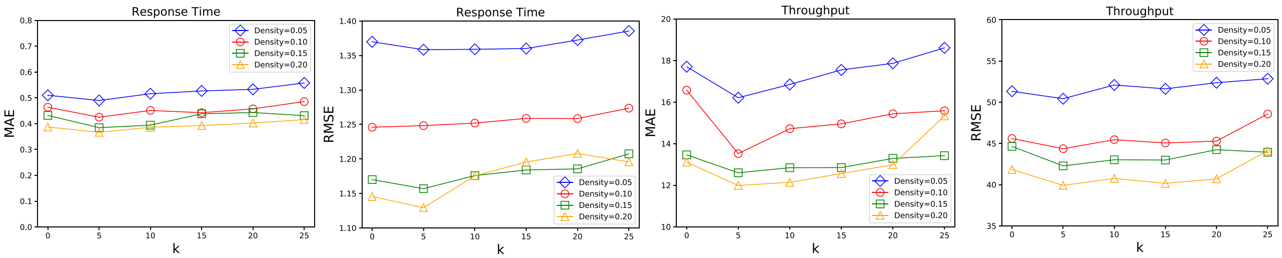
Approaches	Matrix Density=5%		Matrix Density=10%		Matrix Density=15%		Matrix Density=20%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UMEAN	0.8763	1.8529	0.8750	1.8555	0.8751	1.8552	0.8747	1.8554
IMEAN	0.7017	1.5673	0.6879	1.5425	0.6833	1.5327	0.6809	1.5280
UPCC	0.6520	1.6724	0.5478	1.5087	0.5091	1.4577	0.4750	1.4042
IPCC	0.6924	1.3988	0.6660	1.3742	0.5418	1.2567	0.4864	1.2113
UIPCC	0.6253	1.3879	0.5815	1.3302	0.5012	1.2498	0.4498	1.1968
NMF	0.5455	1.4727	**0.4776**	1.2830	0.4467	**1.2022**	0.4274	1.1604
PMF	0.5690	1.5371	0.4866	1.3162	0.4522	1.2205	0.4307	1.1687
LACF	0.6299	1.4393	0.5601	1.3377	0.5102	1.2694	0.4774	1.2220
NIMF	0.5542	1.4791	0.4800	1.2950	**0.4428**	1.2082	**0.4205**	1.1594
NAMF	**0.5384**	**1.3853**	0.4850	**1.2592**	0.4529	1.2071	0.4350	**1.1443**
NeuMF	0.5560	1.4211	0.4998	1.3445	0.4883	1.2450	0.4644	1.2118
LRMF	0.5552	1.4954	0.4848	1.2959	0.4537	1.2091	0.4351	1.1633
NDMF	**0.4880**	**1.3495**	**0.4304**	**1.2349**	**0.3845**	**1.1569**	**0.3665**	**1.1294**
Gains	9.4%	2.6%	9.9%	2.0%	13.2%	3.8%	12.8%	1.3%

TABLE IV
EXPERIMENTAL RESULTS OF SERVICE QOS PREDICTION AMONG COMPETING APPROACHES IN THROUGHPUT

Approaches	Matrix Density=5%		Matrix Density=10%		Matrix Density=15%		Matrix Density=20%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UMEAN	53.8757	110.3693	53.8347	110.3802	53.8155	110.3817	53.8008	110.3822
IMEAN	27.2885	66.1012	26.8596	64.8083	26.7156	64.3864	26.6410	64.1772
UPCC	29.0248	72.6523	22.7842	63.2618	20.0242	57.9214	18.5031	55.1757
IPCC	28.3420	62.6599	28.8850	62.1873	28.0732	58.0547	27.5154	56.6022
UIPCC	26.7568	60.7985	22.3700	54.4563	20.2190	50.7043	18.9276	48.2950
NMF	18.8847	57.5301	15.5785	47.8238	14.3463	44.0323	13.5870	41.7636
PMF	19.0788	57.8884	15.9950	48.0795	14.6821	44.0503	13.9223	41.7244
LACF	22.9737	58.7857	19.4489	52.9270	17.5887	49.5650	16.4580	47.4108
NIMF	18.8752	56.1485	**15.1433**	46.9688	**13.8055**	43.2441	**13.2219**	41.2491
NAMF	**18.0836**	**52.8658**	15.9808	**44.0788**	14.6661	**43.0206**	13.9386	**40.7481**
NeuMF	26.6856	65.7189	19.1881	51.5266	17.6678	51.8070	16.0529	47.3841
LRMF	19.1090	58.0719	15.9494	48.2718	14.5974	44.0682	13.9206	41.7880
NDMF	**16.3818**	**50.9612**	**13.9317**	**43.9095**	**12.5043**	**42.5319**	**11.7204**	**39.9431**
Gains	9.4%	3.6%	8.0%	0.3%	9.4%	1.1%	11.4%	2.0%

state-of-the-art memory-based and model-based approaches under the parameter setting in Table II. In the experiments, we run all these competing approaches with the same training and testing dataset, where the model parameters with the best performance are directly used as they are suggested in the experiments of the references. To avoid the deviations, the experiments are run for several times to guarantee the fairness of the performance comparison between our proposed approach and the baselines.

Table III and Table IV illustrate the experimental results on response time (RT) and throughput (TP) compared with state-of-the-art approaches. Here, lower MAE and RMSE indicate better performance on service QoS prediction. As can be seen from the tables, the proposed NDMF remarkably outperforms the traditional approaches on both RT and TP datasets. In the tables, we make the best results of competing bold and calculate the performance gains on them. The experimental results demonstrate that our approach has a remarkably stepwise improvement on MAE up to 13.2% on RT and 11.4% on TP, respectively. UMEAN and IMEAN as the baseline perform poorly in QoS prediction because they average the historical records directly without mining any latent patterns; UPCC, IPCC and UIPCC improve a lot in QoS prediction performance through CF approach incorporating the user or service neighborhood, and LACF use extra geographical context in neighborhood selection that performs better than previous approaches; NMF and PMF as basic MF approaches



(a)

(b)

(c)

(d)

Fig. 5. The performance impact of service QoS prediction with the changes of user neighbors number in NDMF.

show large improvement compared with CF approach in QoS prediction because CF is vulnerable to the data sparsity, whereas MF is much better in dealing with it; NIMF and NAMF as the traditional neighborhood-incorporated MF approaches have a slight improvement compared with NMF and PMF because they incorporate neighborhood in the model training. Although NeuMF performs well in recommender systems, it does not work well in service QoS prediction because QoS data is more discrete and has bigger data span. Even though LRMF mines latent user reputation and incorporates geographical context in model training, it still does not gain superior performance compared with other MF based QoS prediction approaches. From the results, we can see that NAMF can perform the best among the conventional approaches in some cases with specified matrix densities. However, other approaches dominate the performance of service QoS prediction when the density is set to be 10% in different densities on RT and TP. Due to failing to extract implicit interactions in the traditional approaches, we observed that it is difficult to improve the performance even when neighborhood is taken into account and integrated into matrix factorization. Since deep neural network for the representation of latent user-service feature interaction is used in matrix factorization, our NDMF can overcome the bottleneck of the existing ones, which can achieve better performance for service QoS prediction. More specifically, with the variance of matrix density from 0.05 to 0.20 on RT and TP, we observe that all of the competing approaches can receive better MAE and RMSE, since more QoS values can be used for precisely selecting neighborhood as well as matrix factorization. From the results, the performance of NDMF can become increasingly better and is always superior to other competing approaches along with the increase of matrix density. Consequently, we conclude that the proposed NDMF approach performs the best for service QoS prediction among all of the competing approaches with different matrix densities.

E. Performance Impact of Parameters

In the experiments, three main parameters impact the performance of service QoS prediction in NDMF approach. 1) the number of similar users for collaborative neighborhood selection; 2) the depth of deep neural network for user-service implicit feature interaction; 3) the dimensionality of user and service representation.

1) Impact of the Collaborative User Neighborhood: To study the performance impact of the number of similar users, we denote it as k. It is varied from 0 to 25 with interval of 5. The performance impact of service QoS prediction along with the changes of the number of neighborhood users is shown in Fig. 5. From the results, we observe that as k becomes larger, its prediction accuracy initially increases and then reaches the best at $k = 5$. After that, the performance decreases along the with increase of similar users. The reason is that at first more similar users included in the model training can provide more useful information that leads to better performance. However, as k goes larger and larger, some selected users that are not highly similar with the target user affect the service QoS prediction accuracy.

Due to the constraints of WS-DREAM service QoS dataset, we currently do not take the time factor into account when the service is invoked by users. However, two users connected to the same AP and AS could receive different QoS invocation experiences, when they invoke the same Web service in different periods of a day. Thus, time factor would positively influence collaborative neighborhood selection. If we take the time parameter into account, what would happen on the performance of NDMF? It would help NDMF in making more accurate service QoS prediction.

2) Impact of Depth of Deep Neural Network: Neuron is the basic unit of arithmetic that constitutes a deep neural network(DNN) in NDMF framework. In this work, DNN is implemented as tower structure to better learn more non-linear implicit feature interactions between user and service. The depth of DNN layer influence the accuracy of service QoS prediction. The performance impact of the depth of DNN layer is shown in Fig. 6.

It can be seen from the experimental result, the prediction error is very high both in MAE and RMSE when the depth of DNN layer is set to be 1. As the DNN layer goes deeper, the QoS prediction accuracy significantly improves at first, and then gradually improve to a less extent on MAE and RMSE. DNNs can learn complex and non-linear interaction function of latent vectors, making up the limitations of traditional MF. However, when the depth of DNN layers is too shallow, it cannot fully reach the ability of fitting interaction function well.

3) Impact of Dimensionality of User and Service Representation: Dimensionality of user or service vector determines how many latent factors are utilized to characterize the features of users or services. To test the performance impact of dimensionality on service QoS prediction, we vary the dimensionality d by 2, 4, 8, 16, 32, 64 and set the matrix density as 0.05, 0.10, 0.15 and

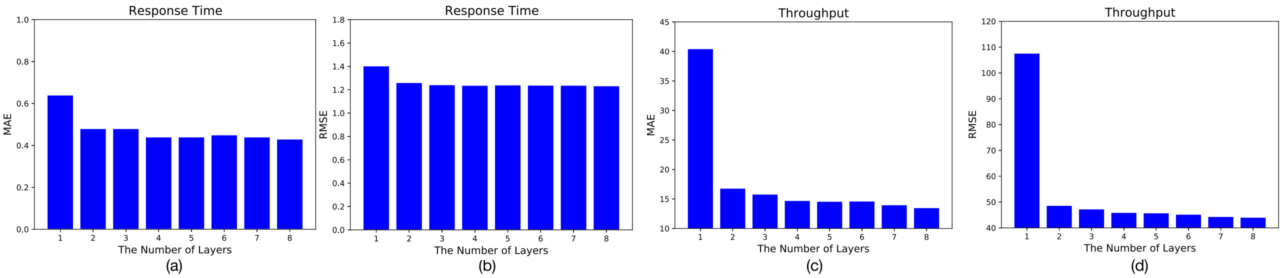


Fig. 6. The performance impact of service QoS prediction with the changes of DNN layers number in NDMF.

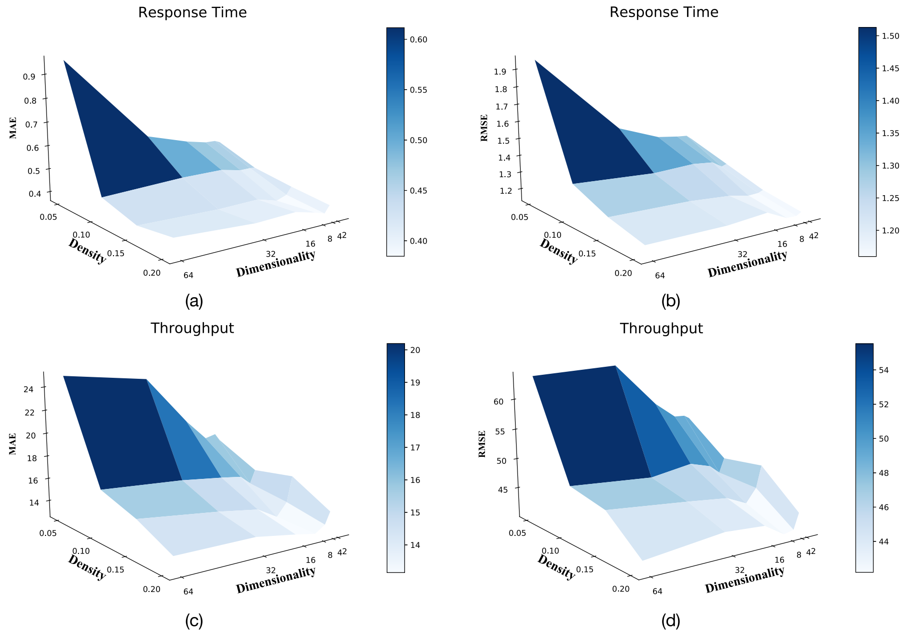


Fig. 7. The performance impact of service QoS prediction with the changes of dimensionality and data density in NDMF.

0.20, respectively. The experimental results of service QoS prediction along with the changes of dimensionality and matrix density on MAE and RMSE are shown in Fig. 7.

It is observed that the acuracy of service QoS prediction improves along with the decrease of user and service vector dimensionality on both the service dataset of RT and TP. The prediction accuracy is dramatically improved as matrix density increases from 0.05 to 0.15 and dimensionality drops from 64 to 8. Nevertheless, it changes slowly when it achieves $density > 0.15$ and $d < 8$. Moreover, the best dimensionality d for service QoS prediction changes along with different matrix densities. The reason of these phenomena is that when service QoS matrix is too sparse, the parameters cannot be fully learned and optimized by model training, so that the implicit features of users and services are poorly represented via deep matrix factorization model; on the contrary, as the service QoS dataset becomes denser, more implicit features can be mined by the NDMF model. In such case, high dimensionality should be used to represent the implicit features for differentiating the users or services, which receives better QoS prediction accuracy.

F. Analysis of Time Consumption

To analyze the time consumption of NDMF, we carry out the experiments under different parameter settings on matrix density, the number of user neighbors and DNN layers. The experiments of competing approaches are run for 20 rounds that are averaged as the final value to guarantee the fairness of the comparison on time consumption. The results are illustrated in Fig. 8.

Along with the increase of matrix density as shown in Fig. 8(a), time consumption of all approaches grows from more data for model training and finding similar users or services. Specifically, memory-based approaches like UPCC, IPCC, UIPCC and some model-based approaches like NMF, NeuMF perform well in time consumption. Since NDMF and NIMF are neighborhood-integrated approaches, it is observed that time consumption of them is much higher than other approaches, where NIMF is even about two times higher than NDMF. More deeply, the experiment results illustrated in Fig. 8(a) consist of the time consumption for model training and all vacant QoS values prediction. Taking matrix density with 0.05 as an example, model training process is performed on approximately 100,000 training samples, which consumes more time than QoS prediction process according to Fig. 8(c). However, model training can always be performed offline in realistic practical scenarios. As for QoS prediction process, it calculates all the vacant service QoS values by using the learned model, which consumes less than 1 minute for almost 200,000 test samples as shown in Fig. 8(c). That is, it takes

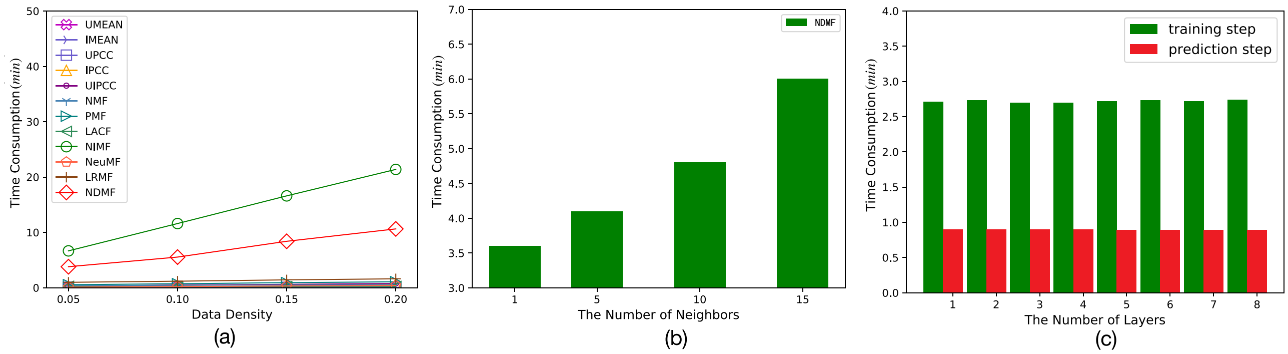


Fig. 8. The time consumption of NDMF against competing approaches with different parameter settings.

around 0.00024s to predict a vacant QoS value, which indicates an example of a user invoking a Web service. Moreover, the computing capability is limited in the experiments and it can be transplanted to a more powerful platform to further shorten the time consumption of model training and QoS prediction in real-world applications. Therefore, NDMF is still efficient and could be potentially deployed and adopted to achieve much better service QoS prediction accuracy in real-world service-oriented applications.

As the increasing number of user neighbors shown in Fig. 8(b), time consumption gradually arises in NDMF. The reason is that more user neighbors are selected in NDMF, it consumes much more time for model training and predicting vacant service QoS values. However, it does not show the trend of explosion on time consumption as the number of user neighbors arises. In many cases, to achieve the best performance on QoS prediction accuracy, the number of user neighbors can not be too large but an appropriate value, whereas the realistic number of user neighbors varies with different datasets.

To analyze the impact on time consumption of network structure, we run NDMF with different setting of DNN layers as shown in Fig. 8(c). Here, training step experiences the whole span of model training by using training set until it converges to the optimal or suboptimal point, which contains several epochs. Based on the learned model, prediction step experiences the whole process of predicting all the vacant QoS values by using test set. The results illustrate that training step and prediction step cost almost cost the same time in different number of DNN layers, respectively. The underlying reason is that although the number of neurons varies in NDMF, the propagation process of DNN is very quick, leading to a stable time consumption in the same data size.

VI. Related Work

A. Memory-Based Approaches for Service QoS Prediction

The memory-based approach for service QoS prediction has been widely studied since the use of CF by Shao *et al.* [1]. It includes user-based [27], item-based [23], and comprehensive-based [2]. By the use of historical QoS invocation records, similar users or services can be selected by similarity calculation approach such as PCC, Cosine *et al.* The crucial step of memory-based approach is to perform similarity calculation on users or services for service QoS prediction. To more accurately calculate the similarity of users or services, enhanced memory-based CF approaches have been proposed. Jiang *et al.* [28] mined personalized features of users and services by analyzing historical QoS records and combined them into the calculation procedure of user and service similarity. Sun *et al.* [3] investigated the QoS distribution and proposed a novel collaborative filtering approach for QoS prediction. It normalizes the QoS values to the same range and then unifies the similarity in multi-dimensional vector spaces. Wu *et al.* [12] proposed a novel ratio-based similarity approach to select neighborhood of users and services. Compared with PCC similarity [29] and cosine similarity [23], it is more precise for predicting the unknown service QoS. Wu *et al.* [30] employed the data smoothing technique on the user-service QoS matrix, and proposed an enhanced CF approach to alleviate the issue of matrix sparsity of user-service QoS invocations.

Moreover, some researchers try to take advantage of extra heuristic information of user and service to assist in boosting the accuracy of service QoS prediction. Xi *et al.* [31] applied the users' IP addresses to find similar users. They assumed that users with similar IP addresses are located in the same region and have similar QoS experiences. Wei *et al.* [32] employed users' longitude and latitude to calculate distance and identify similar users. Tang *et al.* [4] combined users' and services' locations before the similarity calculation, which reduces the scale of finding similar users and services.

Although memory-based collaborative filtering approaches have been widely investigated for service QoS prediction, it is sensitive and vulnerable to the sparsity of QoS records matrix, due to few service invocations by users.

B. Model-Based Approaches for Service QoS Prediction

In recent years, model training by matrix factorization has been widely studied for service QoS prediction. Zheng *et al.* [8] proposed a MF-based prediction approach that integrates users' neighborhood information into the factorization model, where PCC is applied to calculate the similarity and select user neighborhood.

Matrix factorization also employed user and service context in the model training for improving the accuracy of service QoS prediction, such as geographical information.

Li *et al.* [19] proposed an approach of service QoS prediction which incorporates the reputation and location as the heuristic information in model training. By doing so, those dishonest users, who distribute their own services with excellent QoS but bad QoS to services with the same or similar functionality on purpose to attract more users to use their published services, can be effectively identified to improve the QoS prediction accuracy.

Xu *et al.* [33] proposed a location-integrated QoS prediction approach, which finds the similar users within a distance threshold based on their longitudes and latitudes. However, users with close geographical position do not mean they are close in the network path, which indicates they might not have the same or similar network environment. On the contrary, when users are close in network distance, they might share the same network devices or similar network environment that promotes the service QoS prediction accuracy. Upon the assumption, Tang *et al.* [7] proposed a network-aware QoS prediction approach NAMF that integrates neighborhood into matrix factorization. NAMF measures the network distances between users with network map between users and then calculates the neighborhood as heuristics for matrix factorization. However, NAMF calculates user similarity without considering their historical QoS records, which may discard similar users with latent similar service invocations.

In recent years, deep learning techniques have been widely applied to improve the recommendation quality and optimize the QoS management. He *et al.* [10] proposed an effective approach called NeuMF that incorporates traditional matrix factorization with a deep neural network to solve the poor representation of MF in low dimensions. By using deep reinforcement learning, Guo *et al.* [34] proposed a Deep-Q-Network based multi-service QoS optimization approach that helps make decisions to optimally and dynamically schedule the limited radio resource for QoS flows in mobile edge computing (MEC). To provide enhanced QoS with improved data rates in mobile IoT, Zafar *et al.* [35] proposed an approach of threshold percentage dependent interference graph (TPDIG) by deep learning based resource allocation algorithm for city buses mounted with moving small cells (mSCs), where Long–Short Term Memory (LSTM) is applied to predict the locations of city buses with high QoS for interference determination between mSCs. Additionally, Guo *et al.* [36] proposed a deep reinforcement learning based QoS-aware secure routing protocol, which extracts knowledge from history traffic demands and dynamically optimize the routing policy while guaranteeing the QoS in software defined network.

VII. Conclusion and Future Work

In this article, we proposed a novel neighborhood-integrated deep matrix factorization approach called NDMF, which aims at more accurately predicting service QoS. First, it captures the complex non-linear interaction of user and service implicit features by deep neural network, which replaces the inner product interaction function in the traditional matrix factorization. Second, user neighborhood is effectively selected in a collaborative way by both the historical QoS records and users' geographical information. Finally, it integrates user neighborhood as heuristics into the deep matrix factorization model for service QoS prediction. Extensive experiments have been conducted on real-world service invocation QoS datasets to evaluate the performance of NDMF. The results demonstrate that NDMF can significantly improve the accuracy of service QoS prediction compared with state-of-the-art approaches.

In the future, we plan to further extend our model adapting to new users or services added in an incremental learning way, and explore the integration of users' and services' geo-location embeddings into deep neural network to reduce model training time and improve the accuracy of service QoS prediction.

References

[1] L. Shao, Z. Jing, W. Yong, J. Zhao, X. Bing, and M. Hong, "Personalized QoS prediction for Web services via collaborative filtering," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, 2007, pp. 439–446.

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

[3] H. Sun, Z. Zheng, J. Chen, and M. R. Lyu, "Personalized Web service recommendation via normal recovery collaborative filtering," *IEEE Trans. Services Comput.*, vol. 6, no. 4, pp. 573–579, Oct.–Dec. 2013.

[4] M. Tang, Y. Jiang, J. Liu, and X. Liu, "Location-aware collaborative filtering for QoS-based service recommendation," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, 2012, pp. 202–209.

[5] J. Liu, M. Tang, Z. Zheng, X. F. Liu, and S. Lyu, "Location-aware and personalized collaborative filtering for Web service recommendation," *IEEE Trans. Services Comput.*, vol. 9, no. 5, pp. 686–699, Sep./Oct. 2016.

[6] L. Wu, E. Chen, Q. Liu, L. Xu, T. Bao, and L. Zhang, "Leveraging tagging for neighborhood-aware probabilistic matrix factorization," in *Proc. ACM Int. Conf. Inf. Knowl. Manag. (CIKM)*, 2012, pp. 1854–1858.

[7] M. Tang, Z. Zheng, G. Kang, J. Liu, Y. Yang, and T. Zhang, "Collaborative Web service quality prediction via exploiting matrix factorization and network map," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 1, pp. 126–137, Mar. 2016.

[8] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative Web service QoS prediction via neighborhood integrated matrix factorization," *IEEE Trans. Services Comput.*, vol. 6, no. 3, pp. 289–299, Jul.–Sep. 2013.

[9] W. Lo, J. Yin, Y. Li, and Z. Wu, "Efficient Web service QoS prediction using local neighborhood matrix factorization," *Eng. Appl. Artif. Intell.*, vol. 38, pp. 14–23, Feb. 2015.

[10] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. Int. Conf. World Wide Web (WWW)*, 2017, pp. 173–182.

[11] Z. Zheng and M. R. Lyu, "WS-DREAM: A distributed reliability assessment mechanism for Web services," in *Proc. IEEE Int. Conf. Dependable Syst. Netw. FTCS DCC (DSN)*, 2008, pp. 392–397.

[12] 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/Jun. 2017.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013. [Online]. Available: arXiv:1301.3781.

[14] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 1188–1196.

[15] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[16] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. Int. Conf. Comput. Statist. (COMPSTAT)*, 2010, pp. 177–186.

[17] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Trans. Knowl. Discovery Data*, vol. 4, no. 1, pp. 1–24, 2010.

[18] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.

[19] S. Li, J. Wen, F. Luo, T. Cheng, and Q. Xiong, "A location and reputation aware matrix factorization approach for personalized quality of service prediction," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, 2017, pp. 652–659.

[20] W. Xi, L. Huang, C. Wang, Y. Zheng, and J. Lai, "BPAM: Recommendation based on BP neural network with attention mechanism," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2019, pp. 3905–3911.

[21] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T. S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2017, pp. 3119–3125.

[22] X. He and T. S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. Int. ACM SIGIR Conf.*, 2017, pp. 355–364.

[23] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. Int. Conf. World Wide Web (WWW)*, 2001, pp. 285–295.

[24] Z. Zheng, M. Hao, M. R. Lyu, and I. King, "WSRec: A collaborative filtering based Web service recommender system," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, 2009, pp. 437–444.

[25] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2001, pp. 556–562.

[26] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2008, pp. 1257–1264.

[27] J. S. Breese, C. Kadie, and D. Heckerman, "Empirical analysis of predictive algorithms for collaborative filtering," *Uncertainty Artif. Intell.*, vol. 98, no. 7, pp. 43–52, 2013.

[28] Y. Jiang, J. Liu, M. Tang, and X. Liu, "An effective Web service recommendation method based on personalized collaborative filtering," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, 2011, pp. 211–218.

[29] U. Shardanand and P. Maes, "Social information filtering: Algorithms for automatin 'word of mouth,"' " in *Proc. Conf. Human Factors Comput. Syst. (SIGCHI)*, 1995, pp. 210–217.

[30] J. Wu, L. Chen, Y. Feng, Z. Zheng, M. C. Zhou, and Z. Wu, "Predicting quality of service for selection by neighborhood-based collaborative filtering," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 2, pp. 428–439, Mar. 2013.

[31] C. Xi, X. Liu, Z. Huang, and H. Sun, "RegionKNN: A scalable hybrid collaborative filtering algorithm for personalized Web service recommendation," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, 2010, pp. 9–16.

[32] L. Wei, J. Yin, S. Deng, L. Ying, and Z. Wu, "Collaborative Web service QoS prediction with location-based regularization," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, 2012, pp. 464–471.

[33] Y. Xu, J. Yin, W. Lo, and Z. Wu, "Personalized location-aware QoS prediction for Web services using probabilistic matrix factorization," in *Proc. Int. Conf. Web Inf. Syst. Eng. (WISE)*, 2013, pp. 229–242.

[34] B. Guo, X. Zhang, Y. Wang, and H. Yang, "Deep-Q-network-based multimedia multi-service QoS optimization for mobile edge computing systems," *IEEE Access*, vol. 7, pp. 160961–160972, 2019.

[35] S. Zafar, S. Jangsher, O. Bouachir, M. Aloqaily, and J. B. Othman, "QoS enhancement with deep learning-based interference prediction in mobile IoT," *Comput. Commun.*, vol. 148, pp. 86–97, Dec. 2019.

[36] X. Guo, H. Lin, Z. Li, and M. Peng, "Deep-reinforcement-learning-based QoS-aware secure routing for SDN-IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6242–6251, Jul. 2020, doi: 10.1109/JIOT.2019.2960033.

Guobing Zou received the Ph.D. degree in computer science from Tongji University, Shanghai, China, in 2012. He is an Associate Professor and the Dean of the Department of Computer Science and Technology, Shanghai University, China. He worked as a Visiting Scholar with the Department of Computer Science and Engineering, Washington University in St. Louis, USA, from 2009 to 2011. He has published around 80 papers on premier international journals and conferences, including IEEE TRANSACTIONS ON SERVICES COMPUTING, IEEE International Conference on Web Services, International Conference on Service Oriented Computing, and IEEE International Conference on Services Computing.

Jin Chen received the bachelor's degree in computer science and technology from Fujian Normal University in 2017. He is currently pursuing the master's degree with the School of Computer Engineering and Science, Shanghai University, China. He has published a paper on the 18th International Conference on Service Oriented Computing in 2020. His research interests include service QoS prediction, QoS management, and deep learning.

Qiang He received the first Ph.D. degree from the Swinburne University of Technology (SUT), Australia, in 2009, and the second Ph.D. degree in computer science and engineering from the Huazhong University of Science and Technology, China, in 2010. He is currently working as a Senior Lecturer with the Department of Computer Science and Software Engineering, SUT. He has published around 110 papers on premier international journals and conferences, including IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, IEEE TRANSACTIONS ON SERVICES COMPUTING, and IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON CLOUD COMPUTING.

Kuan-Ching Li (Senior Member, IEEE) is currently a Distinguished Professor with the Department of Computer Science and Information Engineering, Providence University, Taiwan. Besides publishing numerous research papers and articles, he is a coauthor/co-editor of several technical professional books published by CRC Press/Taylor & Francis, Springer, McGraw-Hill, and IGI Global. His research interests include parallel and distributed processing, GPU/many-core computing, and big data and cloud. He has been actively involved in several major conferences and workshops in program/general/steering conference chairman positions. He is a member of AAAS, a Life Member of TACC, and a Fellow of IET.

Bofeng Zhang (Member, IEEE) received the Ph.D. degree from Northwestern Polytechnic University, China, in 1997. He is a Full Professor with the School of Computer Engineering and Science, Shanghai University, China. He was a Postdoctoral Researcher with Zhejiang University, China, from 1997 to 1999. He worked as a Visiting Professor with the University of Aizu, Japan, from 2006 to 2007 and Purdue University, USA, from 2014 to 2015. He has published more than 150 papers on international journals and conferences. His research interests include personalized service recommendation, data mining, and human–computer interaction.

Yanglan Gan received the Ph.D. degree in computer science from Tongji University, Shanghai, China, in 2012. She is an Associate Professor with the School of Computer Science and Technology, Donghua University, Shanghai. Her research interests include bioinformatics, Web services, and data mining. She has published more than 30 papers on premier international journals and conferences, including *Bioinformatics*, *BMC Bioinformatics*, IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, IEEE International Conference on Web Services, IEEE International Conference on Service-Oriented Computing, *Neurocomputing*, and *Knowledge-Based Systems*.