TRCF: Temporal Reinforced Collaborative Filtering for Time-aware QoS Prediction

Guobing Zou, Yutao Huang, Shengxiang Hu, Yanglan Gan, Bofeng Zhang, and Yixin Chen, Fellow, IEEE

Abstract—The proliferation of homogeneous web services has necessitated the task of predicting vacant Quality of Service (QoS) for service-oriented downstream tasks. Existing approaches primarily focus on user-service invocations without considering temporal factors, limiting their applicability in QoS fluctuations over time. Moreover, some investigations are conducted to predict temporally missing QoS, which still suffers from two limitations. First, time-aware collaborative filtering (CF) approaches fail to well capture continuous temporal changes, which lowers the performance of time-aware QoS prediction. Second, they have paid less attention to the high sparsity of user-service QoS invocations across sequentially multiple time slices, which affects the calculation of temporal average QoS, thereby further reducing the accuracy of time-aware QoS prediction. To effectively mine the continuous temporal variations and solve the high sparsity of user-service QoS invocations, we propose a novel time-aware QoS prediction approach named Temporal Reinforced Collaborative Filtering (TRCF). We design temporal reinforced RBS and PCC to improve similarity evaluation that leads to better calculation of temporal average QoS and deviation migration for predicting time-aware QoS. We evaluate TRCF on a large-scale real-world temporal dataset WS-DREAM across 64 time slices and the results demonstrate its superior performance in time-aware QoS prediction, both under relatively dense and extremely sparse QoS situations.

Index Terms—Web Service, Time-aware QoS Prediction, Temporal Factor, Collaborative Filtering, Deviation Migration

1 Introduction

With the rapid development of Internet technology, web services are deployed by service vendors in real-world application scenarios, which has greatly promoted the advancements in service selection [1], composition [2], recommendation [3] and mashup creation [4]. The growing popularity of service-oriented architecture (SOA) and the overwhelming web services registered on the Internet has resulted in a huge number of functionally similar or equivalent services. It has led to a homogenization of service functionalities, making it difficult and time-consuming for service consumers to select their desired and suitable services from a large pool of homogeneous service repositories.

Quality of Service (QoS) as the representation of non-functional criterion has been widely used to differentiate those homogeneous web services. However, different service requesters may receive discrepant QoS due to various external factors. Simultaneously, the QoS of the same web service can fluctuate over multiple time slices. Because of the large number of service consumers and web services, it is scarcely possible to monitor and experience all the QoS of user-service invocations, emerging high sparsity of QoS invocations across multiple time slices. Therefore, how to design an effective approach to predicting time-aware vacant QoS has become a fundamental research issue in service-oriented application contexts.

In recent years, QoS prediction has been received many research investigations. Based on whether temporal factors are taken into account or not, it can be classified into non-temporal QoS prediction and time-aware QoS prediction [5]. Non-temporal QoS prediction includes memory-based [6], [7], model-based [8], [9], and deep learning based [10], [11], [12] approaches for predicting missing QoS. With the consideration of the time-series variations of network performance, time-aware QoS prediction is dedicated to integrating the temporal factor into memory-based collaborative filtering (CF) [13], [14], [15], which calculates similar neighbors through historical user-service QoS invocations to perform the task of QoS prediction. Simultaneously, some researchers have proposed new fusion approaches for time-aware QoS prediction by leveraging sequence prediction techniques [16], [17]. Moreover, owing to the applicability of machine learning, more sophisticated investigations such as tensor decomposition of increasing time dimension [18], [19] and deep learning models [20], [21], [22] have been proposed for time-aware QoS prediction.

Despite the progress of existing approaches for partially facilitating time-aware QoS prediction, they still cannot reach satisfactory performance in service-oriented application contexts. Specifically, they mainly suffer from the two following deficiencies. First, most memory-based conventional time-aware CF approaches try to divide the entire temporal intervals into a set of time slices by corresponding two-dimensional matrices for representing user-service QoS invocations and combine the predicted QoS values from the multiple partitioned QoS matrices based on distance coefficient. Since they mechanically merge the isolated QoS predictions of each time slice, ignoring the importance of continuous temporal changes [15], it cannot well uncover

- *G. Zou, Y. Huang, S. Hu are with the School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China. E-mail: {gbzou, hyt2021, shengxianghu}@shu.edu.cn*
- *Y. Gan is with the School of Computer Science and Technology, Donghua University, Shanghai 201620, China. E-mail: ylgan@dhu.edu.cn*
- *B. Zhang is with the School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai 201209, China. E-mail: bfzhang@sspu.edu.cn.*
- *Y. Chen is with the Department of Computer Science and Engineering, Washington University in St. Louis, MO 63130, USA. E-mail:chen@cse.wustl.edu.*

the temporal relationships among users invoking web services. That significantly undermines the accuracy of time-aware QoS prediction. Second, it is observed that existing approaches have paid less attention to how to address the high sparsity of user-service QoS invocations across sequentially multiple time slices, which affects the calculation of temporal average QoS, thereby further reducing the accuracy of time-aware QoS prediction. Thus, current approaches are incapable of achieving superior accuracy of time-aware QoS prediction, due to a lack of effectively mining the continuous temporal variations and solving the high sparsity of user-service QoS invocations.

To address the above two issues, we propose a novel framework for time-aware QoS prediction called Temporal Reinforced Collaborative Filtering (TRCF), including three mutually correlative procedures. When calculating temporal average QoS under densely historical QoS records, it first directly fusions the QoS values of target user-service invocations across multiple time slices; as for the situation of high sparsity of user-service QoS invocations, similar neighbors and their corresponding historical QoS values at different time slices are taken to reinforce the reliability of calculating temporal average QoS. Then, temporal deviation migration is performed by incorporating reliable factor to improve the effectiveness of calculating the aggregated deviations from similar neighbors of a target user or service. Finally, we integrate the temporal average QoS and temporal deviation migration to predict the missing time-aware QoS.

To evaluate the effectiveness of our proposed TRCF, we conduct extensive experiments on a public and large-scale real-world dataset called WS-DREAM, which consists of 4500 real-world web services from 57 regions and 142 users from 22 regions. It involves a total number of 27,392,643 user-service QoS invocations, which are partitioned into a set of independent temporal groups of historical QoS records across 64 time slices. The experimental results demonstrate that TRCF achieves the best performance in multiple evaluation metrics for time-aware QoS prediction compared to several state-of-the-art competing baselines.

The main contributions are summarized as follows:

- We propose a novel framework TRCF for time-aware QoS prediction. It integrates temporal average QoS and temporal deviation migration by continuously temporal QoS vectors across multiple time slices, leading to better accuracy of time-aware QoS prediction.
- With respect to the high sparsity of user-service QoS invocations across multiple time slices, we propose a flexible scheme to reinforce the reliability of calculating temporal average QoS. Ratio-Based Similarity (RBS) [6] is applied to find similar neighbors as hidden heuristics of insufficient target user-service temporal interactive relationships, further boosting the accuracy of time-aware QoS prediction at extremely high QoS sparse situations.
- To validate the performance of the proposed TRCF, we conducted extensive experiments on a real-world dataset. The experimental results show that TRCF receives superior time-aware QoS prediction accuracy over competing baselines, and it comprehensively achieves the best performance under both relatively dense or extremely high sparse QoS invocations.

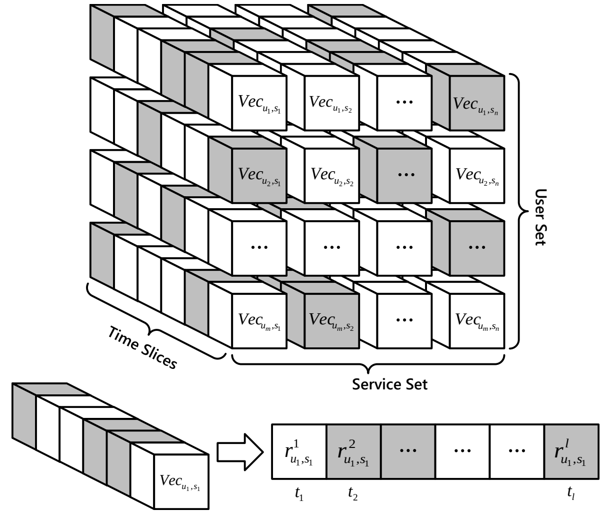


Fig. 1: Temporal QoS matrix. It consists of a set of temporal QoS vectors in terms of multiple user-service pairs at different time slices.

The remainder of this paper is organized as follows. Section 2 formulates the time-aware QoS prediction problem. Section 3 illustrates the overall framework of TRCF and elaborates the approach in detail. Section 4 shows and analyzes the experimental results. Section 5 reviews the related work. Finally, Section 6 concludes the paper and discusses the future work.

2 Problem Formulation

We first focus on the understanding of temporal service ecosystem, and then detailedly define time-aware QoS prediction problem. Table 1 presents all the notations.

Definition 1 (Temporal Service Ecosystem). A temporal service ecosystem is defined as a four-tuple $M =< U, S, T, R >$, where $U = \{u_1, u_2, \ldots\}$ is a set of users, $S = \{s_1, s_2, \ldots\}$ is a set of web services and $T = \{t_1, t_2, \ldots\}$ is a set of continuous time slices. $R = \{r^t_{u,s}\}$ consists of a set of QoS values correlative to different user-service pairs at multiple temporal slices.

Definition 2 (User-Service QoS Invocation). Given a temporal service ecosystem $M =< U, S, T, R >$, a user-service QoS invocation is defined as a four-tuple $r =< u, s, t, r^t_{u,s} >$, where $u \in U$ is a user, $s \in S$ is a web service, $t \in T$ is a temporal slice, and $r^t_{u,s}$ is a QoS value obtained by u invoking s at t.

By aggregating all the QoS values from user-service QoS invocations, we can obtain a three-dimensional temporal QoS matrix R as shown in Fig. 1, where it can be equipped by a set of temporal QoS vectors across multiple time slices.

Definition 3 (Temporal QoS Vector). A temporal QoS vector is defined as a set $V_{u,s} = \{r^{t_1}_{u,s}, r^{t_2}_{u,s}, ..., r^{t_{|T|}}_{u,s}\}$, where $u \in U$ is a user, $s \in S$ is a web service, $t_x \in T$ is a temporal slice of T, and $r^t_{u,s}$ is the QoS value obtained by u invoking s at $t \in T$. A temporal QoS vector represents QoS values of a user-service pair at time slices T, reflecting the temporal fluctuation of QoS sequences.

As depicted in Fig. 1, in a temporal service ecosystem M, R can be formalized by temporal QoS vectors of all the user-service pairs, denoted as $R = V_{u_1,s_1} \cup \ldots \cup V_{u_1,s_n} \cup$

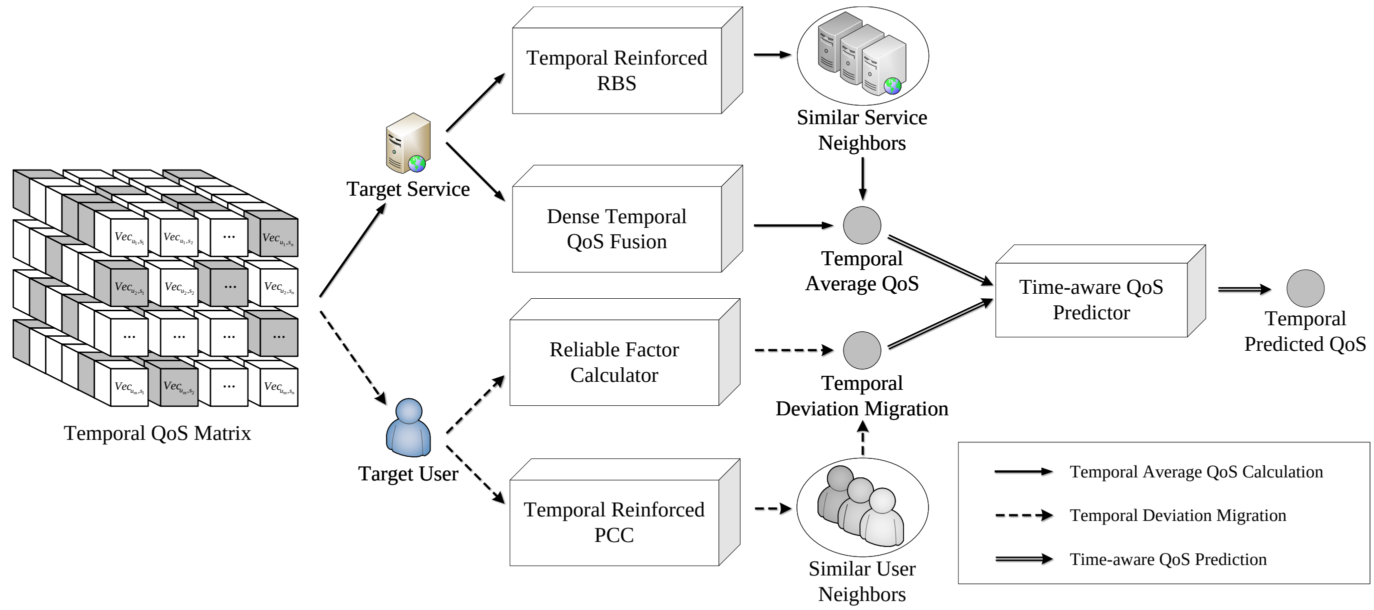


Fig. 2: Overall framework of user-based TRCF for time-aware QoS prediction.

TABLE 1: Notations.

Notation	Description
U	a set of users
S	a set of Web services
T	a set of time slices, time window
$\|T\|$	length of T, window size
$V_{u,s}$	temporal QoS vector of a user u invokes a service s
$V'_{u,s}$	aggregated temporal QoS vector of a user u invokes a service s
θ_{PCC}	threshold of temporal reinforced PCC
θ_{RBS}	threshold of temporal reinforced RBS
$\tilde{r}^t_{u,s}$	aggregated QoS of a user u invokes a service s at time slice t
$\hat{r}^u_{u,s}$	user-based temporal predicted QoS
$\hat{r}^s_{u,s}$	service-based temporal predicted QoS
$\hat{r}^t_{u,s}$	time-aware predicted QoS

$\ldots \cup V_{u_m,s_1} \cup \ldots \cup V_{u_m,s_n}$, where m is the number of U, n is the number of S, and l is the number of time slices.

In real application contexts, given a temporal QoS vector $V_{u,s} = \{r^{t_1}_{u,s}, r^{t_2}_{u,s}, ..., r^{t_{|T|}}_{u,s}\}$, if a user-service pair corresponds to a QoS value at time slice t, we have $r^t_{u,s} \in R$; otherwise, $r^t_{u,s} \notin R$ needs to be predicted for service recommendation, which is defined as below.

Definition 4 (Time-aware QoS Prediction Problem). Given a temporal service ecosystem M, time-aware QoS prediction problem is defined as $\Omega =< M, u, s, t >$, where $u \in U$ is a target user, $s \in S$ is a target service, $t \in T$ is a target time slice and $r^t_{u,s} \notin R$.

The solution to a given Ω can be represented by an element $< u, s, t, \hat{r}^t_{u,s} > \in V_{u,s}$, where $\hat{r}^t_{u,s}$ denotes the predicted missing temporal QoS for the invocation of a target service s by a target user u at a time slice t.

3 Approach

3.1 The Framework of TRCF

Fig. 2 is the overall framework of TRCF for time-aware QoS prediction. The goal of TRCF is to predict an unknown temporal QoS value when a target user aims at invoking a target web service at a specified time slice. It consists of three crucial stages, including temporal average QoS calculation, temporal deviation migration, and time-aware QoS prediction. The processes of the three stages are marked with different arrow types and described as below.

- In the stage of temporal average QoS calculation, the module of dense temporal QoS fusion directly calculates the temporal average QoS by averagely accumulating all the non-zero QoS values of target user-service at multiple time slices. We propose TRCF-TA (Temporal Average) for dense QoS distributions by directly averaging the QoS values from the original temporal QoS vector. For the context of highly sparse user-service QoS invocations, temporal reinforced RBS is designed to find similar neighbors of target service, where the QoS values of target user invoking those similar services at a certain time slice are aggregated to supplement the corresponding vacant historical QoS value of target user-service, for better calculating temporal average QoS. We propose TRCF-RTA (Ratio-based Temporal Average) for highly sparse QoS distributions by indirectly averaging the QoS values from the supplemented temporal QoS vector.
- In the stage of temporal deviation migration, temporal reinforced PCC is designed to find similar neighbors of target user, which is used to calculate the deviation migrations to temporal average QoS of each similar user invoking the target service. Moreover, reliable factor is calculated by Jaccard similarity coefficient under temporal QoS invocations of target user and similar neighbors, further improving the effectiveness of temporal reinforced PCC and enhancing the reliability of temporal deviation migration.

- In the stage of time-aware QoS prediction, we integrate the calculated temporal average QoS and temporal deviation migration to perform the final unknown temporal predicted QoS.

3.2 Temporal Average QoS Calculation

As shown in Fig. 2, dense temporal QoS fusion is provisioned to directly calculate the temporal average QoS for the case of target user-service dense QoS invocations across multiple time slices. Conversely, as for highly sparse QoS invocations, temporal reinforced RBS is taken to find similar neighbors for improving the effectiveness of temporal average QoS.

3.2.1 Density-based Temporal Average

For a target user and web service under the distribution of dense QoS invocations, temporal average QoS can be calculated by averaging the values of user-service temporal QoS vector. Specifically, given a temporal QoS vector $V_{u,s} = \{r_{u,s}^{t_1}, r_{u,s}^{t_2}, ..., r_{u,s}^{t_{|T|}}\}$, density-based temporal average QoS is expressed by:

$$Avg_{u,s}^{u} = \overline{V}_{u,s} = \frac{\sum_{t \in T} r_{u,s}^{t}}{|T_o|} \tag{1}$$

where $Avg_{u,s}^{u}$ represent the user-based temporal average QoS values. $r_{u,s}^{t}$ represents the QoS value of target service s invoked by target user u at time slice $t \in T$. $T_o \subset T$ represents a set of time slices with a non-zero QoS value. That is, for a time slice $t \in T_o$, $r_{u,s}^{t} \in R$.

As to the calculation of service-based temporal average QoS $Avg_{u,s}^{s}$, it can also be expressed by the same formula. Specifically, when calculating temporal average QoS, it is observed that a target user invokes a target service with the same temporal QoS vector for both user-based and service-based TRCF under dense QoS distributions. Consequently, they obtain the same temporal average QoS by averaging the invoked QoS values from the same temporal QoS vector across multiple time slices.

3.2.2 Sparsity-based Temporal Average

To solve the extremely high sparsity when calculating temporal average QoS, we apply RBS technique [6], [7] to temporal scenario for finding similar neighbors of a target user or service. Based on traditional RBS, we extend it to a temporal reinforced RBS for measuring the similarity between two temporal QoS vectors by matching the QoS values at their corresponding time slice. The higher similarity of temporal reinforced RBS, the more usefulness of evaluating two similar users or services in terms of their absolutely historical QoS values. It can collaboratively aggregate the historical QoS from similar neighbors to supplement the vacant QoS values of a target user-service at corresponding time slices, enhancing the reliability of sparsity-base temporal average QoS.

For the user-based temporal average QoS, we obtain a candidate service set that is more similar to the target service through temporal reinforced RBS filtering, by a target user invoking a target service and candidate services, respectively.

$$Sim_{RBS}^{u}(s,g) = \frac{\sum_{t \in T_c} \frac{min(r_{u,s}^{t}, r_{u,g}^{t})}{max(r_{u,s}^{t}, r_{u,g}^{t})}}{|T_c|} \tag{2}$$

where $Sim_{RBS}^{u}(s,g)$ represents the temporal reinforced RBS of a target user u invoking a target service s and candidate service g. T_c represents a set of time slices of u invoking s and g simultaneously, and $|T_c|$ is the number of time slices in T_c. $min(r_{u,s}^{t}, r_{u,g}^{t})$ and $max(r_{u,s}^{t}, r_{u,g}^{t})$ denote the minimum and maximum invoked QoS of $r_{u,s}^{t}$ and $r_{u,g}^{t}$ at time slice t, respectively.

$$S^*(s) = \{g \in S | Sim_{RBS}^{u}(s,g) > \theta_{RBS}\} \tag{3}$$

where $S^*(s)$ represents the selected set of similar neighbors of a target service s, and θ_{RBS} is the specified threshold of temporal reinforced RBS filtering.

We find that when the temporal reinforced RBS between s and g approaches 1, it indicates that u invoking s and g has highly similar QoS values in their temporal QoS vectors, respectively.

Based on the filtering results, we can obtain a QoS value from each similar service, which is invoked by the target user at a time slice. Specifically, given a target user u, a target service s and its similar service $g \in S^*(s)$, the estimated QoS value at time slice t is calculated as:

$$\tilde{r}_{u,s}^{t}(g) = \begin{cases} r_{u,g}^{t} \cdot Sim_{RBS}^{u}(s,g), & \overline{V}_{u,g} \geq \overline{V}_{u,s} \\ r_{u,g}^{t} / Sim_{RBS}^{u}(s,g), & \overline{V}_{u,g} < \overline{V}_{u,s} \end{cases} \tag{4}$$

where $\overline{V}_{u,s}$ and $\overline{V}_{u,g}$ represent the temporal average QoS of user-service pair u, s and u, g, respectively. By comparing $\overline{V}_{u,s}$ and $\overline{V}_{u,g}$, when the temporal average QoS reflected by user-service pair u, s is less than u, g, the QoS of u invoking g at time slice t is reduced by $Sim_{RBS}^{u}(s,g)$ to denote $\tilde{r}_{u,s}^{t}$; otherwise, it is represented by amplifying the QoS value through $Sim_{RBS}^{u}(s,g)$.

By accumulating the estimated QoS from each similar service, a group of QoS values can be obtained for estimating the QoS of the target user invoking the target service at a time slice.

$$P^u(u,s,t) = \{\tilde{r}_{u,s}^{t}(g_1), \tilde{r}_{u,s}^{t}(g_2), \ldots, \tilde{r}_{u,s}^{t}(g_{|S^*(s)|})\} \tag{5}$$

where $P^u(u,s,t)$ is the set of QoS values estimated for the target user u and the target service s on the time slice t, which is assisted by all similar services $g_x \in S^*(s)$.

By using the above $P^u(u,s,t)$, we estimate the missing QoS of u invoking s based on similar services at time slice t.

$$\tilde{r}_{u,s}^{t} = \begin{cases} \frac{\sum_{\tilde{r} \in P^u(u,s,t)} \tilde{r}}{|P_o^u(u,s,t)|}, & r_{u,s}^{t} \notin R \\ r_{u,s}^{t}, & r_{u,s}^{t} \in R \end{cases} \tag{6}$$

where $\tilde{r}_{u,s}^{t}$ is the final estimated QoS of u invoking s at t. $P_o^u(u,s,t)$ is the subset of non-zero values in $P^u(u,s,t)$, and $|P_o^u(u,s,t)|$ is the number of non-zero values.

Here, we note that not every similar service can provide a valid estimated QoS value because some similar services may not be invoked by the target user at time slice t. As a result, if a target user has not invoked a target service at a time slice, we supplement the missing value with the estimated QoS from similar services; otherwise, its original invoked QoS is used to calculate the temporal average QoS.

Consequently, the updated temporal QoS vector $V'_{u,s}$ can be represented by leveraging the estimated QoS values with temporal reinforced RBS of similar services, or its original invoked QoS at each time slice. It is expressed by:

$$V'_{u,s} = \{\tilde{r}^{t_1}_{u,s}, \tilde{r}^{t_2}_{u,s}, \ldots, \tilde{r}^{t_{|T|}}_{u,s}\}_u \tag{7}$$

where $V'_{u,s}$ is the updated temporal QoS vector for a target user u and a target service s, and T is a set of time slices.

By applying the updated temporal QoS vector $V'_{u,s}$ to enhancing the reliability of sparsity-based temporal average QoS. It is expressed by:

$$Avg^u_{u,s} = \overline{V'}_{u,s} = \frac{\sum_{t \in T} \tilde{r}^t_{u,s}}{|T^+_o|} \tag{8}$$

where $Avg^u_{u,s}$ is user-based temporal average QoS, and $\tilde{r}^t_{u,s} \in V'_{u,s}$ denotes the estimated or original QoS of u invoking s at time slice t. $T^+_o \subset T$ represents a set of time slices with non-zero QoS value and estimated QoS with similar services.

Similarly, from the perspective of service-based temporal average QoS calculation, its temporal reinforced RBS for finding similar users is expressed as:

$$Sim^s_{RBS}(u,w) = \frac{\sum_{t \in T_c} \frac{min(r^t_{u,s}, r^t_{w,s})}{max(r^t_{u,s}, r^t_{w,s})}}{|T_c|} \tag{9}$$

where $Sim^s_{RBS}(u,w)$ represents the temporal reinforced RBS of a target u and candidate user w both invoking a target service s, and T_c is a set of time slices of u and w commonly invoking s. $min(r^t_{u,s}, r^t_{w,s})$ and $max(r^t_{u,s}, r^t_{w,s})$ denote the minimum and maximum QoS of $r^t_{u,s}$ and $r^t_{w,s}$ at a time slice t, respectively.

By applying the calculated service-based temporal reinforced RBS, we can generate a set of similar users $U^*(u)$ of a target user u, which are fed to estimate approximative QoS $\tilde{r}^t_{u,s}$ of each $w \in U^*(u)$ invoking a target service s at t.

$$U^*(u) = \{w \in U | Sim^s_{RBS}(u,w) > \theta_{RBS}\} \tag{10}$$

$$\tilde{r}^t_{u,s}(w) = \begin{cases} r^t_{w,s} \cdot Sim^s_{RBS}(u,w), & \overline{V}_{w,s} \geq \overline{V}_{u,s} \\ r^t_{w,s} / Sim^s_{RBS}(u,w), & \overline{V}_{w,s} < \overline{V}_{u,s} \end{cases} \tag{11}$$

$$P^s(u,s,t) = \{\tilde{r}^t_{u,s}(w_1), \tilde{r}^t_{u,s}(w_2), \ldots, \tilde{r}^t_{u,s}(w_{|U^*(u)|})\} \tag{12}$$

where $U^*(u)$ represents the selected set of similar neighbors of a target user u. $\overline{V}_{u,s}$ and $\overline{V}_{w,s}$ denote the temporal average QoS of the user-service pair u, s and w, s, respectively. By accumulating the estimated QoS from each similar user $w_x \in U^*(u)$, $P^s(u,s,t)$ is the set of QoS obtained for estimating the QoS of the target user u invoking the target service s at a time slice t.

Finally, the missing historical QoS value of a target user u invoking a target service s at a specified time slice t is estimated by $P^s(u,s,t)$. It can be used to generated an updated the temporal QoS vector $V'_{u,s}$ for calculating the service-based temporal average QoS.

$$\tilde{r}^t_{u,s} = \begin{cases} \frac{\sum_{\tilde{r} \in P^s(u,s,t)} \tilde{r}}{|P^s_o(u,s,t)|}, & r^t_{u,s} \notin R \\ r^t_{u,s}, & r^t_{u,s} \in R \end{cases} \tag{13}$$

$$V'_{u,s} = \{\tilde{r}^{t_1}_{u,s}, \tilde{r}^{t_2}_{u,s}, \ldots, \tilde{r}^{t_{|T|}}_{u,s}\}_s \tag{14}$$

$$Avg^s_{u,s} = \overline{V'}_{u,s} = \frac{\sum_{t \in T} \tilde{r}^t_{u,s}}{|T^+_o|} \tag{15}$$

where $\tilde{r}^t_{u,s}$ represents the final estimated QoS of u invoking s at t. $P^s_o(u,s,t)$ denotes the subset of non-zero QoS values in $P^s(u,s,t)$. $V'_{u,s}$ is the updated temporal QoS vector generated by similar neighbors of a target user. $Avg^s_{u,s}$ is the service-based temporal average QoS.

3.3 Temporal Deviation Migration

We first design temporal reinforced PCC to find a set of similar neighbors for aggregating the deviations to their temporal average QoS. With the consideration of high QoS sparsity, we then take into account reliable factor to further improve the effectiveness of temporal reinforced PCC, which finally leads to better calculation of temporal deviation migration.

3.3.1 Temporal Reinforced PCC

Existing time-aware QoS prediction approaches typically incorporate temporal factors using neural networks or temporal decay factors after separately calculating QoS matrix of each time slice. In other words, they first mine the two-dimensional QoS invocation relationships among users and services, and then consider the characteristics of temporal factor, ignoring the importance of continuous temporal QoS invocation changes between users and services [15] [23]. Thus, it cannot reveal the temporal QoS relationships among users invoking web services. To solve this issue, TRCF utilizes temporal QoS vector of user-service pair across multiple time slices to perform temporal reinforced PCC that focuses on evaluating the linear relationship and calculating the similarity of two temporal QoS vectors to more accurately find similar neighbors. It enhances the expression of QoS variations along with continuous time slices, reflecting the temporal fluctuation of QoS sequences. Following the observation, TRCF can better capture temporal deviation migration by similar neighbors.

When finding user-based similar neighbors by temporal reinforced PCC, instead of traditional PCC that is used to calculate the similarity between two users jointly invoking multiple services, we incorporate temporal factors by calculating temporal QoS vectors of two users invoking the same web service at multiple time slices. That is, it enables us to calculate the similarity of QoS fluctuations for a target user u and candidate user v invoking the target service s across multiple time slices T_c. It is expressed as:

$$Sim^s_{PCC}(u,v) = \frac{\sum_{t \in T_c}(r^t_{u,s} - \overline{V}_{u,s})(r^t_{v,s} - \overline{V}_{v,s})}{\sqrt{\sum_{t \in T_c}(r^t_{u,s} - \overline{V}_{u,s})^2}\sqrt{\sum_{t \in T_c}(r^t_{v,s} - \overline{V}_{v,s})^2}} \tag{16}$$

where $T_c = T_{u,s} \cap T_{v,s}$ represents the intersection of time slices when u and v have both invoked s. $r^t_{u,s}$ and $r^t_{v,s}$ represent the QoS values of s invoked by u and v at time slice t, respectively. $\overline{V}_{u,s}$ and $\overline{V}_{v,s}$ denote the temporal average QoS of user-service pair u, s and v, s, respectively.

Based on the calculation of temporal reinforced PCC, a set of candidate users whose similarity to the target user is greater than a threshold are selected as similar users.

$$U(u) = \{v \in U | Sim^s_{PCC}(u, v) > \theta_{PCC}\} \tag{17}$$

where $U(u)$ is the selected set of similar users, and θ_{PCC} is the similarity threshold of temporal reinforced PCC.

Likewise, when finding service-based similar neighbors by temporal reinforced PCC, we introduce temporal factors to traditional PCC by calculating temporal QoS vectors of two services invoked by the same user. It can reflect the similarity of QoS fluctuations between a target user u invoking a target service s and candidate service f over a set of time slices T_c. It is expressed as:

$$Sim^u_{PCC}(s, f) = \frac{\sum_{t \in T_c}(r^t_{u,s} - \overline{V}_{u,s})(r^t_{u,f} - \overline{V}_{u,f})}{\sqrt{\sum_{t \in T_c}(r^t_{u,s} - \overline{V}_{u,s})^2}\sqrt{\sum_{t \in T_c}(r^t_{u,f} - \overline{V}_{u,f})^2}} \tag{18}$$

$$S(s) = \{f \in S | Sim^u_{PCC}(s, f) > \theta_{PCC}\} \tag{19}$$

where $T_c = T_{u,s} \cap T_{u,f}$ represents the intersection of time slices when u has jointly invoked s and f. $r^t_{u,s}$ and $r^t_{u,f}$ represent the QoS values of s and f invoked by u at time slice t, respectively. $\overline{V}_{u,s}$ and $\overline{V}_{u,f}$ denote the temporal average QoS of user-service pair u, s and u, f, respectively. $S(s)$ is the selected set of similar services of s for calculating temporal deviation migration.

3.3.2 *Reliable Factor*

When user-service QoS invocations are abundant, temporal reinforced PCC can accurately calculate the similarity between two temporal QoS vectors. However, in case of high QoS sparsity, the impact of vector intersection factors may cause PCC to either underestimate or overestimate the similarity between two QoS vectors [15]. It can result in low reliability of temporal reinforced PCC, which has a negative influence on calculating temporal deviation migration.

To address this issue, we have employed the Jaccard similarity coefficient as a reliable factor to numerically adjust the similarity of temporal reinforced PCC. It is expressed as:

$$J^s(u, v) = \frac{|V_{u,s} \cap V_{v,s}|}{|V_{u,s} \cup V_{v,s}|} \tag{20}$$

where $|V_{u,s} \cap V_{v,s}|$ is the number of time slices that both u and v have invoked s, and $|V_{u,s} \cup V_{v,s}|$ is the total number of time slices that u and/or v have invoked s.

It is observed that under the same density of user-service QoS invocations, as the number of common time slices increases for a target user and candidate user invoking a target service, it strengthens the reliability of calculating the similarity of temporal reinforced PCC, and vice versa. In such case, we integrate the temporal reinforced PCC similarity calculated by Eq. 16 and reliable factor calculated by Eq. 20 to optimize the user-based similarity between a target user u and a candidate user v.

$$Sim^s(u, v) = Sim^s_{PCC}(u, v) \cdot J^s(u, v) \tag{21}$$

where the multiplication of $J^s(u, v)$ can avoid false high similarity that may not truly reflect the similarity between two users due to the small number of intersection of time slices on co-invoked service. Furthermore, even though integrating both similarity calculations may result in relatively small $Sim^s(u, v)$ when temporal deviation migration is performed by Eq. 24, the difference among similar users can still be significantly reflected by the ratio of the similarity of a target user's neighbor to the sum of similarities of all the neighbors. Thus, it can ensure a more reliable and accurate QoS prediction result.

Similarly, the updated similarity of service-based temporal reinforced PCC with reliable factor is expressed as:

$$J^u(s, f) = \frac{|V_{u,s} \cap V_{u,f}|}{|V_{u,s} \cup V_{u,f}|} \tag{22}$$

$$Sim^u(s, f) = Sim^u_{PCC}(s, f) \cdot J^u(s, f) \tag{23}$$

where $|V_{u,s} \cap V_{u,f}|$ is the number of time slices that u has jointly invoked s and f, and $|V_{u,s} \cup V_{u,f}|$ is the total number of time slices that u has invoked s and/or f. $Sim^u_{PCC}(s, f)$ is the similarity of service-based temporal reinforced PCC calculated by Eq. 18.

When calculating the temporal deviation migration, we prioritize the filtration of selecting similar neighbors based on the specified threshold of temporal reinforced PCC, followed by the integration of reliable factor, rather than selecting similar neighbors by the updated similarity together with temporal reinforced PCC and reliable factor. It can be explained by the following example.

Example:

$\mathbf{O} = \{0.7, 0.6, \underline{0.6}, \underline{0.4}\}$

Filtration $\rightarrow$ **Integration**:

$\mathbf{A} = \{0.7, 0.6, \underline{0.6}\}$

$\mathbf{A'} = \{0.7 \times 0.06, 0.6 \times 0.06, \underline{0.6 \times 0.03}\}$

Integration $\rightarrow$ **Filtration**:

$\mathbf{B} = \{0.7 \times 0.06, 0.6 \times 0.06, \underline{0.4 \times 0.09}, \underline{0.6 \times 0.03}\}$

$\mathbf{B'} = \{0.7 \times 0.06, 0.6 \times 0.06, \underline{0.4 \times 0.09}\}$

Suppose we have calculated the similarity set O of temporal reinforced PCC, consisting of four similar neighbors with the corresponding QoS similarity. In the **Filtration** $\rightarrow$ **Integration**, assuming that we specify the threshold to 0.5, it filters out 0.4 and obtains the similar neighbors with the similarity set A. After integrating the reliable factors, we generate the final similarity set A'. Conversely, in the **Integration** $\rightarrow$ **Filtration**, we first integrate O with reliable factors and obtain the updated similarity set B. Assuming that we specify the threshold to 0.030, the updated similarity 0.6×0.03 having higher temporal reinforced PCC and lower reliable factor is filtered out from B, whereas the updated similarity 0.4×0.09 having lower temporal reinforced PCC and higher reliable factor keeps in the final similarity set B'.

From the above analysis, it is more reasonable to first find a set of similar neighbors by the similarity of temporal reinforced PCC, and then further update the similarity of each selected neighbor by integrating reliable factor, when performing the temporal deviation migration. That is, for a similar neighbor, although it has a relatively bigger reliable factor and potentially leads to higher deviation weight, it still may be classified as a dissimilar neighbor that cannot be used for calculating temporal deviation due to its lower temporal reinforced PCC.

3.3.3 *Temporal Deviation Calculation*

By combining temporal reinforced PCC, reliable factor and temporal average QoS of similar neighbors, the user-based temporal deviation migration is calculated as:

$$Dev_{u,s}^{u} = \frac{\sum_{v \in U(u)} Sim^{s}(u,v) \cdot (r_{v,s}^{t} - Avg_{v,s}^{u})}{\sum_{v \in U(u)} Sim^{s}(u,v)} \tag{24}$$

where $Avg_{v,s}^{u}$ represents the temporal average QoS of a candidate user v and a target service s. $r_{v,s}^{t}$ is the true QoS value of s invoked by v in the currently predicted time slice.

Similarly, the service-based temporal deviation migration is calculated as follows:

$$Dev_{u,s}^{s} = \frac{\sum_{f \in S(s)} Sim^{u}(s,f) \cdot (r_{u,f}^{t} - Avg_{u,f}^{s})}{\sum_{f \in S(s)} Sim^{u}(s,f)} \tag{25}$$

where $Avg_{u,f}^{s}$ represents the temporal average QoS of a target user u and a candidate service f. $r_{u,f}^{t}$ is the true QoS value of f invoked by u in the currently predicted time slice.

By applying the above temporal deviation migration, it may result in a negative non-zero temporal predicted QoS. This is primarily due to the fact that the temporal deviation migration of is negative and its absolute value is relatively large, possibly triggering an illegal prediction of missing temporal QoS by adding temporal average QoS. To guarantee the non-negative predicted QoS in real application contexts, we further modify the temporal deviation migration for those cases with negative predicted QoS. They are reconstructed by multiplication scaling for user-based and service-based temporal deviation migration, respectively.

$$Dev_{u,s}^{\prime u} = \frac{\sum_{v \in U(u)} Sim^{s}(u,v) \cdot (r_{v,s}^{t} / Avg_{v,s}^{u})}{\sum_{v \in U(u)} Sim^{s}(u,v)} \tag{26}$$

$$Dev_{u,s}^{\prime s} = \frac{\sum_{f \in S(s)} Sim^{u}(s,f) \cdot (r_{u,f}^{t} / Avg_{u,f}^{s})}{\sum_{f \in S(s)} Sim^{u}(s,f)} \tag{27}$$

3.4 Time-aware QoS Prediction

Based on temporal average QoS and temporal deviation migration, the user-based missing temporal QoS is predicted:

$$\hat{r}_{u,s}^{u} = Avg_{u,s}^{u} + Dev_{u,s}^{u} \tag{28}$$

where $\hat{r}_{u,s}^{u}$ denotes the predicted temporal QoS at a time slice t. $Avg_{u,s}^{u}$ and $Dev_{u,s}^{u}$ represent the user-based temporal average QoS and temporal deviation migration of a target user u invoking a target service s, respectively.

Similarly, the service-based missing temporal QoS is predicted by:

$$\hat{r}_{u,s}^{s} = Avg_{u,s}^{s} + Dev_{u,s}^{s} \tag{29}$$

where $\hat{r}_{u,s}^{s}$ denotes the predicted temporal QoS at a time slice t. $Avg_{u,s}^{s}$ and $Dev_{u,s}^{s}$ represent the service-based temporal average QoS and temporal deviation migration of a target user u invoking a target service s, respectively.

Based on the results of above user-based and service-based predicted temporal QoS values, the finally time-aware predicted QoS is calculated as follows:

$$\hat{r}_{u,s}^{t} = \alpha \cdot \hat{r}_{u,s}^{u} + (1-\alpha) \cdot \hat{r}_{u,s}^{s} \tag{30}$$

where α denotes the adjusting coefficient of user-based and service-based temporal QoS prediction.

TABLE 2: Statistics of Temporal RT in WS-DREAM.

Item	Value
Users	142
Services	4500
Time Slices	64
Range of RT	0-20
Invocation Records	27,392,643
Maximum QoS Density	66.98%

TABLE 3: Parameter Settings of TRCF.

Parameter	Range	Interval Step
Density	5%-20%	5%
Window Size	8-64	8
θ_{PCC}	0.3-0.8	0.1
θ_{RBS}	0.66-0.76	0.02
α	0.0-1.0	0.1

When the predicted temporal QoS value $\hat{r}_{u,s}^{t}$ is less than 0, i.e., $\hat{r}_{u,s}^{t} < 0$, we perform the user-based and service-based missing temporal QoS prediction by the reconstructed temporal deviation migration. They are further combined by adjusting coefficient to predict the finally time-aware missing QoS.

$$\hat{r'}_{u,s}^{u} = Avg_{u,s}^{u} \cdot Dev_{u,s}^{\prime u} \tag{31}$$

$$\hat{r'}_{u,s}^{s} = Avg_{u,s}^{s} \cdot Dev_{u,s}^{\prime s} \tag{32}$$

$$\hat{r'}_{u,s}^{t} = \alpha \cdot \hat{r'}_{u,s}^{u} + (1-\alpha) \cdot \hat{r'}_{u,s}^{s} \tag{33}$$

where $\hat{r'}_{u,s}^{t}$ represents the finally missing temporal QoS for the phenomenon of negative prediction.

Despite the enhancement of information expression through similar user and service neighbors for the reinforcement of linearly temporal user-service invocations across multiple continuous time slices, TRCF still struggles to learn complex nonlinear invocation relationships under time-aware situations. It may partially weaken temporal QoS prediction performance.

4 Experiments

4.1 Experimental Setup and Dataset

All the experiments are carried out on our workstation equipped with two NVIDIA GTX 1080Ti GPUs, an Intel(R) Xeon(R) Gold 6130 @2.60 GHz CPU and 192GB RAM. All the components of TRCF are implemented by python 3.7.1.

To evaluate the effectiveness of TRCF, we have conducted extensive experiments on a publicly available large-scale real-world temporal QoS dataset called WS-DREAM [24]. It has been widely used for time-aware QoS prediction that contains two kinds of QoS criteria, namely response time (RT) and throughput (TP). Since RT can intuitively reflect the networking status of users and services across different time slices, we have chosen RT as the primary experimental temporal QoS dataset. It comprises 142 independent users, 4500 web services, and a total number of 27,392,643 user-service QoS invocations, which is partitioned into a set of independent temporal groups of historical QoS records across 64 time slices. The overall QoS sparsity of WS-DREAM dataset is approximately 66.98%. Table 2 provides the detailed statistics regarding the temporal RT dataset. For the splitting strategy of training data, we have randomly selected QoS invocation records across multiple time slices from the original dataset to generate the experimental dataset that aims to simulate realistic application scenarios as closely as possible. In the experiments, we have divided the dataset into four different QoS densities, including 5%, 10%, 15%, and 20%. For the comparison of temporal QoS prediction performance, the remaining QoS samples at each QoS density are used as testing samples.

To verify the effectiveness of our proposed TRCF, we tune different ranges of parameters. In the experiments, the prediction performance of TRCF is impacted by the parameter settings, which is shown in Table 3.

4.2 Evaluation Metrics

In the experiments, we compare the QoS prediction performance of TRCF and competing baselines by two evaluation metrics, including Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

Let $r^t_{u,s}$ and $\hat{r}^t_{u,s}$ represent the ground truth QoS and predicted time-aware QoS of a target service s invoked by a target user u at a time slice t, respectively. MAE and RMSE are applied to measure the variances between the observed QoS and the predicted missing temporal QoS.

$$MAE = \frac{\sum_{u,s} \left| r^t_{u,s} - \hat{r}^t_{u,s} \right|}{N} \tag{34}$$

$$RMSE = \sqrt{\frac{\sum_{u,s} \left(r^t_{u,s} - \hat{r}^t_{u,s} \right)^2}{N}} \tag{35}$$

where N is the number of test samples of the predicted time-aware QoS. Obviously, we can find that smaller values on MAE and RMSE indicate better accuracy of predicting missing temporal QoS across multiple time slices.

In our experiments, MAE is a linear evaluation metric that equally weights all individual differences, enabling it to demonstrate the accuracy of overall time-aware QoS prediction. Conversely, RMSE enhances the weighting of those individual outliers that is more sensitive to large errors of time-aware QoS prediction.

4.3 Competing Methods

To evaluate the performance of TRCF, we compare it with eight competing baselines, including a benchmark approach UIMean, two traditional CF-based approaches UPCC [25] and IPCC [26], as well as five well-known and state-of-the-art approaches, namely PNCF [20], WSPred [18], TUIPCC [15], PLMF [22], and RNCF [23].

- **UIMean:** It is a hybrid QoS prediction approach that combines the average user-based QoS value from UMEAN and the average service-based QoS value from IMEAN. Here, UMEAN and IMEAN calculate the average QoS of a target user who has invoked all services, and a target service that has been invoked by all users in the current time slice, respectively.
- **UPCC [25]:** It is a user-based QoS prediction approach that involves finding a group of users similar to a target user. The predicted QoS is obtained by combining the average QoS from UMEAN and the deviation migration from similar users.
- **IPCC [26]:** It is a service-based QoS prediction approach that involves finding a set of services similar to a target service. The predicted QoS is obtained by combining the average QoS from IMEAN and the deviation migration from similar services.
- **PNCF [20]:** It is a personalized recommendation model by neural collaborative filtering that can also be used for QoS prediction. It uses a deep neural network to capture user-service nonlinear invocation relationships and obtain the feature representations of users and services by sparse vectors for predicting missing QoS.
- **WSPred [18]:** It is a temporal perception QoS prediction approach that upgrades the temporal dimension feature based on two-dimensional user-service QoS matrix. It expands to three-dimensional tensor decomposition and makes reliable QoS prediction results by adding the dimension of temporal factor.
- **TUIPCC [15]:** It is a temporal QoS prediction approach that combines the average historical QoS value and collaborative QoS value calculated by the selected user or service neighbors based on the similarity of user-service temporal QoS invocations.
- **PLMF [22]:** It is an LSTM-based time-aware QoS prediction approach. It first encodes three-dimensional tensor of user-service-time invocation relationships and obtains the feature representations by one-hot encoding. Then, the encoded eigenvector is reduced by the embedding dimension of a fully connected network. Finally, LSTM is applied to extract the latent temporal characteristics for predicting time-aware QoS.
- **RNCF [23]:** It introduces a multi-layer GRU into the framework of neural collaborative filtering and leverages historical user-service invocations of different time slices to learn the temporal patterns between users and services for superior time-aware QoS prediction.

4.4 Experiment Results and Analyses

In the experiments, we denote our proposed TRCF-TA as the approach of time-aware QoS prediction under dense user-service invocations, and TRCF-RTA for highly sparse QoS invocations. To ensure the fairness of the performance comparison of time-aware QoS prediction, we evaluate the effectiveness among TRCF and the competitive baselines by calculating MAE and RMSE under four different QoS densities of 5%, 10%, 15%, and 20%.

Table 4 shows the experimental results of time-aware QoS prediction under four different matrix densities on temporal RT dataset. Here, the best results of MAE and RMSE of each column within eight competing baselines and TRCF variants are first respectively highlighted in the gray background, and then the best results of each column among all competing approaches are marked in bold. From the results, we can see that our proposed TRCF-TA and TRCF-RTA continue to outperform all competitive approaches in terms of RMSE, and the QoS prediction accuracy of TRCF-TA and TRCF-RTA on MAE is slightly lower than RNCF at QoS density of 10%, but superior to

TABLE 4: Experimental Results of Time-aware QoS Prediction under Multiple QoS Densities on Temporal RT Dataset.

Methods	Density=5%		Density=10%		Density=15%		Density=20%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
UIMean	1.1494	2.6801	1.0755	2.5657	1.0012	2.4427	0.9309	2.3218
UPCC	0.9022	1.9243	0.9587	1.7961	0.8948	1.7041	0.8513	1.6284
IPCC	1.0657	2.0001	0.8938	1.7465	0.8432	1.6807	0.8075	1.6228
PNCF	1.1653	1.8358	1.0891	1.7221	1.0427	1.6533	1.0129	1.6170
WSPred	0.7809	1.7065	0.6894	1.6334	0.6726	1.6076	0.6634	1.5930
TUIPCC	0.7675	1.8025	0.7578	1.7148	0.7427	1.6584	0.7223	1.5947
PLMF	0.7267	1.7059	0.6786	1.6126	0.6582	1.5749	0.6444	1.5525
RNCF	0.6920	1.7582	**0.6007**	1.6685	0.5902	1.6035	0.5559	1.5935
TRCF-TA	0.6771	1.7106	0.6178	1.5135	**0.5656**	**1.3932**	**0.5301**	**1.3162**
TRCF-RTA	**0.6532**	**1.6283**	0.6194	**1.4922**	0.5851	1.4215	0.5655	1.3712
Gains	**5.61%**	**4.55%**	**-2.85%**	**7.47%**	**4.17%**	**11.54%**	**4.64%**	**15.22%**

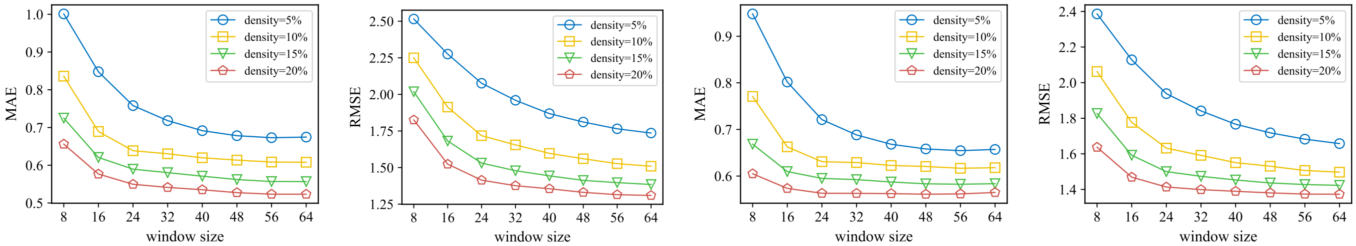
TABLE 5: Performance Comparisons between TRCF-TA and TRCF-RTA under extremely high sparse QoS densities.

Methods	Density=1%		Density=2%		Density=3%		Density=4%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
TRCF-TA	0.8638	2.0834	0.7396	1.9396	0.6900	1.8311	0.6836	1.7732
TRCF-RTA	**0.8341**	**1.9368**	**0.6924**	**1.7711**	**0.6740**	**1.6860**	**0.6671**	**1.6641**

all competitive approaches at other three QoS densities. In the case of 10% QoS density where RNCF achieves slightly better performance than our proposed TRCF-TA and TRCF-RTA in terms of MAE, it may be occurred because the model complexity of RNCF at that QoS density could more precisely match the characteristics of the dataset, allowing the model to fully capture key factors and thereby exhibiting superior QoS prediction performance. Additionally, as QoS density on temporal RT increases from 5% to 20% with an interval step 5%, it is observed that MAE and RMSE gradually become smaller among all competing approaches, indicating more effective QoS prediction performance. The underlying reason is that higher QoS density provides more sufficient user-service temporal invocations that is beneficial to mine temporal characteristics and find similar neighbors, enabling competing approaches to receive better accuracy of time-aware QoS prediction.

More Specifically, UIMEAN, UPCC, and IPCC, as basic and purely traditional CF approaches, have relatively poor QoS prediction performance. The primary reason is that they can only predict missing QoS values through two-dimensional historical QoS records, instead of three-dimensional temporal QoS invocations. That is, these approaches cannot take full advantage of past historical QoS records across multiple time slices. Compared to the three conventional approaches, the other five competing baselines take into account temporal factors, which can boost the accuracy of time-aware QoS prediction on MAE and RMSE. In particular, RNCF that can effectively represent features of different user-service pairs on multiple time slices by neural collaborative filtering performs better among all eight competitive baselines in terms of MAE under different QoS densities, but a little bit worse in terms of RMSE. On the contrary, PLMF that trains the complete time series of user-service pairs through a personalized LSTM model achieves better accuracy of time-aware QoS prediction on RMSE among all eight competitive baselines, whereas relatively lower performance on MAE. Inspired by these competing baselines, we have found that the temporally dynamic changes of user-service invocations play an important role in QoS prediction across multiple time slices, due to the differences of networking conditions among users and services. Following the observation, TRCF captures the continuous QoS fluctuations along multiple time slices by temporal QoS vectors, which are used to measure the similarity between two users or services. As a result, it can find neighbors of a user or service with similar QoS temporal trends, which is used to calculate temporal average QoS and temporal deviation migration for better accuracy of time-aware QoS prediction on both MAE and RMSE.

As for our proposed TRCF-TA and TRCF-RTA, when the temporal QoS matrix is dense, abundant user-service QoS invocations can be available across multiple time slices. In such case, TRCF-TA can straightforwardly adopt sufficient temporal historical records from a target user-service pair for more accurately calculating temporal average QoS. Meanwhile, it can also directly employ temporal QoS vectors of a target user and target service to find similar users and services for more effectively calculating temporal deviation migration. Consequently, TRCF-TA obtains superior MAE and RMSE compared to TRCF-RTA at almost all of the dense QoS situations of 10%, 15% and 20%, as shown in Table 4. That is, due to sufficient historical QoS records of a given target user-service pair, TRCF-TA can effectively reflect temporal QoS vectors for predicting missing temporal QoS. At this moment, it is unnecessary to further rely on similar users or services to supplement missing historical QoS records for a target user invoking a target service at corresponding time slices, which may trigger noisy representation of temporal QoS vector for worsening the QoS prediction accuracy. However, when the density drops, the prediction results of TRCF-TA will rapidly decline. This is because the decline in data volume leads to the unclear expression of network conditions, which also affects the neighbor selection and benchmark value calculation of TRCF-TA.



(a) MAE of TRCF-TA (b) RMSE of TRCF-TA (c) MAE of TRCF-RTA (d) RMSE of TRCF-RTA

Fig. 3: Performance impact of window size on TRCF-TA and TRCF-RTA under different QoS densities.

To improve the prediction accuracy of TRCF at high sparsity, we added an RBS aggregation module when calculating the baseline value and proposed the TRCF-RTA method suitable for extremely-high sparsity. The reason is that TRCF-TA cannot directly capture useful temporal QoS vectors of a target user and target service, which significantly affects its calculation of temporal average QoS and selection of similar neighbors for temporal deviation migration. In such a scene, TRCF-RTA is applied for time-aware QoS prediction at high QoS sparsity, since it finds both similar neighbors of a target user and target service for supplementing historical QoS records that a target user has not invoked a target service at multiple time slices. In this way, TRCF-RTA enriches the originally high sparse temporal QoS vectors for more effectively calculating temporal average QoS and temporal deviation migration, where similar neighbors of a target user or service also take into account the compensation of missing historical QoS in their temporal QoS vectors to solve the high sparsity. Thus, it leads to better time-aware QoS prediction accuracy, as shown at the density of 5% in Table 4.

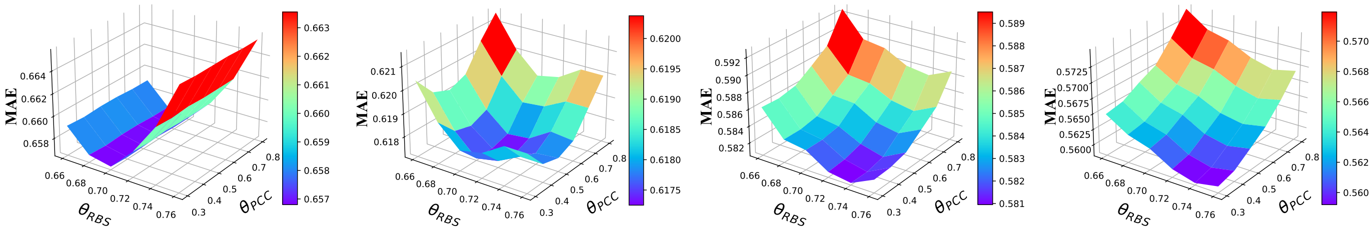
To further testify the advantages of TRCF-RTA over TRCF-TA at extremely low densities of temporal QoS matrix, we make the performance comparisons of time-aware QoS prediction under QoS densities ranging from 1% to 4% with an interval step 1%. Here, TRCF-TA is designed for temporal QoS prediction when there are sufficient historical QoS records from different time slices. TRCF-RTA is designed to enhance the performance of temporal QoS prediction in situations where the QoS density distributions are extremely sparse among multiple time slices. It can facilitate the QoS prediction by incorporating the module of temporal reinforced RBS to find similar service neighbor and supplement the missing QoS invocations in temporal QoS vector. The performance comparisons between TRCF-TA and TRCF-RTA on MAE and RMSE under additional four highly sparse QoS densities of 1%, 2%, 3%, and 4% are shown in Table 5, where the best results of each column are marked in bold and gray background. The results indicate that at each low QoS density, TRCF-RTA consistently outperforms the prediction accuracy of TRCF-TA on both MAE and RMSE. Therefore, our designed temporal reinforced RBS can effectively improve time-aware QoS prediction performance of TRCF under those densities with high QoS sparsity. It demonstrates the usefulness of temporal reinforced RBS for enhancing the reliability of calculating temporal average QoS, which ultimately raises time-aware QoS prediction of TRCF-RTA at extremely low QoS densities.

4.5 Performance Impact of Parameters

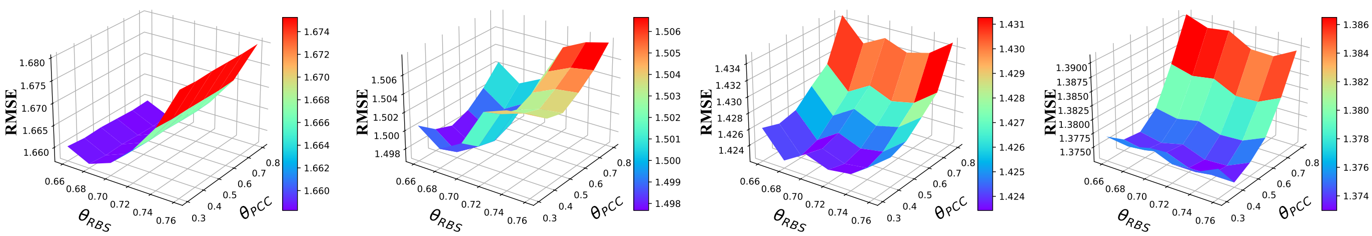
4.5.1 *Impact of Window Size*

In the experiments, window size of user-service historical QoS invocations impacts the performance of time-aware QoS prediction for both TRCF-TA and TRCF-RTA. A window size refers to the length of a temporal QoS vector where a user has invoked a service across a set of time slices. A larger window size reflects a longer temporal QoS vector for a user-service pair and vice versa. A small window size may result in the exclusion of useful historical QoS records, whereas a large window may introduce outdated noisy QoS lowering the prediction accuracy [15]. Therefore, selecting an appropriate window size is crucial to improve the performance of temporal QoS prediction of TRCF. To test the performance impact of window size, we have conducted a series of experiments by setting the parameters of θ_{PCC}=0.5, θ_{RBS}=0.68 and α=0.5, while varying the window size from 8 to 64 with an interval step of 8. The experimental results of performance impact of window size on MAE and RMSE is illustrated in Fig. 3, where each window size is tested on TRCF-TA and TRCF-RTA under four QoS densities of 5%, 10%, 15% and 20%.

It can be observed from the experimental results that when the window size is small, increasing it significantly enhances the performance of time-aware QoS prediction, since a relatively larger window size can provision more useful historical invocation QoS records. More specifically, in the case of low window size, since TRCF-TA only considers the original temporal QoS vector of a target user-service pair, it lacks of sufficient historical temporal QoS records and results in poor prediction accuracy. Comparatively, TRCF-RTA can enrich temporal QoS vector by using similar neighbors' historical QoS records, thereby achieving relatively superior QoS prediction performance in a small window size. However, as the window size becomes larger, TRCF-TA can capture its own original historical QoS records that remarkably improves the prediction accuracy, while TRCF-RTA probably brings in noisy historical QoS invocations by similar neighbors that may has a negative impact of outweighing the benefits of increased number of temporal QoS invocations, thereby weakening the effectiveness of predicting temporal QoS in a large window size. Moreover, as the windows size continues to increase, the final QoS prediction accuracy of TRCF-RTA is inferior to that of TRCF-TA. Thus, the overall experimental curve on MAE and RMSE of TRCF-RTA converges faster than that of TRCF-TA, as shown in Fig. 3. In the temporal RT dataset, larger



(a) Density = 5% (b) Density = 10% (c) Density = 15% (d) Density = 20%



(e) Density = 5% (f) Density = 10% (g) Density = 15% (h) Density = 20%

Fig. 4: Performance impact of θ_{PCC} and θ_{RBS} on MAE and RMSE of TRCF-RTA under different QoS densities.

window sizes lead to better accuracy for TRCF, whereas it takes more additionally computational costs for predicting missing temporal QoS.

4.5.2 Impact of θ_{PCC} and θ_{RBS}

The settings of θ_{PCC} and θ_{RBS} impact the selection of similar neighbors of a target user and service, which are crucial to effectively calculate temporal average QoS and temporal deviation migration for better time-aware QoS prediction. However, a higher or lower threshold of θ_{PCC} and θ_{RBS} affects the experimental results due to the exclusion of potentially useful temporal characteristics or additional noisy user-service invocations that compromise the temporal QoS prediction accuracy.

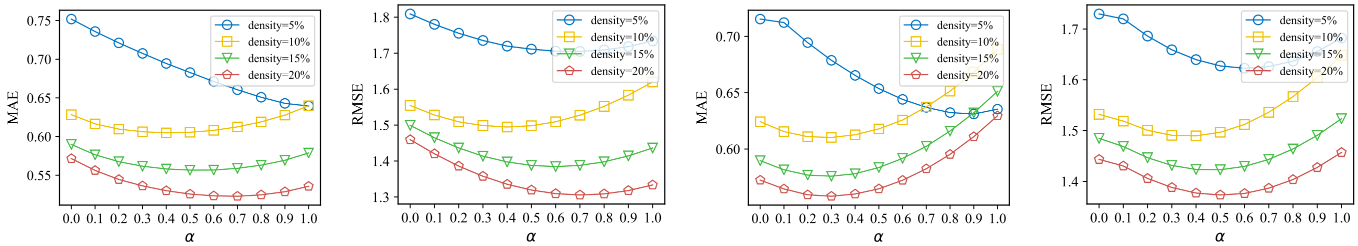
To test the performance impact of two similarity thresholds for optimally finding similar neighbors, we have conducted experiments with a predefined window size of 64 and α=0.5, while θ_{PCC} ranges from 0.3 to 0.8 with an interval step of 0.1, and θ_{RBS} varies from 0.66 to 0.76 with an interval step of 0.02. Fig. 4 illustrates the performance impact of θ_{PCC} and θ_{RBS} on MAE and RMSE, where TRCF-RTA is used to predict temporal QoS under four different densities of 5%, 10%, 15% and 20%, respectively.

From the three-dimensional visualizations on MAE and RMSE as the variations of θ_{PCC} and θ_{RBS}, we can find that when the QoS density starts from 5%, adjusting θ_{PCC} has a weak performance impact on time-aware QoS prediction because of relatively less influence on temporal deviation migration by θ_{PCC} than that of temporal average QoS by θ_{RBS}. Thus, adjusting θ_{RBS} leads to significantly positive changes on MAE and RMSE since TRCF-RTA is sensitive to more accurately calculate temporal average QoS by useful similar neighbors under highly sparse QoS. As the increasing QoS density from 5% to 10% and 15%, TRCF-RTA is impacted more by temporal deviation migration relative to θ_{PCC} than that by temporal average QoS relative to θ_{RBS}, where adjusting both θ_{PCC} and θ_{RBS} has obvious influence on the accuracy of temporal QoS prediction. However, when the QoS density arises at 20%, the influence of adjusting θ_{RBS} has been significantly reduced, particularly on RMSE, while θ_{PCC} can still keep a substantial impact on prediction accuracy. The main reason is that as the QoS density gradually increases, an original temporal QoS vector of a target user-service pair can sufficiently represent their temporally historical invocation patterns, without externally supplementing missing QoS invocations across multiple time slices for better calculating the temporal average QoS. After comprehensive parameter tuning, we set θ_{PCC}=0.5 and θ_{RBS}=0.68 that achieve the best performance of time-aware QoS prediction in temporal RT dataset.

4.5.3 Impact of Adjusting Coefficient α

TRCF combines user-based and service-based time-aware QoS prediction by an adjusting coefficient α, which balances the impact of these two kinds of ways and optimizes the finally missing time-aware predicted QoS. Since the total adjusting coefficients of the two ways are 1, we focus on analyzing the performance impact of α. In the experiments, we set the parameters of window size as 64, θ_{PCC}=0.5, and θ_{RBS}=0.68, while varying the adjusting coefficient α from 0.0 to 1.0 with an interval step of 0.1. The densities of temporal QoS matrix are 5%, 10%, 15% and 20%.

Fig. 5 illustrates the performance impact of adjusting coefficient on time-aware QoS prediction. When α is set to 1.0, TRCF completely degenerates into user-based approach; at the other extreme, when α is set to 0.0, it turns to be a purely service-based temporal approach. It can be observed from Fig. 5 that both TRCF-TA and TRCF-RTA can receive the lowest MAE and RMSE for the best performance with the setting of α at a certain value between 0 and 1. It indicates that both user-based and service-based time-aware QoS prediction make contributions to the improvement of predicting missing temporal QoS under multiple densities. However, it is challenging to identify a fixed α that achieves



(a) MAE of TRCF-TA (b) RMSE of TRCF-TA (c) MAE of TRCF-RTA (d) RMSE of TRCF-RTA

Fig. 5: Performance impact of adjusting coefficient α on TRCF-TA and TRCF-RTA under different QoS densities.

the best performance on MAE and RMSE for TRCF-TA and TRCF-RTA across multiple QoS densities. For example, when QoS density is set to 10% for TRCF-TA, it performs optimally on MAE and RMSE with the setting of α=0.4, while TRCF-TA can receive the best performance by adjusting α=0.7 under the QoS density of 20%. Taking into account the prediction performance on MAE and RMSE, TRCF-TA and TRCF-RTA can receive superior accuracy of temporal QoS prediction when their adjusting coefficients are set by α=0.6 and α=0.5, respectively.

5 Related Work

5.1 Non-temporal QoS Prediction

Non-temporal QoS prediction can be classified into three categories: memory-based, model-based and deep learning based approaches, which are typically performed on a two-dimensional matrix of user-service QoS invocations.

Memory-based approaches mainly employ traditional collaborative filtering (CF) to predict missing QoS. It can be divided into user-based [25], service-based [26], and their linear combination through weight coefficients. The core of memory-based QoS prediction approaches is to identify a group of similar users or services as the neighborhood by similarity calculation, and use them for calculating deviation migration, which is finally combined with average QoS to perform the task of QoS prediction. Some researchers have focused on effectively quantifying the similarity between users and services to recognize similar neighborhoods [27]. Wu et al. [6] proposed a rate-based similarity (RBS) approach to select the neighborhood of users and services, resulting in better QoS prediction. Zou et al. [7] proposed a reinforced CF approach based on both RBS and PCC, which can accurately calculate average QoS and deviation migration.

Model-based and deep learning approaches can partially address the limitation of CF-based ones by extracting implicit linear or nonlinear invocation relationships to enhance QoS prediction performance. Xu et al. [8] proposed two context-aware matrix factorization models for users and services to obtain more accurate QoS prediction results. Wu et al. [9] proposed a general context-sensitive matrix factorization approach to model the interaction between users and services. Additionally, deep learning techniques have been recently used to solve QoS prediction problems since they can better deal with sparsity and learn implicit nonlinear interactions [10], [11]. [28] combined neural networks and matrix factorization, adopting multi-task learning to reduce prediction errors and improve the performance of the predicted QoS. Zou et al. Li et al. [12] proposed topology-aware neural (TAN) model to address the challenge of collaborative QoS prediction by considering the underlying network topology and complex interactions between autonomous systems. [29] designed a location-aware two-tower deep residual network together with collaborative filtering to achieve superior QoS prediction. In the latest advancements, some researches have further improved QoS prediction performance by using expert systems and attention mechanisms [30] or graph neural networks [31] for multiple feature selection, extraction, and interaction from user-service contextual information and QoS invocations.

5.2 Time-aware QoS Prediction

Time-aware QoS prediction can be partitioned into four categories, including temporal factor integrated CF, sequence prediction, tensor decomposition and deep learning.

Hu et al. [13] integrated temporal factor with the CF approach and selected more similar neighbors through a random walk algorithm to alleviate data sparsity and achieve better time-aware QoS prediction. Ma et al. [14] proposed a new vector comparison approach that combines orientation similarity and dimension similarity to implement time series analysis for multi-valued collaborative QoS prediction in cloud computing. Tong et al. [15] proposed an improved time-aware QoS prediction approach based on CF. First, it normalized the historical QoS value and calculated the similarity. Then, it calculated the weight based on the distance of time slices and selected similar neighbors. Finally, the missing QoS was predicted using hybrid CF. These approaches demonstrate the effectiveness of integrating temporal information into QoS prediction, and addressing the limitations of non-temporal QoS prediction approaches.

Due to the correlation between time-aware QoS prediction and sequence prediction analysis, relevant research has used the ARIMA model to enhance the prediction performance of missing temporal QoS. Hu et al. [16] established a QoS prediction model that effectively combines CF and the ARIMA model, and applied the Kalman filtering algorithm to compensate for the shortcomings of the ARIMA model in time-aware QoS prediction. Ding et al. [17] combined the ARIMA model with memory-based CF to capture the temporal characteristics of user similarity, improving the performance of missing predicted temporal QoS. These approaches demonstrate the effectiveness of integrating sequence prediction analysis with QoS prediction, improving the accuracy of time-aware QoS prediction by capturing temporal characteristics of QoS variations.

Compared to non-temporal QoS prediction, incorporating temporal factor requires converting the classic two-dimensional user-service matrix into a three-dimensional tensor representation, where matrix factorization is upgraded to three-dimensional tensor decomposition [18], [19]. Meng et al. [32] proposed a time-aware hybrid collaborative cloud service recommendation approach, which introduced a temporal-aware LFM model-based on CP decomposition and biases model to distinguish temporal QoS metrics from stable QoS ones. Zhang et al. [33] proposed an approach that combines Personalized Gated Recurrent Unit (PGRU) and Generalized Tensor Factorization (GTF) to comprehensively predict unknown time-aware QoS by leveraging long short term dependency patterns. Luo et al. [34] proposed a temporal pattern-aware QoS prediction approach by biased non-negative late factorization of tensors (BNLFTs) model, which extracts time potential factors from dynamic QoS. These approaches demonstrate the effectiveness of incorporating temporal factor into QoS prediction by using tensor representations and factorization techniques.

With regard to deep learning models, RNN and its variants LSTM and GRU have been recently used for time-aware QoS prediction. Wang et al. [35] applied LSTM to create online reliable QoS prediction model for service-oriented systems. Xiong et al. [21] considered multi-dimension context for learning an effective QoS prediction model derived from the past QoS invocation history. Xiong et al. [22] proposed a personalized matrix factorization approach PLMF based on LSTM, which can capture dynamic representations for online QoS prediction. Zou et al. [36] proposed a temporal QoS prediction framework called DeepTSQP, which combines binary features with memory-based similarity to express the characteristics of users or services and feeds them to a GRU model for mining temporal aggregated feature for predicting unknown temporal QoS value. These approaches demonstrate the effectiveness of using deep learning models for temporal QoS prediction by capturing temporal dependencies and patterns.

6 Conclusion and Future Work

In this paper, we propose a novel approach called TRCF, which is dedicated to advancing the performance of time-aware QoS prediction. First, when calculating temporal average QoS with densely historical QoS records, TRCF-TA directly fusions the QoS values of a given target user-service invocations across multiple time slices; especially for high QoS sparsity, TRCF-RTA finds a set of similar neighbors and their corresponding historical QoS values at different time slices are taken by temporal reinforced RBS to enhance the reliability of calculating temporal average QoS. Then, TRCF performs temporal deviation migration by incorporating reliable factor to improve the effectiveness of calculating the aggregated deviations from similar neighbors based on temporal reinforced PCC. Finally, TRCF integrates the temporal average QoS and temporal deviation migration to predict the missing time-aware QoS. The experimental results demonstrate that TRCF achieves the best performance compared with state-of-the-art competing baselines.

In the future, we plan to explore and design new deep neural networks by plugging into similar neighborhoods in TRCF as heuristics to further improve the performance of time-aware QoS prediction.

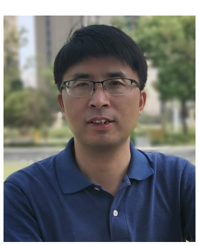
Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 62272290, 62172088) and Shanghai Natural Science Foundation (No. 21ZR1400400).

References

[1] H. Ma, Z. Hu, K. Li, and H. Zhu, "Variation-aware cloud service selection via collaborative QoS prediction," *IEEE Transactions on Services Computing*, vol. 14, no. 6, pp. 1954–1969, 2019.
[2] M. Li, H. Xu, Z. Tu, T. Su, X. Xu, and Z. Wang, "A deep learning based personalized QoE/QoS correlation model for composite services," in *IEEE International Conference on Web Services*, 2022, pp. 312–321.
[3] X. Wu, Y. Fan, J. Zhang, H. Lin, and J. Zhang, "QF-RNN: QI-matrix factorization based RNN for time-aware service recommendation," in *IEEE International Conference on Services Computing (SCC)*, 2019, pp. 202–209.
[4] B. Cao, X. F. Liu, M. M. Rahman, B. Li, J. Liu, and M. Tang, "Integrated content and network-based service clustering and Web APIs recommendation for mashup development," *IEEE Transactions on Services Computing*, vol. 13, no. 1, pp. 99–113, 2017.
[5] S. H. Ghafouri, S. M. Hashemi, and P. C. Hung, "A survey on web service QoS prediction methods," *IEEE Transactions on Services Computing*, vol. 15, no. 4, pp. 2439–2454, 2020.
[6] X. Wu, B. Cheng, and J. Chen, "Collaborative filtering service recommendation based on a novel similarity computation method," *IEEE Transactions on Services Computing*, vol. 10, no. 03, pp. 352–365, 2017.
[7] G. Zou, M. Jiang, S. Niu, H. Wu, S. Pang, and Y. Gan, "QoS-aware web service recommendation with reinforced collaborative filtering," in *International Conference on Service-Oriented Computing (ICSOC)*, 2018, pp. 430–445.
[8] Y. Xu, J. Yin, S. Deng, N. N. Xiong, and J. Huang, "Context-aware QoS prediction for web service recommendation and selection," *Expert Systems with Applications*, vol. 53, pp. 75–86, 2016.
[9] H. Wu, K. Yue, B. Li, B. Zhang, and C. H. Hsu, "Collaborative QoS prediction with context-sensitive matrix factorization," *Future Generation Computer Systems*, vol. 82, pp. 669–678, 2018.
[10] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," in *International Conference on World Wide Web*, 2017, pp. 173–182.
[11] H. Wu, Z. Zhang, J. Luo, K. Yue, and et al., "Multiple attributes QoS prediction via deep neural model with contexts," *IEEE Transactions on Services Computing*, vol. 14, no. 4, pp. 1084–1096, 2021.
[12] J. Li, H. Wu, J. Chen, Q. He, and et al., "Topology-aware neural model for highly accurate QoS prediction," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 7, pp. 1538–1552, 2022.
[13] Y. Hu, Q. Peng, and X. Hu, "A time-aware and data sparsity tolerant approach for web service recommendation," in *IEEE International Conference on Web Services*, 2014, pp. 33–40.
[14] H. Ma, H. Zhu, Z. Hu, W. Tang, and et al., "Multi-valued collaborative QoS prediction for cloud service via time series analysis," *Future Generation Computer Systems*, vol. 68, pp. 275–288, 2017.
[15] E. Tong, W. Niu, and J. Liu, "A missing QoS prediction approach via time-aware collaborative filtering," *IEEE Transactions on Services Computing*, vol. 15, no. 6, pp. 3115–3128, 2021.
[16] Y. Hu, Q. Peng, X. Hu, and R. Yang, "Web service recommendation based on time series forecasting and collaborative filtering," in *IEEE International Conference on Web Services*, 2015, pp. 233–240.
[17] S. Ding, Y. Li, D. Wu, Y. Zhang, and S. Yang, "Time-aware cloud service recommendation using similarity-enhanced collaborative filtering and arima model," *Decision Support Systems*, vol. 107, pp. 103–115, 2018.
[18] Y. Zhang, Z. Zheng, and M. R. Lyu, "WSPred: A time-aware personalized QoS prediction framework for web services," in *IEEE International Symposium on Software Reliability Engineering (ISSRE)*, 2011, pp. 210–219.

[19] W. Zhang, H. Sun, X. Liu, and X. Guo, "Temporal QoS-aware web service recommendation via non-negative tensor factorization," in *International Conference on World Wide Web*, 2014, pp. 585–596.
[20] L. Chen, A. Zheng, Y. Feng, F. Xie, and Z. Zheng, "Software service recommendation base on collaborative filtering neural network model," in *International Conference on Service-Oriented Computing (ICSOC)*, 2018, pp. 388–403.
[21] W. Xiong, Z. Wu, B. Li, and Q. Gu, "A learning approach to QoS prediction via multi-dimensional context," in *IEEE International Conference on Web Services*, 2017, pp. 164–171.
[22] R. Xiong, J. Wang, Z. Li, B. Li, and P. C. Hung, "Personalized LSTM based matrix factorization for online QoS prediction," in *IEEE International Conference on Web Services*, 2018, pp. 34–41.
[23] T. Liang, M. Chen, Y. Yin, L. Zhou, and H. Ying, "Recurrent neural network based collaborative filtering for QoS prediction in IoV," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2400–2410, 2021.
[24] Z. Zheng, Y. Zhang, and M. R. Lyu, "Distributed QoS evaluation for real-world web services," in *IEEE International Conference on Web Services*, 2010, pp. 83–90.
[25] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS prediction for web services via collaborative filtering," in *IEEE International Conference on Web Services*, 2007, pp. 439–446.
[26] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *International Conference on World Wide Web*, 2001, pp. 285–295.
[27] Y. Wang, J. Deng, J. Gao, and P. Zhang, "A hybrid user similarity model for collaborative filtering," *Information Sciences*, vol. 418, pp. 102–118, 2017.
[28] J. Xu, L. Xiao, Y. Li, M. Huang, Z. Zhuang, T. H. Weng, and W. Liang, "NFMF: neural fusion matrix factorisation for QoS prediction in service selection," *Connection Science*, vol. 33, no. 3, pp. 753–768, 2021.
[29] G. Zou, S. Wu, S. Hu, C. Cao, Y. Gan, B. Zhang, and Y. Chen, "NCRL: Neighborhood-based collaborative residual learning for adaptive QoS prediction," *IEEE Transactions on Services Computing*, vol. 16, no. 3, pp. 2023–2043, 2023.
[30] H. Lian, J. Li, H. Wu, Y. Zhao, L. Zhang, and X. Wang, "Towards effective personalized service QoS prediction from the perspective of multi-task learning," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 2587–2597, 2023.
[31] Z. Wu, D. Ding, Y. Xiu, Y. Zhao, and J. Hong, "Robust QoS prediction based on reputation integrated graph convolution network," *IEEE Transactions on Services Computing*, DOI: 10.1109/TSC.2023.3317642, 2023.
[32] S. Meng, Z. Zhou, T. Huang, D. Li, S. Wang, F. Fei, W. Wang, and W. Dou, "A temporal-aware hybrid collaborative recommendation method for cloud service," in *IEEE International Conference on Web Services*, 2016, pp. 252–259.
[33] Y. Zhang, C. Yin, Z. Lu, D. Yan, M. Qiu, and Q. Tang, "Recurrent tensor factorization for time-aware service recommendation," *Applied Soft Computing*, vol. 85, 105762, 2019.
[34] X. Luo, H. Wu, H. Yuan, and M. Zhou, "Temporal pattern-aware QoS prediction via biased non-negative latent factorization of tensors," *IEEE Transactions on Cybernetics*, vol. 50, no. 5, pp. 1798–1809, 2019.
[35] H. Wang, Z. Yang, and Q. Yu, "Online reliability prediction via long short term memory for service-oriented systems," in *IEEE International Conference on Web Services*, 2017, pp. 81–88.
[36] G. Zou, T. Li, M. Jiang, S. Hu, C. Cao, B. Zhang, Y. Gan, and Y. Chen, "DeepTSQP: Temporal-aware service QoS prediction via deep neural network and feature integration," *Knowledge-Based Systems*, vol. 241, 108062, 2022.



Guobing Zou is a full professor and vice dean of the School of Computer Engieering and Science, Shanghai University, China. He received his PhD degree in Computer Science from Tongji University, Shanghai, China, 2012. He has worked as a visiting scholar in the Department of Computer Science and Engineering at Washington University in St. Louis from 2009 to 2011, USA. His current research interests mainly focus on services computing, edge computing, data mining and intelligent algorithms, recommender systems. He has published more than 100 papers on premier international journals and conferences, including IEEE Transactions on Services Computing, IEEE Transactions on Network and Service Management, IEEE ICWS, ICSOC, IEEE SCC, AAAI, Information Sciences, Expert Systems with Applications, Knowledge-Based Systems, Applied Intelligence, etc.

Yutao Huang is currently a master student in the School of Computer Engineering and Science, Shanghai University, China. Before that, he received a Bachelor degree in Computer Science and Technology at Shanghai University, China, 2021. His research interests include temporal QoS prediction, collaborative filtering, and deep learning. He has led a research and development group to successfully design, implement and maintain a big data service system for intelligent delivery of recyclables, which can effectively identify and visualize real-time delivery data of large-scale recyclables, resulting in significant economic and social benefits.

Shengxiang Hu is currently a PhD candidate in the School of Computer Engineering and Science, Shanghai University, China. Before that, he received a Bachelor degree in 2018 and Master degree in 2021 both in Computer Science and Technology at Shanghai University, respectively. His research interests include QoS prediction, graph neural network and natural language processing. He has published more than five papers on IEEE Transactions on Services Computing, IEEE Knowledge-Based Systems, International Conference on Web Services (IEEE ICWS), International Conference on Service-Oriented Computing (ICSOC), International Conference on Parallel Problem Solving from Nature (PPSN).

Yanglan Gan is a full professor in the School of Computer Science and Technology, Donghua University, Shanghai, China. She received her PhD in Computer Science from Tongji University, Shanghai, China, 2012. Her research interests include bioinformatics, service computing, and data mining. She has published more than 50 papers on premier international journals and conferences, including Bioinformatics, Briefings in Bioinformatics, BMC Bioinformatics, IEEE/ACM Transactions on Computational Biology and Bioinformatics, IEEE Transactions on Services Computing, IEEE Transactions on Network and Service Management, IEEE ICWS, ICSOC, Neurocomputing, and Knowledge-Based Systems.

Bofeng Zhang is a full professor and dean of the School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai, China. He received his PhD degree from the Northwestern Polytechnic University (NPU) in 1997, China. He experienced a Postdoctoral Research at Zhejiang University from 1997 to 1999, China. He worked as a visiting professor at the University of Aizu from 2006 to 2007, Japan. He worked as a visiting scholar at Purdue University from 2013 to 2014, US. His research interests include personalized service recommendation, intelligent human-computer interaction, and data mining. He has published more than 200 papers on international journals and conferences.

Yixin Chen received the PhD degree in computer science from the University of Illinois at Urbana Champaign, in 2005. He is currently a full professor of Computer Science at Washington University in St. Louis, MO, USA. His research interests include artificial intelligence, data mining, deep learning, and big data analytics. He has published more than 210 papers on premier international journals and conferences, including AIJ, JAIR, IEEE TPDS, IEEE TKDE, IEEE TSC, IEEE TC, IEEE TII, IJCAI, AAAI, ICML, KDD, etc. He won the Best Paper Award at AAAI and a best paper nomination at KDD. He received an Early Career Principal Investigator Award from the US Department of Energy and a Microsoft Research New Faculty Fellowship. He was an Associate Editor for the ACM TIST, IEEE TKDE, and JAIR. He is a Fellow of the IEEE.